## 2 Simplification of 3-way Networks

In the present paper, the case of a tripartite network is considered as an example to show how the proposed network data simplification method works. In particular, we consider the real case study of student mobility paths in Italian universities. The MOBYSU.IT dataset[1] enables reconstruction of network data structures considering student mobility flows among territorial units and universities.

More formally, given $\mathcal{V}_P \equiv \{p_1, \ldots, p_i, \ldots, p_I\}$, the set of $I$ provinces of residence; $\mathcal{V}_U \equiv \{u_1, \ldots, u_j, \ldots, u_J\}$, the set of $J$ Italian universities, and $\mathcal{V}_E \equiv \{e_1, \ldots, e_k, \ldots, e_K\}$, the set of $K$ educational programmes, a weighted tripartite 3-uniform hyper-graph $\mathcal{T}$ can be defined, consisting of a triple $(\mathcal{V}, \mathcal{L}, \mathcal{W})$, with $\mathcal{V} = \{\mathcal{V}_P, \mathcal{V}_U, \mathcal{V}_E\}$ the collection of three sets of vertices, one for each mode, and being $\mathcal{L} = \{\mathcal{L}_{PUE}\}$, $\mathcal{L}_{PUE} \subseteq \mathcal{V}_P \times \mathcal{V}_U \times \mathcal{V}_E$, the collection of hyper-edges, with generic term $(p_i, u_j, e_k)$, which is the link joining the $i$-th province, the $j$-th university, and the $k$-th educational programme. Finally, $\mathcal{W}$ is the set of weights, obtained by the function $w : \mathcal{L}_{PUE} \rightarrow \mathbb{N}$, and $w(p_i, u_j, e_k) = w_{ijk}$ is the number of students moving from a province $p_i$ towards a university $u_j$ in an educational programme $e_k$. Such a network structure can be described as a three-way array $\mathbb{A} = (a_{ijk})$, with $a_{ijk} \equiv w_{ijk}$, and it has been called a 3-way network [3].

To deal with such a complex network structure and aiming at obtaining communities in which three modes are mixed, we wish to simplify the tripartite nature of the graph, without losing any significant information. In statistical terms, the array $\mathbb{A}$ can be interpreted as a 3-way contingency table, and then the statistical techniques to evaluate the association among variables (i.e. the modes) can be exploited [1]. Because a 3-way contingency table is a cross-classification of observations by the levels of three categorical variables, we are defining a network structure where the sets of nodes are the levels of the categorical variables. Specifically, we assume that if two modes are jointly associated –as are, for their own nature, universities and educational programmes– the tripartite network can be logically simplified into a bipartite one. In the student mobility network, we join the pair of nodes in $\mathcal{V}_U$ and in $\mathcal{V}_E$, and then we deal with the relationships between these *dyads* and the nodes in $\mathcal{V}_P$.

Following this assumption, the sets of nodes $\mathcal{V}_U$ and $\mathcal{V}_E$ are put together into a set of joint nodes, namely $\mathcal{V}_{UE}$. The tripartite network $\mathcal{T}$ can now be represented as a bipartite network $\mathcal{B}$ given by the triple $\{\mathcal{V}^*, \mathcal{L}^*, \mathcal{W}^*\}$, with $\mathcal{V}^* = \{\mathcal{V}_P, \mathcal{V}_{UE}\}$. The set of hyper-edges $\mathcal{L}$ is thus simplified into a set of edges $\mathcal{L}^* = \{\mathcal{L}_{P,UE}\}$, $\mathcal{L}_{P,UE} \subseteq \mathcal{V}_P \times \mathcal{V}_{UE}$. The new edges $(p_i, (u_j; e_k))$ connect a province $p_i$ with an educational programme $e_k$ running in a given university $u_j$. The weights $\mathcal{W}^*$ are the same as in the hyper-graph $\mathcal{T}$, i.e., $w^*_{ij,k} = w_{ijk}$. Note that the weights contained in the 3-way array $\mathbb{A}$ are preserved, but are now organized in a rectangular matrix $\mathbf{A}$ of $I$ rows and $(J \times K)$ columns.

---

[1] Database MOBYSU.IT [Mobilità degli Studi Universitari in Italia], research protocol MUR - Universities of Cagliari, Palermo, Siena, Torino, Sassari, Firenze, Cattolica and Napoli Federico II, Scientific Coordinator Massimo Attanasio (UNIPA), Data Source ANS-MUR/CINECA.

Taking advantage of this method, we aim to analyse weighted bipartite graphs adopting clustering methods. Among others, we use the Infomap community detection algorithm [9, 4] to study the flows' patterns in network structures instead of modularity optimization proposed in topological approaches [18, 5]. Indeed, the rationale of this algorithm –*map equation*– takes advantage of the duality between finding communities and minimizing the length –*codelength*– of a random walker's movement on a network. The partition with the shortest path length is the one that best captures the community structure in the bipartite data. Formally, the algorithm defines a module partition **M** of $n$ vertices into $m$ modules such that each vertex is assigned to one and only one module. The Infomap algorithm looks for the best **M** partition that minimizes the expected *codelength*, $L(M)$, of a random walker, given by the following map equation:

$$L(M) = q_{\curvearrowright} H(\mathscr{Q}) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(\mathscr{P}^i) \tag{1}$$

In equation (1), $q_{\curvearrowright} H(\mathscr{Q})$ represents the entropy of the movement between modules weighed for the probability that the random walker switches modules on any given step ($q_{\curvearrowright}$), and $\sum_{i=1}^{m} p^i_{\circlearrowright} H(\mathscr{P}^i)$ is the entropy of movements within modules weighed for the fraction of within-module movements that occur in module $i$, plus the probability of exiting module $i$ ($p^i_{\circlearrowright}$), such that $\sum_{i=1}^{m} p^i_{\circlearrowright} = 1 + q_{\curvearrowright}$ [9].

In our case, the Infomap algorithm is adopted to discover communities of students characterized by similar mobility patterns. Indeed, to analyse mobility data, where links represent patterns of student movement among territorial units and universities, flow-based approaches are likely to identify the most important features. Finally, in our student mobility network, to focus only on relevant student flows, a filtering procedure is adopted by considering the Empirical Cumulative Density Function (ECDF) of links' weights distribution.

## 2.1 Main Findings

Students' cohorts enrolled in Italian universities in four academic years (a.y.) 2008–09, 2011–12, 2014–15, and 2017–18 are analysed. The number of nodes for the sets $\mathcal{V}_P$ (107 provinces), $\mathcal{V}_U$ (79-80 universities), and $\mathcal{V}_E$ (45 educational programmes), and the number of students involved in the four cohorts are quite stable over time (Table 1). Furthermore, the percentage of movers (i.e., students enrolled in a university outside of their region of residence) increased, from 16.4% in the a.y. 2008–09 to 20.6% in the a.y. 2017–18, and it is higher for males than females.

**Table 1** Percentage of students according to their mobility status by cohort and gender.

| Cohort | Gender | | Mover status | |
|---|---|---|---|---|
| | | | Stayers% | Movers% |
| 2008–09 | F | 136,381 | 84.2 | 15.8 |
| | M | 106,950 | 82.8 | 17.2 |
| | Total | 243,331 | 83.6 | 16.4 |
| 2011–12 | F | 126,606 | 81.7 | 18.3 |
| | M | 102,479 | 80.9 | 19.1 |
| | Total | 229,085 | 81.0 | 19.0 |
| 2014–15 | F | 121,121 | 80.5 | 19.5 |
| | M | 102,358 | 80.4 | 19.6 |
| | Total | 223,479 | 80.5 | 19.5 |
| 2017–18 | F | 134,315 | 79.1 | 20.9 |
| | M | 113,496 | 79.8 | 20.2 |
| | Total | 247,811 | 79.4 | 20.6 |

Following the network simplification approach, the tripartite networks –one for each cohort– are simplified into bipartite networks, and the four ECDFs of links' weights are considered to filter relevant flows. The distributions suggest that more than 50% of links between pairs of nodes have weights equal to 1 (i.e., flows of only one student), and about 95% of flows are characterized by flows not greater than a digit. Thus, networks holding links with a value greater or equal to 10 are further analysed.

To reveal groups of universities and educational programmes attracting students, the Infomap community detection algorithm is applied. Looking at Table 2, we notice a reduction of the number of communities from the first to the last student cohort, suggesting a sort of stabilization in the trajectories of movers towards brand universities of the center-north with also an increase in the north-north mobility [20], and a relevant dichotomy between scientific and humanistic educational programmes. Network visualizations by groups (Figures 1 and 2) confirm that the more attractive universities are located in the north of Italy, especially for educational programmes in economics and engineering (the Bocconi University, the Polytechnic of Turin and the Cattolica University).

**Table 2** Number of communities, codelength, and relative saving codelength per cohort.

| Cohort | Communities | Codelength | Relative saving codelength |
|---|---|---|---|
| 2008–09 | 14 | 0.96 | 83% |
| 2011–12 | 17 | 1.72 | 70% |
| 2014–15 | 3 | 5.23 | 12% |
| 2017–18 | 3 | 1.00 | 83% |

**Fig. 1** Network visualization by groups, student cohort a.y. 2008–09.

**Fig. 2** Network visualization by groups, student cohort a.y. 2017–18.

## 3 Concluding Remarks

The proposed simplification network strategy on tripartite graphs defined for student mobility data provides interesting insights for the phenomenon under analysis. The main attractive destinations still remain the northern universities for educational programmes, such as engineering and business. Besides the well-known south-to-north route, other interregional routes in the northern area appear. In addition, the reduction in the number of communities suggests a sort of stabilization in terms of mobility roots of movers towards brand universities, highlighting student university destination choices close to the labor market demand.

Hyper-graphs and multipartite networks still remain very active areas for research and challenging tasks for scholars interested in discovering the complexities underlying these kinds of data. Specific tools for such complex network structures should be designed combining network analysis and other statistical techniques. As future lines of research, the comparison of community detection algorithms that better represent the structural constraints of the phenomena under analysis and the assessment of other backbone approaches to filter the significant links will be developed.

**Acknowledgements** The contribution has been supported from Italian Ministerial grant PRIN 2017 "From high school to job placement: micro-data life course analysis of university student mobility and its impact on the Italian North-South divide", n. 2017 HBTK5P - CUP B78D19000180001.

## References

1. Agresti, A.: Categorical Data Analysis (Vol. 482). John Wiley & Sons, New York (2003)
2. Barber, M. J.: Modularity and community detection in bipartite networks. Phys. Rev. E, **76**, 066102 (2007)
3. Batagelj, V., Ferligoj, A., Doreian, P.: Indirect Blockmodeling of 3-Way Networks. In: Brito P., Cucumel G., Bertrand P., de Carvalho F. (eds) Selected Contributions in Data Analysis and Classification. Studies in Classification, Data Analysis, and Knowledge Organization, pp. 151–159. Springer, Berlin, Heidelberg (2007)
4. Blöcker, C., Rosvall, M.: Mapping flows on bipartite networks. Phys. Rev. E, **102**, 052305 (2020)
5. Blondel, V. D., Guillaume, J. L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech.-Theory E, **10**, P10008 (2008)
6. Borgatti, S. P., Everett, M. G.: Regular blockmodels of multiway, multimode matrices. Soc. Networks, **14**, 91–120 (1992)
7. Columbu, S., Porcu, M., Primerano, I., Sulis, I., Vitale, M.P.: Geography of Italian student mobility: A network analysis approach. Socio. Econ. Plan. Sci. **73**, 100918 (2021)
8. Columbu, S., Porcu, M., Primerano, I., Sulis, I., Vitale, M. P.: Analysing the determinants of Italian university student mobility pathways. Genus, **77**, 34 (2021)
9. Edler, D., Bohlin, L., Rosvall, M.: Mapping higher-order network flows in memory and multilayer networks with infomap. Algorithms, **10**, 112 (2017)
10. Everett, M. G., Borgatti, S.: Partitioning multimode networks. In: Doreian, P., Batagelj, V., Ferligoj, A. (eds.) Advances in Network Clustering and Blockmodeling, pp. 251-265, John Wiley & Sons, Hoboken, USA (2020)

11. Fararo, T. J., Doreian, P.: Tripartite structural analysis: Generalizing the Breiger-Wilson formalism. Soc. Networks, **6**, 141–175 (1984)
12. Genova, V. G., Tumminello, M., Aiello, F., Attanasio, M.: Student mobility in higher education: Sicilian outflow network and chain migrations. Electronic Journal of Applied Statistical Analysis, **12**, 774–800 (2019)
13. Genova, V. G., Tumminello, M., Aiello, F., Attanasio, M.: A network analysis of student mobility patterns from high school to master's. Stat. Method. Appl., **30**, 1445–1464 (2021)
14. Ikematsu, K., Murata, T.: A fast method for detecting communities from tripartite networks. In: International Conference on Social Informatics, pp. 192-205. Springer, Cham (2013)
15. Melamed, D., Breiger, R. L., West, A. J.: Community structure in multi-mode networks: Applying an eigenspectrum approach. Connections, **33**, 18–23 (2013)
16. Murata, T.: Detecting communities from tripartite networks. In: Proceedings of the 19th international conference on world wide web, pp. 1159-1160. (2010)
17. Neubauer, N., Obermayer, K.: Tripartite community structure in social bookmarking data. New Rev. Hypermedia M., **17**, 267-294 (2011)
18. Newman, M. E., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E, **69**, 026113 (2004)
19. Newman, M. E.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences, **103**, 8577-8582 (2006)
20. Rizzi, L., Grassetti, L. Attanasio, M.: Moving from North to North: how are the students' university flows? Genus **77**, 1–22 (2021)
21. Santelli, F., Scolorato, C., Ragozini, G.: On the determinants of student mobility in an interregional perspective: A focus on Campania region. Statistica Applicata - Italian Journal of Applied Statistics, **31**, 119–142 (2019)
22. Santelli, F., Ragozini, G., Vitale, M. P.: Assessing the effects of local contexts on the mobility choices of university students in Campania region in Italy. Genus, **78**, 5 (2022)




# Clustering Brain Connectomes Through a Density-peak Approach

Riccardo Giubilei

**Abstract** The density-peak (DP) algorithm is a mode-based clustering method that identifies cluster centers as data points being surrounded by neighbors with lower density and far away from points with higher density. Since its introduction in 2014, DP has reaped considerable success for its favorable properties. A striking advantage is that it does not require data to be embedded in vector spaces, potentially enabling applications to arbitrary data types. In this work, we propose improvements to overcome two main limitations of the original DP approach, i.e., the unstable density estimation and the absence of an automatic procedure for selecting cluster centers. Then, we apply the resulting method to the increasingly important task of graph clustering, here intended as gathering together similar graphs. Potential implications include grouping similar brain networks for ability assessment or disease prevention, as well as clustering different snapshots of the same network evolving over time to identify similar patterns or abrupt changes. We test our method in an empirical analysis whose goal is clustering brain connectomes to distinguish between patients affected by schizophrenia and healthy controls. Results show that, in the specific analysis, our method outperforms many existing competitors for graph clustering.

**Keywords:** nonparametric statistics, mode-based clustering, networks, graph clustering, kernel density estimation

## 1 Introduction

Clustering is the task of grouping elements from a set in such a way that elements in the same group, also defined as *cluster*, are in some sense similar to each other, and dissimilar to those from other groups. Mode-based clustering is a nonparametric approach that works by first estimating the density, and then identifying in some

Riccardo Giubilei (✉)
Luiss Guido Carli, Rome, Italy, e-mail: rgiubilei@luiss.it



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_18

way its modes and the corresponding clusters. An effective method to find modes and clusters is through the density-peak (DP) algorithm [12], which has drawn considerable attention since its introduction in 2014. One of the striking advantages of DP is that it does not require data to be embedded in vector spaces, implying that it can be applied to arbitrary data types, provided that a proper distance is defined. In this work, we focus on its application to clustering graph-structured data objects.

The expression *graph clustering* can refer either to *within-graph clustering* or to *between-graph clustering*. In the first case, the elements to be grouped are the vertices of a single graph; in the second, the objects are distinct graphs. Here, *graph clustering* is intended as *between-graph clustering*. Between-graph clustering is an emerging but increasingly important task due to the growing need of analyzing and comparing multiple graphs [10, 4]. Potential applications include clustering: brain networks of different people for ability assessment, disease prevention, or disease evaluation; online social ego networks of different users to find people with similar social structures; different snapshots of the same network evolving over time to identify similar patterns, cycles, or abrupt changes.

Heretofore, the task of between-graph clustering has not been exhaustively investigated in the literature, implying a substantial lack of well-established methods. The goal of this work is to improve and adapt the density-peak algorithm to define a fairly general method for between-graph clustering. For validation and comparison purposes, the resulting procedure and its main competitors are applied to grouping brain connectomes of different people to distinguish between patients affected by schizophrenia and healthy controls.

## 2 Related Work

Existing techniques for between-graph clustering can be divided into two main categories: 1) transforming graph-structured data objects into Euclidean feature vectors in order to apply standard clustering algorithms; 2) using the distances between the original graphs in distance-based clustering methods.

The most common technique within the first category is the use of classical clustering techniques on the vectorized adjacency matrices [10]. Nonetheless, more advanced numerical summaries have been proposed to better capture the structural properties of the graphs and to decrease feature dimensionality. Examples include: shell distribution [1], traces of powers of the adjacency matrix [10], and graph embeddings such as *graph2vec* [11]; see [4] for a longer list. Techniques from the first category share an important drawback: the transformation into feature vectors necessarily implies loss of information. Additionally, methods for extracting features may be domain-specific.

The second category features Partitioning Around Medoids (PAM) [7], or k-medoids, which finds representative observations by iteratively minimizing a cost function based on the distances between data objects, and assigns other observations to the closest medoid. PAM's main limitations are that it requires the number of

clusters in advance and can only identify convex-shaped groups. Density-based spatial clustering of applications with noise [3], or DBSCAN, overcomes these two constraints by computing the density of data points starting from their distances, and defining clusters as samples of high density that are close to each other (and surrounded by areas of lower density). A similar approach is the DP, which is described in greater detail in Section 3.1. Alternatively, hierarchical clustering can be applied to distances between graphs, as in [13], where a spectral Laplacian-based distance is proposed and used. Finally, $k$-groups [8] is a clustering technique within the Energy Statistics framework [14] where the goal is minimizing the total within-cluster Energy distance, which is computed starting from the distances between original observations.

## 3 Methods

In this section, we first describe the original DP approach; then, we introduce the DP-KDE method, which is partly named after Kernel Density Estimation; finally, we discuss how to employ it for graph clustering.

### 3.1 Original DP

The density-peak algorithm [12] is based on a simple idea: since cluster centers are identified as the distribution's modes, they must be 1) surrounded by neighbors with lower density, and 2) at a relatively large distance from points with higher density. Consequently, two quantities are computed for each observation $x_i$: the local density $\rho_i$, and the minimum distance $\delta_i$ from other data points with higher density. The local density $\rho_i$ of $x_i$ is defined as:

$$\rho_i = \sum_j I_{(d_{ij}-d_c)}, \tag{1}$$

where $I_{(\cdot)}$ is the indicator function, $d_{ij} = d(x_i, x_j)$ is the distance between $x_i$ and $x_j$, and $d_c$ is a cutoff distance. In simple terms, $\rho_i$ is the number of points that are closer than $d_c$ to $x_i$. The DP algorithm is robust with respect to $d_c$, at least with large datasets [12]. Once the density is computed, the definition of the minimum distance $\delta_i$ between point $x_i$ and any other point $x_j$ with higher density is straightforward:

$$\delta_i = \min_{j:\rho_j>\rho_i} (d_{ij}). \tag{2}$$

By convention, the point with highest density has $\delta_i = max_j(d_{ij})$. The interpretation of $\delta_i$ reflects the algorithm's core idea: data points that are not local or global maxima have their $\delta_i$ constrained by other points within the same cluster, hence cluster centers have large values of $\delta_i$. However, this is not sufficient: they also need to have a large $\rho_i$

because otherwise the point could be merely distant from any other. After identifying cluster centers, other observations are assigned to the same cluster as their nearest neighbor of higher density.

The density-peak algorithm has many favorable properties: it manages to detect nonspherical clusters, it does not require the number of clusters in advance or data to be embedded in vector spaces, it is computationally fast because it does not maximize explicitly each data point's density field and it performs cluster assignment in a single step, it estimates a clear population quantity, and it has only one tuning parameter (the cutoff distance $d_c$).

## 3.2 DP-KDE

The density-peak approach also has drawbacks. Over the last few years, many articles have proposed improvements to overcome two main critical points: the unstable density estimation and the absence of an automatic procedure for selecting cluster centers. In this work, we explicitly tackle these two aspects.

The unstable density estimation induced by Equation (1) has been widely shown [9, 16, 15]. Although many solutions have been proposed, we espouse the research line suggesting the use of Kernel Density Estimation (KDE) to compute $\rho_i$ [9, 15]:

$$\rho_i = \frac{1}{nh} \sum_{j=1}^{n} K\left(\frac{x_i - x_j}{h}\right). \tag{3}$$

In Equation (3), $h$ is the *bandwidth*, which is a smoothing parameter, and $K(\cdot)$ is the *kernel*, which is a non-negative function weighting the contribution of each data point to the density of the $i$-th observation. We use the Epanechnikov kernel, which is normalized, symmetric, and optimal in the Mean Square Error sense [2]:

$$K(u) = \begin{cases} 3/4(1-u^2), & |u| \leq 1 \\ 0, & |u| > 1 \end{cases}. \tag{4}$$

Equation (4) implies a null contribution of observation $j$ to the $i$-th density whenever $|(x_i - x_j)/h| \geq 1$, while, in the opposite case, it results in a positive weight depending quadratically on $(x_i - x_j)/h$. Consequently, $h$ may be regarded as the cutoff distance for the DP-KDE method.

The automatic selection of cluster centers involves many aspects: the cutoff distance, the number of clusters, and which data points to select. In the following, we use a cutoff distance $h$ such that the average number of neighbors is between 1 and 2% of the sample size, as suggested by [12]. The number of clusters $k$ is here considered as a given parameter, leaving the search for its optimal value for future work. Finally, the method for selecting data points as cluster centers is obtained refining an intuition contained in [12], where candidates are observations with sufficiently large values of $\gamma_i = \delta_i \rho_i$. However, this quantity has two major drawbacks: first, if

$\delta_i$ and $\rho_i$ are not defined over the same scale, results could be misleading; second, it implicitly assumes that $\delta_i$ and $\rho_i$ shall be given the same weight in the decision. We overcome these two limitations by first normalizing both $\delta_i$ and $\rho_i$ between 0 and 1, and then giving them different weights that are based on their informativeness. We measure the latter using the Gini coefficient of the two (normalized) quantities, under the assumption that the least concentrated distribution between the two is the most informative. Specifically, each observation is given a measure of importance that is defined as:

$$\gamma_i^G = \delta_{01,i}^{G(\delta_{01})} \rho_{01,i}^{G(\rho_{01})}, \tag{5}$$

where $\delta_{01}$ and $\rho_{01}$ are the normalized versions of $\delta$ and $\rho$ respectively, $\delta_{01,i}$ and $\rho_{01,i}$ are the corresponding $i$-th values, and $G(x)$ denotes the Gini coefficient of $x$. Then, the selected cluster centers are the top $k$ observations in terms of $\gamma_i^G$. Assigning observations to the same cluster as their nearest neighbor of higher density is what concludes the DP-KDE method.

### 3.3 Graph Clustering

A *graph* is a mathematical object composed of a collection of *vertices* linked by *edges* between them. Formally, a graph is denoted with $\mathcal{G} = (V, E)$, where $V$ is the set of vertices and $E$ is the set of edges. If $e \in E$ joins vertices $u, v \in V$, i.e., $e = \{u, v\}$, then $u$ and $v$ are *adjacent* or *neighbors*. The number of edges incident with any vertex $v$ is the *degree* of $v$. Each edge $e \in E$ is represented through a numerical value $w_e$ called *edge weight*: if weights are equal to 1 for all and only the existent edges, and 0 for the others, $\mathcal{G}$ is *unweighted*; when existent edges have real-valued weights, $\mathcal{G}$ is *weighted*. If $w_{\{u,v\}} = w_{\{v,u\}}$ for all $u, v \in V$, the graph $\mathcal{G}$ is *undirected*; otherwise, it is *directed*. The entire information about $\mathcal{G}$'s connectivity is stored in a $|V| \times |V|$ *adjacency matrix* $\mathbf{A}$ whose generic entry in the $u$-th row and $v$-th column is $w_e$, where $e = \{u, v\}$ and $u, v \in V$.

The DP-KDE method can be used for graph clustering if a proper distance between graphs is defined. In this work, we employ the Edge Difference Distance [6], which is defined as the Frobenius norm of the difference between the two graphs' adjacency matrices. The choice is motivated by many factors: a flexible definition that can be directly applied also to directed and weighted graphs, the reasonable results it yields when node correspondence is a concern, and its limited computational time complexity. Formally, the Edge Difference Distance between two graphs $x_i$ and $x_j$ is defined as:

$$d_{ED}(x_i, x_j) = ||\mathbf{A}^i - \mathbf{A}^j||_F := \sqrt{\sum_p \sum_q |A_{pq}^i - A_{pq}^j|^2}, \tag{6}$$

where $\mathbf{A}^i$ and $\mathbf{A}^j$ are the adjacency matrices of $x_i$ and $x_j$ respectively, and $||\cdot||_F$ denotes the Frobenius norm.

Consequently, the two fundamental quantities of the DP-KDE method are computed in the following as:

$$\rho_i = \sum_{j=1}^{n} K\left(\frac{d_{ED}(x_i, x_j)}{h}\right), \tag{7}$$

where $K(\cdot)$ is the Epanechnikov kernel defined in Equation (4) and the normalizing constant is omitted because we are simply interested in the ranking between the densities, and:

$$\delta_i = \min_{j:\rho_j > \rho_i} (d_{ED}(x_i, x_j)). \tag{8}$$

Finally, cluster centers are selected as the observations with the largest values of $\gamma_i^G$, as defined in Equation (5), and other observations are assigned to the same cluster as their nearest neighbor in terms of $\delta_i$.

## 4 Empirical Analysis

The DP-KDE method for graph clustering is employed in an unsupervised empirical analysis where the ground truth is known, and its performance is compared in terms of accuracy both with natural competitors and with a method treating the problem as supervised. The ultimate goal is clustering brain connectomes, one for each individual, correctly distinguishing between patients affected by schizophrenia (SZ) and healthy controls.

We use publicly available[1] data from a recent study [5] whose aim is finding relevant links between Regions of Interest (ROIs) for predicting schizophrenia from multimodal brain connectivity data. The cohort is composed of 27 schizophrenic patients and 27 age-matched healthy participants acting as control subjects. In the current work, we focus only on this cohort's functional Magnetic Resonance Imaging (fMRI) connectomes. Functional connectivity matrices have been computed starting from fMRI scans, treating them as time series, and computing Pearson's correlation coefficient between time series for distinct ROIs. The resulting matrices are weighted, undirected, and made of 83 nodes.

The aforementioned study [5] treats every functional connectivity matrix as a single multivariate realization of $(83 \cdot 82)/2 = 3403$ numeric variables, each representing a connection between two of the 83 ROIs. They reduce feature dimensionality by performing Recursive Feature Elimination based on Support Vector Machines (SVM-RFE), and tackle the classification problem as supervised using 20 repetitions of nested 5-fold cross-validation. When using only functional connectivity data, they achieve an average accuracy of 68.28%[2] over the resulting 100 test sets.

[1] `https://doi.org/10.5281/zenodo.3758534.`

[2] This exact figure is not included in the article, but the analysis is fully reproducible since the authors made their source code available at `https://github.com/leoguti85/BiomarkersSCHZ`.

The approach we adopt in this work is rather different. First, graphs are analyzed in their original form, without any simplification to numeric variables, resulting in only one graph-structured variable. Observations are 54, each one representing the functional connectome of a different individual. We tackle the problem with an unsupervised classification approach seeking to cluster connectomes into two groups: schizophrenic and healthy. To this end, we use the DP-KDE method for graph clustering described in Section 3.3. Starting from the 54 connectomes, each observation's local density $\rho_i$ and minimum distance $\delta_i$ are computed using Equations (7) and (8), respectively. The centers of the two clusters are those whose $\gamma_i^G$ is largest. Then, other observations are assigned to the same cluster as their nearest neighbor of higher density. Finally, the clustering performance is evaluated by comparing the algorithm's assignment to the ground truth. The DP-KDE method achieves an accuracy of 70.37%, which is more than 2% higher than the one obtained in [5].

Table 1 includes the performance in terms of accuracy of both the DP-KDE and the SVM-RFE methods, as well as that of other graph clustering competitors. Specifically, we consider: the classical DP algorithm on the original data objects, with the same cutoff distance as in DP-KDE and manually selected cluster centers; k-means clustering on the 3403 numeric variables obtained from vectorizing the adjacency matrices; DBSCAN on the original data objects, with parameters $\varepsilon = 20.2$ and 15 as the minimum number of points required to form a dense region; PAM and $k$-groups on the original data objects. In all these cases, the number of clusters has been kept fixed to $k = 2$. The method that yields the best accuracy in the specific problem is the DP-KDE.

**Table 1** Accuracy for DP-KDE and some of its possible competitors.

| *Method* | DP-KDE | SVM-RFE | DP | k-means | DBSCAN | PAM | $k$-groups |
|---|---|---|---|---|---|---|---|
| *Accuracy* | 70.37 | 68.28 | 62.96 | 62.96 | 61.11 | 62.96 | 62.96 |

## 5 Concluding Remarks

After explaining the importance of graph clustering and briefly reviewing some existing methods to perform this task, we have considered the possibility of adopting a density-peak approach. We have improved the original DP algorithm by using a more robust definition of the density $\rho_i$, and by automatically selecting cluster centers based on the quantity $\gamma_i^G$ we have introduced. We have also selected a proper distance between graphs, namely, the Edge Difference Distance. Finally, we have used the resulting method in an empirical analysis with the goal of clustering brain connectomes to distinguish between schizophrenic patients and healthy controls. Our method outperforms another one treating the specific task as supervised, and it is by far the best one with respect to many graph clustering competitors.

An initial idea for future work is the search for the optimal number of clusters. This may be achieved either by fixing a threshold for $\gamma_i^G$ or by selecting all the data points after the largest increase in terms of $\gamma_i^G$. Also the cutoff distance could be tuned, possibly maximizing in some way the dispersion of points in the bivariate distribution of $\rho$ and $\delta$. Then, the DP-KDE method needs to be extended beyond the univariate case. Finally, other distances between graphs could be considered to better reflect alternative application-specific needs, e.g., when graphs are not defined over the same set of nodes.

**Acknowledgements** The author would like to thank Pierfancesco Alaimo Di Loro, Federico Carlini, Marco Perone Pacifico, and Marco Scarsini for several engaging and stimulating discussions.

## References

1. Carmi, S., Havlin, S., Kirkpatrick, S., Shavitt, Y., Shir, E.: A model of Internet topology using k-shell decomposition. Proc. Natl. Acad. Sci. **104**, 11150–11154 (2007)
2. Epanechnikov, V.: Non-parametric estimation of a multivariate probability density. Theory Probab. Its Appl. **14**, 153–158 (1969)
3. Ester, M., Kriegel, H., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. KDD-96 **34** 226–231 (1996)
4. Gutiérrez-Gómez, L., Delvenne, J.: Unsupervised network embeddings with node identity awareness. Appl. Netw. Sci. **4**, 1–21 (2019)
5. Gutiérrez-Gómez, L., Vohryzek, J., Chiêm, B., Baumann, P., Conus, P., Do Cuenod, K., Hagmann, P., Delvenne, J.: Stable biomarker identification for predicting schizophrenia in the human connectome. NeuroImage Clin. **27** 102316 (2020)
6. Hammond, D., Gur, Y., Johnson, C.: Graph diffusion distance: A difference measure for weighted graphs based on the graph Laplacian exponential kernel. IEEE GlobalSIP 2013, pp. 419–422 (2013)
7. Kaufmann, L., Rousseeuw, P.: Clustering by means of medoids. Proc. of the Statistical Data Analysis based on the L1 Norm Conference, Neuchatel, Switzerland, pp. 405–416 (1987)
8. Li, S., Rizzo, M.: K-groups: A generalization of k-means clustering. ArXiv Preprint ArXiv:1711.04359 (2017)
9. Mehmood, R., Zhang, G., Bie, R., Dawood, H., Ahmad, H.: Clustering by fast search and find of density peaks via heat diffusion. Neurocomputing. **208**, 210–217 (2016)
10. Mukherjee, S., Sarkar, P., Lin, L.: On clustering network-valued data. NIPS2017, pp. 7074–7084 (2017)
11. Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., Jaiswal, S.: graph2vec: Learning distributed representations of graphs. ArXiv Preprint ArXiv:1707.05005 (2017)
12. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**, 1492–1496 (2014)
13. Shimada, Y., Hirata, Y., Ikeguchi, T., Aihara, K.: Graph distance for complex networks. Sci. Rep. **6**, 1–6 (2016)
14. Székely, G., Rizzo, M.: The energy of data. Annu. Rev. Stat. Appl. **4**, 447–479 (2017)
15. Wang, X., Xu, Y.: Fast clustering using adaptive density peak detection. Stat. Methods Med. Res. **26**, 2800–2811 (2017)
16. Xie, J., Gao, H., Xie, W., Liu, X., Grant, P.: Robust clustering by detecting density peaks and assigning points based on fuzzy weighted K-nearest neighbors. Inf. Sci. **354**, 19–40 (2016)

# Similarity Forest for Time Series Classification

Tomasz Górecki, Maciej Łuczak, and Paweł Piasecki

**Abstract** The idea of similarity forest comes from Sathe and Aggarwal [19] and is derived from random forest. Random forests, during already 20 years of existence, proved to be one of the most excellent methods, showing top performance across a vast array of domains, preserving simplicity, time efficiency, still being interpretable at the same time. However, its usage is limited to multidimensional data. Similarity forest does not require such representation – it is only needed to compute similarities between observations. Thus, it may be applied to data, for which multidimensional representation is not available. In this paper, we propose the implementation of similarity forest for time series classification. We investigate 2 distance measures: Euclidean and dynamic time warping (DTW) as the underlying measure for the algorithm. We compare the performance of similarity forest with 1-nearest neighbor and random forest on the UCR (University of California, Riverside) benchmark database. We show that similarity forest with DTW, taking into account mean ranks, outperforms other classifiers. The comparison is enriched with statistical analysis.

**Keywords:** time series, time series classification, random forest, similarity forest

Tomasz Górecki (✉)
Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Uniwersytetu Poznańskiego 4, Poznań, Poland, e-mail: tomasz.gorecki@amu.edu.pl

Maciej Łuczak
Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Uniwersytetu Poznańskiego 4, Poznań, Poland, e-mail: maciej.luczak@amu.edu.pl

Paweł Piasecki
Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Uniwersytetu Poznańskiego 4, Poznań, Poland, e-mail: pawel.piasecki@amu.edu.pl



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_19

## 1 Introduction

Time series classification is a well-developing research field, that gained much attention from researchers and business during the last two decades apparently by the fact that more and more data around us seems to be located in the time domain – and thus fulfilling the definition of time series. Predictive maintenance [18], quality monitoring [22], stock market analysis [20] or sales forecasting [17] are just a few exemplar nowadays problems where time series are indeed present. The reason why we usually apply to time series different methods from regular (non-time series) data is the fact, that time series are ordered in time (or some other space with ordering) and it is beneficial to use the information conveyed by the ordering.

In recent years, one could observe many advances on the field of time series classification. In 2017, Bagnall et al. presented a comprehensive comparison of time series classification algorithms [2], showing that despite there are dozens of far more complex methods, 1-Nearest Neighbour (1NN) [6, 11] coupled with DTW [3] distance constitutes a good baseline. In fact, it has been outperformed by several classifiers, with Collective of Transformation Ensembles (COTE) [1] as the most efficient one. Furthermore, COTE was extended with Hierarchical Vote system, first to HIVE-COTE [13] and then finally to HIVE-COTE 2.0 [15] – a current state of the art classifier for time series. In general, the success of COTE-family classifiers is based on the observation, that in the case of time series it is highly beneficial to use different data representations. For example, HIVE-COTE 1.0 utilizes five ensembles based on different data transformation domains. However, a common criticism of such an approach is its time complexity. In the case of HIVE-COTE, it equals $O(n^2 l^4)$, where $n$ is a number of observations and $l$ is a length of series. Another drawback, especially significant for practitioners is the complex structure of the model ensembles that makes it hard to use HIVE-COTE without spending a decent amount of time studying its components beforehand.

As an alternative to such complex models may be trying to achieve possibly slightly worse performance in favour of model simplicity and reduced computation time. A group of classifiers that seems to hold a great potential are those inspired by Random Forest (RF) [4]. This already 20-years old algorithm remains in the classifiers' forefront, showing extremely good performance and robustness across multiple domains. Fernandez-Delgado et al. [10] performed a comparison of 179 classifiers on 121 non-time series data sets originated from UCI Machine Learning Repository [9], concluding RF to be the most accurate one. Unfortunately, the usage of RF is essentially limited to multidimensional data, as they sample features from original space while creating each node of decision trees.

In this paper, we propose a method for extending RF to work with time series using similarity forests (SF). We significantly extend the applicability of the RF method to time series data. Furthermore, the approach even outperforms traditional classifiers for time series. The main goal of this paper is to enrich the pool of time series classifiers by Similarity Forest for time series classification. SF was initially proposed by Sathe and Aggarwal in 2017 [19], as a method extending Random Forests to deal with arbitrary data sets, provided that we are able to compute similarities

between observations. We would like to implement and tune the method to time series data. We investigate the performance of the model using two distance measures (the algorithm's hyper-parameter): Euclidean and DTW. Also, a comparison with other selected time series classifiers is provided. We compare its performance against 1NN-ED, 1NN-DTW and RF.

The rest of the paper is structured as follows. In Section 2, we provide details of similarity forest and we give more details about random forests. Additionally, we discuss how similarity forest is related to random forest. Section 3 describes data sets that we used and the comparison methodology. The corresponding results are presented in Section 4. Finally, in Section 5 we give a brief summary of our research.

## 2 Classification Methods Used in Comparison

In the paper, we compare the standard random forest and the similarity forest with the distance measure: ED (Euclidian distance) and DTW (dynamic time warping distance). As benchmark methods, we also use the nearest neighbor method (1NN) with distance measure ED and DTW. 1NN-ED and 1NN-DTW are very common classification methods for time series classification [2]. For a review of these methods refer to [14].

### 2.1 General Method of Random Forest Construction

Random forest consists of random decision trees. For the construction of a random forest we usually take decision trees as simple as possible — without special criteria for stopping, pruning, etc.

When building a decision tree, we start at a node $N$, which contains the entire data set (bootstrap sample). Then, according to an established criterion, we split the node $N$ into two subnodes $N_1$ and $N_2$. In each subnode there are data subsets of the data set from node $N$. We make this split in a way that is optimal for a given split method. In each node, we write down how the split occurred. Then, proceeding recursively, we split next nodes into subnodes until the stop criterion occurs. In our case we take the simplest such criterion, namely we stop the split of a given node when only elements of the same class are included in a node. We call such a node a leaf and assign it a label which elements of the node (leaf) have.

Having built a tree, we can now use it (in the testing phase) to classify a new observation. We pass this observation through the trained tree — starting from the node $N$ selecting each time one of the subnodes, according to the condition stored in the node. We do this until we reach one of the leaves, and then we assign the test observation to the class of the leaf.

Now, constructing the random forest, we collect a certain number of decision trees, train them independently according to the above method and, in the test phase,

use each of the trees to test new observation. Thus, each tree assigns a label to the test observation. The final label (for the entire forest) we construct by voting, we choose the most frequently appearing label among the decision trees.

## 2.2 Classical Random Forest

To create a (classical) random tree and a random forest [4], we proceed as described above using the following node split method:

To obtain split conditions for a single tree, we select randomly a certain number of features ($\sqrt{k}$ for classification, $k$ — number of features), and for each feature we create a feature vector (column, variable) made of all elements of the data set (bootstrap sample). For a given feature vector (variable), we determine a threshold vector. First, we sort values of the feature vector (uniquely — without repeating values). Let us name this sorted feature vector as $\boldsymbol{V} = (V_1, V_2, \dots)$. Then we take the values of the split as means of successive values of the vector $\boldsymbol{V}$:

$$v_i = \frac{V_i + V_{i+1}}{2} \quad i = 1, 2, \dots . \tag{1}$$

Each splitting value divides the data set in node $N$ into two subsets — the one (left) in which we have elements with feature values smaller than $v_i$ and the second (right) with other elements. Then we check the quality of such a split.

The splitting point is chosen such that it minimizes the Gini index of the children nodes. If $p_1, p_2 \dots p_c$ are the fractions of data points belonging to the $c$ different classes in node $N$, then the Gini index of that node is given by: $G(N) = 1 - \sum_{i=1}^{c} p_i^2$.

Then, if the node $N$ is split into two children nodes $N_1$ and $N_2$, with $n_1$ and $n_2$ points, respectively, the Gini quality of the children nodes is given by:

$$GQ(N_1, N_2) = \frac{n_1 G(N_1) + n_2 G(N_2)}{n_1 + n_2}.$$

Quality of the split is given by: $GQ(N) = G(N) - GQ(N_1, N_2)$.

## 2.3 Similarity Forest

The similarity forest [19] differs from the ordinary (classical) random forest only in the way we split nodes of trees. Instead of selecting a certain number of features, we select randomly a pair of elements $e_1$, $e_2$ with different classes. Then, for each element $e$ of the subset of elements in a given node, we calculate the difference of the squared distances to the elements $e_1$ and $e_2$:

$$w(e) = d(e, e_1)^2 - d(e, e_2)^2,$$

where $d$ is any fixed distance measure of the elements of the data set. We sort the vector $\boldsymbol{w}$ uniquely (without duplicates) creating the vector $\boldsymbol{V}$ and continue as for the classical decision tree. We calculate values of the split $v_i$ (1), calculate the quality of the node split using the Gini index (2.2) and choose the best split. In the learning phase, we remember in each node how the optimal split occurred (elements $e_1$, $e_2$, $w(e)$). In the learning phase, in each node we write down the optimal split — elements $e_1$, $e_2$, and value $w(e)$).

### 2.4 Random Forest vs Similarity Forest

The difference between a classical random tree and a similarity tree is that instead of selecting $\sqrt{k}$ of the features, we select only one pair of elements $e_1$, $e_2$. Generally, we have much fewer possible node splits, which has a very good effect on the computation time.

The second important difference is that in the similarity tree we use any distance measure between elements of the data set. Therefore, we can use distance measures specific to a data set. For example, for time series we can use the DTW distance, much better suited for calculating the distance between time series, instead of the Euclidean distance.

## 3 Experimental Setup

We investigated the performance of similarity forest on UCR time series repository [7] (128 data sets). The latest update of the UCR database introduced several data sets with missing observations and uneven sample lengths. However, the repository includes a standardized version of the database without these impediments, and that is the version we used.

All data sets are split into a training and testing subset, and all parameter optimization is conducted on the training set only. We combined both parts and in the next step, we used 100 random train/test splits.

## 4 Results

The error rates for each classifier can be found on the accompanying website[1]. In the Table 1 we show a short summary of results, including a number of wins (draw is not counted as a win) and mean ranks. Taking into account mean ranks, SF-DTW is the best classifier, sightly ahead of RF (mean ranks correspondingly equal 2.64

[1] https://github.com/ppias/similarity_forest_for_tsc

**Table 1** Number of wins (clearly wins) and mean ranks for examined methods.

| Method | 1NN-ED | 1NN-DTW | RF | SF-ED | SF-DTW |
|---|---|---|---|---|---|
| Wins | 12 | 28 | **38** | 10 | 31 |
| Mean rank | 3.59 | 2.89 | 2.69 | 3.19 | **2.64** |

and 2.89). Figure 1 demonstrates comparison of error rates and ranks for classifiers. These results lead to a conclusion that even though there is no clear winner, the top efficient distances are dominated by RF and SF-based classifiers. Figure 2 shows scatter plots of errors for pairs of classifiers.

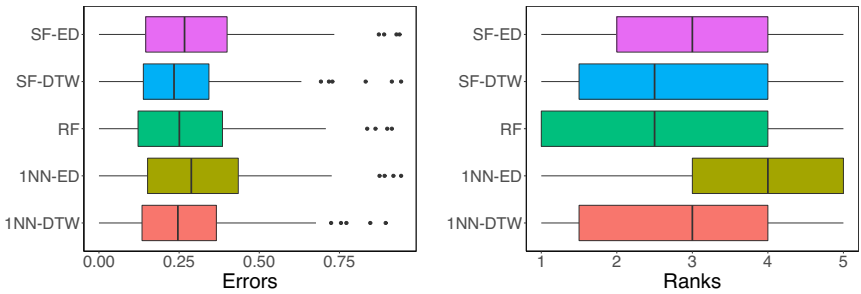


**Fig. 1** Comparison of error rates and ranks.

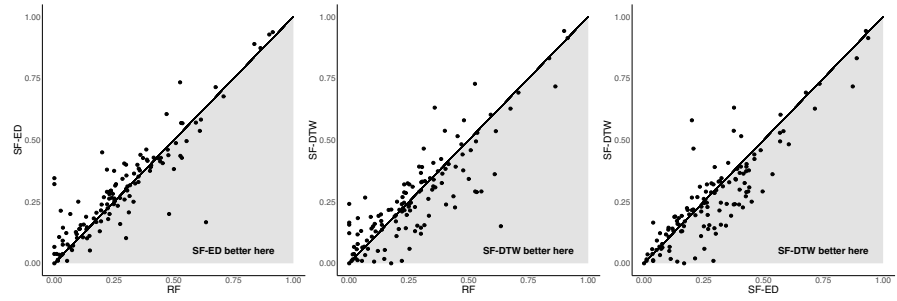


**Fig. 2** Comparison of error rates.

To identify differences between the classifiers, we present a detailed statistical comparison. In the beginning, we test the null hypothesis that all classifiers perform the same and the observed differences are merely random. The Friedman test with Iman & Davenport extension is probably the most popular omnibus test, and it is usually a good choice when comparing different classifiers [12]. The $p$-value from this test is equal to 0. The obtained $p$-value indicates that we can safely reject the null hypothesis that all the algorithms perform the same. We can therefore proceed

with the post-hoc tests in order to detect significant pairwise differences among all of the classifiers.

Demšar [8] proposes the use of the Nemenyi's test [16] that compares all the algorithms pair-wise. For a significance level $\alpha$ the test determines the critical difference (CD). If the difference between the average ranking of two algorithms is greater than CD the null hypothesis that the algorithms have the same performance is rejected. Additionally, Demšar [8] creates a plot to visually check the differences, the CD plot. In the plot, those algorithms that are not joined by a line can be regarded as different.

In our case, with a significance of $\alpha = 0.05$ any two algorithms with a difference in the mean rank above 0.54 will be regarded as non equal (Figure 3). We can see that we have three groups of methods. In the first group we have SF-DTW, RF and 1NN-DTW, in the second we have RF, 1NN-DTW and SF-ED and in the last group we have SF-ED and 1NN-ED. Unfortunately, groups are not disjoint. The first group is the group with the highest accuracy of classification. Hence, SF-DTW does not statistically outperform RF. However, we can recommend it over RF because of statistically the same quality and much better computational properties.

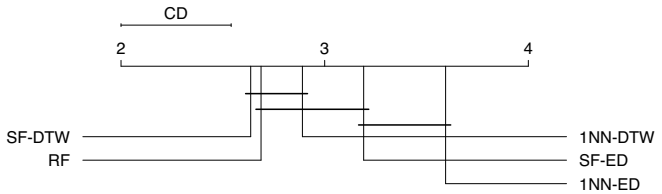


**Fig. 3** Critical difference plot.

## 5 Conclusions

Our contribution is to implement similarity forest for time series classification using two distance measures: Euclidean and DTW. Comparison based on the recently updated UCR data repository (128 data sets) was presented. We showed that SF-DTW outperforms other classifiers, including 1NN-DTW which has been considered as a strong baseline hard to beat for years. The statistical comparison showed, that RF and SF-DTW are statistically insignificantly different, however taking into account mean ranks the latter one is the best one.

There are many improvements that could be applied to the implementation that we propose. For example, we could test other distance measures such as LCSS [21] or ERP [5] that were successfully used in time series tasks. Another idea could be to investigate the usage of boosting algorithm.

**Acknowledgements** The research work was supported by grant No. 2018/31/N/ST6/01209 of the National Science Centre.

## References


1. Bagnall, A., Lines, J., Hills, J., Bostrom A.: Time-series classification with COTE: The collective of transformation-based ensembles. IEEE Trans. on Knowl. and Data Eng. **27**, 2522–2535 (2015)
2. Bagnall, A., Lines, J., Bostrom, A., Large J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. and Knowl. Discov. **31**, 606–660 (2017)
3. Berndt, D. J., Clifford, J.: Using dynamic time warping to find patterns in time series. Proc. of the 3rd Int. Conf. on Knowl. Discov. and Data Min., pp. 359–370 (1994)
4. Brieman, L.: Random forests. J. Mach. Learn. Arch. **45**, 5–32 (2001)
5. Chen, L., Ng, R.: On the marriage of $L_p$-norms and edit distance. Proc. of the 30th Int. Conf. on Very Large Data Bases **30**, pp. 792–803 (2004)
6. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. on Inf. Theor. **13**, 21–27 (1967)
7. Dau, H.A., Keogh, E., Kamgar, K., Yeh, Chin-Chia M., Zhu, Y.,Gharghabi, S., Ratanamahatana, C.A., Yanping, C., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G., Hexagon-ML: The UCR time series classification archive (2019) `https://www.cs.ucr.edu/\string~eamonn/time\_series\_data\_2018`
8. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. of Mach. Learn. Res. **7**, 1–30 (2006).
9. Du,a D., Graff, C.: UCI Machine Learning Repository. `http://archive.ics.uci.edu/ml`
10. Fernandez-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems?. J. of Mach. Learn. Res. **15**, 3133–3181 (2014)
11. Fix, E, Hodges, J. L.: Discriminatory analysis: nonparametric discrimination, consistency properties. Techn. Rep. **4**, (1951)
12. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental Analysis of Power. Inf. Sci. **180**, 2044–2064 (2010)
13. Lines, J., Taylor S., Bagnall, A.: HIVE-COTE: The hierarchical vote collective of transformation based ensembles for time series classification. IEEE Int. Conf. on Data Min., pp. 1041–1046 (2016)
14. Maharaj, E. A., D'Urso, P., Caiado, J.: Time Series Clustering and Classification. Chapman and Hall/CRC. (2019)
15. Middlehurst, M., Large, J., Flynn, M., Lines, J., Bostrom, A., Bagnall, A.: HIVE-COTE 2.0: a new meta ensemble for time series classification. (2021) `https://arxiv.org/abs/2104.07551`
16. Nemenyi, P.:Distribution-free multiple comparisons. PhD thesis at Princeton University (1963)
17. Pavlyshenko, B. M.: Machine-learning models for sales time series forecasting. Data **4**, 15 (2019)
18. Rastogi, V., Srivastava, S., Mishra, M., Thukral, R.: Predictive maintenance for SME in industry 4.0. 2020 Glob. Smart Ind. Conf., pp. 382–390 (2020)
19. Sathe, S., Aggarwal, C. C.: Similarity forests. Proc. of the 23rd ACM SIGKDD, pp. 395–403 (2017)
20. Tang, J., Chen, X.: Stock market prediction based on historic prices and news titles. Proc. of the 2018 Int. Conf. on Mach. Learn. Techn., pp. 29–34 (2018)
21. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. Proc. 18th Int. Conf. on Data Eng., pp. 673–684 (2002)
22. Wuest, T., Irgens, C., Thoben, K. D.: An approach to quality monitoring in manufacturing using supervised machine learning on product state data. J. of Int. Man. **25**, 1167–1180 (2014)


# Detection of the Biliary Atresia Using Deep Convolutional Neural Networks Based on Statistical Learning Weights via Optimal Similarity and Resampling Methods

Kuniyoshi Hayashi, Eri Hoshino, Mitsuyoshi Suzuki, Erika Nakanishi, Kotomi Sakai, and Masayuki Obatake

**Abstract** Recently, artificial intelligence methods have been applied in several fields, and their usefulness is attracting attention. These methods are techniques that correspond to models using batch and online processes. Because of advances in computational power, as represented by parallel computing, online techniques with several tuning parameters are widely accepted and demonstrate good results. Neural networks are representative online models for prediction and discrimination. Many online methods require large training data to attain sufficient convergence. Thus, online models may not converge effectively for low and noisy training datasets. For such cases, to realize effective learning convergence in online models, we introduce statistical insights into an existing method to set the initial weights of deep convolutional neural networks. Using an optimal similarity and resampling method, we proposed an initial weight configuration approach for neural networks. For a practice example, identification of biliary atresia (a rare disease), we verified the usefulness

Kuniyoshi Hayashi (✉)
Graduate School of Public Health, St. Luke's International University, 3-6 Tsukiji, Chuo-ku, Tokyo, Japan, 104-0045, e-mail: khayashi@slcn.ac.jp

Eri Hoshino · Kotomi Sakai
Research Organization of Science and Technology, Ritsumeikan University, 90-94 Chudoji Awata-cho, Shimogyo Ward, Kyoto, Japan, 600-8815,
e-mail: erihoshino119@gmail.com;koto.sakai1227@gmail.com

Mitsuyoshi Suzuki
Department of Pediatrics, Juntendo University Graduate School of Medicine, 2-1-1 Hongo, Bunkyo-ku, Tokyo, Japan, 113-8421, e-mail: msuzuki@juntendo.ac.jp

Erika Nakanishi
Department of Palliative Nursing, Health Sciences, Tohoku University Graduate School of Medicine, 2-1 Seiryo-machi, Aoba-ku, Sendai, Japan, 980-8575,
e-mail: nakanishi.erika.q3@dc.tohoku.ac.jp

Masayuki Obatake
Department of Pediatric Surgery, Kochi Medical School, 185-1 Kohasu, Oko-cho, Nankoku-shi, Kochi, Japan, 783-8505, e-mail: mobatake@kochi-u.ac.jp



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_20

of the proposed method by comparing existing methods that also set initial weights of neural networks.

**Keywords:** AUC, bootstrap method, leave-one-out cross-validation, projection matrix, rare disease, sensitivity and specificity

## 1 Introduction

The core technique in deep learning corresponds to neural networks, including the convolutional process. Since 2012, deep learning architectures have been frequently used for image classification [1, 2]. More so, deep convolution neural networks (DCNN) are representative nonlinear classification methods for pattern recognition. The DCNN technique is used as a powerful framework for the entirety of image processing [3]. The clinical medicine field presents many opportunities to perform diagnoses using imaging data from patients. Therefore, DCNN techniques are applied to enhance diagnostic quality, e.g., applying a DCNN to a chest X-ray dataset to classify pneumonia [2] and detecting breast cancer [4]. However, DCNN architectures involve many parameters to be learned using training data. Therefore, effective and efficient model development must realize effective learning convergence for such parameters. Notably, it is important to set the initial parameter values to achieve better learning convergence. Furthermore, several methods have been proposed to set initial parameter values in the artificial intelligence (AI) field [5, 6]. However, there are no clear guidelines for determining which existing methods should be used in different situations. Thus, we propose an efficient initial weight approach using existing methods from the viewpoints of optimal similarity and resampling methods. Using a real-world clinical biliary atresia (BA) dataset, we evaluate the performance of the proposed method compared with existing DCNNs. Additionally, we show the usefulness of the proposed method in terms of learning convergence and prediction accuracy.

## 2 Background

BA is a rare disease that occurs in children and is fatal unless treated early. Previous studies have investigated models to identify BA by applying neural networks to patient data [7] and using an ensemble deep learning model to detect BA [8]. However, these models were essentially for use in medical institutions, e.g., hospitals. Generally, certain stool colors in infants and children are highly correlated with BA [9]. In Japan, the maternal and child health handbook includes a stool color card so parents can compare their child's stool color to the information on the card. Such fecal color cards are widely used to detect BA because of their easy accessibility outside the clinical environments. However, this stool color card screening approach for BA is

subjective; thus, accurate and objective diagnoses are not always possible. Previously, we developed a mobile application to classify BA and non-BA stools using baby stool images captured using an iPhone [10]. Here, a batch type classification method was used, i.e., the subspace method, originating from the pattern recognition field. Since BA is a rare disease, the number of events in the case group is generally less. Thus, when we set the explanatory variables of the target observation as the pixel values of a target image, the number of explanatory variables is much higher than the number of observations, especially the disease group. With the subspace method, we can efficiently discriminate such high-dimensional small-sample data. For example, our previous study using the subspace method to classify BA and non-BA stools showed that BA could be discriminated with reasonable accuracy by applying the proposed method to image pixel data of the stool image data captured by a mobile phone [10]. This application was an automated version of the stool color card from the maternal and child health handbook. Unlike previous studies by [7, 8], the application is widely available outside hospital environments. As described previously, DCNNs are useful for image classification, including the automatic classification of stool images for early BA detection.

## 3 Proposed Method

Dimension reduction and discrimination processing can be realized using the subspace method and DCNN techniques. In DCNN, layers based on padding, convolution, and pooling correspond to the dimension reduction functions, and the affine layer performs the discrimination. The primary motivation of this study is to propose a method that properly sets the initial weights of the parameters in a DCNN using statistical approaches. Our secondary motivation is to apply the proposed method to real-world, high-dimensional, and small-sample clinical data.

### 3.1 Description of Related Procedures of the Convolution

For image discrimination in pattern recognition and machine learning fields, the pixel values of the image data are set as the explanatory variables for the target outcome. Here, the data to be classified correspond to a high-dimensional observation. To improve efficiency and demonstrate the feasibility of discriminant processing, the dimensionality must be reduced to a manageable size before classification. The most representative dimensionality reduction method is convolution in pattern recognition and machine learning, which involves padding, convolution, and pooling operations. After converting the input image to a pixel data matrix, the pixel data matrix is surrounded with a numeric value of 0. Using a convolution filter, we reconstruct the pixel data matrix while considering pixel adjacency information. Generally, the size and convolution filter type are parameters that need optimization to realize sufficient

prediction accuracy. However, some representative convolution filters that exhibit good performance are known in the AI field, and we can essentially fix the size and type of the convolution filter. Finally, pooling is performed to reduce the size of the pixel data matrix after convolution. Here, we refer to the sequence of processing from padding to pooling as the layer for feature selection.

### 3.2 Setting Conditions Assumed in This Study

We denote the input pattern matrices comprising numerical pixel values in hue (H), saturation (S), and value (V) as $\mathbf{X}^H (\in \mathbb{R}^{p\times q})$, $\mathbf{X}^S (\in \mathbb{R}^{p\times q})$, and $\mathbf{X}^V (\in \mathbb{R}^{p\times q})$, respectively. First, we performed padding for the input pattern matrices in H, S, and V, respectively, and then, performed a convolution in each signal pattern matrix using a convolution filter. Next, we then applied max pooling to each pattern matrix after convolution. Here, we denote the pattern matrices after the padding, convolution, and max pooling as $\tilde{\mathbf{X}}^H (\in \mathbb{R}^{p'\times q'})$, $\tilde{\mathbf{X}}^S (\in \mathbb{R}^{p'\times q'})$, and $\tilde{\mathbf{X}}^V (\in \mathbb{R}^{p'\times q'})$, respectively, where $p'$ and $q'$ are less than $p$ and $q$. Therefore, we combine the component values of each pattern matrix after padding, convolution, and max pooling into a single pattern matrix by simply adding them together. The combined pattern matrix after applying the feature selection layer is expressed as $\tilde{\mathbf{X}}(\in \mathbb{R}^{p'\times q'})$. Next, we applied convolution and max pooling to the combined pattern matrix $k$ times. Additionally, the input vector after performing the convolution and max pooling $k$ times is denoted by $\mathbf{x}(\in \mathbb{R}^{\ell\times 1})$, and the output of the DCNN and the label vectors are denoted $\mathbf{y}(\in \mathbb{R}^{1\times 1})$ and $\mathbf{t}(\in \mathbb{R}^{1\times 1})$, respectively. In this study, we evaluated the difference between $\mathbf{y}$ and $\mathbf{t}$ according to the mean square error function, i.e., $L(\mathbf{y}, \mathbf{t}) = \frac{1}{\ell} \parallel \mathbf{t} - \mathbf{y} \parallel_2^2$. Here, we consider a simple neural network with three layers. Concretely, between the first and second layers, we perform a linear transformation using $\mathbf{W}_1(\in \mathbb{R}^{2\times \ell})$ and $\mathbf{b}_1(\in \mathbb{R}^{2\times 1})$. Then, a linear transformation is performed using $\mathbf{W}_2(\in \mathbb{R}^{1\times 2})$ and $\mathbf{b}_2(\in \mathbb{R}^{1\times 1})$ between the second and third layers. Next, we defined $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ as $\mathbf{W}_1\mathbf{x} + \mathbf{b}_1$ and $\mathbf{W}_2 f_1(\mathbf{x}) + \mathbf{b}_2$, respectively. Note that we assume $\eta_2$ is a nonlinear transformation between the second and third layers, and we calculated the output $\mathbf{y}$ as $\eta_2(f_2 \circ f_1(\mathbf{x}))$. Generally, $\mathbf{y}$ is calculated as a continuous value. For example, with classification and regression tree methods, we can determine the optimal cutoff point of $\mathbf{y}s$ from a prediction perspective.

### 3.3 General Approach to Update Parameters in CNNs

Here, we denote $f_1(\mathbf{x})$ and $f_2 \circ f_1(\mathbf{x})$ in the previous subsection as $\mathbf{u}_1$ and $\mathbf{u}_2$, respectively. By performing the partial derivative of $L(\mathbf{y}, \mathbf{t})$ with respect to $\mathbf{W}_2$, we obtain $\frac{\partial L}{\partial \mathbf{W}_2^T} = \frac{\partial L}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{u}_2} \frac{\partial \mathbf{u}_2}{\partial \mathbf{W}_2^T}$ where $\frac{\partial L}{\partial \mathbf{y}} = -\frac{2}{\ell}(\mathbf{t} - \mathbf{y})$, $\frac{\partial \mathbf{y}}{\partial \mathbf{u}_2} = \frac{\partial \eta_2(\mathbf{u}_2)}{\partial \mathbf{u}_2}$, and $\frac{\partial \mathbf{u}_2}{\partial \mathbf{W}_2^T} = \mathbf{u}_1$. Additionally, we calculate $\eta_2(\mathbf{u}_2)$ as $1/(1 + \exp(-\mathbf{u}_2))$ using the representative

sigmoid function. Then, $\frac{\partial \mathbf{y}}{\partial \mathbf{u}_2}$ is calculated as $\eta_2(\mathbf{u}_2)(1-\eta_2(\mathbf{u}_2))$. Therefore, we obtain $\frac{\partial L}{\partial \mathbf{W}_2^T} = -\frac{2}{\ell}(\mathbf{t}-\mathbf{y})\eta_2(\mathbf{u}_2)(1-\eta_2(\mathbf{u}_2))\mathbf{u}_1$. With the learning coefficient of $\gamma_2$, we update $\mathbf{W}_2^T$ to $\mathbf{W}_2^T - \gamma_2 \frac{\partial L}{\partial \mathbf{W}_2^T}$. Then, when performing the partial derivative of $L(\mathbf{y}, \mathbf{t})$ with respect to $\mathbf{W}_1$, we can obtain $\frac{\partial L}{\partial \mathbf{W}_1} = \frac{\partial L}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{u}_2} \frac{\partial \mathbf{u}_2}{\partial \mathbf{u}_1} \frac{\partial \mathbf{u}_1}{\partial \mathbf{W}_1}$ where $\frac{\partial L}{\partial \mathbf{y}} = -\frac{2}{\ell}(\mathbf{t}-\mathbf{y})$, $\frac{\partial \mathbf{y}}{\partial \mathbf{u}_2} = \eta_2(\mathbf{u}_2)(1-\eta_2(\mathbf{u}_2))$, $\frac{\partial \mathbf{u}_2}{\partial \mathbf{u}_1} = \mathbf{W}_2^T$, and $\frac{\partial \mathbf{u}_1}{\partial \mathbf{W}_1} = 2\mathbf{x}^T$. Thus, we then obtain $\frac{\partial L}{\partial \mathbf{W}_1} = -\frac{4}{\ell}(\mathbf{t}-\mathbf{y})\eta_2(\mathbf{u}_2)(1-\eta_2(\mathbf{u}_2))\mathbf{W}_2^T\mathbf{x}^T$. With the learning coefficient of $\gamma_1$, we update $\mathbf{W}_1$ to $\mathbf{W}_1 - \gamma_1 \frac{\partial L}{\partial \mathbf{W}_1}$.

### 3.4 Setting the Initial Weight Matrix in the Affine Layer

To ensure proper learning convergence in situations with limited training datasets, we proposed a method using optimal similarity and bootstrap methods. Here, the number of training data and the training dataset are denoted $n$ and $S(\ni \mathbf{x}_j)$, respectively, where $\mathbf{x}_j$ is the $j$-th training observation ($j$ takes values 1 to $n$). Additionally, we normalized each observation vector, such that its norm is one. By considering the discrimination problem of two groups whose outcomes are 0 and 1, respectively, we divided $\{\mathbf{x}_j\}$ into $\{\mathbf{x}_j|\mathbf{y}_j = 0\}$ and $\{\mathbf{x}_j|\mathbf{y}_j = 1\}$. Next, we defined $\{\mathbf{x}_j|\mathbf{y}_j = 0\}$ and $\{\mathbf{x}_j|\mathbf{y}_j = 1\}$ as $S_0$ and $S_1$, respectively. First, we calculated the autocorrelation matrix with the observations belonging to $S_0$. Then, using the eigenvalues ($\hat{\lambda}_{s_0}$) and eigenvectors ($\hat{\mathbf{u}}_{s_0}$) for the autocorrelation matrix, we calculated the following projection matrix:

$$\hat{P}_0 := \sum_{s_0=1}^{\ell_0'} \hat{\mathbf{u}}_{s_0}\hat{\mathbf{u}}_{s_0}^T, \tag{1}$$

where $\ell_0'$ takes values 1 to $\ell$ in Equation (1). Similarly, we calculated the autocorrelation matrix with the observations belonging to $S_1$. Then, with eigenvalues ($\hat{\lambda}_{s_1}$) and eigenvectors ($\hat{\mathbf{u}}_{s_1}$) for the autocorrelation matrix, we calculate the following projection matrix:

$$\hat{P}_1 := \sum_{s_1=1}^{\ell_1'} \hat{\mathbf{u}}_{s_1}\hat{\mathbf{u}}_{s_1}^T, \tag{2}$$

where $\ell_1'$ takes values 1 to $\ell$ in Equation (2). Here, if the value of $\mathbf{x}^T(\hat{P}_1 - \hat{P}_0)\mathbf{x} > 0$, we classify $\mathbf{x}$ into $S_1$; otherwise, we classify $\mathbf{x}$ into $S_0$.

From a prediction perspective, using the leave-one-out cross-validation [11], we determined the optimal $\hat{\ell}_0'$ and $\hat{\ell}_1'$, which are minimum values satisfying $\tau < (\sum_{s_0=1}^{\ell_0'} \hat{\lambda}_{s_0})/(\sum_{s_0=1}^{\ell} \hat{\lambda}_{s_0})$ and $\tau < (\sum_{s_1=1}^{\ell_1'} \hat{\lambda}_{s_1})/(\sum_{s_1=1}^{\ell} \hat{\lambda}_{s_1})$, respectively. Here, $\tau$ is a tuning parameter to be optimized using the leave-one-out cross-validation. In the second step, based on $\hat{P}_1$, we estimated $\hat{\mathbf{y}}_j$ as $\mathbf{x}_j^T\hat{P}_1\mathbf{x}_j$. In the third step, using existing approaches [5, 6], we generated , we generated normal random numbers and set an initial matrix, vector, and scalar as $\hat{\mathbf{W}}_2$, $\hat{\mathbf{b}}_1$, and $\hat{\mathbf{b}}_2$, respectively. Next, we extracted

$m$ observations randomly using the bootstrap method [12]. Using $\hat{\mathbf{W}}_2$, $\hat{\mathbf{b}}_1$, $\hat{\mathbf{b}}_2$, and a bootstrap sample of size $m$, we estimated $\mathbf{W}_2\mathbf{W}_1$ as follows:

$$\hat{\mathbf{W}}_2\hat{\mathbf{W}}_1 = \frac{1}{m}\sum_{i=1}^{m}(\eta_2^{-1}(\hat{\mathbf{y}}_i) - (\hat{\mathbf{W}}_2\hat{\mathbf{b}}_1 + \hat{\mathbf{b}}_2))\mathbf{x}_i^T(\mathbf{x}_i\mathbf{x}_i^T)^{-1}, \tag{3}$$

where we estimate the inverse of $\mathbf{x}_i\mathbf{x}_i^T$ in Equation (3) using the naive approach from the diagonal elements in $\mathbf{x}_i\mathbf{x}_i^T$. Additionally, using the generalized inverse approach, we obtained $\hat{\mathbf{W}}_1$ in the basis of $\hat{\mathbf{W}}_2$ and $\hat{\mathbf{W}}_2\hat{\mathbf{W}}_1$. Finally, $\hat{\mathbf{b}}_1$, $\hat{\mathbf{b}}_2$, $\hat{\mathbf{W}}_1$, and $\hat{\mathbf{W}}_2$ were used as initial vectors and matrices to update the parameters of the convolutional neural network.

## 4 Analysis Results on Real-world Data

In this paper, all analyses were performed using R version 4.1.2 (R Foundation for Statistical Computing). We applied the proposed method to a real BA dataset. Here, stool image data with objects, such as diapers partially photographed on the image were used. In this numeric experiment, we randomly divided 35 data into 15 training and 20 test data, respectively. Next, we compared the proposed and existing methods relative to the learning convergence and prediction accuracy on the training and test data, respectively. Here, we set the values of the learning coefficients $\gamma_1$ and $\gamma_2$ to 0.1, respectively. Also, we prepared a single feature selection layer and performed the convolution and max pooling process seven times. Each time an initial value was set randomly, learning was performed 1000 times using the 15 training data, and it was judged that learning converged when the value obtained by dividing the sum of the absolute values of the difference between $\hat{\mathbf{y}}_j$ and $\mathbf{t}_j$ by 1000 became less than 0.01. We repeated to randomly divide 35 data into 15 training and 20 test data five times. As a result, we created five datasets. For each dataset, the sensitivity, specificity, and AUC values of the training and test data were calculated using the parameters ($\hat{\mathbf{b}}_1$, $\hat{\mathbf{b}}_2$, $\hat{\mathbf{W}}_1$, and $\hat{\mathbf{W}}_2$) at the time the learning first converged in the existing and our proposed methods. Figure 1 shows the average of the five absolute values of the difference between the correct label and the predicted value at each step when learning was first converged for each method. We can observe that the error decreased steadily as the proposed method progressed compared to the existing methods. When the model was constructed using the weights at the learning convergence point and applied to 15 training data every time, the average values of sensitivity and specificity were 100.0%, and that of the AUC value was 1.000 for all methods. However, a difference was observed among the compared methods on the test data. For the method by [5], the average values of sensitivity, specificity, and AUC in the test data were 83.3%, 42.5%, and 0.629, respectively. Also, for that of [6], the average values of sensitivity, specificity, and AUC in the test data were 85.0%, 40.0%, and 0.625, respectively. With the proposed method, the average values of sensitivity, specificity, and AUC obtained on the test data were 85.0%, 67.5%, and 0.763, respectively.

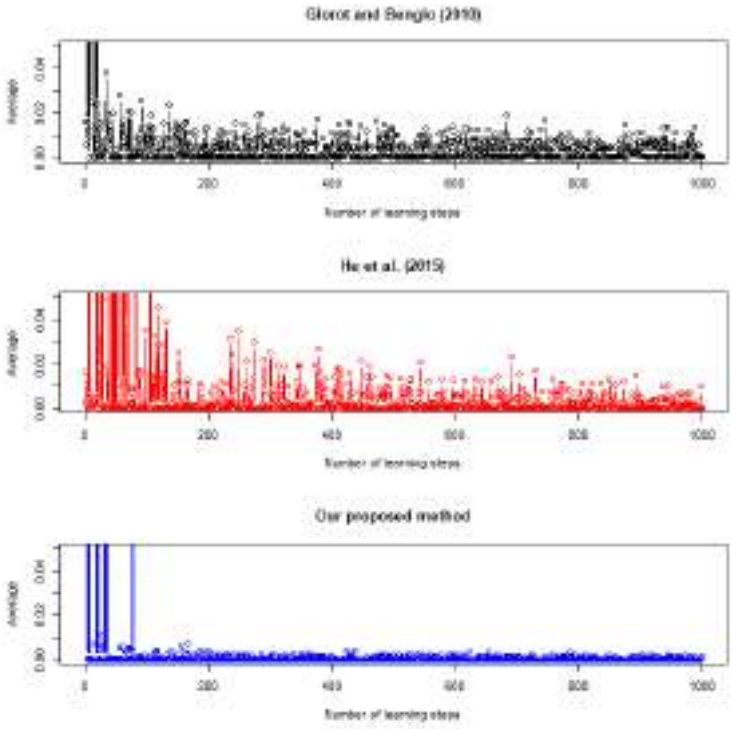


**Fig. 1** Transition of learning in each method.

## 5 Conclusion and Limitations

In this paper, we considered a discrimination problem using a DCNN for high-dimensional small sample data and proposed a method by setting the initial weight matrix in the affine layer. In situations of limited learning data, although transfer learning can be used, we proposed an efficient learning method using the DCNN method. In terms of learning convergence and results obtained from the test data, we confirm that the proposed method is good. However, the results presented in this paper are limited and the proposed method needs to be examined in more detail. Therefore, in the future, through large-scale simulation studies and other real-world data applications, we plan to investigate the differences between the proposed method and existing methods by changing the number of feature selection layers and using different convolution filters. We also plan to investigate the proposed method by considering robustness and setting outliers on the simulation data.

**Acknowledgements** We thank Shinsuke Ito, Takashi Taguchi, Dr. Yusuke Yamane, Ms. Saeko Hishinuma, and Dr. Saeko Hirai for their advice. In addition, we acknowledge the biliary atresia patients' community (BA no kodomowo mamorukai) for their generous support of this project. This work was supported by the Mitsubishi Foundation.

## References


1. Rawat, W., Wang, Z.: Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput. **29**(9), 2352–2449 (2017)
2. Yadav, S. S., Jadhav, S.M.: Deep convolutional neural network based medical image classification for disease diagnosis. J. Big Data (2019) doi: 10.1186/s40537-019-0276-2
3. Huang, J., Xu, Z.: Cell detection with deep learning accelerated by sparse kernel. In: Lu, L. et al. (eds.) Advances in Computer Vision and Pattern Recognition, pp. 137-157. Springer, Switzerland (2017)
4. Abdelhafiz, D., Yang, C., Ammar, R., Nabavi, S.: Deep convolutional neural networks for mammography: advances, challenges and applications. BMC Bioinform. (2019) doi: 10.1186/s12859-019-2823-4
5. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249-256. (2010)
6. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1026-1034. (2015)
7. Liu, J., Dai, S., Chen, G., Sun, S., Jiang, J., Zheng, S., Zheng, Y., Dong, R.: Diagnostic value and effectiveness of an artificial neural network in biliary atresia. Front. Pediatr. (2020) doi: 10.3389/fped.2020.00409
8. Zhou, W., Yang, Y., Yu, C., Liu, J., Duan, X., Weng, Z., Chen, D., Liang, Q., Fang, Q., Zhou, J., Ju, H., Luo, Z., Guo, W., Ma, X., Xie, X., Wang, R., Zhou, L.: Ensembled deep learning model outperforms human experts in diagnosing biliary atresia from sonographic gallbladder images. Nat. Commun. (2021) doi: 10.1038/s41467-021-21466-z
9. Gu, Y.H., Yokoyama, K., Mizuta, K., Tsuchioka, T., Kudo, T., Sasaki, H., Nio, M., Tang, J., Ohkubo, T., Matsui, A.: Stool color card screening for early detection of biliary atresia and long-term native liver survival: a 19-year cohort study in Japan. J. Pediatr. **166**(4), 897–902 (2015)
10. Hoshino, E., Hayashi, K., Suzuki, M., Obatake, M., Urayama, K.Y., Nakano, S., Taura, Y., Nio, M., Takahashi, O.: An iPhone application using a novel stool color detection algorithm for biliary atresia screening. Pediatr. Surg. Int. **33**(10), 1115–1121 (2017)
11. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer, New York (2009)
12. Efron, B.: Bootstrap methods: another look at the jackknife. Ann. Stat. **7**(1), 1–26 (1979)




# Some Issues in Robust Clustering

Christian Hennig

**Abstract** Some key issues in robust clustering are discussed with focus on the Gaussian mixture model based clustering, namely the formal definition of outliers, ambiguity between groups of outliers and clusters, the interaction between robust clustering and the estimation of the number of clusters, the essential dependence of (not only) robust clustering on tuning decisions, and shortcomings of existing measurements of cluster stability when it comes to outliers.

**Keywords:** Gaussian mixture model, trimming, noise component, number of clusters, user tuning, cluster stability

## 1 Introduction

Cluster analysis is about finding groups in data. Robust statistics is about methods that are not affected strongly by deviations from the statistical model assumptions or moderate changes in a data set. Particular attention has been paid in the robustness literature to the effect of outliers. Outliers and other model deviations can have a strong effect on cluster analysis methods as well. There is now much work on robust cluster analysis, see [1, 19, 9] for overviews.

There are standard techniques of assessing robustness such as the influence function and the breakdown point [15] as well as simulations involving outliers, and these have been applied to robust clustering as well [19, 9].

Here I will argue that due to the nature of the cluster analysis problem, there are issues with the standard reasoning regarding robustness and outliers.

The starting point will be clustering based on the Gaussian mixture model, for details see [3]. For this approach, $n$ observations are assumed i.i.d. with density

Christian Hennig (✉)
Dipartimento di Scienze Statistiche "Paolo Fortunati", University of Bologna, Via delle Belle Arti 41, 40126 Bologna, Italy, e-mail: christian.hennig@unibo.it



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_21

$$f_\eta(x) = \sum_{k=1}^{K} \pi_k \varphi_{\mu_k,\Sigma_k}(x),$$

$x \in \mathbb{R}^p$, with $K$ mixture components with proportions $\pi_k$, $\varphi_{\mu_k,\Sigma_k}$ being the Gaussian density with mean vectors $\mu_k$, covariance matrices $\Sigma_k$, $k = 1, \ldots, K$, $\eta$ being a vector of all parameters. For given $K$, $\eta$ can be estimated by maximum likelihood (ML) using the EM-algorithm, as implemented for example in the R-package "mclust". A standard approach to estimate $K$ is the optimisation of the Bayesian Information Criterion (BIC). Normally, mixture components are interpreted as clusters, and observations $x_i$, $i = 1, \ldots, n$, can be assigned to clusters using the estimated posterior probability that $x_i$ was generated by mixture component $k$. A problem with ML estimation is that the likelihood degenerates if all observations assigned to a mixture component lie on a lower dimensional hyperplane, i.e, a $\Sigma_k$ has an eigenvalue of zero. This can be avoided by placing constraints on the eigenvalues of the covariance matrices [8]. Alternatively, a non-degenerate local optimum of the likelihood can be used, and if this cannot be found, constrained covariance matrix models (such as $\Sigma_1 = \ldots = \Sigma_K$) can be fitted instead, as is the default of mclust. Several issues with robustness that occur here are also relevant for other clustering approaches.

## 2 Outliers vs Clusters

It is well known that the sample mean and sample covariance matrix as estimators of the parameters of a single Gaussian distribution can be driven to breakdown by a single outlier [15]. Under a Gaussian mixture model with fixed $K$, an outlier must be assigned to a mixture component $k$ and will break down the estimators of $\mu_k, \Sigma_k$ (which are weighted sample means and covariance matrices) for that component in the same manner; the same holds for a cluster mean in $k$-means clustering.

Addressing this issue, and dealing with more outliers in order to achieve a high breakdown point, is a starting point for robust clustering. Central ideas are trimming a proportion of observations [7], adding a "noise component" with constant density to catch the outliers [4, 3], mixtures with more robust component-wise estimators such as mixtures of heavy-tailed distributions (Sec. 7 of [18]).

But cluster analysis is essentially different from estimating a homogeneous population. Given a data set with $K$ clear Gaussian clusters and standard ML-clustering, consider adding a single outlier that is far enough away from the clusters. Assuming a lower bound on covariance matrix eigenvalues, the outlier will form a one-point cluster, the mean of which will diverge with the added outlier, and the original clusters will be merged to form $K - 1$ clusters [10].

The same will happen with a group of several outliers being close together, once more added far enough away from the Gaussian clusters. "Breakdown" of an estimator it is usually understood as the estimator becoming useless. It is questionable that this is the case here. In fact, the "group of outliers" can well be interpreted as a cluster in its own right, and putting all these points together in a cluster could be

seen as desirable behaviour of the ML estimator, at least if two of the original $K$ clusters are close enough to each other that merging them will produce a cluster that is fairly well fitted by a single Gaussian distribution; note that the Gaussian mixture model does not assume strong separation between components, and a mixture of two Gaussians may be unimodal and in fact very similar to a single Gaussian. A breakdown point larger than a given $\alpha$, $0 < \alpha < \frac{1}{2}$ may not be seen as desirable in cluster analysis if there can be clusters containing a proportion of less than $\alpha$ of the data, as a larger breakdown point will stop a method from taking such clusters (when added in large distance from the rest of the data) appropriately into account.

The core problem is that it is not clear what distinguishes a group of outliers from a legitimate cluster. I am not aware of any formal definition of outliers and clusters in the literature that allows this distinction. Even a one-point cluster is not necessarily invalid. Here are some possible and potentially conflicting aspects of such a distinction.

- A certain minimum size may be required for a cluster; smaller groups of points may be called outliers.
- Groups of points in low density areas of the data may be called outliers. Note that this particularly means that very widely spread Gaussian mixture components would also be defined as outliers, deviating from the standard interpretation of Gaussian mixture components as clusters.
- Members of non-Gaussian mixture components may be called outliers. This does not seem to be a good idea, because Gaussianity cannot be assessed for too small groups of observations, and furthermore in practice model assumptions are never perfectly fulfilled, and it may be desirable to interpret homogeneous or unimodal non-Gaussian parts of the data as "cluster" and fit them by a Gaussian component.
- The term "outlier" suggests that outliers lie far away from most other observations, so it may be required that outliers are farther away from the clusters than the clusters are from each other. But this would be in conflict with the intuition that strong separation is usually seen as a desirable feature for well interpretable clusters. It may only be reasonable in applications in which there is prior information that there is limited variation even between clusters, as is implied by certain Bayesian approaches to clustering [17].
- The term "cluster" may be seen as flexible enough that a definition of an outlier is not required. Clustering should accommodate whatever is "outlying" by fitting it by one or more further clusters, if necessary of size one (single linkage clustering can be useful for outlier detection, even though it is inappropriate for most clustering problems).

Most of these items require specific decisions that cannot be made in any objective and general manner, but only taking into account subject matter information, such as the minimum size of valid clusters or the density level below which observations are seen as outliers (potentially compared to density peaks in the distribution). This implies that an appropriate treatment of outliers in cluster analysis cannot be expected to be possible without user tuning.

## 3 Robustness and the Number of Clusters

The last item suggests that there is an interplay between outlier identification and the number of clusters, and that adding clusters might be a way of dealing with outliers; as long as clusters are assumed to be Gaussian, a single additional component may not be enough. More generally, concentrating robustness research on the case of fixed $K$ may be seen as unrealistic, because $K$ is rarely known, although estimating $K$ is a notoriously difficult problem even without worrying about outliers [13].

The classical robustness concepts, breakdown point and influence function, assume parameters from $\mathbb{R}^q$ with fixed $q$. If $K$ is not fixed, the number of parameters is not fixed either, and the classical concepts do not apply.

As an alternative to the breakdown point, [11] defined a "dissolution point". Dissolution is measured in terms of cluster memberships of points rather than in terms of parameters, and is therefore also applicable to nonparametric clustering methods. Furthermore, dissolution applies to individual clusters in a clustering; certain clusters may dissolve, i.e., there may be no sufficiently similar cluster in a new clustering computed after, e.g., adding an outlier; and others may not dissolve. This does not require $K$ to be fixed; the definition is chosen so that if a clustering changes from $K$ to $L < K$ clusters, at least $K - L$ clusters dissolve.

Hennig [10, 11] showed that when estimating $K$ using the BIC and standard ML estimation, reasonably well separated clusters do not dissolve when adding possibly even a large percentage of outliers (this does not hold for every method to estimate the number of clusters, see [11]). Furthermore, [11] showed that no method with fixed $K$ can be robust for data in which $K$ is misspecified - already [7] had found that robustness features in clustering generally depend on the data.

An implication of these results is that even in the fixed $K$ problem, the standard ML method can be a valid competitor regarding robustness if it comes with a rule that allows to add one or possibly more clusters that can then be used to fit the outliers (this is rarely explored in the literature, but [18], Sec. 7.7, show an example in which adding a single component does not work very well).

An issue with adding clusters to accommodate outliers is that in many applications it is appropriate to distinguish between meaningful clusters, and observations that cannot be assigned to such clusters (often referred to as "noise"). Even though adding clusters of outliers can formally prevent the dissolution of existing clusters, it may be misleading to interpret the resulting clusters as meaningful, and a classification as outliers or noise can be more useful. This is provided by the trimming and noise component approaches to robust clustering. Also some other clustering methods such as the density-based DBSCAN [5] provide such a distinction. On the other hand, modelling clusters by heavy-tailed distributions such as in mixtures of t-distributions will implicitly assign outlying observations to clusters that potentially are quite far away. For this reason, [18], Sec. 7.7, provide an additional outlier identification rule on top of the mixture fit. [6] even distinguish between "mild" outliers that are modelled as having a larger variance around the same mean, and "gross" outliers to be trimmed. The variety of approaches can be connected to the different meanings that outliers can have in applications. They can be erroneous, they can be irrelevant

noise, but they can also be caused by unobserved but relevant special conditions (and would as such qualify as meaningful clusters), or they could be valid observations legitimately belonging to a meaningful cluster that regularly produces observations further away from the centre than modelled by a Gaussian distribution.

Even though currently there is no formal robustness property that requires both the estimation of $K$ and an identification or downweighting of outliers, there is demand for a method that can do both.

Estimating $K$ comes with an additional difficulty that is relevant in connection with robustness. As mentioned before, in clustering based on the Gaussian mixture model normally every mixture component will be interpreted as a cluster. In reality, however, meaningful clusters are not perfectly Gaussian. Gaussian mixtures are very flexible for approximating non-Gaussian distributions. Using a consistent method for estimating $K$ means that for large enough $n$ a non-Gaussian cluster will be approximated by several Gaussian mixture components. The estimated $K$ will be fine for producing a Gaussian mixture density that fits the data well, but it will overestimate the number of interpretable clusters. The estimation of $K$, if interpreted as the number of clusters, relies on precise Gaussianity of the clusters, and is as such itself riddled with a robustness problem; in fact slightly non-Gaussian clusters may even drive the estimated $K \to \infty$ if $n \to \infty$ [12, 14].

This is connected with the more fundamental problem that there is no unique definition of a cluster either. The cluster analysis user needs to specify the cluster concept of interest even before robustness considerations, and arguably different clustering methods imply different cluster concepts [13]. A Gaussian mixture model defines clusters by the Gaussian distributional shape (unless mixture components are merged to form clusters [12]). Although this can be motivated in some real situations, robustness considerations require that distributional shapes fairly close to the Gaussian should be accepted as clusters as well, but this requires another specification, namely how far from a Gaussian a cluster is allowed to be, or alternatively how separated Gaussian components have to be in order to count as separated clusters. A similar problem can also occur in nonparametric clustering; if clusters are associated with density modes or level sets, the cluster concept depends on how weak a mode or gap between high level density sets is allowed to be to be treated as meaningful.

Hennig and Coretto [14] propose a parametric bootstrap approach to simultaneously estimate $K$ and assign outliers to a noise component. This requires two basic tuning decisions. The first one regards the minimum percentage of observations so that a researcher is willing to add another cluster if the noise component can be reduced by this amount. The second one specifies a tolerance that allows a data subset to count as a cluster even though it deviates to some extent from what is expected under a perfectly Gaussian distribution. There is a third tuning parameter that is in effect for fixed $K$ and tunes how much of the tails of a non-Gaussian cluster can be assigned to the noise in order to improve the Gaussian appearance of the cluster. One could even see the required constraints on covariance matrix eigenvalues as a further tuning decision. Default values can be provided, but situations in which matters can be improved deviating from default values are easy to construct.

## 4 More on User Tuning

User tuning is not popular, as it is often difficult to make appropriate tuning decisions. Many scientists believe that subjective user decisions threaten scientific objectivity, and also background knowledge dependent choices cannot be made when investigating a method's performance by theory and simulations. The reason why user tuning is indispensable in robust cluster analysis is that it is required in order to make the problem well defined. The distinction between clusters and outliers is an interpretative one that no automatic method can make based on the data alone. Regarding the number of clusters, imagine two well separated clusters (according to whatever cluster concept of interest), and then imagine them to be moved closer and closer together. Below what distance are they to be considered a single cluster? This is essentially a tuning decision that the data cannot make on their own.

There are methods that do not require user tuning. Consider the mclust implementation of Gaussian mixture model based clustering. The number of clusters is by default estimated by the BIC. As seen above, this is not really appropriate for large data sets, but its derivation is essentially asymptotic, so that there is no theoretical justification for it for small data sets either. Empirically it often but not always works well, and there is little investigation of whether it tends to make the "right" decision in ambiguous situations where it is not clear without user tuning what it even means to be "right". Covariance matrix constraints in mclust are not governed by a tuning of eigenvalues or their ratios to be specified by the user. Rather the BIC decides between different covariance matrix models, but this can be erratic and unstable, as it depends on whether the EM-algorithm gets caught in a degenerate likelihood maximum or not, and in situations where two or more covariance matrix models have similar BIC values (which happens quite often), a tiny change in the data can result in a different covariance matrix model being selected, and substantial changes in the clustering. A tunable eigenvalue condition can result in much smoother behaviour. When it comes to outlier identification, mclust offers the addition of a uniform "noise" mixture component governed by the range of the data, again supposedly without user tuning. This starts from an initial noise estimation that requires tuning (Sec. 3.1.2 of [3]) and is less robust in terms of breakdown and dissolution than trimming and the improper noise component, both of which require tuning [10, 11]. The ICL, an alternative to the BIC (Sec. 2.6 of [3]), on the other hand, is known to merge different Gaussian mixture components already at a distance at which they intuitively still seem to be separated clusters. Similar comments apply to the mixture of t-distributions; it requires user tuning for identifying outliers, scatter matrix constraints, and it has the same issues with BIC and ICL as the Gaussian mixture.

Summarising, both the identification of and robustness against outliers and the estimation of the number of clusters require tuning in order to be well defined problems; user tuning can only be avoided by taking tuning decisions out of the user's hands and making them internally, which will work in some situations and fail in others, and the impression of automatic data driven decision making that a user may have is rather an illusion. This, however, does not free method designers from the necessity to provide default tunings for experimentation and cases in which

the users do not feel able to make the decisions themselves, and tuning guidance for situations in which more information is available. A decision regarding the smallest valid size of a cluster is rather well interpretable; a decision regarding admissible covariance matrix eigenvalues is rather difficult and abstract.

## 5 Stability Measurement

Robustness is closely connected to stability. Both experimental and theoretical investigation of the stability of clusterings require formal stability measurements, usually comparing two clusterings on the same data (potentially modified by replacing or adding observations). Not assuming any parametric model, proximity measures such as the Adjusted Rand Index (ARI; [16]), the Hamming distance (HD; [2]), or the Jaccard distance between individual clusters [11] can be used. Note that [2], standard reference on cluster stability in the machine learning community, state that stability and instability are caused in the first place by ambiguities in the cluster structure of the data, rather than by a method's robustness or lack of it. Although the outlier problem is ignored in that paper, it is true that cluster analysis can have other stability issues that are as serious as or worse than gross outliers.

To my knowledge, none of the measures currently in use allow for a special treatment of a set of outliers or noise; either these have to be ignored, or treated just as any other cluster. Both ARI and HD, comparing clusterings $C_1$ and $C_2$, consider pairs of observations $x_i, x_j$ and check whether those that are in the same cluster in $C_1$ are also in the same cluster in $C_2$. An appropriate treatment of noise sets $N_1 \in C_1, N_2 \in C_2$ would require that $x_i, x_j \in N_1$ are not just in the same cluster in $C_2$ but rather in $N_2$, i.e., whereas the numberings of the regular clusters do not have to be matched (which is appropriate because cluster numbering is meaningless), $N_1$ has to be matched to $N_2$. Corresponding re-definitions of these proximities will be useful to robustness studies.

## 6 Conclusion

Key practical implications of the above discussions are:

- Outliers can be treated as forming their own clusters, or be collected in outlier/noise or trimmed sets, or be integrated in clusters of non-outliers. Which of these is appropriate depends on the nature of outliers in a given application.
- Methods that do not identify outliers but add clusters in order to accommodate them are valid competitors of robust clustering methods, as are nonparametric density-based methods.
- Cluster analysis involving estimating the number of clusters and robustness require tuning in order to define the problem they are meant to solve well. Method

developers need to provide sensible defaults, but also to guide the users regarding a meaningful interpretation of the tuning decisions.

## References


1. Banerjee, A., Davé, R. N.: Robust clustering. WIREs Data Mining Knowl. Discov. **2**, 29–59 (2012)
2. Ben-David, S., von Luxburg, U., Pál, D.: A sober look at clustering stability. In: Proceedings of the 19th annual conference on Learning Theory (COLT'06), pp. 5–19, Springer, Berlin (2006)
3. Bouveyron, C., Celeux, G., Murphy, T. B., Raftery, A. E.: Model-based clustering and classification for data science. Cambridge University Press, Cambridge MA (2019)
4. Coretto, P., Hennig, C.: Consistency, breakdown robustness, and algorithms for robust improper maximum likelihood clustering. J. Mach. Learn. Res.**18**, 1–39 (2017)
5. Ester, M., Kriegel, H. P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226-231, AAAI Press, Portland OR (1996)
6. Farcomeni, A., Punzo, A.: Robust model-based clustering with mild and gross outliers. TEST **29**, 989–1007 (2020)
7. García-Escudero, L. A., Gordaliza, A.: Robustness properties of k-means and trimmed k-means. J. Am. Stat. Assoc. **94**, 956–969 (1999)
8. García-Escudero, L. A., Gordaliza, A., Greselin, F., Ingrassia, S., Mayo-Iscar, A.: Eigenvalues and constraints in mixture modeling: geometric and computational issues. Adv. Data Anal. Classi. **12**, 203–233 (2018)
9. García-Escudero, L. A., Gordaliza, A., Matrán, C., Mayo-Iscar, A., Hennig, C.: Robustness and outliers. In: Hennig, C., Meila, M., Murtagh, F., Rocci, R. (eds.) Handbook of Cluster Analysis, pp. 653–678. Chapman & Hall/CRC, Boca Raton FL (2016)
10. Hennig, C.: Breakdown points for maximum likelihood estimators of location-scale mixtures. Ann. Stat. **32**, 1313–1340 (2004)
11. Hennig, C.: Dissolution point and isolation robustness: robustness criteria for general cluster analysis methods. J. Multivariate Anal. **99**, 1154–1176 (2008)
12. Hennig, C.: Methods for merging Gaussian mixture components. Adv. Data Anal. Classi. **4**, 3–34 (2010)
13. Hennig, C.: Clustering strategy and method selection. In: Hennig, C., Meila, M., Murtagh, F., Rocci, R. (eds.) Handbook of Cluster Analysis, pp. 703–730. Chapman & Hall/CRC, Boca Raton FL (2016)
14. Hennig, C., Coretto, P.: An adequacy approach for deciding the number of clusters for OTRIMLE robust Gaussian mixture-based clustering. Aust. N. Z. J. Stat. (2021) doi: 10.1111/anzs.12338
15. Huber, P. J., Ronchetti, E. M.: Robust Statistics (2nd ed.). Wiley, Hoboken NJ (2009)
16. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)
17. Malsiner-Walli, G., Frühwirth-Schnatter, S., Grün, B.: Identifying mixtures of mixtures using Bayesian estimation. J. Comput. Graph. Stat. **26**, 285–295 (2017)
18. McLachlan, G. J., Peel, D.: Finite Mixture Models. Wiley, New York (2000)
19. Ritter, G.: Robust cluster analysis and variable selection. Chapman & Hall/CRC, Boca Raton FL (2015)


# Robustness Aspects of Optimized Centroids

Jan Kalina and Patrik Janáček

**Abstract** Centroids are often used for object localization tasks, supervised segmentation in medical image analysis, or classification in other specific tasks. This paper starts by contributing to the theory of centroids by evaluating the effect of modified illumination on the weighted correlation coefficient. Further, robustness of various centroid-based tools is investigated in experiments related to mouth localization in non-standardized facial images or classification of high-dimensional data in a matched pairs design. The most robust results are obtained if the sparse centroid-based method for supervised learning is accompanied with an intrinsic variable selection. Robustness, sparsity, and energy-efficient computation turn out not to contradict the requirement on the optimal performance of the centroids.

**Keywords:** image processing, optimized centroids, robustness, sparsity, low-energy replacements

## 1 Introduction

Methods based on centroids (templates, prototypes) are simple yet widely used for object localization or supervised segmentation in image analysis tasks and also within other supervised or unsupervised methods of machine learning. This is true e.g. in various biomedical imaging tasks [1], where researchers typically cannot afford a too large number of available images [3]. Biomedical applications also benefit from the interpretability (comprehensibility) of centroids [11].

This paper is focused on the question how are centroid-based methods influenced by data contamination. Section 2 recalls the main approaches to centroid-based object localization in images, as well as a recently proposed method of [6] for op-

Jan Kalina (✉) · Patrik Janáček
The Czech Academy of Sciences, Institute of Computer Science, Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic, e-mail: kalina@cs.cas.cz;janacekpatrik@gmail.com



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_22

timizing centroids and their weights. The performance of these methods to data contamination (non-standard conditions) has not been however sufficiently investigated. Particularly, we are interested in the performance of low-energy replacements of the optimal centroids and in the effect of posterior variable selection (pixel selection). Section 2.1 presents novel expressions for images with a changed illumination. Numerical experiments are presented in Section 3. These are devoted to mouth localization over raw facial images as well as over artificially modified images; other experiments are devoted to high-dimensional data in a matched pairs design. The optimized centroids of [6] and especially their modification proposed here turn out to have remarkable robustness properties. Section 4 brings conclusions.

## 2 Centroid-based Classification (Object Localization)

Commonly used centroid-based approaches to object localization (template matching) in images construct the centroid simply as the average of the positive examples and typically use Pearson product-moment correlation coefficient $r$ as the most common measure of similarity between a centroid $\mathbf{c}$ and a candidate part of the image (say $\mathbf{x}$). While the centroid and candidate areas are matrices of size (say) $I \times J$ pixels, they are used in computations after being transformed to vectors of length $d := IJ$. This allows us to use the notation $\mathbf{c} = (c_1, \ldots, c_d)^T$ and $\mathbf{x} = (x_1, \ldots, x_d)^T$.

*Assumptions* $\mathcal{A}$: We assume the whole image to have size $N_R \times N_C$ pixels. We assume the centroid $\mathbf{c} = (c)_{i,j}$ with $i = 1, \ldots, I$ and $j = 1, \ldots, J$ to be a matrix of size $I \times J$ pixels. A candidate area $\mathbf{x}$ and nonnegative weights $\mathbf{w}$ with $\sum_i \sum_j w_{ij} = 1$ are assumed to be matrices of the same size as $\mathbf{c}$.

For a given image, $\mathsf{E}$ will denote the set of its rectangular candidate areas of size $I \times J$. The candidate area fulfilling

$$\arg\max_{\mathbf{x} \in \mathsf{E}} r(\mathbf{x}, \mathbf{c}) \tag{1}$$

or (less frequently)

$$\arg\min_{\mathbf{x} \in \mathsf{E}} ||\mathbf{x} - \mathbf{c}||_2 \tag{2}$$

are classified to correspond to the object (e.g. mouth).

Let us consider here replacing $r$ by the weighted correlation coefficient $r_w$

$$\arg\max_{\mathbf{x} \in \mathsf{E}} r_w(\mathbf{x}, \mathbf{c}; \mathbf{w}) \tag{3}$$

with given non-negative weights $\mathbf{w} = (w_1, \ldots, w_d)^T \in \mathcal{R}^p$ with $\sum_{i=1}^n w_i = 1$, where $\mathcal{R}$ denotes the set of all real numbers. Let us further use the notation $\bar{x}_w = \sum_{j=1}^d w_j x_j = \mathbf{w}^T \mathbf{x}$ and $\bar{c}_w = \mathbf{w}^T \mathbf{c}$. We may recall $r_w$ between $\mathbf{x}$ and $\mathbf{c}$ to be defined as

$$r_W(\mathbf{x}, \mathbf{c}; \mathbf{w}) = \frac{\sum_{i=1}^d w_i (x_i - \bar{x}_w)(c_i - \bar{c}_w)}{\sqrt{\sum_{i=1}^d [w_i (x_i - \bar{x}_w)^2] \sum_{i=1}^d [w_i (c_i - \bar{c}_w)^2]}}. \tag{4}$$

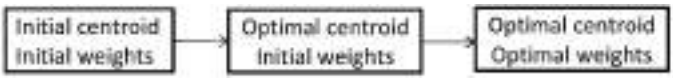


**Fig. 1** The workflow of the optimization procedure of [6].

A detailed study of [2] investigated theoretical foundations of centroid-based classification, however for the rare situation when (1) is replaced by

The sophisticated centroid optimization method of [6], outlined in Figure 1, requires to minimize a nonlinear loss function corresponding to a regularized margin-like distance (exploiting $r_w$) evaluated for the worst pair from the worst image over the training database (i.e. the worst with respect to the loss function). Subsequently, optimization of the weights may be also performed, ensuring many pixels to obtain zero weights (i.e. yielding a sparse solution). The optimal centroid may be used as such, even without any weights at all; still, optimization of the weights leads to a further improvement of the classification performance. In the current paper, we always consider a linear (i.e. approximate) approach to centroid optimization, although a nonlinear optimization is also successful as revealed in the comparisons in [6].

## 2.1 Centroid-Based Object Localization: Asymmetric Modification of the Candidate Area

In the context of object localization as described above, our aim is to express $r_w(\mathbf{x}^*, \mathbf{c}; \mathbf{w})$ under modified candidate areas (say $\mathbf{x}^*$) of the image $\mathbf{x}$; we stress that the considered modification of the image does not allow to modify the centroid $\mathbf{c}$ and weights $\mathbf{w}$. These considerations are useful for centroid-based object localization, when asymmetric illumination is present in the whole image or its part. The weighted variance $S_w^2(\mathbf{x}; \mathbf{w})$ of $\mathbf{x}$ with weights $\mathbf{w}$ and the weighted covariance $S_w(\mathbf{x}, \mathbf{c})$ between $\mathbf{x}$ and $\mathbf{c}$ are denoted as

$$S_w^2(\mathbf{x}) = \sum_{i,j} w_{ij}(x_{ij} - \bar{x}_w)^2, \quad S_w(\mathbf{x}, \mathbf{c}) = \sum_{i,j} w_{ij}(x_{ij} - \bar{x}_w)(c_{ij} - \bar{c}_w). \tag{5}$$

Further, the notation $\mathbf{x} + a$ with $\mathbf{x} = (x_{ij})_{i,j}$ is used to denote the matrix $(x_{ij} + a)_{i,j}$ for a given $a \in \mathcal{R}$. We also use the following notation. The image $\mathbf{x}$ is divided to two parts $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T \in \mathcal{R}^d$, where $\sum_I$ or $\sum_{II}$ denote the sum over the pixels of the first or second part, respectively.

**Theorem 1** *Under Assumptions $\mathcal{A}$, the following statements hold.*

1. *For $\mathbf{x}^* = \mathbf{x} + \varepsilon$, it holds $r_w(\mathbf{x}^*, \mathbf{c}) = r_w(\mathbf{x}, \mathbf{c})$ for $\varepsilon > 0$.*
2. *For $\mathbf{x}^* = k\mathbf{x}$ with $k > 0$, it holds $r_w(\mathbf{x}^*, \mathbf{c}) = r_w(\mathbf{x}, \mathbf{c})$.*
3. *For $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T$ and $\mathbf{x}^* = (\mathbf{x}_1, \mathbf{x}_2 + \varepsilon)^T$, it holds $r_w(\mathbf{x}^*, \mathbf{c}) =$*

$$= \frac{S_w(\mathbf{x},\mathbf{c}) + \varepsilon \sum_{II} w_{ij} c_{ij} - \varepsilon v_2 \bar{c}_w}{S_w(\mathbf{c}) \sqrt{S_w^2(\mathbf{x}) + v_2(1-v_2)\varepsilon^2 + 2\varepsilon(2v_2-1)(\sum_{II} w_{ij} x_{ij} - v_2 \bar{x}_w)}}, \quad (6)$$

*where* $v_2 = \sum_{II} w_{ij}$ *and* $\varepsilon \in \mathcal{R}$.

4. *For* $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)^T$ *and* $\mathbf{x}^* = (\mathbf{x}_1, k\mathbf{x}_2)^T$ *with* $k > 0$, *it holds*

$$r_w(\mathbf{x}^*, \mathbf{c}) = r_w(\mathbf{x}, \mathbf{c}) \frac{S_w(\mathbf{x})}{S_w^*(\mathbf{x})} + \frac{(k-1)\sum_{II} w_{ij} x_{ij} (c_{ij} - \bar{c}_w)}{S_w(\mathbf{c}) S_w^*(\mathbf{x})}, \quad (7)$$

*where*

$$\left(S_w^*(\mathbf{x})\right)^2 = S_w^2(\mathbf{x}) + (k^2-1) \sum_{II} w_{ij} x_{ij}^2 - \frac{k^2-1}{n} \left( \sum_{II} w_{ij} x_{ij} \right)^2 -$$

$$-\frac{2}{n}(k-1) \left( \sum_{I} w_{ij} x_{ij} \right) \left( \sum_{II} w_{ij} x_{ij} \right). \quad (8)$$

The proofs of the formulas are technical but straightforward exploiting known properties of $r_w$. The theorem reveals $r_w$ to be vulnerable to the modified illumination, i.e. all the methods based on centroids of Section 2 may be too influenced by the data modification.

## 3 Experiments

### 3.1 Data

Three datasets are considered in the experiments. In the first dataset, the task is to localize the mouth in the database containing 212 grey-scale 2D facial images of faces of healthy individuals of size $192 \times 256$ pixels. The database previously analyzed in [6] was acquired at the Institute of Human Genetics, University of Duisburg-Essen, within research of genetic syndrome diagnostics based on facial images [1] under the projects BO 1955/2-1 and WU 314/2-1 of the German Research Council (DFG). We consider the training dataset to consist of the first 124 images, while the remaining 88 images represent an independent test set acquired later but still under the same standardized conditions fulfilling assumptions of unbiased evaluation. The centroid described below is used with $I = 26$ and $J = 56$.

Using always raw training images, the methods are applied not only to the raw test set, but also to the test set after being artificially modified using models inspired by Section 2.1. On the whole, five different versions of the test database are considered; the modifications required that we first manually localized the mouths in the test images:

1. Raw images.

2. Illumination. If we consider a pixel $[i, j]$ with intensity $x_{ij}$ in an image (say) $f$, then the grey-scale intensity $f_{ij}$ will be

$$f^*_{ij} = f_{ij} + \lambda |j - j_0|, \quad i = 1, \ldots, I, \quad j = 1, \ldots, J, \tag{9}$$

where $[i_0, j_0]$ are the coordinates of the mouth and $\lambda = 0.002$.
3. A more severe version of the modification (ii) with $\lambda = 0.004$.
4. Asymmetry. In every test image, each true mouth $\mathbf{x}$ of size $26 \times 56$ pixels with intensities $x_{ij}$ is replaced by

$$x^*_{ij} = \begin{cases} x_{ij} + 0.2, & i = 1, \ldots, 26, \ j = 1, \ldots, 15, \\ x_{ij}, & i = 1, \ldots, 26, \ j = 16, \ldots, 41, \\ x_{ij} + 0.1, & i = 1, \ldots, 26, \ j = 42, \ldots, 56. \end{cases} \tag{10}$$

5. Rotation. Such candidate area is classified as the mouth in the given image, which maximizes the loss (1) or (3) over the three versions of the image, namely after rotations by +5, 0, and −5 degrees.
6. Image denoising (for raw images). The LWS-filter [5], replacing each grey value by the least weighted squares estimate [7] computed from a circular neighborhood with radius 4 pixels, was applied to each test image.

The optimized centroids were explained in [6] to be applicable also to classification tasks for other data than images, if they follow a matched pairs design. We use two datasets from [6] in the experiments and their classification accuracies are reported in a 10-fold cross validation.

- AMI. The gene expressions of 4000 genes over 92 individuals in two versions (raw or contaminated by outliers). The aim is to learn a classification rule allowing to assign a new individual to one of the two given groups (controls or patients with acute myocardial infarction (AMI)).
- Simulated data. The design mimicks a 1:1 matched case-control study with 2500 variables over 60 individuals in two versions (raw or contaminated by outliers) and the aim is again to classify between two given groups (patients and controls).

**Fig. 2** The average centroid used as the initial choice for the centroid optimization.

### 3.2 Methods

The following methods are compared in the experiments; standard methods are computed using R software and we use our own C++ implementation of centroid-based methods. The average centroid is obtained as the average of all mouths of the training set, or the average across all patients. The centroid optimization starts with the average centroid as the initial one, and the optimization of weights starts with equal weights as the initial ones:

A. Centroid-based method (2).
B. Centroid-based method (1) with average centroid (Figure 2) and equal weights.
C. Centroid-based method (1) with average centroid, replacing $r_w$ by cosine similarity defined for $\mathbf{x} \in \mathcal{R}^d$ and $\mathbf{y} \in \mathcal{R}^d$ as

$$\cos\,\theta = \frac{\mathbf{x}^T\mathbf{y}}{||\mathbf{x}||_2||\mathbf{y}||_2} = \frac{\sum_{i=1}^d x_i y_i}{\left(\sum_{i=1}^d x_i^2\right)^{1/2}(\sum_{j=1}^d y_j^2)^{1/2}}. \tag{11}$$

D. Centroid-based method (1) with optimal centroid and equal weights [6].
E. Centroid-based method (1) with optimal centroid and optimal weights as in [6] (optimizing the centroid and only after that the weights), i.e. with posterior variable selection (pixel selection).
F. Centroid-based method (1) as in [6], where however the weights are optimized first, and then the centroid is optimized.
G. Centroid-based method (1) as in [6], where however each step of centroid optimization is immediately followed by optimization of the weights; this method performs (in contrary to [6]) intrinsic variable selection.
H. Centroid-based method (1) as in [6], where however each optimization step proceeds over 10 worst images (instead of the very worst image).
I. Centroid-based method (1) with average centroid, where $r_w$ is used as $r_{\mathsf{LWS}}$ [7] with weight function

$$\psi_1(t) = \exp\left\{-\frac{t^2}{2\tau^2}\right\} 1\left[t < \frac{3}{4}\right], \quad t \in [0,1], \tag{12}$$

corresponding to a (trimmed) density of the Gaussian $\mathsf{N}(0,1)$ distribution; 1 denotes an indicator function. To explain, the computation of $r_{\mathsf{LWS}}(x,y)$ starts by fitting the LWS estimator in the linear regression of $y$ as the response of $x$, and $r_w$ is used with the weights determined by the LWS estimator.
J. The method (I) with the weight function $\psi_2(t) = 1\left[t < \frac{3}{4}\right]$ for $t \in [0,1]$.
K. The approach of [12] that is meaningful however only for the mouth localization dataset.

**Table 1** Classification accuracy for three datasets. For the mouth localization data, modifications of the test images are described in Section 3: (i) None (raw images); (ii) Illumination; (iii) Asymmetry; (iv) Rotation; (v) Image denoising. A detailed description of the methods is given in Section 3.2.

| | Dataset | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Mouth localization | | | | | | AMI | | Simul. | |
| Method | (i) | (ii) | (iii) | (iv) | (v) | (vi) | Raw | Cont. | Raw | Cont. |
| A | 0.90 | 0.86 | 0.81 | 0.88 | 0.81 | 0.93 | 0.73 | 0.66 | 0.71 | 0.67 |
| B | 0.93 | 0.90 | 0.86 | 0.92 | 0.86 | 0.95 | 0.76 | 0.70 | 0.77 | 0.70 |
| C | 0.89 | 0.84 | 0.74 | 0.89 | 0.84 | 0.93 | 0.72 | 0.61 | 0.70 | 0.64 |
| D | 1.00 | 0.98 | 0.95 | 0.99 | 0.93 | 0.98 | 0.85 | 0.83 | 0.80 | 0.77 |
| E | 1.00 | 1.00 | 0.98 | 1.00 | 0.95 | 0.98 | 0.87 | 0.85 | 0.83 | 0.80 |
| F | 1.00 | 0.98 | 0.96 | 1.00 | 0.89 | 0.97 | 0.86 | 0.82 | 0.79 | 0.73 |
| G | 1.00 | 0.96 | 0.95 | 1.00 | 0.93 | 0.99 | 0.88 | 0.85 | 0.86 | 0.82 |
| H | 1.00 | 1.00 | 0.98 | 1.00 | 0.92 | 0.96 | 0.86 | 0.83 | 0.84 | 0.79 |
| I | 0.96 | 0.96 | 0.93 | 0.99 | 0.94 | 0.96 | 0.77 | 0.72 | 0.75 | 0.71 |
| J | 0.94 | 0.93 | 0.89 | 0.95 | 0.89 | 0.93 | 0.74 | 0.69 | 0.72 | 0.66 |
| K | 1.00 | 1.00 | 0.97 | 0.95 | 0.97 | 0.96 | Not meaningful | | | |

### 3.3 Results

The results as ratios of correctly classified cases are presented in Table 1. For the mouth localization, the optimized centroids of methods D, F, and H turn out to outperform simple centroids (A, B, and C); the novel modifications E and G performing intrinsic variable selection yield the best results. Simple standard centroids (A, B, and C) are non-robust to data contamination; this follows from Section 2.1 and from analogous considerations for other types of contaminating the images. On the other hand, the robustness of optimized centroids is achieved by their optimization (but not by using $r_w$ as such). Methods E and G are even able to overcome methods I and J based on $r_{\mathsf{LWS}}$. We recall that $r_{LWS}$ is globally robust in terms of the breakdown point [4]), is computationally very demanding, and does not seem to allow any feasible optimization. Other results reported previously in [6] revealed that also numerous standard machine learning methods are too vulnerable (non-robust) with respect to data contamination, if measuring the similarity by $r$ or $r_w$.

For the AMI dataset, methods E and G with variable selection perform the best results for raw as well as contaminated datasets. For the simulated data, the method G yields the best results and the method E stays only slightly behind as the second best method.

## 4 Conclusions

Understanding the robustness of centroids represents a crucial question in image processing with applications for convolutional neural networks (CNNs), because centroids are very versatile tools that may be based on deep features learned by deep

learning. We focus on small datasets, for which CNNs cannot be used [10]. This paper is interested in performance of centroid-based object localization over small databases with non-standardized images, which commonly appear e.g. in medical image analysis.

The requirements on robustness with respect to modifications of the images turn out not to contradict the requirements on optimality of the centroids. The method G applying an intrinsic variable selection on the optimal centroid and weights [6] can be interpreted within a broader framework of robust dimensionality reduction (see [8] for an overview) or low-energy approximate computation. Additional results not presented here reveal the method based on optimized centroids to be robust also to small shift. Neither the theoretical part of this paper nor the experiments exploit any specific properties of faces. The presented robust method has potential also for various other applications, e.g. for deep fake detection by centroids, robust template matching by CNNs [9], or applying filters in convolutional layers of CNNs.

**Acknowledgements** The research was supported by the grant 22-02067S of the Czech Science Foundation.

## References

1. Böhringer, S., de Jong, M. A.: Quantification of facial traits. Frontiers in Genetics **10**, 397 (2019)
2. Delaigle, A., Hall, P.: Achieving near perfect classification for functional data. Journal of the Royal Statistical Society **74**, 267–286 (2012)
3. Gao, B., Spratling, M. W.: Robust template matching via hierarchical convolutional features from a shape biased CNN. ArXiv:2007.15817 (2021)
4. Jurečková, J., Picek, J., Schindler, M.: Robust statistical methods with R. 2nd edn. CRC Press, Boca Raton (2019)
5. Kalina, J.: A robust pre-processing of BeadChip microarray images. Biocybernetics and Biomedical Engineering **38**, 556–563 (2018)
6. Kalina, J., Matonoha, C.: A sparse pair-preserving centroid-based supervised learning method for high-dimensional biomedical data or images. Biocybernetics and Biomedical Engineering **40**, 774–786 (2020)
7. Kalina, J., Schlenker, A.: A robust supervised variable selection for noisy high-dimensional data. BioMed Research International **2015**, 320385 (2015)
8. Rousseeuw, P. J., Hubert, M.: Anomaly detection by robust statistics. WIREs Data Mining and Knowledge Discovery **8**, e1236 (2018)
9. Sun, L., Sun, H., Wang, J., Wu, S., Zhao, Y., Xu, Y.: Breast mass detection in mammography based on image template matching and CNN. Sensors **2021**, 2855 (2021)
10. Sze, V., Chen, Y. H., Yang, T. J., Emer, J. S.: Efficient processing of deep neural networks. Morgan & Claypool Publishers, San Rafael (2020)
11. Watanuki, S.: Watershed brain regions for characterizing brand equity-related mental processes. Brain Sciences **11**, 1619 (2021)
12. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. IEEE Conference on Computer Vision and Pattern Recognition 2012. IEEE, New York, pp. 2879–2886 (2012)

# Data Clustering and Representation Learning Based on Networked Data

Lazhar Labiod and Mohamed Nadif

**Abstract** To deal simultaneously with both, the attributed network embedding and clustering, we propose a new model exploiting both content and structure information. The proposed model relies on the approximation of the relaxed continuous embedding solution by the true discrete clustering. Thereby, we show that incorporating an embedding representation provides simpler and easier interpretable solutions. Experiment results demonstrate that the proposed algorithm performs better, in terms of clustering, than the state-of-art algorithms, including deep learning methods devoted to similar tasks.

**Keywords:** networked data, clustering, representation learning, spectral rotation

## 1 Introduction

In recent years, *Networks* [4] and *Attributed Networks* (AN) [8] have been used to model a large variety of real-world networks, such as academic and health care networks where both node links and attributes/features are available for analysis. Unlike plain networks in which only node links and dependencies are observed, with AN, each node is associated with a valuable set of features. In other words, we have $\mathbf{X}$ and $\mathbf{W}$ obtained/available independently of $\mathbf{X}$. More recently, the learning representation has received a significant amount of attention as an important aim in many applications including social networks, academic citation networks and protein-protein interaction networks. Hence, *Attributed network Embedding* (ANE) [2] aims to seek a continuous low-dimensional matrix representation for nodes in a network, such that original network topological structure and node attribute proximity can be preserved in the new low-dimensional embedding.

Although, many approaches have emerged with *Network Embedding* (NE), the research on ANE (Attributed Network Embedding) still remains to be explored

Lazhar Labiod (✉) · Mohamed Nadif
Centre Borelli UMR9010, Université Paris Cité, 75006-Paris, France,
e-mail: lazhar.labiod@u-paris.fr, e-mail: mohamed.nadif@u-paris.fr



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_23

[3]. Unlike NE that learns from plain networks, ANE aims to capitalize both the proximity information of the network and the affinity of node attributes. Note that, due to the heterogeneity of the two information sources, it is difficult for the existing NE algorithms to be directly applied to ANE. To sum up, the learned representation has been shown to be helpful in many learning tasks such as network clustering [13], Therefore ANE is a challenging research problem due to the high-dimensionality, sparsity and non-linearity of the graph data.

The paper is organized as follows. In Section 2 we formulate the objective function to be optimized, describe the different matrices used, and present a *Simultaneous Attributed Network Embedding and Clustering* (SANEC) framework for embedding and clustering. Section 3 is devoted to numerical experiments. Finally, the conclusion summarizes the advantages of our contribution.

## 2 Proposed Method

In this section, we describe the SANEC method. We will present the formulation of an objective function and an effective algorithm for data embedding and clustering. But first, we show how to construct two matrices $\mathbf{S}$ and $\mathbf{M}$ integrating both types of information –content and structure information– to reach our goal.

### 2.1 Content and Structure Information

An attributed network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ consists of $\mathcal{V}$ the set of nodes, $E \subseteq \mathcal{V} \times \mathcal{V}$ the set of links, and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ where $n = |\mathcal{V}|$ and $\mathbf{x}_i \in \mathbb{R}^d$ is the feature/attribute vector of the node $v_i$. Formally, the graph can be represented by two types of information, the content information $\mathbf{X} \in \mathbb{R}^{n \times d}$ and the structure information $\mathbf{A} \in \mathbb{R}^{n \times n}$, where $\mathbf{A}$ is an adjacency matrix of $G$ and $a_{ij} = 1$ if $e_{ij} \in E$ otherwise 0; we consider that each node is a neighbor of itself, then we set $a_{ii} = 1$ for all nodes. Thereby, we model the nodes proximity by an $(n \times n)$ transition matrix $\mathbf{W}$ given by $\mathbf{W} = \mathbf{D}^{-1}\mathbf{A}$, where $\mathbf{D}$ is the degree matrix of $\mathbf{A}$ defined by $d_{ii} = \sum_{i'=1}^{n} a_{i'i}$.

In order to exploit additional information about nodes similarity from $\mathbf{X}$, we preprocessed the above dataset $\mathbf{X}$ to produce similarity graph input $\mathbf{W}_\mathbf{X}$ of size $(n \times n)$; we construct a K-Nearest-Neighbor (`KNN`) graph. To this end, we use the heat kernel and $L_2$ distance, `KNN` neighborhood mode with $K = 15$ and we set the width of the neighborhood $\sigma = 1$. Note that any appropriate distance or dissimilarity measure can be used. Finally we combine in an $(n \times n)$ matrix $\mathbf{S}$, nodes proximity from both content information $\mathbf{X}$ and structure information $\mathbf{W}$. In this way, we intend to perturb the similarity $\mathbf{W}$ by adding the similarity from $\mathbf{W}_\mathbf{X}$; we choose to take $\mathbf{S}$ defined by $\mathbf{S} = \mathbf{W} + \mathbf{W}_\mathbf{X}$ (Figure 1).

As we aim to perform clustering, we propose to integrate it in the formulation of a new data representation by assuming that nodes with the same label tend to have

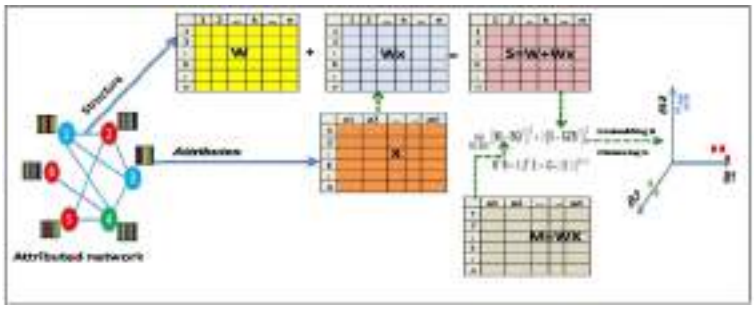


**Fig. 1** Model and objective function of SANEC.

similar social relations and similar node attributes. This idea is inspired by the fact that, the labels are strongly influenced by both content and structure information and inherently correlated to both these information sources. Thereby the new data representation referred to as $\mathbf{M} = (m_{ij})$ of size $(n \times d)$ can be considered as a multiplicative integration of both $\mathbf{W}$ and $\mathbf{X}$ by replacing each node by the centroid of their neighborhood (barycenter): i.e, $\mathbf{m}_{ij} = \sum_{k=1}^{n} \mathbf{w}_{ik}\mathbf{x}_{kj}, \forall i, j$ or $\mathbf{M} = \mathbf{WX}$. In this way, given a graph $G$, a graph clustering aims to partition the nodes in $G$ into $k$ disjoint clusters $\{C_1, C_2, \ldots, C_k\}$, so that: (1) nodes within the same cluster are close to each other while nodes in different clusters are distant in terms of graph structure; and (2) the nodes within the same cluster are more likely to have similar attribute values.

## 2.2 Model, Optimization and Algorithm

Let $k$ be the number of clusters and the number of components into which the data is embedded. With $\mathbf{M}$ and $\mathbf{S}$, the SANEC method that we propose aims to obtain the maximally informative embedding according to the clustering structure in the attributed network data. Therefore, we propose to optimize

$$\min_{\mathbf{B},\mathbf{Z},\mathbf{Q},\mathbf{G}} \left\|\mathbf{M} - \mathbf{B}\mathbf{Q}^\top\right\|^2 + \lambda\left\|\mathbf{S} - \mathbf{G}\mathbf{Z}\mathbf{B}^\top\right\|^2 \quad \mathbf{B}^\top\mathbf{B} = \mathbf{I}, \mathbf{Z}^\top\mathbf{Z} = \mathbf{I}, \mathbf{G} \in \{0,1\}^{n\times k} \quad (1)$$

where $\mathbf{G} = (g_{ij})$ of size $(n \times k)$ is a cluster membership matrix, $\mathbf{B} = (b_{ij})$ of size $(n \times k)$ is the embedding matrix and $\mathbf{Z} = (z_{ij})$ of size $(k \times k)$ is an orthonormal rotation matrix which most closely maps $\mathbf{B}$ to $\mathbf{G} \in \{0,1\}^{n\times k}$. $\mathbf{Q} \in \mathbb{R}^{d\times k}$ is the features embedding matrix. Finally, The parameter $\lambda$ is a non-negative value and can be viewed as a regularization parameter. The intuition behind the factorization of $\mathbf{M}$ and $\mathbf{S}$ is to encourage the nodes with similar proximity, those with higher similarity in both matrices, to have closer representations in the latent space given by $\mathbf{B}$. In doing so, the optimisation of (1) leads to a clustering of the nodes into $k$ clusters given by $\mathbf{G}$. Note that, both tasks –embedding and clustering– are performed

simultaneously and supported by $\mathbf{Z}$; it is the key to attaining good embedding while taking into account the clustering structure. To infer the latent factor matrices $\mathbf{Z}$, $\mathbf{B}$, $\mathbf{Q}$ and $\mathbf{G}$, we shall derive an alternating optimization algorithm. To this end, we rely on the following proposition.

**Proposition 1.** Let be $\mathbf{S} \in \mathbb{R}^{n\times n}$, $\mathbf{G} \in \{0,1\}^{n\times k}$, $\mathbf{Z} \in \mathbb{R}^{k\times k}$, $\mathbf{B} \in \mathbb{R}^{n\times k}$, we have

$$\left\|\mathbf{S} - \mathbf{G}\mathbf{Z}\mathbf{B}^\top\right\|^2 = \left\|\mathbf{S} - \mathbf{B}\mathbf{B}^\top\mathbf{S}\right\|^2 + \|\mathbf{S}\mathbf{B} - \mathbf{G}\mathbf{Z}\|^2 \tag{2}$$

**proof.** We first expand the matrix norm of the left term of (2)

$$\left\|\mathbf{S} - \mathbf{G}\mathbf{Z}\mathbf{B}^T\right\|^2 = \|\mathbf{S}\|^2 + \left\|\mathbf{G}\mathbf{Z}\mathbf{B}^\top\right\|^2 - 2Tr(\mathbf{S}\mathbf{G}\mathbf{Z}\mathbf{B}^\top) \tag{3}$$

In a similar way, we obtain from the two terms of the right term of (2)

$$\left\|\mathbf{S} - \mathbf{S}\mathbf{B}\mathbf{B}^T\right\|^2 = ||\mathbf{S}||^2 - ||\mathbf{S}\mathbf{B}||^2 \quad \text{due to } \mathbf{B}^\top\mathbf{B} = \mathbf{I} \tag{4}$$

$$\text{and} \quad \|\mathbf{S}\mathbf{B} - \mathbf{G}\mathbf{Z}\|^2 = \|\mathbf{S}\mathbf{B}\|^2 + \|\mathbf{G}\mathbf{Z}\|^2 - 2Tr(\mathbf{S}\mathbf{B}\mathbf{Z}\mathbf{G}^\top).$$

Due also to $\mathbf{B}^\top\mathbf{B} = \mathbf{I}$, we have

$$\|\mathbf{S}\mathbf{B} - \mathbf{G}\mathbf{Z}\|^2 = ||\mathbf{S}\mathbf{B}||^2 + ||\mathbf{G}\mathbf{Z}\mathbf{B}^\top||^2 - 2Tr(\mathbf{S}\mathbf{G}\mathbf{Z}\mathbf{B}^\top) \tag{5}$$

Summing the two terms of (4) and (5) leads to the left term of (2).

$$\|\mathbf{S}\|^2 + \|\mathbf{G}\mathbf{Z}\|^2 - 2Tr(\mathbf{S}\mathbf{G}\mathbf{Z}\mathbf{B}^\top) = \left\|\mathbf{S} - \mathbf{G}\mathbf{Z}\mathbf{B}^T\right\|^2 \text{ due to } \|\mathbf{G}\mathbf{Z}\|^2 = \left\|\mathbf{G}\mathbf{Z}\mathbf{B}^\top\right\|^2$$

**Compute Z**. Fixing $\mathbf{G}$ and $\mathbf{B}$ the problem which arises in (1) is equivalent to $\min_{\mathbf{Z}} \|\mathbf{S} - \mathbf{G}\mathbf{Z}\mathbf{B}^\top\|^2$. From Proposition 1, we deduce that

$$\min_{\mathbf{Z}} \left\|\mathbf{S} - \mathbf{G}\mathbf{Z}\mathbf{B}^\top\right\|^2 \Leftrightarrow \min_{\mathbf{Z}} \left\|\mathbf{S} - \mathbf{B}\mathbf{B}^\top\mathbf{S}\right\|^2 + \|\mathbf{S}\mathbf{B} - \mathbf{G}\mathbf{Z}\|^2 \tag{6}$$

which can be reduced to $\max_{\mathbf{Z}} Tr(\mathbf{G}^\top\mathbf{S}\mathbf{B}\mathbf{Z})$ s.t. $\mathbf{Z}^\top\mathbf{Z} = \mathbf{I}$. As proved in page 29 of [1], let $\mathbf{U}\Sigma\mathbf{V}^\top$ be the SVD for $\mathbf{G}^\top\mathbf{S}\mathbf{B}$, then $\mathbf{Z} = \mathbf{U}\mathbf{V}^\top$.

**Compute Q.** Given $\mathbf{G}$, $\mathbf{Z}$ and $\mathbf{B}$, the opimization problem (1) is equivalent to $\min_{\mathbf{Q}} \|\mathbf{M} - \mathbf{B}\mathbf{Q}^\top\|^2$, and we get

$$\mathbf{Q} = \mathbf{M}^\top\mathbf{B}. \tag{7}$$

Thereby $\mathbf{Q}$ is somewhere an embedding of attributes.

**Compute B**. Given $\mathbf{G}$, $\mathbf{Q}$ and $\mathbf{Z}$, the problem (1) is equivalent to

$$\max_{\mathbf{B}} Tr((\mathbf{M}^\top\mathbf{Q} + \lambda\mathbf{S}\mathbf{G}\mathbf{Z})\mathbf{B}^\top) \quad \text{s.t.} \quad \mathbf{B}^\top\mathbf{B} = \mathbf{I}.$$

In the same manner for the computation of $\mathbf{Z}$, let $\hat{\mathbf{U}}\hat{\Sigma}\hat{\mathbf{V}}^\top$ be the SVD for $(\mathbf{M}^\top\mathbf{Q} + \lambda\mathbf{SGZ})$, we get

$$\mathbf{B} = \hat{\mathbf{U}}\hat{\mathbf{V}}^\top. \tag{8}$$

It is important to emphasize that, at each step, $\mathbf{B}$ exploits the information from the matrices $\mathbf{Q}$, $\mathbf{G}$, and $\mathbf{Z}$. This highlights one of the aspects of the simultaneity of embedding and clustering.

**Compute G**: Finally, given $\mathbf{B}$, $\mathbf{Q}$ and $\mathbf{Z}$, the problem (1) is equivalent to $\min_{\mathbf{G}} \|\mathbf{SB} - \mathbf{GZ}\|^2$. As $\mathbf{G}$ is a cluster membership matrix, its computation is done as follows: We fix $\mathbf{Q}$, $\mathbf{Z}$, $\mathbf{B}$. Let $\tilde{\mathbf{B}} = \mathbf{SB}$ and calculate

$$g_{ik} = 1 \text{if } k = \arg\min_{k'} ||\tilde{\mathbf{b}}_i - \mathbf{z}_{k'}||^2 \text{ and 0 otherwise .} \tag{9}$$

In summary, the steps of the SANEC algorithm relying on $\mathbf{S}$ referred to as $\text{SANEC}_\mathbf{S}$ can be deduced in Algorithm 1. The convergence of $\text{SANEC}_\mathbf{S}$ is guaranteed and depends on the initialization to reach only a local optima. Hence, we start the algorithm several times and select the best result which minimizes the objective function (1).

**Algorithm 1** : $\text{SANEC}_\mathbf{S}$ algorithm

**Input:** $\mathbf{M}$ and $\mathbf{S}$ from structure matrix $\mathbf{W}$ and content matrix $\mathbf{X}$;
**Initialize:** $\mathbf{B}$, $\mathbf{Q}$ and $\mathbf{Z}$ with arbitrary orthonormal matrix;
**repeat**
(a) - Compute $\mathbf{G}$ using (9)
(b) - Compute $\mathbf{B}$ using (8)
(c) - Compute $\mathbf{Q}$ using (7)
(d) - Compute $\mathbf{Z}$ using (6)
**until** convergence
**Output:** $\mathbf{G}$: clustering matrix, $\mathbf{Z}$: rotation matrix, $\mathbf{B}$: nodes embedding and $\mathbf{Q}$: attributes embedding

## 3 Numerical Experiments

In the following, we compare SANEC with some competitive methods described later. The performances of all clustering methods are evaluated using challenging real-world datasets commonly tested with ANE where the clusters are known. Specifically, we consider three public citation network data sets, Citeseer, Cora and Wiki, which contain sparse bag-of-words feature vector for each document and a list of citation links between documents. Each document has a class label. We treat documents as nodes and the citation links as the edges. The characteristics of the used datasets are summarized in Table 1. The balance coefficient is defined as the ratio of the number of documents in the smallest class to the number of documents in the largest class while *nz* denotes the percentage of sparsity.

**Table 1** Description of datasets (#: the cardinality).

| datasets | # Nodes | # Attributes | # Edges | #Classes | $nz(\%)$ | Balance |
|---|---|---|---|---|---|---|
| Cora | 2708 | 1433 | 5294 | 7 | 98.73 | 0.22 |
| Citeseer | 3312 | 3703 | 4732 | 6 | 99.14 | 0.35 |
| Wiki | 2405 | 4973 | 17981 | 17 | 86.46 | 0.02 |

In our comparison we include standard methods and also recent deep learning methods; these differ in the way they use available information. Some of them (such as `K-means`) use only $\mathbf{X}$ as the baseline, while others use more recent algorithms based on $\mathbf{X}$ and $\mathbf{W}$. All the compared methods are: `TADW` [14], `DeepWalk` [7] and `Spectral Clustering` [11]. Using $\mathbf{X}$ and $\mathbf{W}$ we evaluated `GAE` and `VGAE` [5], `ARVGA` [6], `AGC` [15] and `DAEGC` [12].

With the SANEC model, the parameter $\lambda$ controls the role of the second term $||\mathbf{S}-\mathbf{GZB}^\top||^2$ in (1). To measure its impact on the clustering performance of $\texttt{SANEC}_\mathbf{S}$, we vary $\lambda$ in $\{0, 10^{-6}, 10^{-3}, 10^{-1}, 10^{0}, 10^{1}, 10^{3}\}$. Through, many experiments, as illustrated in Figure 2 we choose to take $\lambda = 10^{-3}$. The choice of $\lambda$ warrants in-depth evaluation.

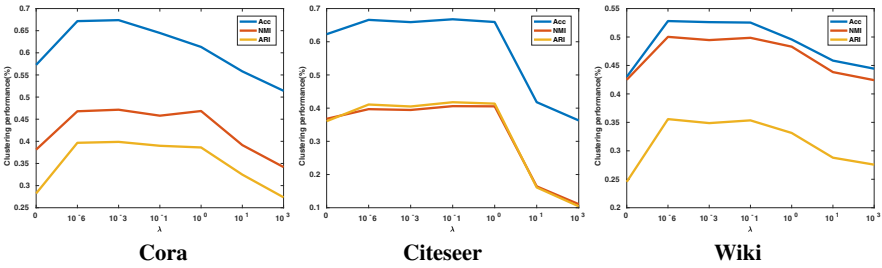


**Fig. 2** Sensitivity analysis of $\lambda$ using `ACC`, `NMI` and `ARI`.

Compared to the true available clusters, in our experiments the clustering performance is assessed by *accuracy* (ACC), *normalized mutual information* (NMI) and *adjusted rand index* (ARI). We repeat the experiments 50 times, with different random initialization and the averages (mean) are reported in Table 2; the best performance for each dataset is highlighted in bold.

First, we observe the high performances of methods integrating information from $\mathbf{W}$. For instance, `RTM` and `RMSC` are better than classical methods using only either $\mathbf{X}$ or $\mathbf{W}$. On the other hand, all methods including deep learning algorithms relying on $\mathbf{X}$ and $\mathbf{W}$ are better yet. However, regarding `SANEC` with both versions relying on $\mathbf{W}$, referred to as $\texttt{SANEC}_\mathbf{W}$ or $\mathbf{S}$ referred to as $\texttt{SANEC}_\mathbf{S}$, we note high performances for all the datasets and with $\texttt{SANEC}_\mathbf{S}$, we remark the impact of $\mathbf{W}_\mathbf{X}$; it learns low-dimensional representations while suits the clustering structure.

To go further in our investigation and given the sparsity of $\mathbf{X}$ we proceeded to standardization tf-idf followed by $L_2$, as it is often used to process document-term

matrices; see e.g, [9, 10], while in the construction of $\mathbf{W_X}$ we used the cosine metric. In Figure 3 are reported the results where we observe a slight improvement.

**Table 2** Clustering performances (ACC % , NMI % and ARI %).

| Methods | Input | Datasets | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Cora | | | Citeseer | | | Wiki | | |
| | | ACC | NMI | ARI | ACC | NMI | ARI | ACC | NMI | ARI |
| K-means | **X** | 49.22 | 32.10 | 22.96 | 54.01 | 30.54 | 27.86 | 41.72 | 44.02 | 15.07 |
| Spectral | **W** | 36.72 | 12.67 | 03.11 | 23.89 | 05.57 | 01.00 | 22.04 | 18.17 | 01.46 |
| DeepWalk | **W** | 48.40 | 32.70 | 24.27 | 33.65 | 08.78 | 09.22 | 38.46 | 32.38 | 17.03 |
| RTM | **X**, **W** | 43.96 | 23.01 | 16.91 | 45.09 | 23.93 | 20.26 | 43.64 | 44.95 | 13.84 |
| RMSC | **X**, **W** | 40.66 | 25.51 | 08.95 | 29.50 | 13.87 | 04.88 | 39.76 | 41.50 | 11.16 |
| TAWD | **X**, **W** | 56.03 | 44.11 | 33.20 | 45.48 | 29.14 | 22.81 | 30.96 | 27.13 | 04.54 |
| VGAE | **X**, **W** | 50.20 | 32.92 | 25.47 | 46.70 | 26.05 | 20.56 | 45.09 | 46.76 | 26.34 |
| ARGE | **X**, **W** | 64.0 | 44.9 | 35.2 | 57.3 | 35.0 | 34.1 | 47.34 | 47.02 | 28.16 |
| ARVGE | **X**, **W** | 63.8 | 45.0 | 37.74 | 54.4 | 26.1 | 24.5 | 46.45 | 47.8 | 29.65 |
| $\texttt{SANEC}_\mathbf{W}$ | **X**, **W** | 64.47 | 43.30 | 36.19 | 64.71 | 38.61 | 39.20 | 46.21 | 42.83 | 28.30 |
| $\texttt{SANEC}_\mathbf{S}$ | **X**, **S** | **67.38** | **47.14** | **39.88** | **66.77** | **40.60** | **41.78** | **52.80** | **50.02** | **35.57** |

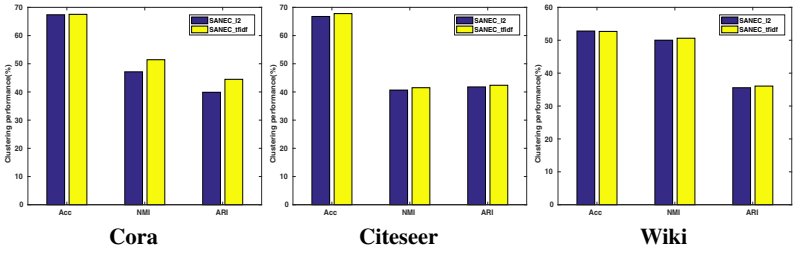


**Fig. 3** Evaluation of $\texttt{SANEC}_\mathbf{S}$ using tf-idf normalization of **X** and cosine metric for $\mathbf{W_X}$.

## 4 Conclusion

In this paper, we proposed a novel matrix decomposition framework for simultaneous attributed network data embedding and clustering. Unlike known methods that combine the objective function of AN embedding and the objective function of clustering separately, we proposed a new single framework to perform $\texttt{SANEC}_\mathbf{S}$ for AN embedding and nodes clustering. We showed that the optimized objective function can be decomposed into three terms, the first is the objective function of a kind of PCA applied to **X**, the second is the graph embedding criterion in a low-dimensional space, and the third is the clustering criterion. We also integrated a discrete rotation functionality, which allows a smooth transformation from the relaxed continuous embedding to a discrete solution, and guarantees a tractable optimization problem with a discrete solution. Thereby, we developed an effective algorithm capitalizing

on learning representation and clustering. The obtained results show the advantages of combining both tasks over other approaches. $\texttt{SANEC}_{\mathbf{S}}$ outperforms all recent methods devoted to the same tasks including deep learning methods which require deep models pretraining. However, there are other points that warrant in-depth evaluation, such as the choice of $\lambda$ and the complexity of the algorithm in terms of network size. The proposed framework offers several perspectives and investigations. We have noted that the construction of $\mathbf{M}$ and $\mathbf{S}$ is important, it highlights the introduction of $\mathbf{W}$. As for the $\mathbf{W_X}$ we have observed that it is fundamental as it makes possible to link the information from $\mathbf{X}$ to the network; this has been verified by many experiments. First, we would like to be able to measure the impact of each matrix $\mathbf{W}$ and $\mathbf{W_X}$ in the construction of $\mathbf{S}$ by considering two different weights for $\mathbf{W}$ and $\mathbf{W_X}$ as follows: $\mathbf{S} = \alpha\mathbf{W} + \beta\mathbf{W_X}$. Finally, as we have stressed that $\mathbf{Q}$ is an embedding of attributes, this suggests to consider also a simultaneously ANE and co-clustering.

## References


1. Ten Berge, J. M. F: Least Squares Optimization in Multivariate Analysis. DSWO Press, Leiden University Leiden, (1993)
2. Cai, H. Y., Zheng, V. W., Chang, K. C. C.: A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE Trans. Knowl. Data Eng. **30**(9), 1616-1637 (2018)
3. Chang, S., Han, W., Qi, G. J., Aggarwal, C. C., Huang, T.S.: Heterogeneous network embedding via deep architectures. In SIGKDD, pp. 119–128 (2015)
4. Doreian, P., Batagelj, V., Ferligoj, A.: Advances in network clustering and blockmodeling. John Wiley & Sons (2020)
5. Kipf, T. N., Welling, M.: Variational graph auto-encoders. In NIPS Workshop on Bayesian Deep Learning, (2016)
6. Pan, S., Hu, R., Long, G., Jiang, J., Yao, L., Zhang, C.: Adversarially regularized graph autoencoder for graph embedding. In IJCAI, pp. 2609-2615, (2018)
7. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In SIGKDD, pp. 701-710 (2014)
8. Qi, G.J., Aggarwal, C. C., Tian, Q., Ji, H., Huang, T. S.: Exploring context and content links in social media: A latent space method. IEEE Trans. Pattern Anal. Mach. Intell. **34**(5), 850-862 (2012)
9. Salah, A., Nadif, M.: Model-based von Mises-Fisher co-clustering with a conscience. In SDM, pp. 246–254. SIAM (2017)
10. Salah, A., Nadif, M.: Directional co-clustering. Data Analysis and Classification. **13**(3), 591-620 (2019)
11. Tang, L., Liu, H.: Leveraging social media networks for classification. Data mining and knowledge discovery. **23**(3), 447-478 (2011)
12. Wang, C., Pan, S., Hu, R., Long, G., Jiang, J., Zhang, C.: Attributed graph clustering: A deep attentional embedding approach. arXiv preprint arXiv:1906.06532 (2019) Available via . `https://arxiv.org/pdf/1906.06532.pdf`
13. Wang, C., Pan, S., Long, G., Zhu,X., Jiang, J.: Mgae: Marginalized graph autoencoder for graph clustering. In CIKM, pp. 889-898, (2017)
14. Yang, C., Liu, Z., Zhao, D., Sun, M., Chang, E. Y.: Network representation learning with rich text information. In IJCAI, pp. 2111-2117 (2015)
15. Zhang, X., Liu, H., Li, Q., Wu, X. M.: Attributed graph clustering via adaptive graph convolution. arXiv preprint arXiv:1906.01210, (2019) Available via . `https://arxiv.org/pdf/1906.01210.pdf?ref=https://githubhelp.com`

# Towards a Bi-stochastic Matrix Approximation of $k$-means and Some Variants

Lazhar Labiod and Mohamed Nadif

**Abstract** The $k$-means algorithm and some $k$-means variants have been shown to be useful and effective to tackle the clustering problem. In this paper we embed $k$-means variants in a bi-stochastic matrix approximation (BMA) framework. Then we derive from the $k$-means objective function a new formulation of the criterion. In particular, we show that some $k$-means variants are equivalent to algebraic problem of bi-stochastic matrix approximation under some suitable constraints. For optimizing the derived objective function, we develop two algorithms; the first one consists in learning a bi-stochastic similarity matrix while the second seeks for the optimal partition which is the equilibrium state of a Markov chain process. Numerical experiments on real data-sets demonstrate the interest of our approach.

**Keywords:** $k$-means, reduced $k$-means, factorial $k$-means, bi-stochastic matrix

## 1 Introduction

These last decades unsupervised learning and specifically clustering, have received a significant amount of attention as an important problem with many application in data science. Let $A = (a_{ij})$ be a $n \times m$ continuous data matrix where the set of rows (objects, individuals) is denoted by $I$ and the set of columns (attributes, features) by $J$. Many clustering methods such as hierarchical or not aim to construct an optimal partition of $I$ or, sometimes of $J$.

In this paper we show how some $k$-means variants can be presented as a bi-stochastic matrix approximation problem under some suitable constraints generated by the properties of the reached solution. To reach this goal, we first demonstrate that some variants of $k$-means are equivalent to learning a bi-stochastic similarity matrix having a diagonal block structure. Based on this formulation, referred to as BMA, we derive two iterative algorithms, the first algorithm learns a bi-stochastic $n \times n$ similarity matrix while the second directly seeks an optimal clustering solution.

Our main contribution is to establish the theoretical connection of the conventional $k$-means and some of its variants to BMA framework. The implications of the reformulation of $k$-means as a BMA problem are multi-folds:

Lazhar Labiod (✉) · Mohamed Nadif
Centre Borelli UMR9010, Université Paris Cité, 75006-Paris, France,
e-mail: lazhar.labiod@u-paris.fr, e-mail: mohamed.nadif@u-paris.fr



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_24

- It makes connections with recent clustering methods like spectral clustering and subspace clustering.
- It learns a well normalized (bi-stochastic normalization) similarity matrix, beneficial for spectral clustering [12].
- Unlike existing spectral and subspace methods which combine in a sequential way, the steps of similarity learning and clustering derivation, our proposed method jointly learns a block diagonal bi-stochastic affinity matrix which naturally expresses a clustering structure.

The rest of paper is organized as follows. Section 2 introduces some variants of $k$-means. Section 3 provides *Matrix Factorization* (MF) and BMA formulations of $k$-means variants. Section 4 discusses the BMA clustering algorithm and section 5 is devoted to numerical experiments. Finally, the conclusion summarizes the interest of our contribution.

## 2 Variants of $k$-Means

Given a data matrix $A = (a_{ij}) \in R^{n \times m}$, the aim of clustering is to cluster the rows or the columns of $A$, so as to optimize the difference between $A = (a_{ij})$ and the clustered matrix revealing significant block structure. More formally, we seek to partition the set of rows $I = \{1, \ldots, n\}$ into $k$ clusters $C = \{C_1, \ldots, C_l, \ldots, C_k\}$. The partitioning naturally induce clustering index matrix $R = (r_{il}) \in \mathbb{R}^{n \times k}$, defined as binary classification matrix such as we have $r_{il} = 1$, if the row $a_i \in C_l$, and 0 otherwise. On the other hand, we note $S \in \mathbb{R}^{m \times k}$ a reduced matrix specifying the cluster representation. The detection of homogeneous clusters of objects can be reached by looking for the two matrices $R$ and $S$ minimizing the total squared residue measure

$$\mathcal{J}_{KM}(R, S) = ||A - RS^\top||^2 \tag{1}$$

The term $RS^\top$ characterizes the information of $A$ that can be described by the clusters structure. The clustering problem can be formulated as a matrix approximation problem where the clustering aims to minimize the approximation error between the original data $A$ and the reconstructed matrix based on the cluster structures.

Factorial $k$-means analysis (FKM) [9] and Reduced $k$-means analysis (RKM) [1] are clustering methods that aim at simultaneously achieving a clustering of the objects and a dimension reduction of the features. The advantage of these methods is that both clustering of objects and low-dimensional subspace capturing the cluster structure are simultaneously obtained. To achieve this objective, RKM is defined by the minimizing problem of the following criterion

$$\mathcal{J}_{RKM}(R, S, Q) = ||A - RS^\top Q^\top||^2 \tag{2}$$

and FKM is defined by the minimizing problem of the following criterion

$$\mathcal{J}_{FKM}(R, S, Q) = ||AQ - RS^\top||^2 \tag{3}$$

where $S \in \mathbb{R}^{p\times k}$ with RKM and FKM, and $Q$ is an $m$ by $p$ column-wise orthonormal loading matrix.

## 3 Bi-stochastic Matrix Approximation of $k$-Means Variants

### 3.1 Low-rank Matrix Factorization (MF)

By considering $k$-means as a lower rank matrix factorization with constraints, rather than a clustering method, we can formulate constraints to impose on MF formulation. Let $D_r^{-1} \in \mathcal{R}^{k\times k}$ be diagonal matrix defined as follow $D_r^{-1} = Diag(r_1^{-1}, \ldots, r_k^{-1})$. Using the matrices $D_r, A$ and $R$, the matrix summary $S$ can be expressed as $S^T = D_r^{-1}R^\top A$. Plugging $S$ into the objective function in equation, (1) leads to optimize $||A - R(D_r^{-1}R^\top A)||^2$ equal to

$$\mathcal{J}_{MF-KM}(\mathbf{R}) = ||A - \mathbf{R}\mathbf{R}^\top A||^2, \text{ where } \mathbf{R} = RD_r^{-0.5}. \tag{4}$$

On the other hand, it is easy to verify that the approximation $\mathbf{R}\mathbf{R}^\top A$ of $A$ is formed by the same value in each block $A_{l,(l=1,\ldots,k)}$. Specifically, the matrix $\mathbf{R}^\top A$, equal to $S^T$, plays the role of a summary of $A$ and absorbs the different scales of $A$ and $\mathbf{R}$. Finally $\mathbf{R}\mathbf{R}^\top A$ gives the row clusters mean vectors. Note that it is easy to show that $\mathbf{R}$ verifies the following properties

$$\mathbf{R} \geq 0, \mathbf{R}^\top\mathbf{R} = I_k, \mathbf{R}\mathbf{R}^\top\mathbb{1} = \mathbb{1}, Trace(\mathbf{R}\mathbf{R}^\top) = k, (\mathbf{R}\mathbf{R}^\top)^2 = \mathbf{R}\mathbf{R}^\top \tag{5}$$

Next, in similar way, we can derive a MF formulation of FKM,

$$\mathcal{J}_{MF-FKM}(\mathbf{R}) = ||AQ - \mathbf{R}\mathbf{R}^\top AQ||^2, \tag{6}$$

$$\text{and of RKM, } \mathcal{J}_{MF-RKM}(\mathbf{R}) = ||A - \mathbf{R}\mathbf{R}^\top AQQ^\top||^2. \tag{7}$$

### 3.2 BMA Formulation

Let $\mathbf{\Pi} = \mathbf{R}\mathbf{R}^\top$ be a bi-stochastic similarity matrix, before giving the BMA formulation of $k$-means variants, we need first to spell out the good properties of $\mathbf{\Pi}$. Indeed, by construction from $\mathbf{R}$, $\mathbf{\Pi}$ has at least the following properties reported below that can be easily proven.

$$\mathbf{\Pi} \geq 0, \mathbf{\Pi}^\top = \mathbf{\Pi}, \mathbf{\Pi}\mathbb{1} = \mathbb{1}, Trace(\mathbf{\Pi}) = k, \mathbf{\Pi}\mathbf{\Pi}^\top = \mathbf{\Pi}, Rank(\mathbf{\Pi}) = k \tag{8}$$

Given a data matrix $A$ and $k$ row clusters, we can hope to discover the cluster structure of $A$ from $\mathbf{\Pi}$. Notice that from (8) $\mathbf{\Pi}$ is nonnegative, symmetric, bi-stochastic (doubly

stochastic) and idempotent. By setting the $k$means in the BMA framework, the problem of clustering is reformulated as the learning of a structured bi-stochastic similarity matrix $\mathbf{\Pi}$ by minimizing the following $k$-means variants objective,

$$\mathcal{J}_{BMA-kM}(\mathbf{\Pi}) = ||A - \mathbf{\Pi}A||^2, \tag{9}$$

$$\mathcal{J}_{BMA-FKM}(\mathbf{\Pi}) = ||AQ - \mathbf{\Pi}AQ||^2, \tag{10}$$

$$\mathcal{J}_{BMA-RKM}(\mathbf{\Pi}) = ||A - \mathbf{\Pi}AQQ^\top||^2, \tag{11}$$

with respect to the following constraints on $\mathbf{\Pi}$

$$\mathbf{\Pi} \geq 0, \mathbf{\Pi} = \mathbf{\Pi}^\top, \mathbf{\Pi}\mathbb{1} = \mathbb{1}, Tr(\mathbf{\Pi}) = k, \mathbf{\Pi}\mathbf{\Pi}^\top = \mathbf{\Pi} \tag{12}$$

$$\text{and } Q^\top Q = I \quad \text{for equations (10) and (11).}$$

In the rest of the paper, we will consider only non-negativity, symmetry and bi-stochastic constraints.

### 3.3 The Equivalence Between BMA and $k$-Means

The theorem below demonstrates that the optimization of the $k$-means objective and the BMA objective under some suitable constraints are equivalent. The equation (13) establishes the equivalence between $k$-means and the BMA formulation. Then, solving the BMA objective function (9) is equivalent to finding a global solution of the $k$-means criterion (1).

**Theorem 1**

$$\arg\min_{R,S} ||A - RS^\top||^2 \Leftrightarrow \arg \min_{\{\mathbf{\Pi}\geq 0, \mathbf{\Pi}=\mathbf{\Pi}^\top, \mathbf{\Pi}\mathbb{1}=\mathbb{1}, Tr(\mathbf{\Pi})=k, \mathbf{\Pi}\mathbf{\Pi}^\top=\mathbf{\Pi}\}} ||A - \mathbf{\Pi}A||^2 \tag{13}$$

The proof of this equivalence is given in the appendix. Note that this new formulation gives some interesting highlights on $k$-means and its variants:

- First, this shows that $k$-means is equivalent to learning a structured bi-stochastic similarity matrix which is normalized bi-stochastic matrix with block diagonal structure.
- Secondly, it establishes very interesting connections of $k$-means to many state-of-the-art subspace clustering methods [10, 5]. Moreover, this formulation combines the traditional two-step process used by subspace clustering methods, which consist in first constructing an affinity matrix between data points and then applying spectral clustering to this affinity. This allows joint learning of a similarity matrix that better reflects the clustering structure by its block diagonal shape.
- Finally, it allows to apply the spirit of $k$-means for graph or similarity data.

## 4 BMA Clustering Algorithm

First, we establish the relationship between our objective function and that used in [12, 11]. From $||A - \mathbf{\Pi}A||^2 = Trace(AA^\top) + Trace(\mathbf{\Pi}AA^\top\mathbf{\Pi}) - 2Trace(AA^\top\mathbf{\Pi})$ and by using the idempotent property, $\mathbf{\Pi}\mathbf{\Pi}^\top = \mathbf{\Pi}$ , we can show that

$$\arg\min_{\mathbf{\Pi}} ||A - \mathbf{\Pi}A||^2 \Leftrightarrow \arg\min_{\mathbf{\Pi}} ||AA^\top - \mathbf{\Pi}||^2 \Leftrightarrow \arg\max_{\Pi} Trace(AA^\top\mathbf{\Pi}).$$

The algorithm for learning similarity matrix is summarized in Algorithm 1 as in [12, 11]. Once the bi-stochastic similarity matrix $\mathbf{\Pi}$ is obtained, the basic idea of `BMA` is based on the following steps:

**Algorithm 1** : Learning similarity matrix

**Input:** data $A$
**Output:** similarity matrix $\mathbf{\Pi}$
**Initialize:** $t = 0$ and $\mathbf{\Pi}^{(0)} = AA^\top$
**repeat**
$\Pi^{(t+1)} \leftarrow [\mathbf{\Pi}^{(t)} + \frac{1}{n}(I - \mathbf{\Pi}^{(t)} + \frac{\mathbb{1}\mathbb{1}^\top\mathbf{\Pi}^{(t)}}{n})\mathbb{1}\mathbb{1}^\top - \frac{1}{n}\mathbb{1}\mathbb{1}^\top\mathbf{\Pi}^{(t)}]$
**until** Satisfied convergence condition

1. Estimating iteratively $A$ by applying at each time the matrix $\mathbf{\Pi}$ on the current $A$ using the following update $\hat{A}^{(t+1)} = \mathbf{\Pi}A^{(t)}$. This process converges to an equilibrium (steady) state. Let $k$ be the multiplicity of the eigenvalue of matrix $\mathbf{\Pi}$ equal to 1, $\hat{A}$ is composed of $k << n$ quasi-similar rows, where each row is represented by its prototype.
2. Extracting the first left singular vectors $\pi$ of $\hat{A}$ using the Power method [4]; it is a well-known technique used for computing the largest left eigenvector of data matrix. The numerical computation of the leading left singular vector of $\hat{A}$, consists in starting with an arbitrary vector $\pi^{(0)}$, repeatedly performing updates of $\pi$ until stabilization of $\pi$ as follow: $\pi^{(t+1)} = \hat{A}\hat{A}^\top\pi^{(t)}$and $\pi^{(t)} \leftarrow \frac{\pi^{(t)}}{||\pi^{(t)}||}$. We stop the Power method if, $|\gamma^{(t+1)} - \gamma^{(t)}| \simeq \epsilon$ where $\gamma^{(t+1)} \leftarrow ||\pi^{(t+1)} - \pi^{(t)}||$.

**Why does this work?** At first glance, this process might seem uninteresting since it eventually leads to a vector with all rows and columns coincide for any starting vector. However our practical experience shows that, first the vectors $\pi$ very quickly collapse into rows blocks and these blocks move towards each other relatively slowly. If we stop the Power method iteration at this point, the algorithm would have a potential application for data visualization and clustering. The structure of $\pi$ during short-run stabilization makes the discovery of rows data ordering straightforward. The key is to look for values of $\pi$ that are approximately equal and reordering rows and columns data accordingly. The `BMA` algorithm involves a reorganization of the rows of data matrix $\hat{A}$ according to sorted $\pi$. It also allows to locate the points corresponding to an abrupt change in the curve of the first left singular vector $\pi$, and then assess the number of clusters and the rows belonging to each cluster.

## 5 Experiments Analysis

In this subsection we first ran our algorithm on two real world data set, the 16 townships data which consists of the characteristics (rows) of 16 townships (columns), each cell indicates the presence 1 or absence 0 of a characteristic on a township . This example has been used by Niermann [7] for data ordering task and the author aims to reveal a block diagonal form. The second data called Mero data, comes from archaeological data on Merovingian buckles found in north eastern France. This data matrix consists of 59 buckles characterized by 26 attributes of description (see Marcotorchino for more details [6]). Figure 1 shows in order, $A$, $\hat{A}$, $SR = AA^T$ reorganized according to the sorted $\pi$ and the sorted $\pi$ plot for both data sets. We also evaluated

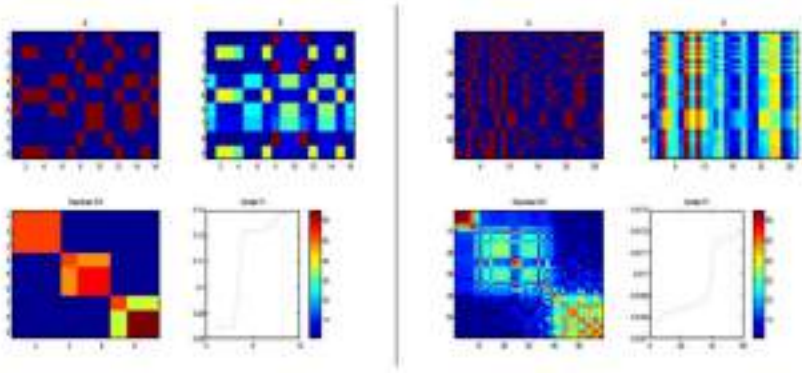

**Fig. 1** left: 16 Townships data - right: Mero data.

the performances of `BMA` on some real challenging datasets described in Table1. We compared the performance of `BMA` with the spectral co-clustering (`SpecCo`) [2], Non-negative Matrix Factorization (`NMF`) and Orthgogonal Non-negative Matrix Tri-Factorization (`NMTF`) [3] by using two evaluation metrics: accuracy (ACC) corresponding to the percentage of well-classified elements and the normalized mutual information (NMI) [8]. In Table 1, we observe that `BMA` outperforms all compared algorithms for all tested datasets.

**Table 1** Clustering Accuracy and Normalized Mutual Information (%).

| datasets | # samples | # features | # classes | per | $k$-means | NMF | ONMTF | SpecCO | BMA |
|---|---|---|---|---|---|---|---|---|---|
| Classic3 | 3891 | 4303 | 3 | ACC | 88.6 | 73.33 | 70.10 | 97.89 | 98.30 |
| | | | | NMI | 74.9 | 51.46 | 51.46 | 91.17 | 91.91 |
| CSTR | 476 | 1000 | 4 | ACC | 76.3 | 75.30 | 77.41 | 80.21 | 90.73 |
| | | | | NMI | 65.4 | 66.40 | 67.30 | 66.36 | 77.86 |
| Webkb4 | 4199 | 1000 | 4 | ACC | 60.10 | 66. 30 | 67.10 | 61.68 | 68.8 |
| | | | | NMI | 45.7 | 42.70 | 45.36 | 48.64 | 49 |
| Leukemia | 38 | 5000 | 3 | ACC | 72.2 | 89.21 | 90.32 | 94.73 | 97.36 |
| | | | | NMI | 19.4 | 75.42 | 80.50 | 82 | 90.69 |

## 6 Conclusion

In this paper we have presented a new reformulation of some variants of $k$-means as a unified BMA framework and established the equivalence between $k$-means and `BMA` under suitable constraints. By doing so, $k$-means leads to learning a structured bi-stochastic matrix which is beneficial for clustering task. The proposed approach, not only learns a similarity matrix from data matrix, but uses this matrix in an iterative process that converges to a matrix $\hat{A}$ in which each row is represented by its prototype. The clustering solution is given by the first left eigenvector of $\hat{A}$ while overcoming the knowledge of the number of clusters. We expect for future work to integrate the idempotent and trace constraints on $\mathbf{\Pi}$ to make the approximate similarity matrix fits the best the case of a block diagonal structure.

## Appendix

From the BMA's formulation, we know that one can easily construct a feasible solution for $k$-means from a feasible solution of BMA's formulation. Therefore, it remains to show that from a global solution of BMA's formulation, we can obtain a feasible solution of $k$-means. In order to show the equivalence between the optimization of $k$-means formulation and the BMA formulation, we first consider the following lemma.

**Lemma** If $\Pi$ is a symmetric and positive semi-definite matrix, then we have

$$\begin{cases} (a)\pi_{ii'} \leq \sqrt{\pi_{ii}\pi_{i'i'}} & \text{(geometric mean) } \forall i, i' \\ (b)\pi_{ii'} \leq \frac{1}{2}(\pi_{ii} + \pi_{i'i'}) & \text{(arithmetic mean) } \forall i, i' \\ (c)\ \max_{ii'} \pi_{ii'} = \max_i \pi_{ii} & \\ (d)\pi_{ii} = 0 \Rightarrow \pi_{ii'} = \pi_{i'i} = 0\ \forall i, i' & \end{cases}$$

**Proposition.** Any positive semi-definite matrix $\Pi$ satisfying the constraints:

$$\begin{cases} \pi_{ii'} = \pi_{i'i}\ \ \forall i, i' & \text{(symmetry)} \\ \pi_{ii'} = \sum_{i''} \pi_{ii''}\pi_{i'i''}\ \ \forall i, i' & \text{(idempotence)} \\ \sum_{i'} \pi_{ii'} = 1\ \ \forall i & \\ \sum_i \pi_{ii} = k & \end{cases}$$

is a matrix partitioned into $k$ blocks $\Pi = diag(\Pi^1, \ldots, \Pi^l, \ldots, \Pi^k)$ with $\Pi^l = \frac{1}{n_l}\mathbb{1}_l\mathbb{1}_l^t$, $trace(\Pi^l) = 1\ \forall l$ and $\sum_{l=1}^k n_l = n$; $\mathbb{1}_l$ denotes the vector of appropriate dimension with all its values are 1.

**Proof.** Since $\Pi$ is idempotent ($\Pi^2 = \Pi$), we have: $\forall i;\ \pi_{ii} = \sum_{i'} \pi_{ii'}^2$ From the Lemma above, we know that there exist; $i^0 \in \{1, 2, \ldots, n\}$ such as $\max_{ii'} \pi_{i'i} = \pi_{i^0 i^0} > 0$. Consider the set $A_{i^0}$ defined by $A_{i^0} = \{i | \pi_{i^0 i} > 0\}$, we can rewrite; $\forall i \in A_{i^0};\ \pi_{ii} = \sum_{i' \in A_{i^0}} \pi_{i'i}^2$

$$\forall i \in A_{i^0};\quad \sum_{i' \in A_{i^0}} \pi_{i'i} = \sum_{i' \in I} \pi_{i'i} = 1 \tag{14}$$

and,

$$\sum_{i' \in A_{i^0}} \sum_{i \in A_{i^0}} \pi_{i'i} = \sum_{i \in A_{i^0}} \pi_{i.} = \sum_{i \in A_{i^0}} 1 = |A_{i^0}| \tag{15}$$

$$\forall i \ \ \pi_{ii} = \sum_{i'} \pi_{ii'}^2 \Rightarrow \forall i \in A_{i^0}; \quad \sum_{i' \in A_{i^0}} \frac{\pi_{ii'}^2}{\pi_{ii}} = \sum_{i' \in A_{i^0}} (\frac{\pi_{ii'}}{\pi_{ii}})\pi_{ii'} = 1. \tag{16}$$

From (14) and (16), we deduce that $\forall i \in A_{i^0}; \ \ \sum_{i' \in A_{i^0}} \pi_{i'i} = \sum_{i' \in A_{i^0}} (\frac{\pi_{ii'}}{\pi_{ii}})\pi_{ii'}$, implying that: $\pi_{ii'} = \pi_{ii}, \ \ \forall i, i' \in A_{i^0}$. Substituting in (15) $\pi_{ii'}$ by $\pi_{ii}$ for all $i, i' \in A_{i^0}$ leads to $\sum_{i' \in A_{i^0}} \pi_{ii'} = \sum_{i' \in A_{i^0}} \pi_{ii} = |A_{i^0}|\pi_{ii} = 1, \ \ \forall i \in A_{i^0}$. From this we can deduce that $\pi_{ii} = \pi_{ii'} = \frac{1}{|A_{i^0}|}, \ \ \forall i, i' \in A_{i^0}$. We can therefore rewrite the matrix $\Pi$ in the form of a block diagonal matrix $\Pi = \begin{pmatrix} \Pi^0 & 0 \\ 0 & \bar{\Pi}^0 \end{pmatrix}$ where $\Pi^0$ is a block matrix whose general term is defined by $\Pi^0_{ii'} = \frac{1}{|A_{i^0}|}, \ \forall i, i' \in A_{i^0}$ and $trace(\Pi^0) = 1$. The matrix $\bar{\Pi}^0$ is a positive semi-definite matrix which also verified the constraints $(\bar{\Pi}^0)^t = \bar{\Pi}^0, \ \ \bar{\Pi}^0 \mathbb{1} = \mathbb{1}, \ \ (\bar{\Pi}^0)^2 = \bar{\Pi}^0$ and $trace(\bar{\Pi}^0) = k - 1$.
By repeating the same process $k - 1$ times, we get the block diagonal form of $\Pi$. $\Pi = diag(\Pi^0, \Pi^1, \ldots, \Pi^l, \ldots, \Pi^{k-1})$ with, $\Pi^l = \frac{1}{n_l} \mathbb{1}_l \mathbb{1}_l^t, trace(\Pi^l) = 1 \forall l$ and $\sum_{l=0}^{k-1} n_l = n$.

## References


1. De Soete, G., Carroll, J. D.: K-means clustering in a low-dimensional euclidean space. In: E. Diday et al. (eds.) New Approaches in Classification and Data Analysis, pp. 212–219. Springer-Verlag Berlin (1994)
2. Dhillon, I. S.: Co-clustering documents and words using bipartite spectral graph partitioning. In SIGKDD, pp. 269-274 (2001)
3. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix trifactorizations for clustering. In SIGKDD, pp. 126-135 (2006)
4. Golub, G. H., van Loan, C. F.: Matrix Computations (3rd ed.). Johns Hopkins University Press (1996)
5. Lim, D., Vidal, R., Haeffele, B. D.: Doubly stochastic subspace clustering. ArXiv, abs/2011.14859, 2020. Available via ArXiv. `https://arxiv.org/abs/2011.14859`
6. Marcotorchino, J. F.: Seriation problems: an overview. Appl. Stoch. Model. D. A., **7(2)**, 139–151 (1991)
7. Niermann, S.: Optimizing the ordering of tables with evolutionary computation. American Statistician. **59**(1), 41-46 (2005)
8. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research, **3**, 583-617 (2002)
9. Vichi, M., Kiers, H. A.: Factorial $k$-means analysis for two-way data. CSDA, **37**(1), 49-64 (2001)
10. Vidal, R.: Subspace clustering. IEEE Signal Processing Magazine **28**(2), 52-68 (2011)
11. Wang, F., Li, P., König, A. C.: Improving clustering by learning a bi-stochastic data similarity matrix. Knowl. Inf. Syst. **32**(2), 351-382 (2012)
12. Zass, R., Shashua, A.: A unifying approach to hard and probabilistic clustering. In ICCV, pp. 294-301 (2005)


# Clustering Adolescent Female Physical Activity Levels with an Infinite Mixture Model on Random Effects

Amy LaLonde, Tanzy Love, Deborah R. Young, and Tongtong Wu

**Abstract** Physical activity trajectories from the Trial of Activity in Adolescent Girls (TAAG) capture the various exercise habits over female adolescence. Previous analyses of this longitudinal data from the University of Maryland field site, examined the effect of various individual-, social-, and environmental-level factors impacting the change in physical activity levels over 14 to 23 years of age. We aimed to understand the differences in physical activity levels after controlling for these factors. Using a Bayesian linear mixed model incorporating a model-based clustering procedure for random deviations that does not specify the number of groups *a priori*, we find that physical activity levels are starkly different for about 5% of the study sample. These young girls are exercising on average 23 more minutes per day.

**Keywords:** Bayesian methodology, Markov chain Monte Carlo, mixture model, reversible jump, split-merge procedures

## 1 Introduction

Physical activity and diet are arguably the two main controllable factors having the greatest impact on our health. Whereas we have little to no control over factors like our genetic predisposition to disease or exposure to environmental toxins, we have

Amy LaLonde
University of Rochester, NY, USA, e-mail: amylalonde2@gmail.com

Tanzy Love (✉)
University of Rochester, NY, USA, e-mail: tanzy_love@urmc.rochester.edu

Deborah Rohm Young
University of Maryland, MD, USA, e-mail: dryoung@umd.edu

Tongtong Wu
University of Rochester, NY, USA, e-mail: tongtong_wu@urmc.rochester.edu



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_25

much greater control over our diet and activity levels. Despite our ability to choose to engage in healthy behaviors such as exercising and eating a healthy diet, these choices are plagued with the complexity of human psychology and the modern demands and distractions that pervade our lives today. Several factors influence levels of physical activity; we explore the factors impacting female adolescents using longitudinal data.

The University of Maryland, one of the six initial university field centers of the Trial of Activity in Adolescent Girls (TAAG), selected to follow its 2006 $8^{th}$ grade cohort for two additional time points over adolescence: $11^{th}$ grade and 23 years of age. The females were therefore measured roughly at ages 14, 17, and 23. In these waves, there was no intervention as this observational longitudinal study aimed at exploring the patterns of physical activity levels and associated factors over time.

The model presented in Wu et al. [1] motivates the current work. We fit a similar linear mixed model controlling for the same variables. Rather than cluster the raw physical activity trajectories to identify groups, we cluster the females within the model-fitting procedure based on the values of the subject-specific deviations from the adjusted physical activity levels. Fitting a Bayesian linear mixed model, we simultaneously explore the subject groups through the use of reversible jump Markov chain Monte Carlo (MCMC) applied to the random effects. Bayesian model-based clustering methods have been applied within linear mixed models to identify groups by clustering the fitted values of the dependent variable. For example, [2] fits cluster-specific linear mixed models to the gene expression outcome using an EM algorithm and [3] clusters gene expression in a similar fashion, except using Bayesian methods. In contrast, we perform the clustering on the random effects, which allows us to investigate the variability that is unexplained by the covariates of interest. This methodology is advantageous because of its ability to jointly estimate all effects, while also exploring the infinite space of group arrangements.

## 2 Bayesian Mixture Models for Heterogeneity of Random Effects

Let $\mathbf{y}_i = (y_{i,1}, \ldots, y_{i,T})$ be the $i^{th}$ subject's average daily moderate-to-vigorous physical activity (MVPA) at each of the $T = 3$ time points. The MVPA was collected from ActiGraph accelerometers (Manufacturing Technologies Inc. Health Systems, Model 7164, Shalimar, FL) worn for seven consecutive days. Accelerometers offered a great alternative to self-report for tracking physical activity levels, and measuring over seven days helped to account for differences in activity patterns during weekdays and weekends. Wu et al. [1] analyzed this cohort using mixed models that accounted for the subject-specific variability. We let $\mathbf{X}_i$ represent the $i^{th}$ subject's values for covariates.

Furthermore, let $\mathbf{r} = (r_1, \ldots, r_n)$ represent the subject-specific random effects for the $n$ subjects. The simple linear mixed model is written in terms of each subject as

$$\mathbf{y}_i = \mathbf{X}_i\boldsymbol{\beta} + r_i\mathbf{1}_T + \boldsymbol{\epsilon}_i \tag{1}$$

where $\boldsymbol{\beta}$ represents the coefficients for the covariate effects and $\boldsymbol{\epsilon}_i = (\epsilon_{i,1}, \ldots, \epsilon_{i,T})$ are the residuals. We assume independence and normality in the residuals and the random effects; hence, $r_i \sim N(0, \sigma_r^2)$ and $\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I}_T)$ for $i = 1, \ldots, n$.

Fitting the mixed model demonstrates substantial heterogeneity in the residuals, the variability increases as the fitted values increase. A traditional approach to fixing this violation would re-fit the model to the log-transformed MVPA values. Plots of residuals versus fitted values in this model approach also exhibited evidence of heterogeneity in the model; thus, still violating a core assumption of the regression framework. Given the changes adolescents experience as they grow into young adults, we expect to see heterogeneity in the physical activity patterns across this duration of follow-up time. However, the inability of the model to capture such changes over time at these higher levels of physical activity suggests the need for model improvements. The purpose of this analysis is to present our adjustments to previous analyses in order to investigate underlying characteristics across different groups of females formed based on deviations from adjusted physical activity levels.

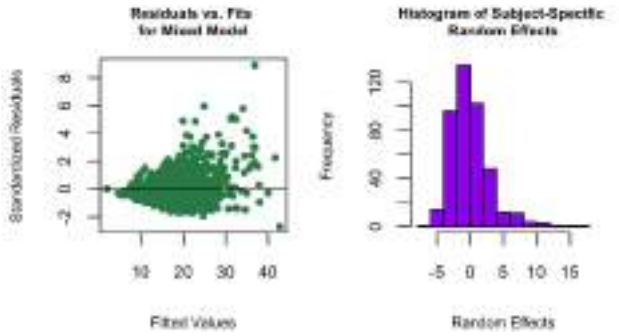


**Fig. 1** The plot on the left depicts the residuals versus fitted values for the linear mixed model in Eq. (1); they demonstrate severe heteroscedasticity. The variance increases as the fitted values increase. The plot on the right depicts the distribution of the random effects.

We fit the mixed model in Eq. (1) to the sample of female adolescents. The heteroscedasticity depicted in Figure 1 reveals an increase in variance with predicted minutes of moderate-to-vigorous physical activity, which we would expect. The plot on the right in Figure 1 demonstrates that the distribution of the random effects do not appear to follow our assumption of normally distributed and centered around zero. The random effects do appear to follow a normal distribution over the lower range of deviations with a subset of the subjects having larger positive deviations from the estimated adjusted physical activity levels.

To capture the heterogeneity and allow the random effects to follow a non-normal distribution, we assign the random effects a Gaussian mixture distribution. Before introducing the model for heterogeneity, we note the likelihood distribution for the observed outcomes, $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)'$. The moderate-to-vigorous physical activity distribution is

$$p(\mathbf{Y}|\boldsymbol{\beta}, \mathbf{r}, \sigma_\epsilon^2) = \prod_{i=1}^{n}\prod_{t=1}^{T}\left(2\pi\sigma_\epsilon^2\right)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma_\epsilon^2}(y_{i,t} - X_{i,t}\boldsymbol{\beta} - r_i)^2\right\}. \quad (2)$$

Then to account for the heterogeneity across subjects, the probability density for the subject-specific deviations in physical activity is expressed as a mixture of one-dimensional normal densities,

$$p(r_i|\boldsymbol{\mu}, \boldsymbol{\sigma_r^2}) = \sum_{g=1}^{G}\pi_g\left(2\pi\sigma_{r,g}^2\right)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma_{r,g}^2}(r_i - \mu_g)^2\right\}. \quad (3)$$

Here, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_G)'$ defines the group-specific mean deviations, $\boldsymbol{\sigma}_r^2 = (\sigma_{r,1}^2, \ldots, \sigma_{r,G}^2)'$ characterizes the variances of the group-specific deviations, and $\pi = (\pi_1, \ldots, \pi_G)'$ is the probability of membership in each group $g$.

The model in Eqs. (2) and (3) can be fit using either an EM or Bayesian MCMC procedures. Both require specification of a fixed number of $G$-groups. While we may hypothesize that there are only two groups–one that is normally distributed and centered at zero and another that is normally distributed and centered at a larger mean–the assumption hinges on what we have seen from plots like those in Figure 1. The random effects in the aforementioned histogram, however, are being shrunk towards zero by assumption; while a mixture model will allow the data to more accurately depict the deviations observed in the girl's physical activity levels. The assumption of $G$ groups can strongly influence the results of our model fitting. To circumvent the issues associated with selecting $G$ in either an EM algorithm or a Bayesian finite mixture model framework, we implement a Bayesian mixture model that incorporates $G$ as an additional unknown parameter.

## 2.1 Bayesian Mixed Models With Clustering

Richardson and Green [4] adapts the reversible jump methodology to univariate normal mixture models. In addition to being able to characterize the distribution of $G$, this Bayesian framework has the ability to simultaneously explore the posterior distribution for the covariate effects of interest. Furthermore, we will have the posterior distributions of the group-defining parameters rather than just point estimates. Since we are interested in the physical activity differences in subjects when controlling for these covariates, we use Eq. (1) as the basis of our model.

The foundation of our clustering model is a finite mixture model on the random effects, $r_i$, as shown in Eq. (3). For all $i = 1, \ldots, n$ and $g = 1, \ldots, G$, $r_i|c_i, \boldsymbol{\mu} \sim F_r(\mu_{c_i}, \sigma_{r,c_i}^2)$, $(c_i = g)|\boldsymbol{\pi}, G \sim \text{Categorical}(\pi_1, \ldots, \pi_G)$, $\mu_g|\tau \sim N(\mu_0, \tau)$, $\sigma_{r,g}^2|c, \delta \sim IG(c, \delta)$, $\boldsymbol{\pi}|G \sim \text{Dirichlet}(\alpha, \ldots, \alpha)$, $G \sim \text{Uniform}[1, G_{max}]$, where $c_i$ is the latent grouping variable tracking the assignment of $r_i$ into any one of the $G$ clusters. The *likelihood function* for these subject-specific deviations, given the group assignment, $c_i$, is simply $p(r_i|c_i = g, \mu_g, \sigma_{r,g}^2) = \left(2\pi\sigma_{r,g}^2\right)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2\sigma_{r,g}^2}(r_i - \mu_g)^2\right\}$.

This replaces the typical independent and identically distributed assumption of $r_i \sim N(0, \sigma_r^2)$ for all $i$ with a normal distribution that is now conditional on group assignment. The remainder of the model formulation follows closely to the framework constructed in [4], except we have an additional layer of unknown parameters defining the linear mixed model in Eq. (1).

We select conjugate priors so that the the posterior distributions of the unknown parameters are analytically tractable. The prior on the mixing probabilities, $\boldsymbol{\pi}$, is a symmetric Dirichlet distribution, reflecting the prior belief that belonging to any one cluster is equally likely. To use the sampling methods of [4], we select a discrete uniform prior on $G$ that reflects our uncertainty on the number of groups, and impose an a priori ordering of the $\mu_g$, such that for any given value $G$, $\mu_1 < \mu_2 < \cdots < \mu_G$, to remove label switching. Thus, in the prior for the clustering parameters,

$$p(\boldsymbol{\mu}) = G! \prod_{g=1}^{G} \sqrt{(2\pi\tau)^{-\frac{1}{2}}} \exp\left\{-\frac{1}{2\tau}(\mu_g - \mu_0)^2\right\}$$

$$p(\sigma_{r,g}^2) = \frac{\delta^c}{\Gamma(c)}(\sigma_{r,g}^2)^{-c-1} \exp\left\{-\frac{\delta}{\sigma_{r,g}^2}\right\}$$

$$p(G) = \frac{1}{G_{max}} \mathbf{1}\{G \in [1, G_{max}]\},$$

where $G_{max}$ is set to be reasonably large and $\mathbf{1}\{G \in [1, G_{max}]\}$ is a discrete indicator function, equal to 1 on the interval $[1, G_{max}]$ and 0 elsewhere.

The capacity of our sampler to move between dimensions is essential to our ability to explore the grouping of the observations while simultaneously exploring the parameters describing the relationships between the covariates and the outcome. This means that we can allow the number of components of our mixture model on the random effects to increase or decrease at each state of our MCMC chain. Such changes impact the dimension of the parameters of the mixture model, $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\sigma}_r^2, G, \boldsymbol{\pi}, \mathbf{c})$.

Let $\boldsymbol{\theta}$ denote the current state of the parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma}_r^2, G, \boldsymbol{\pi}, \mathbf{c})$ when proposing move $m$ where $m \in \{S, M, B, D\}$ corresponds to a split, merge, birth and death, respectively. Given the current state, $\boldsymbol{\theta}$, and move $m$, we propose a new state, $\boldsymbol{\theta}^m$, under move $m$. The acceptance probability is written as $acc_m(\boldsymbol{\theta}^m, \boldsymbol{\theta}) = \min\left[1, \frac{p(\boldsymbol{\theta}^m|\mathbf{r})q(\boldsymbol{\theta}^m|m^{-1})}{p(\boldsymbol{\theta}|\mathbf{r})q(\boldsymbol{\theta}|m)}|J|\right]$ where $p(\cdot)$ and $q(\cdot)$ denote the target and proposal distribution, respectively. In our case, the target distribution is the posterior distribution of our group-specific parameters, $(\boldsymbol{\mu}, \boldsymbol{\sigma}_r^2, \boldsymbol{\pi}, \mathbf{c})$, given the data, $\mathbf{r}$, which are the random effects. Each proposed move changes the dimension of the parameters in $\boldsymbol{\theta}$ by 1, adding or deleting group-specific parameters. The ratio $q(\boldsymbol{\theta}^m|m^{-1})/q(\boldsymbol{\theta}|m)$ ensures "dimension balancing", as explained in [4]. For moves increasing in dimension, the Jacobian, $|J|$, is computed as $|\delta\boldsymbol{\theta}^m/\delta(\boldsymbol{\theta}, \mathbf{u})|$ because moving from $\boldsymbol{\theta}$ to $\boldsymbol{\theta}^m$ will require additional parameters, $\mathbf{u}$ to appropriately match dimensions. The opposite is true for moves decreasing in dimension. This is what we refer to as the reversible jump mechanism; each time a split is proposed, we must also design the reversible move that would result in the currently merged component, and vice versa.

Split and merge moves are implemented for our model. These moves update $\pi$, $\mu$, and $\sigma$ for two adjacent groups or create two adjacent groups using three Beta-distributed additional parameters, $u$, for dimension balancing in a similar way to [4]. Within our context of random effects, births and deaths are not appropriate. A singleton causes issues of identifiability because the $r_i$ is no longer defined as random. We do not allow for birth and death moves in our reversible jump methods.

## 3 Trial of Activity in Adolescent Girls (TAAG) and Model Results

Our analysis focuses only on these girls from the University of Maryland site of the TAAG study who were measured at all three follow-up time points, beginning in 2006. After excluding girls with missing outcomes, the final sample consisted of 428 girls measured in 2006, 2009, and 2014. Missing covariate values were imputed for four subjects using the values from the nearest time point.

We determine the group assignments using an MCMC sampler having 10,000 iterations, with a burn-in of 500 draws. The posterior distribution for $G$ was extremely peaked at $G = 2$. Summarization of the posterior distribution of the group assignments via the least squares clustering method delivers the final arrangement, $\hat{\mathbf{c}}_{LS}$, of girls into two groups describing their physical activity levels [5]. Since our sampler explores several models for which group assignments and $G$ can vary, we sample additional draws from the posterior distribution of the remaining parameters of interest using an MCMC sampler with the model specification of Eq. (1) with groups fixed at our posterior assignment, $\hat{\mathbf{c}}_{LS}$, for the subject-specific random effects. This additional chain was run for 10,000 iterations with a burn-in 500 draws, yielding the results summarized below. Convergence diagnostics indicated that 10,000 iterations sufficiently met the effective sample size threshold for estimating the coefficients for the covariate effects, $\boldsymbol{\beta}$, and the group-specific means, $\boldsymbol{\mu}$, describing the deviations of the girls' physical activity levels [6].

After controlling for covariates believed to best describe the variation in the physical activity levels of females, our method finds that there is a small subset of the females who are much more active than the remainder of the sample. Every subject in the more active group has fitted trajectories above the recommended 30 minutes of exercise. Most of the population does not get the recommended allowance of daily physical activity and this is well-supported in our analysis. All but two subjects in the less active group have fitted trajectories that never pass the recommended 30 minutes of exercise. The random effects from this model better fit a normal distribution (not centered at 0) for each of the two groups and do not show as much heteroscedasticity over time as the one group model depicted in Figure 1.

Given these differences are observed even after controlling for the aforementioned variables, we would like to further examine the characteristics that may set these highly active females apart from the rest of the girls in our sample. To do this, we look at a number of other covariates that were either excluded during the variable selection process or were not measured at all time points. We use simple Wilcoxon

tests on the available time points of the additional variables and on all time points for covariates we adjusted for in the initial model.

We first note that the median BMI of the subset of highly active girls is significantly lower than that of the remaining girls consistently at each TAAG wave. Similarly, mother's education level is also consistently significant at each time point. These values are measured at each time point to reflect changes as the mother pursues additional education, or as the girls become more aware of their mother's education. The majority of the highly active girls have mother's who have completed college or higher (75% or higher at each time point); whereas, the remainder of the sample has mother's with a range of education levels (less than high school through college or more). The number of parks within a one-mile radius of the home is significantly different among the high and low groups in the middle school and high school years, when the girls are likely to be living at home. This variable may be an indicator of socioeconomic status as families with more money may live in neighborhoods nearer to parks. Finally, in the high school and college-aged years, the self-management strategies among the highly active girls are significantly higher rated than the remainder of the population.

In high school, the subset of highly active girls tend to have better self-described health, participate in more sports teams, have access to more physical education classes, and have been older at the time of their first menstrual period. At the college age, these girls still have higher self-described health; however, the higher levels of the global physical activity score and self-esteem scores are now significantly improved in the subset of highly active females.

## 4 Discussion

We extended the mixed models of [1] with the application still focused on the same 428 girls from the TAAG, TAAG 2, and TAAG 3 studies. Within the Bayesian linear mixed model, we implemented a clustering procedure aimed at clustering girls into groups based on deviations from the adjusted physical activity levels. These groups reflected the tendency for small subsets of females to be highly active. Not surprisingly, only 24 girls (5% of our sample) were classified as highly active.

This group of highly active girls differs in several ways. These girls are more active, and thus we expect that the age at first menstrual period will be higher. We may also expect that the highly active girls are involved in more sports teams and that they will have higher global physical activity scores. Some other interesting characteristics of these girls, however, is their increased self-management strategies, self-esteem scores, and self-described health. This may suggest that interventions focusing on time management and emphasizing self-efficacy could impact adolescent female physical activity levels. In doing so, we could aim to increase self-esteem and self-described health.

The ability to account for heterogeneity in the subject-specific deviations from an adjusted model allows us to keep the outcome on the original scale while still

improving model assumptions. Our model estimates model parameters while identifying groups of observations with differing activity levels. In contrast, a frequentist approach could be taken using EM algorithm; however, we would lose the ability for the data to give statistical inference on the appropriate number of groups and to incorporate posterior samples with different numbers of groups into the estimated class label.

The current analysis looks only at identifying groups based on deviations from the overall adjusted minutes of MVPA for the females. A natural extension would be to look at clustering on the slope for time to begin to understand the various patterns we observe among adolescent females over time. Furthermore, we may want to incorporate a variable selection procedure into the fixed portion of the model. The groups we find by either clustering on subject-specific intercepts and/or slopes would be sensitive to the covariates selected, depending on the variability captured by this fixed portion of the model. Physical activity, like most human behavior, varies widely for a multitude of reasons, many of which we may not think to or are unable to measure. Identifying groups when a traditional mixed model constructed using standard variable selection methods suggests lack of fit can be a useful step towards better understanding differences through post-hoc analyses of the groups' characteristics.

**Acknowledgements** Research reported in this publication was supported by the National Institutes of Health (NIH) under award numbers T32ES007271 and R01HL119058. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

## References

1. Young, D. R., Mohan, Y. D., Saksvig, B. I., Sidell, M., Wu, T. T., Cohen, D.: Longitudinal predictors of moderate to vigorous physical activity among adolescent girls and young women. Under review. (2017)
2. Ng, S. K., McLachlan, G. J., Wang, K., Ben-Tovim Jones, L., Ng, S. W.: A mixture model with random-effects components for clustering correlated gene-expression profiles. Bioinformatics **22**(14), 1745 (2006)
3. Zhou, C., Wakefield, J.: A Bayesian mixture model for partitioning gene expression data. Biometrics **62**(2), 515–525 (2006)
4. Richardson, S., Green, P. J.: On Bayesian analysis of mixtures with an unknown number of components (with discussion). J. Roy. Stat. Soc. B **59**(4), 731–792 (1997)
5. Dahl, D. B.: Model-based clustering for expression data via a Dirichlet process mixture model. Bayesian Inference for Gene Expression and Proteomics. 201–218 (2006)
6. Flegal, J. M., Hughes, J., Vats, D.: mcmcse: Monte Carlo standard errors for MCMC. R package version 1.2-1 (2016)


# Unsupervised Classification of Categorical Time Series Through Innovative Distances

Ángel López-Oriona, José A. Vilar, and Pierpaolo D'Urso

**Abstract** In this paper, two novel distances for nominal time series are introduced. Both of them are based on features describing the serial dependence patterns between each pair of categories. The first dissimilarity employs the so-called association measures, whereas the second computes correlation quantities between indicator processes whose uniqueness is guaranteed from standard stationary conditions. The metrics are used to construct crisp algorithms for clustering categorical series. The approaches are able to group series generated from similar underlying stochastic processes, achieve accurate results with series coming from a broad range of models and are computationally efficient. An extensive simulation study shows that the devised clustering algorithms outperform several alternative procedures proposed in the literature. Specifically, they achieve better results than approaches based on maximum likelihood estimation, which take advantage of knowing the real underlying procedures. Both innovative dissimilarities could be useful for practitioners in the field of time series clustering.

**Keywords:** categorical time series, clustering, association measures, indicator processes

## 1 Introduction

Clustering of time series concerns the challenge of splitting a set of unlabeled time series into homogeneous groups, which is a pivotal problem in many knowledge discovery tasks [1]. Categorical time series (CTS) are a particular class of time series exhibiting a qualitative range which consists of a finite number of categories. Most of the classical statistical tools used for real-valued time series (e.g., the autocorrelation function) are not useful in the categorical case, so different types of measures than the standard ones are needed for a proper analysis of CTS. CTS

Ángel López-Oriona (✉), José A. Vilar
Research Group MODES, Research Center for Information and Communication Technologies (CITIC), University of A Coruña, Spain,
e-mail: oriona38@hotmail.com; jose.vilarf@udc.es

Pierpaolo D'Urso
Department of Social Sciences and Economics, Sapienza University of Rome, Italy,
e-mail: pierpaolo.durso@uniroma1.it



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_26

arise in an extensive assortment of fields [2, 3, 7, 8, 9]. Since only a few works have addressed the problem of CTS clustering [4, 5], the main goal of this paper is to introduce novel clustering algorithms for CTS.

## 2 Two Novel Feature-based Approaches for Categorical Time Series Clustering

Consider a set of $s$ categorical time series $\mathcal{S} = \{X_t^{(1)}, \ldots, X_t^{(s)}\}$, where the $j$-th element $X_t^{(j)}$ is a $T_j$-length partial realization from any categorical stochastic process $(X_t)_{t\in\mathbb{Z}}$ taking values on a number $r$ of unordered qualitative categories, which are coded from 1 to $r$ so that the range of the process can be seen as $\mathcal{V} = \{1, \ldots, r\}$. We suppose that the process $(X_t)_{t\in\mathbb{Z}}$ is bivariate stationary, i.e., the pairwise joint distribution of $(X_{t-k}, X_t)$ is invariant in $t$. Our goal is to perform clustering on the elements of $\mathcal{S}$ in such a way that the series assumed to be generated from identical stochastic processes are placed together. To that aim, we propose two distance metrics which are based on feature extraction.

### 2.1 Descriptive Features for Categorical Processes

Let $\{X_t, t \in \mathbb{Z}\}$ be a bivariate stationary categorical stochastic process with range $\mathcal{V} = \{1, \ldots, r\}$. Denote by $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_r)$ the marginal distribution of $X_t$, which is, $P(X_t = j) = \pi_j > 0$, $j = 1, \ldots, r$. Fixed $l \in \mathbb{N}$, we use the notation $p_{ij}(l) = P(X_t = i, X_{t-l} = j)$, with $i, j \in \mathcal{V}$, for the lagged bivariate probability and the notation $p_{i|j}(l) = P(X_t = i|X_{t-l} = j) = p_{ij}(l)/\pi_j$ for the conditional bivariate probability.

To extract suitable features characterizing the serial dependence of a given CTS, we start by defining the concepts of perfect serial independence and dependence for a categorical process. We have perfect serial independence at lag $l \in \mathbb{N}$ if and only if $p_{ij}(l) = \pi_i\pi_j$ for any $i, j \in \mathcal{V}$. On the other hand, we have perfect serial dependence at lag $l \in \mathbb{N}$ if and only if the conditional distribution $p_{\cdot|j}(l)$ is a one-point distribution for any $j \in \mathcal{V}$. There are several association measures which describe the serial dependence structure of a categorical process at lag $l$. One of such measures is the so-called Cramer's $v$, which is defined as

$$v(l) = \sqrt{\frac{1}{r-1}\sum_{i,j=1}^{r}\frac{(p_{ij}(l) - \pi_i\pi_j)^2}{\pi_i\pi_j}}. \tag{1}$$

Cramer's $v$ summarizes the serial dependence patterns of a categorical process for every pair $(i, j)$ and $l \in \mathbb{N}$. However, this quantity is not appropriate for characterizing a given stochastic process, since two different processes can have the same value of $v(l)$. A better way to characterize the process $X_t$ is by considering the matrix $\boldsymbol{V}(l) = \left(V_{ij}(l)\right)_{1\leq i,j\leq r}$, where $V_{ij}(l) = \frac{(p_{ij}(l)-\pi_i\pi_j)^2}{\pi_i\pi_j}$. The elements of the matrix

$\boldsymbol{V}(l)$ give information about the so-called *unsigned* dependence of the process. However, it is often useful to know whether a process tends to stay in the state it has reached or, on the contrary, the repetition of the same state after $l$ steps is infrequent. This motivates the concept of *signed* dependence, which arises as an analogy of the autocorrelation function of a numerical process, since such quantity can take either positive or negative values. Provided that perfect serial dependence holds, we have perfect *positive* (*negative*) serial dependence if $p_{i|i}(l) = 1$ ($p_{i|i}(l) = 0$) for all $i \in \mathcal{V}$.

Since $\boldsymbol{V}(l)$ does not shed light on the signed dependence structure, it would be valuable to complement the information contained in $\boldsymbol{V}(l)$ by adding features describing signed dependence. In this regard, a common measure of signed serial dependence at lag $l$ is the Cohen's $\kappa$, which takes the form

$$\kappa(l) = \frac{\sum_{j=1}^{r}(p_{jj}(l) - \pi_j^2)}{1 - \sum_{j=1}^{r}\pi_j^2}. \tag{2}$$

Proceeding as with $v(l)$, the quantity $\kappa(l)$ can be decomposed in order to obtain a complete representation of the signed dependence pattern of the process. In this way, we consider the vector $\boldsymbol{\mathcal{K}}(l) = (\mathcal{K}_1(l), \ldots, \mathcal{K}_r(l))$, where each $\mathcal{K}_i$ is defined as

$$\mathcal{K}_i(l) = \frac{p_{ii}(l) - \pi_i^2}{1 - \sum_{j=1}^{r}\pi_j^2}, \tag{3}$$

$i = 1, \ldots, r$.

In practice, the matrix $\boldsymbol{V}(l)$ and the vector $\boldsymbol{\mathcal{K}}(l)$ must be estimated from a $T$-length realization of the process, $\{X_1, \ldots X_T\}$. To this aim, we consider estimators of $\pi_i$ and $p_{ij}(l)$, $\widehat{\pi}_i$ and $\widehat{p}_{ij}(l)$, respectively, defined as $\widehat{\pi}_i = \frac{N_i}{T}$ and $\widehat{p}_{ij}(l) = \frac{N_{ij}(l)}{T-l}$, where $N_i$ is the number of variables $X_t$ equal to $i$ in the realization $\{X_1, \ldots X_T\}$, and $N_{ij}(l)$ is the number of pairs $(X_t, X_{t-l}) = (i, j)$ in the realization $\{X_1, \ldots X_T\}$. Hence, estimates of $\boldsymbol{V}(l)$ and $\boldsymbol{\mathcal{K}}(l)$, $\widehat{\boldsymbol{V}}(l)$ and $\widehat{\boldsymbol{\mathcal{K}}}(l)$, respectively, can be obtained by plugging in the estimates $\widehat{\pi}_i$ and $\widehat{p}_{ij}(l)$ in (2) and (3), respectively. This leads directly to estimates of $v(l)$ and $\kappa(l)$, denoted by $\widehat{v}(l)$ and $\widehat{\kappa}(l)$.

An alternative way of describing the dependence structure of the process $\{X_t, t \in \mathbb{Z}\}$ is to take into consideration its equivalent representation as a multivariate binary process. The so-called *binarization* of $\{X_t, t \in \mathbb{Z}\}$ is constructed as follows. Let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_r \in \{0, 1\}^r$ be unit vectors such that $\boldsymbol{e}_k$ has all its entries equal to zero except for a one in the $k$-th position, $k = 1, \ldots, r$. Then, the binary representation of $\{X_t, t \in \mathbb{Z}\}$ is given by the process $\{\boldsymbol{Y}_t = (Y_{t,1}, \ldots, Y_{t,r})^\top, t \in \mathbb{Z}\}$ such that $\boldsymbol{Y}_t = \boldsymbol{e}_j$ if $X_t = j$. Fixed $l \in \mathbb{N}$ and $i, j \in \mathcal{V}$, consider the correlation $\phi_{ij}(l) = Corr(Y_{t,i}, Y_{t-l,j})$, which measures linear dependence between the $i$-th and $j$-th categories with respect to the lag $l$. The following proposition provides some properties of the quantity $\phi_{ij}(l)$.

**Proposition 1**

Let $\{X_t, t \in \mathbb{Z}\}$ be a bivariate stationary categorical process with range $\mathcal{V} = \{1, \ldots, r\}$. Then the following properties hold:

1. For every $i, j \in \mathcal{V}$, the function $\phi_{ij} : \mathbb{N} \to [-1, 1]$ given by $l \to \phi_{ij}(l) = Corr(Y_{t,i}, Y_{t-l,j})$ is well-defined.
2. $\phi_{ij}(l) = 0 \Leftrightarrow p_{ij}(l) = \pi_i \pi_j$.
3. $\phi_{ij}(l) = \pm 1 \Leftrightarrow p_{ij}(l) = \pm\sqrt{\pi_i(1-\pi_i)\pi_j(1-\pi_j)} + \pi_i\pi_j$.
4. $\phi_{ij}(l) = \sqrt{\frac{\pi_j(1-\pi_i)}{\pi_i(1-\pi_j)}} \Leftrightarrow p_{i|j}(l) = 1$.

The proof of Proposition 1 is quite straightforward and it is not shown in the manuscript for the sake of brevity. According to Proposition 1, the quantity $\phi_{ij}(l)$ can be used to explain both types of dependence, signed and unsigned, within the underlying process. In fact, in the case of perfect unsigned independence at lag $l$, we have that $p_{ij}(l) = \pi_i\pi_j$ for all $i, j \in \mathcal{V}$ so that $\phi_{ij}(l) = 0$ for all $i, j \in \mathcal{V}$ in accordance with Property 2 of Proposition 1. Under perfect positive dependence at lag $l$, $p_{i|i}(l) = 1$ for all $i \in \mathcal{V}$. Then $\phi_{ii}(l) = 1$ for all $i \in \mathcal{V}$ by following Property 4 of Proposition 1. The same property allows to conclude that $\phi_{ii}(l) = -\pi_i/(1-\pi_i)$ for all $i \in \mathcal{V}$ in the case of perfect negative dependence. In sum, $\phi_{ij}(l)$ evaluates unsigned dependence when $i \neq j$ and signed dependence when $i = j$. The previous quantities can be encapsulated in a matrix $\mathbf{\Phi}(l) = (\phi_{ij}(l))_{1\leq i,j\leq r}$, which can be directly estimated by means of $\widehat{\mathbf{\Phi}}(l) = (\widehat{\phi}_{ij}(l))_{1\leq i,j\leq r}$, where each $\widehat{\phi}_{ij}(l)$ is computed as $\widehat{\phi}_{ij}(l) = \frac{\widehat{p}_{ij}(l) - \widehat{\pi}_i\widehat{\pi}_j}{\sqrt{\widehat{\pi}_i(1-\widehat{\pi}_i)\widehat{\pi}_j(1-\widehat{\pi}_j)}}$ (this is derived from the proof of Proposition 1).

## 2.2 Two Innovative Dissimilarities Between CTS

In this section we introduce two distance measures between categorical series based on the features described above. Suppose we have a pair of CTS $X_t^{(1)}$ and $X_t^{(2)}$, and consider a set of $L$ lags, $\mathcal{L} = \{l_1, \ldots, l_L\}$. A dissimilarity based on Cramer's $v$ and Cohen's $\kappa$, so-called $d_{CC}$, is defined as

$$d_{CC}(X_t^{(1)}, X_t^{(2)}) = \sum_{k=1}^{L} \Big[ \Big\| vec\big(\widehat{\boldsymbol{V}}(l_k)^{(1)} - \widehat{\boldsymbol{V}}(l_k)^{(2)}\big) \Big\|^2 + \Big\| \widehat{\boldsymbol{\mathcal{K}}}(l_k)^{(1)} - \widehat{\boldsymbol{\mathcal{K}}}(l_k)^{(2)} \Big\|^2 \Big] + \Big\| \widehat{\boldsymbol{\pi}}^{(1)} - \widehat{\boldsymbol{\pi}}^{(2)} \Big\|^2,$$

where the superscripts (1) and (2) are used to indicate that the corresponding estimations are obtained with respect to the realizations $X_t^{(1)}$ and $X_t^{(2)}$, respectively.

An alternative distance measure relying on the binarization of the processes, so-called $d_B$, is defined as

$$d_B(X_t^{(1)}, X_t^{(2)}) = \sum_{k=1}^{L} \left\| vec\left(\widehat{\mathbf{\Phi}}(l_k)^{(1)} - \widehat{\mathbf{\Phi}}(l_k)^{(2)}\right)\right\|^2 + \left\|\widehat{\boldsymbol{\pi}}^{(1)} - \widehat{\boldsymbol{\pi}}^{(2)}\right\|^2 .$$

For a given set of categorical series, the distances $d_{CC}$ and $d_B$ can be used as input for traditional clustering algorithms. In this manuscript we consider the *Partition Around Medoids* (PAM) algorithm.

## 3 Partitioning Around Medoids Clustering of CTS

In this section we examine the performance of both metrics $d_{CC}$ and $d_B$ in the context of hard clustering (i.e., each series is assigned to exactly one cluster) of CTS through a simulation study.

### 3.1 Experimental Design

The simulated scenarios encompass a broad variety of generating processes. In particular, three setups were considered, namely clustering of (i) Markov Chains (MC), (ii) Hidden Markov Models (HMM) and (iii) New Discrete ARMA (NDARMA) processes. The generating models with respect to each class of processes are given below.

**Scenario 1**. Clustering of MC. Consider four three-state MC, so-called $MC_1$, $MC_2$, $MC_3$ and $MC_4$, with respective transition matrices $\boldsymbol{P}_1^1$, $\boldsymbol{P}_2^1$, $\boldsymbol{P}_3^1$ and $\boldsymbol{P}_4^1$ given by

$$\begin{aligned}
\boldsymbol{P}_1^1 &= Mat^3(0.1, 0.8, 0.1, 0.5, 0.4, 0.1, 0.6, 0.2, 0.2),\\
\boldsymbol{P}_2^1 &= Mat^3(0.1, 0.8, 0.1, 0.6, 0.3, 0.1, 0.6, 0.2, 0.2),\\
\boldsymbol{P}_3^1 &= Mat^3(0.05, 0.90, 0.05, 0.05, 0.05, 0.90, 0.90, 0.05, 0.05),\\
\boldsymbol{P}_4^1 &= Mat^3(1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3),
\end{aligned}$$

where the operator $Mat^k$, $k \in \mathbb{N}$ transforms a vector into a square matrix of order $k$ by sequentially placing the corresponding numbers by rows.

**Scenario 2**. Clustering of HMM. Consider the bivariate process $(X_t, Q_t)_{t\in\mathbb{Z}}$, where $Q_t$ stands for the hidden states and $X_t$ for the observable random variables. Process $(Q_t)_{t\in\mathbb{Z}}$ constitutes an homogeneous MC. Both $(X_t)_{t\in\mathbb{Z}}$ and $(Q_t)_{t\in\mathbb{Z}}$ are assumed to be count processes with range $\{1, \ldots, r\}$. Process $(X_t, Q_t)_{t\in\mathbb{Z}}$ is assumed to verify the three classical assumptions of a HMM. Based on previous considerations, let $HMM_1$, $HMM_2$, $HMM_3$ and $HMM_4$ be four three-state HMM with respective transition matrices $\boldsymbol{P}_1^2$, $\boldsymbol{P}_2^2$, $\boldsymbol{P}_3^2$ and $\boldsymbol{P}_4^2$ and emission matrices $\boldsymbol{E}_1^2$, $\boldsymbol{E}_2^2$, $\boldsymbol{E}_3^2$ and $\boldsymbol{E}_4^2$ given by

$$\boldsymbol{P}_1^2 = Mat^3(0.05, 0.90, 0.05, 0.05, 0.05, 0.90, 0.90, 0.05, 0.05), \boldsymbol{P}_2^2 = \boldsymbol{P}_1^2,$$
$$\boldsymbol{P}_3^2 = Mat^3(0.1, 0.7, 0.2, 0.4, 0.4, 0.2, 0.4, 0.3, 0.3),$$
$$\boldsymbol{P}_4^2 = Mat^3(1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3), \boldsymbol{E}_1^2 = \boldsymbol{P}_1^2,$$
$$\boldsymbol{E}_2^2 = Mat^3(0.1, 0.8, 0.1, 0.5, 0.4, 0.1, 0.6, 0.2, 0.2), \boldsymbol{E}_3^2 = \boldsymbol{E}_2^2,$$
$$\boldsymbol{E}_4^2 = Mat^3(1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3, 1/3).$$

**Scenario 3**. Clustering of NDARMA processes. Let $(X_t)_{t\in\mathbb{Z}}$ and $(\epsilon_t)_{t\in\mathbb{Z}}$, be two count processes with range $\{1, \ldots, r\}$ following the equation

$$X_t = \alpha_{t,1}X_{t-1} + \ldots + \alpha_{t,p}X_{t-p} + \beta_{t,0}\epsilon_t + \ldots + \beta_{t,q}\epsilon_{t-q},$$

where $(\epsilon_t)_{t\in\mathbb{Z}}$ is i.i.d with $P(\epsilon_t = i) = \pi_i$, independent of $(X_s)_{s<t}$, and the i.i.d multinomial random vectors

$$(\alpha_{t,1}, \ldots, \alpha_{t,p}, \beta_{t,0}, \ldots, \beta_{t,q}) \sim \text{MULT}(1; \phi_1, \ldots, \phi_p, \varphi_0, \ldots, \varphi_q),$$

are independent of $(\epsilon_t)_{t\in\mathbb{Z}}$ and $(X_s)_{s<t}$. The considered models are three three-state NDARMA(2,0) processes and one three-state NDARMA(1,0) process with marginal distribution $\boldsymbol{\pi}^3 = (2/3, 1/6, 1/6)$, and corresponding probabilities in the multinomial distribution given by

$$(\phi_1, \phi_2, \varphi_0)_1^3 = (0.7, 0.2, 0.1), (\phi_1, \phi_2, \varphi_0)_2^3 = (0.1, 0.45, 0.45),$$
$$(\phi_1, \phi_2, \varphi_0)_3^3 = (0.5, 0.25, 0.25), (\phi_1, \varphi_0)_4^3 = (0.2, 0.8).$$

The simulation study was carried out as follows. For each scenario, 5 CTS of length $T \in \{200, 600\}$ were generated from each process in order to execute the clustering algorithms twice, thus allowing to analyze the impact of the series length. The resulting clustering solution produced by each considered algorithm was stored. The simulation procedure was repeated 500 times for each scenario and value of $T$. The computation of $d_{CC}$ and $d_B$ was carried out by considering $\mathcal{L} = \{1\}$ in Scenarios 1 and 2, and $\mathcal{L} = \{1, 2\}$ in Scenario 3. This way, we adapted the distances to the maximum number of significant lags existing in each setting.

## 3.2 Alternative Metrics and Assessment Criteria

To better analyze the performance of both metrics $d_{CC}$ and $d_B$, we also obtained partitions by using alternative techniques for clustering of categorical series. The considered procedures are described below.

- *Model-based approach using maximum likelihood estimation (MLE).* The distance between two CTS is defined as the squared Euclidean distance between the corresponding vectors of fitted coefficients via MLE ($d_{MLE}$).
- *Model-based approach using mixtures.* [4] propose to group a set of CTS by using a mixture of first order Markov models via the EM algorithm ($d_{CZ}$).
- *An hybrid framework for clustering CTS.* [6] presents a dissimilarity between categorical series which evaluates both closeness between raw categorical values and proximity between dynamic patterns ($d_{MV}$).

Note that the approach based on the distance $d_{MLE}$ can be seen as a strict benchmark in the evaluation task. The effectiveness of the clustering approaches was assessed by comparing the clustering solution produced by the algorithms with the true clustering partition, so-called ground truth. The latter consisted of $C = 4$ clusters in all scenarios, each group including the five CTS generated from the same process. The value $C = 4$ was provided as input parameter to the PAM algorithm in the case of $d_{CC}$, $d_B$, $d_{MLE}$ and $d_{MV}$. As for the approach $d_{CZ}$, a number of 4 components were considered for the mixture model. Experimental and true partitions were compared by using three well-known external clustering quality indexes, the Adjusted Rand Index (ARI), the Jaccard Index (JI) and the Fowlkes-Mallows index (FMI).

## 3.3 Results and Discussion

Average values of the quality indexes by taking into account the 500 simulation trials are given in Tables 1, 2 and 3 for Scenarios 1, 2 and 3, respectively.

**Table 1** Average results for Scenario 1.

| | $T = 200$ | | | $T = 600$ | | |
|---|---|---|---|---|---|---|
| Method | ARI | JI | FMI | ARI | JI | FMI |
| $d_{CC}$ | **0.774** | **0.710** | **0.830** | **0.916** | **0.886** | **0.935** |
| $d_B$ | 0.729 | 0.661 | 0.792 | 0.861 | 0.878 | 0.893 |
| $d_{MLE}$ | 0.704 | 0.633 | 0.772 | 0.841 | 0.792 | 0.876 |
| $d_{CZ}$ | 0.712 | 0.648 | 0.786 | 0.915 | **0.886** | 0.934 |
| $d_{MV}$ | 0.406 | 0.363 | 0.665 | 0.379 | 0.363 | 0.650 |

The results in Table 1 indicate that the dissimilarity $d_{CC}$ is the best performing one when dealing with MC, outperforming the MLE-based metric $d_{MLE}$. The distance $d_B$ is also superior to $d_{MLE}$. The measure $d_{CZ}$ attains in Scenario 1 similar results than $d_{CC}$, specially for $T = 600$. The good performance of $d_{CZ}$ was expected, since the assumption of first order Markov models considered by this metric is fulfilled in Scenario 1. Table 2 shows a completely different picture, indicating that the metrics $d_{CC}$ and $d_B$ exhibit a significantly better effectiveness than the rest of the dissimilarities. Finally, the quantities in Table 3 reveal that the model-based distance $d_{MLE}$ attains the best results when $T = 200$, but is defeated by $d_B$ when

**Table 2** Average results for Scenario 2.

| | $T = 200$ | | | $T = 600$ | | |
|---|---|---|---|---|---|---|
| Method | ARI | JI | FMI | ARI | JI | FMI |
| $d_{CC}$ | 0.707 | 0.639 | 0.777 | 0.856 | 0.810 | 0.888 |
| $d_B$ | **0.760** | **0.701** | **0.812** | **0.963** | **0.949** | **0.971** |
| $d_{MLE}$ | 0.354 | 0.342 | 0.512 | 0.299 | 0.310 | 0.478 |
| $d_{CZ}$ | 0.645 | 0.577 | 0.739 | 0.703 | 0.638 | 0.779 |
| $d_{MV}$ | 0.089 | 0.175 | 0.323 | 0.062 | 0.175 | 0.301 |

**Table 3** Average results for Scenario 3.

| | $T = 200$ | | | $T = 600$ | | |
|---|---|---|---|---|---|---|
| Method | ARI | JI | FMI | ARI | JI | FMI |
| $d_{CC}$ | 0.627 | 0.563 | 0.715 | 0.875 | 0.837 | 0.903 |
| $d_B$ | 0.680 | 0.612 | 0.754 | **0.925** | **0.901** | **0.941** |
| $d_{MLE}$ | **0.727** | **0.656** | **0.788** | 0.872 | 0.828 | 0.900 |
| $d_{CZ}$ | 0.586 | 0.562 | 0.693 | 0.647 | 0.577 | 0.738 |
| $d_{MV}$ | 0.035 | 0.167 | 0.292 | -0.028 | 0.138 | 0.251 |

$T = 600$. The metric $d_{CZ}$ suffers again from model misspecification. In summary, the numerical experiments carried out throughout this section show the excellent ability of both measures $d_{CC}$ and $d_B$ to discriminate between a broad variety of categorical processes. Specifically, these metrics either outperform or show similar behavior than distances based on estimated model coefficients, which take advantage of knowing the true underlying models.

It is worth highlighting that the methods proposed in this paper could have promising applications in some fields as the clustering of genetic data sequences.

## References

1. Liao, T. W.: Clustering of time series data: A survey. Pattern Recogn. **38**, 1857-1874 (2005)
2. Churchill, G. A.: Stochastic models for heterogeneous DNA sequences. Bull. Math. Biol. **51**, 79-94 (1989)
3. Fokianos, K., Kedem, B.: Regression theory for categorical time series. Stat. Sci. **18**, 357-376 (2003)
4. Cadez, I., Heckerman, D., Meek, C., Smyth, P., White, S.: Model-based clustering and visualization of navigation patterns on a web site. Data Min. Knowl. Discov. **7**, 399-424 (2003)
5. Fruhwirth-Schnatter, S., Pamminger, C.: Model-based clustering of categorical time series. Bayesian Analysis. **5**, 345-368 (2010)
6. García Magariños, M., Vilar, J. A.: A framework for dissimilarity-based partitioning clustering of categorical time series. Data Min. Knowl. Discov. **29**, 466-502 (2015)
7. Elzinga, C. H.: Combinatorial representations of token sequences. J. Classif. **22**, 87-118 (2005)
8. Elzinga, C. H.: Sequence similarity: a nonaligning technique. Socio. Meth. Res. **32**, 3-22 (2003)
9. Elzinga, C. H.: Sequence analysis: Metric representations of categorical time series. Socio. Meth. Res. (2006)


# Fuzzy Clustering by Hyperbolic Smoothing

David Masís, Esteban Segura, Javier Trejos, and Adilson Xavier

**Abstract** We propose a novel method for building fuzzy clusters of large data sets, using a smoothing numerical approach. The usual sum-of-squares criterion is relaxed so the search for good fuzzy partitions is made on a continuous space, rather than a combinatorial space as in classical methods [8]. The smoothing allows a conversion from a strongly non-differentiable problem into differentiable subproblems of optimization without constraints of low dimension, by using a differentiable function of infinite class. For the implementation of the algorithm, we used the statistical software *R* and the results obtained were compared to the traditional fuzzy *C*–means method, proposed by Bezdek [1].

**Keywords:** clustering, fuzzy sets, numerical smoothing

## 1 Introduction

Methods for making groups from data sets are usually based on the idea of disjoint sets, such as the classical crisp clustering. The most well known are hierarchical and $k$-means [8], whose resulting clusters are sets with no intersection. However, this restriction may not be natural for some applications, where the condition for

David Masís
Costa Rica Institute of Technology, Cartago, Costa Rica, e-mail: dmasis@itcr.ac.cr

Esteban Segura
CIMPA & School of Mathematics, University of Costa Rica, San José, Costa Rica,
e-mail: esteban.seguraugalde@ucr.ac.cr

Javier Trejos (✉)
CIMPA & School of Mathematics, University of Costa Rica, San José, Costa Rica,
e-mail: javier.trejos@ucr.ac.cr

Adilson E. Xavier
Universidade Federal de Rio de Janeiro, Brazil, e-mail: adilson.xavier@gmail.com



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_27

some objects may be to belong to two or more clusters, rather than only one. Several methods for constructing overlapping clusters have been proposed in the literature [4, 5, 8]. Since Zadeh introduced the concept of fuzzy sets [17], the principle of belonging to several clusters has been used in the sense of a degree of membership to such clusters. In this direction, Bezdek [1] introduced a fuzzy clustering method that became very popular since it solved the problem of representation of clusters with centroids and the assignment of objects to clusters, by the minimization of a well-stated numerical criterion. Several methods for fuzzy clustering have been proposed in the literature; a survey of these methods can be found in [16].

In this paper we propose a new fuzzy clustering method based on the numerical principle of hyperbolic smoothing [15]. Fuzzy $C$-Means method is presented in Section 2 and our proposed Hyperbolic Smoothing Fuzzy Clustering method in Section 3. Comparative results between these two methods are presented in Section 4. Finally, Section 5 is devoted to the concluding remarks.

## 2 Fuzzy Clustering

The most well known method for fuzzy clustering is the original Bezdek's $C$-means method [1] and it is based on the same principles of $k$-means or dynamical clusters [2], that is, iterations on two main steps: i) class representations by the optimization of a numerical criterion, and ii) assignment to the closest class representative in order to construct clusters; these iterations are made until a convergence is reached to a local minimum of the overall quality criterion.

Let us introduce the notation that will be used and the numerical criterion for optimization. Let $\mathbf{X}$ be an $n \times p$ data matrix containing $p$ numerical observations over $n$ objects. We look for a $K \times p$ matrix $\mathbf{G}$ that represents centroids of $K$ clusters of the $n$ objects and an $n \times K$ membership matrix with elements $\mu_{ik} \in [0, 1]$, such that the following criterion is minimized:

$$\begin{aligned} W(\mathbf{X}, \mathbf{U}, C) &= \sum_{i=1}^{n} \sum_{k=1}^{K} (\mu_{ik})^m \, \|\mathbf{x}_i - \mathbf{g}_k\|^2 \\ \text{subject to } & \textstyle\sum_{k=1}^{K} \mu_{ik} = 1, \text{ for all } i \in \{1, 2, \ldots, n\} \\ & 0 < \textstyle\sum_{i=1}^{n} \mu_{ik} < n, \text{ for all } k \in \{1, 2, \ldots, K\}, \end{aligned} \tag{1}$$

where $\mathbf{x}_i$ is the $i$-th row of $\mathbf{X}$ and $\mathbf{g}_k$ is the $k$-th row of $\mathbf{G}$, representing in $\mathbb{R}^p$ the centroid of the $k$-th cluster.

The parameter $m \neq 1$ in (1) controls the fuzzyness of the clusters. According to the literature [16], it is usual to take $m = 2$, since greater values of $m$ tend to give very low values of $\mu_{ik}$, tending to the usual crisp partitions such as in $k$-means. We also assume that the number of clusters, $K$, is fixed.

Minimization of (1) represents a non linear optimization problem with constraints, which can be solved using Lagrange multipliers as presented in [1]. The solution, for each row of the centroids matrix, given a matrix $\mathbf{U}$, is:

$$\mathbf{g}_k = \sum_{i=1}^{n} (\mu_{ik})^m \mathbf{x}_i \Bigg/ \sum_{i=1}^{n} (\mu_{ik})^m \ . \tag{2}$$

The solution for the membership matrix, given a matrix centroids $\mathbf{G}$, is [1]:

$$\mu_{ik} = \left[ \sum_{j=1}^{K} \left( \frac{||\mathbf{x}_i - \mathbf{g}_k||^2}{||\mathbf{x}_i - \mathbf{g}_j||^2} \right)^{1/(m-1)} \right]^{-1} . \tag{3}$$

The following pseudo-code shows the mains steps of Bezdek's Fuzzy $C$-Means method [1].

**Bezdek's Fuzzy c-Means (FCM) Algorithm**

1. Initialize fuzzy membership matrix $\mathbf{U} = [\mu_{ik}]_{n \times K}$
2. Compute centroids for fuzzy clusters according to (2)
3. Update membership matrix $\mathbf{U}$ according to (3)
4. If improvement in the criterion is less than a threshold, then stop; otherwise go to Step 2.

Fuzzy $C$-Means method starts from an initial partition that is improved in each iteration, according to (1), applying Steps 2 and 3 of the algorithm. It is clear that this procedure may lead to local optima of (1) since iterative improvement in (2) and (3) is made by a local search strategy.

## 3 Algorithm for Hyperbolic Smoothing Fuzzy Clustering

For the clustering problem of the $n$ rows of data matrix $\mathbf{X}$ in $K$ clusters, we can seek for the minimum distance between every $\mathbf{x}_i$ and its class center $\mathbf{g}_k$:

$$z_i^2 = \min_{\mathbf{g}_k \in \mathbf{G}} \|\mathbf{x}_i - \mathbf{g}_k\|_2^2$$

where $\|\cdot\|_2$ is the Euclidean norm. The minimization can be stated as a sum-of-squares:

$$\min \sum_{i=1}^{n} \min_{\mathbf{g}_k \in \mathbf{G}} \|\mathbf{x}_i - \mathbf{g}_k\|_2^2 = \min \sum_{i=1}^{n} z_i^2$$

leading to the following constrained problem:

$$\min \sum_{i=1}^{n} z_i^2 \text{ subject to } z_i = \min_{\mathbf{g}_k \in \mathbf{G}} \|\mathbf{x}_i - \mathbf{g}_k\|_2, \text{ with } i = 1, \ldots, n.$$

This is equivalent to the following minimization problem:

$$\min \sum_{i=1}^{n} z_i^2 \text{ subject to } z_i - \|\mathbf{x}_i - \mathbf{g}_k\|_2 \leq 0, \text{ with } i = 1, \ldots, n \text{ and } k = 1, \ldots, K.$$

Considering the function: $\varphi(y) = \max(0, y)$, we obtain the problem:

$$\min \sum_{i=1}^{n} z_i^2 \text{ subject to } \sum_{k=1}^{K} \varphi(z_i - \|\mathbf{x}_i - \mathbf{g}_k\|_2) = 0 \text{ for } i = 1, \ldots, n.$$

That problem can be re-stated as the following one:

$$\min \sum_{i=1}^{n} z_i^2 \text{ subject to } \sum_{k=1}^{K} \varphi(z_i - \|\mathbf{x}_i - \mathbf{g}_k\|_2) > 0, \text{ for } i = 1, \ldots, n.$$

Given a perturbation $\epsilon > 0$ it leads to the problem:

$$\min \sum_{i=1}^{n} z_i^2 \text{ subject to } \sum_{k=1}^{K} \varphi(z_i - \|\mathbf{x}_i - \mathbf{g}_k\|_2) \geq \epsilon \text{ for } i = 1, \ldots, n.$$

It should be noted that function $\varphi$ is not differentiable. Therefore, we will make a smoothing procedure in order to formulate a differentiable function and proceed with a minimization by a numerical method. For that, consider the function: $\psi(y, \tau) = \frac{y+\sqrt{y^2+\tau^2}}{2}$, for all $y \in \mathbb{R}$, $\tau > 0$, and the function: $\theta(\mathbf{x}_i, \mathbf{g}_k, \gamma) = \sqrt{\sum_{j=1}^{p} (x_{ij} - g_{kj})^2 + \gamma^2}$, for $\gamma > 0$. Hence, the minimization problem is transformed into:

$$\min \sum_{i=1}^{n} z_i^2 \text{ subject to } \sum_{k=1}^{K} \psi(z_i - \theta(\mathbf{x}_i, \mathbf{g}_k, \gamma), \tau) \geq \epsilon, \text{ for } i = 1, \ldots, n.$$

Finally, according to the Karush–Kuhn–Tucker conditions [10, 11], all the constraints are active and the final formulation of the problem is:

$$\begin{aligned} &\min \sum_{i=1}^{n} z_i^2 \\ &\text{subject to } h_i(z_i, \mathbf{G}) = \sum_{k=1}^{K} \psi(z_i - \theta(\mathbf{x}_i, \mathbf{g}_k, \gamma), \tau) - \epsilon = 0, \text{ for } i = 1, \ldots, n, \\ &\epsilon, \tau, \gamma > 0. \end{aligned} \tag{4}$$

Considering (4), in [15] it was stated the Hyperbolic Smoothing Clustering Method presented in the following algorithm.

**Hyperbolic Smoothing Clustering Method (HSCM) Algorithm**

1. Initialize cluster membership matrix $\mathbf{U} = [\mu_{ik}]_{n\times K}$
2. Choose initial values: $\mathbf{G}^0, \gamma^1, \tau^1, \epsilon^1$
3. Choose values: $0 < \rho_1 < 1, 0 < \rho_2 < 1, 0 < \rho_3 < 1$
4. Let $l = 1$
5. Repeat steps 6 and 7 until a stop condition is reached:
6. Solve problem (P): $\min f(\mathbf{G}) = \sum_{i=1}^{n} z_i^2$ with $\gamma = \gamma^l$, $\tau = \tau^l$ and $\epsilon = \epsilon^l$, $\mathbf{G}^{l-1}$ being the initial value and $\mathbf{G}^l$ the obtained solution
7. Let $\gamma^{l+1} = \rho_1\gamma^l$, $\tau^{l+1} = \rho_2\tau^l$, $\epsilon^{l+1} = \rho_3\epsilon^l$ and $l = l + 1$.

The most relevant task in the hyperbolic smoothing clustering method is finding the zeroes of the function $h_i(z_i, \mathbf{G}) = \sum_{k=1}^{K} \psi(z_i - \theta(\mathbf{x}_i, \mathbf{g}_k, \gamma), \tau) - \epsilon = 0$ for for $i = 1, \ldots, n$. In this paper, we used the Newton-Raphson method for finding these zeroes [3], particularly the BFGS procedure [12]. Convergence of the Newton-Raphson method was successful, mainly, thank to a good choice of initial solutions. In our implementation, these initial approximations were generated by calculating the minimum distance between the $i$-th object and the $k$-th centroid for a given partition. Once the zeroes $z_i$ of the functions $h_i$ are obtained, it is implemented the hyperbolic smoothing. The final solution for this method consists on solving a finite number of optimization subproblems corresponding to problem (P) in Step 6 of the HSCM algorithm. Each one of these subproblems was solved with the R routine *optim* [13], a useful tool for solving optimization problems in non linear programming. As far as we know there is no closed solution for solving this step. For the future, we can consider writing a program by our means, but for this paper we are using this R routine.

Since we have that: $\sum_{k=1}^{K} \psi(z_i - \theta(\mathbf{x}_i, \mathbf{g}_k, \gamma), \tau) = \epsilon$, then each entry $\mu_{ik}$ of the membership matrix is given by: $\mu_{ik} = \frac{\psi(z_i - d_k, \tau)}{\epsilon}$. It is worth to note that fuzzyness is controlled by parameter $\epsilon$.

The following algorithm contains the main steps of the Hyperbolic Smoothing Fuzzy Clustering (HSFC) method.

**Hyperbolic Smoothing Fuzzy Clustering (HSFC) Algorithm**

1. Set $\epsilon > 0$
2. Choose initial values for: $\mathbf{G}^0$ (centroids matrix), $\gamma^1$, $\tau^1$ and $N$ (maximum number of iterations)
3. Choose values: $0 < \rho_1 < 1$, $0 < \rho_2 < 1$
4. Set $l = 1$
5. While $l \leq N$:
6. Solve the problem (P): Minimize $f(\mathbf{G}) = \sum_{i=1}^{n} z_i^2$ with $\gamma = \gamma^{(l)}$ and $\tau = \tau^{(l)}$, with an initial point $\mathbf{G}^{(l-1)}$ and $\mathbf{G}^{(l)}$ being the obtained solution
7. Set $\gamma^{(l+1)} = \rho_1\gamma^{(l)}$,$\tau^{(l+1)} = \rho_2\tau^{(l)}$, y $l = l + 1$
8. Set $\mu_{ik} = \psi(z_i - \theta(\mathbf{x}_i, \mathbf{g}_k, \gamma), \tau)/\epsilon$ for $i = 1, \ldots, n$ and $k = 1, \ldots, K$.

## 4 Comparative Results

Performance of the HSFC method was studied on a data table well known from the literature, the Fisher's iris [7] and 16 simulated data tables built from a semi-Monte Carlo procedure [14].

For comparing FCM and HSFC, we used the implementation of FCM in R package *fclust* [6]. This comparison was made upon the within class sum-of-squares: $W(P) = \sum_{k=1}^{K} \sum_{i=1}^{n} \mu_{ik} \|\mathbf{x}_i - \mathbf{g}_k\|^2$. Both methods were applied 50 times and the best value of $W$ is reported. For simplicity here, for HSFC we used the following parameters: $\rho_1 = \rho_2 = \rho_3 = 0.25$, $\epsilon = 0.01$ and $\gamma = \tau = 0.001$ as initial values. In Table 1 the results for Fisher's iris are shown, in which case HSFC performs slightly better. It contains the Adjusted Rand Index (ARI) [9] between HSFC and the best FCM result among 100 runs; ARI compares fuzzy membership matrices crisped into hard partitions.

**Table 1** Minimum sum-of-squares (SS) reported for the Fisher's iris data table with HSFC and FCM, $K$ being the number of clusters, ARI comparing both methods. In bold best method.

| Table | $K$ | SS for HSFC | SS for FCM | ARI |
|---|---|---|---|---|
| Fisher's iris | 2 | **152.348** | 152.3615 | 1 |
| | 3 | **78.85567** | 78.86733 | 0.994 |
| | 4 | 57.26934 | 57.26934 | 0.980 |

Simulated data tables were generated in a controlled experiment as in [14], with random numbers following a Gaussian distribution. Factors of the experiment were:

- The number of objects (with 2 levels, $n = 105$ and $n = 525$).
- The number of clusters (with levels $K = 3$ and $K = 7$).
- Cardinality (card) of clusters, with levels i) all with the same number of objects (coded as card(=)), and ii) one large cluster with 50% of objects and the rest with the same number (coded as card(≠)).
- Standard deviation of clusters, with levels i) all Gaussian random variables with standard deviation (SD) equal to one (coded as SD(=)), and ii) one cluster with SD=3 and the rest with SD=1 (coded as SD(≠)).

Table 2 contains codes for simulated data tables according to the codes we used.

Table 3 contains the minimum values of the sum-of-squares obtained for our HSFC and Bezdek's FCM methods; the best solution of 100 random applications for FCM in presented and one run of HSFC. It also contains the ARI values for comparing HSFC solution with that best solution of FCM. It can be seen that, generally, HSFC method tends to obtain better results than FCM, with only few exceptions. In 23 cases HSFC obtains better results, FCM is better in 5 cases, and results are in same in 17 cases. However, ARI shows that partitions tend to be very similar with both methods.

**Table 2** Codes and characteristics of simulated data tables; $n$: number of objects, $K$: number of clusters, card: cardinality, DS: standard deviation.

| Table | Characteristcs | Table | Characteristcs |
|---|---|---|---|
| T1 | $n = 525$, $K = 3$, card(=), SD(=) | T9 | $n = 525$, $K = 3$, card(≠), DS(=) |
| T2 | $n = 525$, $K = 7$, card(=), SD(=) | T10 | $n = 525$, $K = 7$, card(≠), DS(=) |
| T3 | $n = 105$, $K = 3$, card(=), SD(=) | T11 | $n = 105$, $K = 3$, card(≠), DS(=) |
| T4 | $n = 105$, $K = 7$, card(=), SD(=) | T12 | $n = 105$, $K = 7$, card(≠), DS(=) |
| T5 | $n = 525$, $K = 3$, card(=), SD(≠) | T13 | $n = 525$, $K = 3$, card(≠), DS(≠) |
| T6 | $n = 525$, $K = 7$, card(=), SD(≠) | T14 | $n = 525$, $K = 7$, card(≠), DS(≠) |
| T7 | $n = 105$, $K = 3$, card(=), SD(≠) | T15 | $n = 105$, $K = 3$, card(≠), DS(≠) |
| T8 | $n = 105$, $K = 7$, card(=), SD(≠) | T16 | $n = 105$, $K = 7$, card(≠), DS(≠) |

**Table 3** Minimum sum-of-squares (SS) reported for HSFC and FCM methods on the simulated data tables. Best method in bold.

| Table | $K$ | SS for HSFC | SS for FCM | ARI | Table | $K$ | SS for HSFC | SS for FCM | ARI |
|---|---|---|---|---|---|---|---|---|---|
| | 2 | **7073.402** | 7073.814 | 0.780 | | 2 | 12524.31 | 12524.31 | 0.900 |
| T1 | 3 | 3146.119 | 3146.119 | 1 | T9 | 3 | **9269.361** | 9269.611 | 1 |
| | 4 | 2983.651 | 2983.651 | 1 | | 4 | 6298.47 | **6298.368** | 1 |
| | 2 | **16987.19** | 16987.71 | 0.764 | | 2 | **5466.893** | 5466.912 | 0.890 |
| T2 | 3 | 11653.22 | 11653.22 | 1 | T10 | 3 | 2977.58 | 2977.58 | 1 |
| | 4 | **7776.855** | 7777.396 | 1 | | 4 | **2745.721** | 2746.671 | 1 |
| | 2 | **3923.051** | 3923.062 | 0.763 | | 2 | **2969.247** | 2969.32 | 0.860 |
| T3 | 3 | 2917.13 | 2917.13 | 0.754 | T11 | 3 | 1912.323 | 1912.323 | 1 |
| | 4 | 2287.523 | **2256.298** | 0.993 | | 4 | 1401.394 | 1401.394 | 1 |
| | 2 | **1720.365** | 1720.374 | 0.992 | | 2 | 1816.056 | 1816.056 | 1 |
| T4 | 3 | 569.3112 | 569.3112 | 1 | T12 | 3 | 525.7118 | 525.7118 | 1 |
| | 4 | 535.5491 | **535.3541** | 1 | | 4 | **477.0593** | 477.2696 | 1 |
| | 2 | 15595.67 | 15595.67 | 0.910 | | 2 | **12804.03** | 12805.05 | 0.920 |
| T5 | 3 | **11724.93** | 11725.28 | 1 | T13 | 3 | **8816.805** | 8817.702 | 1 |
| | 4 | 8409.738 | 8409.738 | 0.984 | | 4 | **6293.774** | 6293.951 | 1 |
| | 2 | 11877.96 | 11877.96 | 0.970 | | 2 | **16228.07** | 16228.98 | 0.920 |
| T6 | 3 | **8299.779** | 8300.718 | 1 | T14 | 3 | **7255.113** | 7255.423 | 1 |
| | 4 | **7212.611** | 7213.725 | 1 | | 4 | 6427.313 | 6427.313 | 1 |
| | 2 | **4336.261** | 4336.507 | 0.955 | | 2 | **2616.286** | 2616.943 | 1 |
| T7 | 3 | 3041.076 | 3041.076 | 1 | T15 | 3 | **1978.017** | 1978.233 | 1 |
| | 4 | **2395.683** | 2421.333 | 1 | | 4 | **1526.895** | 1526.953 | 1 |
| | 2 | 1767.43 | 1767.43 | 1 | | 2 | 2226.923 | **2226.212** | 0.962 |
| T8 | 3 | **1380.766** | 1381.019 | 1 | T16 | 3 | **1232.074** | 1232.124 | 1 |
| | 4 | 1215.302 | **1211.235** | 1 | | 4 | **982.7074** | 982.9721 | 1 |

## 5 Concluding Remarks

In hyperbolic smoothing, parameters $\tau$, $\gamma$ and $\epsilon$ tend to zero, so the constraints in the subproblems make that problem (P) tends to solve (1). Parameter $\epsilon$ controls the fuzzyness degree in clustering; the higher it is, the solution becomes more and more fuzzy; the less it is, the clustering is more and more crisp. In order to compare results and efficiency of the HSFC method, zeroes of functions $h_i$ can be obtained with any method for solving equations in one variable or a predefined routine. According to the results we obtained so far and the implementation of the hyperbolic smoothing for fuzzy clustering, we can conclude that, generally, the HSFC method has a slightly better performance than original Bezdek's FCM on small real and simulated data tables. Further research is required for testing performance of HSFC method on very large data sets, with measures of efficiency, quality of solutions and running time. We are also considering to study further comparisons between HSFC and FCM with different indices, and writing the program for solving Step 6 in HSFC algorithm, that is the minimization of $f(G)$, by our means, instead of using the *optim* routine in R.

## References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
2. H.-H. Bock: Origins and extensions of the k-means algorithm in cluster analysis. Electronic Journal for History of Probability and Statistics **4** (2008)
3. Burden, R., Faires, D.: Numerical analysis, 9th ed. Brooks/Cole, Pacific Grove (2011)
4. Diday, E.: Orders and overlapping clusters by pyramids. In J.De Leeuw et al. (eds.) Multidimensional Data Analysis, DSWO Press, Leiden (1986)
5. Dunn, J. C.: A fuzzy relative of the ISODATA process and its use in detecting compact, well separated clusters. J. Cybernetics **3**, 32–57 (1974)
6. Ferraro, M. B., Giordani, P., Serafini, A.: fclust: An R Package for Fuzzy Clustering. The R Journal **11**(1), 198-210 (2019) doi: 10.32614/RJ-2019-017
7. Fisher, R. A.: The use of multiple measurements in taxonomic problems. Annals of Eugenics **7**: 179-188 (1936)
8. Hartigan, J. A.: Clustering Algorithms. Wiley, New York, NY (1975)
9. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**(1), 193-218 (1985)
10. Karush, W.: Minima of Functions of Several Variables with Inequalities as Side Constraints. Master's Thesis, Dept. of Mathematics, University of Chicago, Chicago, Illinois (1939)
11. Kuhn, H., Tucker, A.: Nonlinear programming, Proc. 2nd Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, pp. 481-492 (1951)
12. Li, D., Fukushima, M.: On the global convergence of the BFGS method for nonconvex unconstrained optimization problems. SIAM J. Optim. **11**, 1054-1064 (2001)
13. R Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2021)
14. Trejos, J., Villalobos, M. A.: Partitioning by particle swarm optimization. In: Brito, P. Bertrand, P., Cucumel G., de Carvalho, F. (eds.) Selected Contributions in Data Analysis and Classification, pp. 235-244. Springer, Berlin (2007)
15. Xavier, A.: The hyperbolic smoothing clustering method, Pattern Recognit. **43**, 731-737 (2010)
16. Yang, M. S.: A survey of fuzzy clustering. Math. Comput. Modelling **18**, 1-16 (1993)
17. Zadeh, L. A.: Fuzzy sets. Information and Control **8**(3), 338-353 (1965)


# Stochastic Collapsed Variational Inference for Structured Gaussian Process Regression Networks

Rui Meng, Herbert K. H. Lee, and Kristofer Bouchard

**Abstract** This paper presents an efficient variational inference framework for a family of structured Gaussian process regression network (SGPRN) models. We incorporate auxiliary inducing variables in latent functions and jointly treat both the distributions of the inducing variables and hyper-parameters as variational parameters. Then we take advantage of the collapsed representation of the model and propose structured variational distributions, which enables the decomposability of a tractable variational lower bound and leads to stochastic optimization. Our inference approach is able to model data in which outputs do not share a common input set, and with a computational complexity independent of the size of the inputs and outputs to easily handle datasets with missing values. Finally, we illustrate our approach on both synthetic and real data.

**Keywords:** stochastic optimization, Gaussian process, variational inference, multivariate time series, time-varying correlation

## 1 Introduction

Multi-output regression problems arise in various fields. Often, the processes that generate such datasets are nonstationary. Modern instrumentation has resulted in increasing numbers of observations, as well as the occurrence of missing values. This motivates the development of scalable methods for forecasting in such datasets.

Multi-ouput Gaussian process models or multivariate Gaussian process models (MGP) generalise the powerful Gaussian process predictive model to vector-valued

Rui Meng (✉) · Kristofer Bouchard
Biological Systems and Engineering Division, Lawrence Berkeley National Laboratory, USA, e-mail: rmeng@lbl.gov;kebouchard@lbl.gov

Herbert K. H. Lee
University of California, Santa Cruz, USA, e-mail: herbie@ucsc.edu



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_28

random fields [1]. Those models demonstrate improved prediction performance compared with independent univariate Gaussian processes (GP) because MGPs express correlations between outputs. Since the correlation information of data is encoded in the covariance function, modeling the flexible and computationally efficient cross-covariance function is of interest. In the literature of multivariate processes, many approaches are proposed to build valid cross-covariance functions including the linear model of coregionalization (LMC) [2], kernel convolution techniques [3], B-spline based coherence functions [4]. However, most of these models are designed for modelling low-dimensional stationary processes, and require Monte Carlo simulations, making inference in large datasets computationally intractable.

Modelling the complicated temporal dependencies across variables is addressed in [5, 6] by several adaptions of stochastic LMC. Such models can handle input-varying correlation across multivariate outputs. Especially for multivariate time series, [6] propose a SGPRN that captures time-varying scale, correlation, and smoothness. However, the inference in [6] is difficult to handle in applications where either the number of observations and dimension size are large or where missing data exist.

Here, we propose an efficient variational inference approach for the SGPRN by employing the inducing variable framework on all latent processes [7], taking advantage of its collapsed representation where nuisance parameters are marginalized out [8] and proposing a tractable variational bound amenable to doubly stochastic variational inference. We call our approach variational SGPRN (VSGPRN). This variational framework allows the model to handle missing data without increasing the computational complexity of inference. We numerically provide evidence of the benefits of simultaneously modeling time-varying correlation, scale and smoothness in both a synthetic experiment and a real-world problem.

The main contributions of this work are threefold:

- Learning structured Gaussian process regression networks using inducing variables on both mixing coefficients and latent functions.
- Employing doubly stochastic variational inference for structured Gaussian process regression networks by taking advantage of its collapsed representation and constructing a tractable lower bound of the loglikelihood, making it suitable for mini-batching learning.
- Demonstrating that our proposed algorithm succeeds in handling time-varying correlation on missing data under different scenarios in both synthetic data and real data.

## 2 Model

Assume $\mathbb{y}(\mathbb{x}) \in \mathbb{R}^D$ is a vector-valued function of $\mathbb{x} \in \mathbb{R}^P$, where $D$ is the dimension size of the outputs and $P$ is the dimension size of the inputs. SGPRN assumes that noisy observations $\mathbb{y}(\mathbb{x})$ are the linear combination of latent variables $\mathbb{g}(\mathbb{x}) \in \mathbb{R}^D$, corrupted by Gaussian noise $\epsilon(\mathbb{x})$. The coefficients $\mathbb{L}(\mathbb{x}) \in \mathbb{R}^{D\times D}$ of the latent functions are assumed to be a stochastic lower triangular matrix with

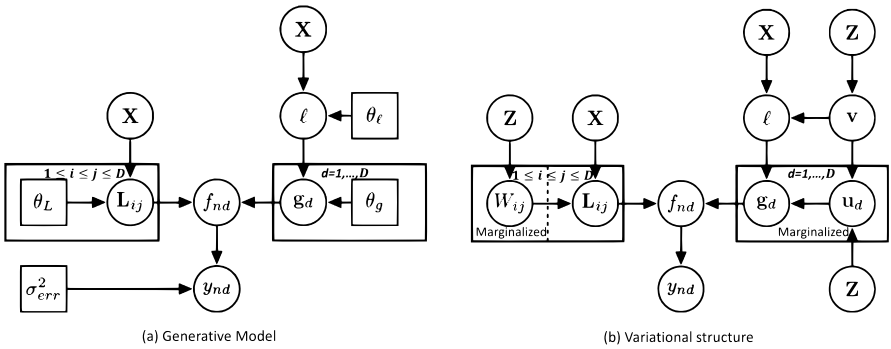


**Fig. 1** Graphical model of VSGPRN. Left: Illustration of the generative model. Right: Illustration of the variational structure. The dashed (red) block means that we marginalize out those latent variables in the variational inference framework.

positive values on the diagonal for model identification [9, 6]. Thus, SGPRN is defined in the generative model of Figure 1 and it is $\mathbb{y}(\mathbb{x}) = \mathbb{f}(\mathbb{x}) + \epsilon(\mathbb{x}), \quad \mathbb{f}(\mathbb{x}) = \mathbb{L}(\mathbb{x})\mathbb{g}(\mathbb{x})$ with independent white noise $\epsilon(\mathbb{x}) \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_{err} I)$. We note that each latent function $g_d$ in $\mathbb{g}$ is independently sampled from a GP with a non-stationary kernel $K^g$ and the stochastic coefficients are modeled via a structured GP based prior as proposed in [9] with a stationary kernel $K^l$ such that $g_d \overset{iid}{\sim} \mathrm{GP}(0, K^g)\,, d = 1, \ldots, D$ , and $l_{ij} \sim \begin{cases} \mathrm{GP}(0, K^l)\,, & i > j\,, \\ \mathrm{logGP}(0, K^l)\,, & i = j\,, \end{cases}$ where logGP denotes the log Gaussian process [10]. $K^g$ is modelled as a Gibbs correlation function $K^g(\mathbb{x}, \mathbb{x}') = \sqrt{\frac{2\ell(\mathbb{x})\ell'(\mathbb{x})}{\ell(\mathbb{x})^2 + \ell(\mathbb{x}')^2}} \exp\left(-\frac{\|\mathbb{x}-\mathbb{x}'\|^2}{\ell(\mathbb{x})^2+\ell(\mathbb{x}')^2}\right), \ell \sim \mathrm{logGP}(0, K^\ell)$ , where $\ell$ determines the input-dependent length scale of the shared correlations in $K^g$ for all latent functions $g_d$. The varying length-scale process $\ell$ plays an important role in modelling nonstationary time series as illustrated in [11, 6].

Let $\mathbb{X} = \{\mathbb{x}_i\}_{i=1}^N$ be the set of observed inputs and $\mathbb{Y} = \{\mathbb{y}_i\}_{i=1}^N$ be the set of observed outputs. Denote $\eta$ as the concatenation of all coefficients and all log length-scale parameters, i.e., $\eta = (\mathbb{l}, \tilde{\ell})$ evaluated at training inputs $\mathbb{X}$. Here, $\mathbb{l}$ is a vector including the entries below the main diagonal and the entries on the diagonal in the log scale and $\tilde{\ell} = \log \ell$ is the length-scale parameters in log scale. Also, denote $\theta = (\theta_l, \theta_\ell, \sigma^2_{err})$ as all hyper-parameters, where $\theta_l$ and $\theta_\ell$ are the hyper-parameters in kernel $K_l$ and $K_\ell$. We note that directly inferring the posterior of the latent variables $p(\eta|\mathbb{Y}, \theta) \propto p(\mathbb{Y}|\eta, \sigma^2_{err}) p(\eta|\theta_l, \theta_\ell)$ is computationally intractable in general because the computational complexity of $p(\eta|\mathbb{Y}, \theta)$ is $\mathcal{O}(N^3 D^3)$. To overcome this issue, we propose an efficient variational inference to significantly reduce the computational burden in the next section.

## 3 Inference

We introduce a shared set of inducing inputs $\mathbb{Z} = \{\mathbb{z}_m\}_{m=1}^M$ that lie in the same space as the inputs $\mathbb{X}$ and a set of shared inducing variables $\mathbb{w}_d$ for each latent function $g_d$ evaluated at the inducing inputs $\mathbb{Z}$. Likewise, we consider inducing variables $\mathbb{u}_{ii}$ for the function $\log L_{ii}$ when $i = j$, $\mathbb{u}_{ij}$ for function $L_{ij}$ when $i > j$, and inducing variables $\mathbb{v}$ for function $\log \ell(\mathbb{x})$ evaluated at inducing inputs $\mathbb{Z}$. We denote those collective variables as $\mathbb{l} = \{\mathbb{l}_{ij}\}_{i\geq j}$, $\mathbb{u} = \{\mathbb{u}_{ij}\}_{i\geq j}$, $\mathbb{g} = \{\mathbb{g}_d\}_{d=1}^D$, $\mathbb{w} = \{\mathbb{w}_d\}_{d=1}^D$, $\ell$ and $\mathbb{v}$. Then we redefine the model parameters $\eta = (\mathbb{l}, \mathbb{u}, \mathbb{g}, \mathbb{w}, \ell, \mathbb{v})$, and the prior of those model parameters is $p(\eta) = p(\mathbb{l}|\mathbb{w})p(\mathbb{w})p(\mathbb{g}|\mathbb{u}, \ell, \mathbb{v})p(\mathbb{u})p(\ell|\mathbb{v})p(\mathbb{v})$.

The core assumption of inducing point-based sparse inference is that the inducing variables are sufficient statistics for the training and testing data in the sense that the training and testing data are conditionally independent given the inducing variables. In the context of our model, this means that the posterior processes of $L$, $g$ and $\ell$ are sufficiently determined by the posterior distribution of $\mathbb{u}$, $\mathbb{w}$ and $\mathbb{v}$. We propose a structured variational distribution and its corresponding variational lower bound. Due to the nonconjugacy of this model, instead of doing expectation in the evidence lower bound (ELBO), as is normally done in the literature, we perform the marginalization on inducing variables $\mathbb{u}$, $\mathbb{w}$ and $\mathbb{g}$, and then use the reparameterization trick to apply end-to-end training with stochastic gradient descent. We will also discuss a procedure for missing data inference and prediction.

To capture the posterior dependency between the latent functions, we propose a structured variational distribution of the model parameters $\eta$ used to approximate its posterior distribution as $q(\eta) = p(\mathbb{l}|\mathbb{u})p(\mathbb{g}|\mathbb{w}, \ell, \mathbb{v})p(\ell|\mathbb{v})q(\mathbb{u}, \mathbb{w}, \mathbb{v})$ . This variational structure is illustrated in Figure 1. The variational distribution of the inducing variables $q(\mathbb{u}, \mathbb{w}, \mathbb{v})$ fully characterizes the distribution of $\mathbb{q}(\eta)$. Thus, the inference of $q(\mathbb{u}, \mathbb{w}, \mathbb{v})$ is of interest. We assume the parameters $\mathbb{u}$, $\mathbb{w}$, and $\mathbb{v}$ are Gaussian and mutually independent.

Given the definition of Gaussian process priors for the SGPRN, the conditional distributions $p(\mathbb{l}|\mathbb{u})$, $p(\mathbb{g}|\mathbb{w}, \tilde{\ell}, \mathbb{v})$, and $p(\ell|\mathbb{v})$ have closed-form expressions and all are Gaussian, except for $p(\ell|\mathbb{v})$, which is log Gaussian. The ELBO of the log likelihood of observations under our structured variational distribution $q(\eta)$ is derived using Jensen's inequality as:

$$\log p(\mathbb{Y}) \geq E_{q(\eta)}\left[\log\left(\frac{p(\mathbb{Y}|\mathbb{g}, \mathbb{l})p(\mathbb{u})p(\mathbb{w})p(\mathbb{v})}{q(\mathbb{u}, \mathbb{w}, \mathbb{v})}\right)\right] = R + A\,, \tag{1}$$

where $R = \sum_{n=1}^N \sum_{d=1}^D E_{q(\mathbb{g}_n, \mathbb{l}_n)} \log(p(y_{nd}|\mathbb{g}_n, \mathbb{l}_n))$ is the reconstruction term and $A = \mathrm{KL}(q(\mathbb{u})||p(\mathbb{u})) + \mathrm{KL}(q(\mathbb{w})||p(\mathbb{w})) + \mathrm{KL}(q(\mathbb{v})||p(\mathbb{v}))$ is the regularization term. $\mathbb{g}_n = \{g_{dn} = (\mathbb{g}_d)_n\}_{d=1}^D$ and $\mathbb{l}_n = \{l_{ijn} = (\mathbb{l}_{ij})_n\}_{i\geq j}$ are latent variables.

The structured decomposition trick for $q(\eta)$ has also been used by [12] to derive variational inference for the multivariate output case. The benefit of this structure is that all conditional distributions in $q(\eta)$ can be cancelled in the derivation of the lower bound in (1), which alleviates the computational burden of inference. Because of the conditional independence of the reconstruction term in (1) given $\mathbb{g}$ and $\mathbb{l}$, the

lower bound decomposes across both inputs and outputs and this enables the use of stochastic optimization methods. Moreover, due to the Gaussian assumption in the prior and variational distributions of the inducing variables, all KL divergence terms in the regularization term $A$ are analytically tractable. Next, instead of directly computing expectation, we leverage stochastic inference [13].

Stochastic inference requires sampling of $\mathbb{l}$ and $\mathbb{g}$ from the joint variational posterior $q(\eta)$. Directly sampling them would introduce much uncertainty from intermediate variables and thus make inference inefficient. To tackle this issue, we marginalize unnecessary intermediate variables $\mathbb{u}$ and $\mathbb{w}$ and obtain the marginal distributions $q(\mathbb{l}) = \prod_{i=j} \log \mathcal{N}(\mathbb{l}_{ii}|\tilde{\mu}^l_{ii}, \tilde{\Sigma}^l_{ii}) \prod_{i>j} \mathcal{N}(\mathbb{l}_{ij}|\tilde{\mu}^l_{ij}, \tilde{\Sigma}^l_{ij})$ and $q(\mathbb{g}|\ell, \mathbb{v}) = \prod_{d=1}^{D} \mathcal{N}(\mathbb{g}_d|\tilde{\mu}^g_d, \tilde{\Sigma}^g_d)$ with a joint distribution $q(\ell, \mathbb{v}) = p(\ell|\mathbb{v})q(\mathbb{v})$, where the conditional mean and covariance matrix are easily derived. The corresponding marginal distributions $q(\mathbb{l}_n)$ and $q(\mathbb{g}_n|\ell, \mathbb{v})$ at each $n$ are also easy to derive. Moreover, we conduct collapsed inference by marginalizing the latent variables $\mathbb{g}_n$, so then the individual expectation is

$$\mathrm{E}_{q(\mathbb{g}_n, \mathbb{l}_n)} \log(p(y_{nd}|\mathbb{g}_n, \mathbb{l}_n)) = \int (L_{nd}) q(\ell_n, \mathbb{v}) q(\mathbb{l}_{d\cdot n}) d(\mathbb{l}_{d\cdot n}, \ell_n, \mathbb{v})), \quad (2)$$

where $L_{nd} = \log \mathcal{N}(y_{nd}| \sum_{j=1}^{D} l_{djn} \tilde{\mu}^g_{jn}, \sigma^2_{err}) - \frac{1}{2\sigma^2_{err}} \sum_{j=1}^{D} l^2_{djn} \tilde{\sigma}^{g2}_{jn}$ measure the reconstruction performance for observations $\mathbb{y}_{nd}$.

Directly evaluating the ELBO is still challenging due to the non-linearities introduced by our structured prior. Recent progress in black box variational inference [13] avoids this difficulty by computing noisy unbiased estimates of the gradient of ELBO, via approximating the expectations with unbiased Monte Carlo estimates and relying on either score function estimators [14] or reparameterization gradients [13] to differentiate through a sampling process. Here we leverage the reparameterization gradients for stochastic optimization for model parameters. We note that evaluating ELBO (1) involves two sources of stochasticity from Monte Carlo sampling in (2) and from data sub-sampling stochasticity [15]. The prediction procedure is based on Bayes' rule and replaces the posterior distribution by the inferred variational distribution. In the case of missing data, the only modification in (1) is in the reconstruction term, where we sum up the likelihoods of observed data instead of complete data.

## 4 Experiments

This section illustrates the performance of our model on multivariate time series. We first show that our approach can model the time-varying correlation and smoothness of outputs on 2D synthetic datasets in three scenarios with respect to different types of frequencies but the same missing data mechanism. Then, we compare the imputation performance on missing data with other inducing-variable based sparse multivariate Gaussian process models on a real dataset.

We conduct experiments on three synthetic time series with low frequency (LF), high frequency (HF) and varying frequency (VF) respectively. They are generated from the system of equations $y_1(t) = 5\cos(2\pi w t^s) + \epsilon_1(t)$, $y_2(t) = 5(1-t)\cos(2\pi w t^s) - 5t\cos(2\pi w t^s) + \epsilon_2(t)$, where $\{\epsilon_i(t)\}_{i=1}^2$ are independent standard white noise processes. The value of $w$ refers to the frequency and the value of $s$ characterizes the smoothness. The LF and HF datasets use the same $s = 1$, implying the smoothness is invariant across time. But they employ different frequencies, $w = 2$ for LF and $w = 5$ for HF (i.e., two periods and five periods in a unit time interval respectively). The VF dataset takes $s = 2$ and $w = 5$, so that the frequency of the function is gradually increasing as time increases. For all three datasets, the system shows that as time $t$ increases from 0 to 1, the correlation between $y_1(t)$ and $y_2(t)$ gradually varies from positive to negative. Within each dataset, we randomly select 200 training data points, in which 100 time stamps are sampled on the interval $(0, 0.8)$ for the first dimension and the other 100 time stamps sampled on the interval $(0.2, 1)$ for the second dimension. For the test inputs, we randomly select 100 time stamps on the interval $(0, 1)$ for each dimension.

**Table 1** Prediction measurements on three synthetic datasets and different models. LF, HF and VF refer to low-frequency, high-frequency, and time-varying datasets. Three prediction measures are root mean square error (RMSE), average length of confidence interval (ALCI), and coverage rate (CR). All three measurements are summarized by the mean and standard deviation across 10 runs with different random initializations.

| Data | Model | RMSE | ALCI | CR |
|---|---|---|---|---|
| LF | IGPR [16] | 2.25(1.33e-13) | 2.18(1.88e-13) | 0.835(0) |
| | ICM [17] | 2.26(2.54e-5) | 2.18(1.22e-5) | 0.835(0) |
| | CMOGP [12] | 1.43(6.12e-2) | 1.36(1.98e-1) | 0.651(3.00e-2) |
| | VGPRN [18] | 1.01(0.31) | - | - |
| | VSGPRN | **1.00(1.43e-1)** | 2.21(6.56e-2) | **0.892(1.63e-2)** |
| HF | IGPR [16] | 1.51(6.01e-14) | 3.17(1.30e-13) | 0.915(2.22e-16) |
| | ICM [17] | 1.52(1.01e-5) | 3.17(1.19e-5) | 0.910(0) |
| | CMOGP [12] | 1.29(3.04e-2) | 2.34(3.31e-1) | 0.729(3.07e-2) |
| | VGPRN [18] | 1.11(0.25) | - | - |
| | VSGPRN | **1.10(1.98e-1)** | 2.74(7.94e-2) | **0.930(1.14e-2)** |
| VF | IGPR [16] | 1.64(8.17e-14) | 3.19(3.02e-13) | 0.875(0) |
| | ICM [17] | 1.66(2.37e-3) | 3.16(1.49e-3) | 0.880(1.50e-3) |
| | CMOGP [12] | 2.24(3.08e-1) | 2.56(9.29e-1) | 0.697(1.56e-1) |
| | VGPRN [18] | 1.04(0.67) | - | - |
| | VSGPRN | **1.24(1.33e-1)** | 2.92(1.21e-1) | **0.887(9.80e-3)** |

We quantify the model performance in terms of root mean square error (RMSE), average length of confidence interval (ALCI), and coverage rate (CR) on the test set. A smaller RMSE corresponds to better predictive performance of the model, and a smaller ALCI implies a smaller predictive uncertainty. As for CR, the better the model prediction performance is, the closer CR is to the percentile of the credible band. Those results are reported by the mean and standard deviation with 10 different random initializations of model parameters. Quantitative comparisons relating

to all three datasets are in Table 1. We compare with independent Gaussian process regression (IGPR) [16], the intrinsic coregionalization model (ICM) [17], Collaborative Multi-Output Gaussian Processes (CMOGP) [12] and variational inference of Gaussian process regression networks [18] on three synthetic datasets. In both CMOGP and VSGPRN approaches, we use 20 inducing variables. We further examined model predictive performance on a real-world dataset, the PM2.5 dataset from the UCI Machine Learning Repository [19]. This dataset tracks the concentration of fine inhalable particles hourly in five cities in China, along with meteorological data, from Jan 1st, 2010 to Dec 31st, 2015. We compare our model with two sparse Gaussian process models, i.e., independent sparse Gaussian process regression (ISGPR) [20] and the sparse linear model of corregionalization (SLMC) [17]. In the dataset, we consider six important attributes and use 20% of the first 5000 standardized multivaritate for training and use the others for testing. The RMSEs on the testing data are shown in Table 2, illustrating that VSGPRN had better prediction performance compared with ISGPR and SLMC, even when using fewer inducing points.

**Table 2** Empirical results for PM2.5 dataset. Each model's performance is summarized by its RMSE on the testing data. The number of equi-spaced inducing points is given in parentheses.

| Data | ISGPR (100) [20] | SLMC (100) [17] | VSGPRN (50) | VSGPRN (100) | VSGPRN (200) |
|---|---|---|---|---|---|
| PM2.5 | 0.994 | 0.948 | 0.840 | 0.708 | 0.625 |

## 5 Conclusions

We propose a novel variational inference approach for structured Gaussian process regression networks named the variational structured Gaussian process regression network, VSGPRN. We introduce inducing variables and propose a structured variational distribution to reduce the computational burden. Moreover, we take advantage of the collapsed representation of our model and construct a tractable lower bound of the log likelihood to make it suitable for doubly stochastic inference and easy to handle missing data. In our method, the computation complexity is independent of the size of the inputs and the outputs. We illustrate the superior predictive performance for both synthetic and real data.

Our inference approach, VSGPRN can be widely used for high dimensional time series to model complicated time-varying dependence across multivariate outputs. Moreover, due to its scalability and flexibility, it can be widely applied for irregularly sampled incomplete large datatsets that widely exist in various research fields including healthcare, environmental science and geoscience.

## References


1. Álvarez, M., Lawrence, N.: Computationally efficient convolved multiple output Gaussian processes. J. Mach. Learn. Res. **12**, 1459-1500 (2011)
2. Goulard, M., Voltz, M.: Linear coregionalization model: tools for estimation and choice of cross-variogram matrix. Math. Geol. **24**, 269-286 (1992)
3. Gneiting, T., Kleiber, W., Schlather, M.: Matérn cross-covariance functions for multivariate random fields. J. Am. Stat. Assoc. **105**, 1167-1177 (2010)
4. Qadir, G., Sun, Y.: Semiparametric estimation of cross-covariance functions for multivariate random fields. Biom. **77**, 547-560 (2021)
5. Gelfand, A., Schmidt, A., Banerjee, S., Sirmans, C.: Nonstationary multivariate process modeling through spatially varying coregionalization. Test. **13**, 263-312 (2004)
6. Meng, R., Soper, B., Lee, H., Liu, V., Greene, J., Ray, P.: Nonstationary multivariate Gaussian processes for electronic health records. J. Biom. Inform. **117**, 103698 (2021)
7. Titsias, M., Lawrence, N.: Bayesian Gaussian process latent variable model. Int. Conf. Artif. Intell. Stat. 844-851 (2010)
8. Teh, Y., Newman, D., Max Welling, M.: A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In: Schölkopf, B., Platt, J., Hofmann, T. (eds.) Advances in Neural Information Processing Systems **19**, (2006)
9. Guhaniyogi, R., Finley, A., Banerjee, S., Kobe, R.: Modeling complex spatial dependencies: Low-rank spatially varying cross-covariances with application to soil nutrient data. J. Agric. Biol. Environ. Stat. **18**, 274-298 (2013)
10. Møller, J., Syversveen, A., Waagepetersen, R.: Log Gaussian Cox processes. Scand. J. Stat. **25**, 451-482 (1998)
11. Remes, S., Heinonen, M., Kaski, S.: Non-stationary spectral kernels. Adv. Neural Inf. Process. Syst. **30** (2017), https://proceedings.neurips.cc/paper/2017/file/c65d7bd70fe3e5e3a2f3de681edc193d-Paper.pdf
12. Nguyen, T., Bonilla, E., et al.: Collaborative multi-output Gaussian processes. Uncertain. Artif. Intell. 643-652 (2014)
13. Titsias, M., Lázaro-Gredilla, M.: Doubly stochastic variational Bayes for non-conjugate inference. Int. Conf. Mach. Learn. 1971-1979 (2014)
14. Ranganath, R., Gerrish, S., Blei, D.: Black box variational inference. Int. Conf. Artif. Intell. Stat. 814-822 (2014)
15. Hoffman, M., Blei, D., Wang, C., Paisley, J.: Stochastic variational inference. J. Mach. Learn. Res. **14**, 1303-1347 (2013)
16. Rasmussen, C., Kuss, M.: Gaussian processes in reinforcement learning. Adv. Neural Inf. Process. Syst. 751-759 (2004)
17. Wackernagel, H.: Multivariate geostatistics: an introduction with applications. Springer Science & Business Media (2013)
18. Nguyen, T., Bonilla, E.: Efficient variational inference for Gaussian process regression networks. Int. Conf. Artif. Intell. Stat. 472-480 (2013)
19. Liang, X., Zou, T., Guo, B., Li, S., Zhang, H., Zhang, S., Huang, H., Chen, S.: Assessing Beijing's PM2.5 pollution: severity, weather impact, APEC and winter heating. Proc. R. Soc. A: Math. Phys. Eng. Sci. **471**, 20150257 (2015) https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2015.0257
20. Snelson, E., Ghahramani, Z.: Sparse Gaussian processes using pseudo-inputs. Adv. Neural Inf. Process. Syst. 1257-1264 (2006), http://papers.nips.cc/paper/2857-sparse-gaussian-processes-using-pseudo-inputs.pdf


# An Online Minorization-Maximization Algorithm

Hien Duy Nguyen, Florence Forbes, Gersende Fort, and Olivier Cappé

**Abstract** Modern statistical and machine learning settings often involve high data volume and data streaming, which require the development of online estimation algorithms. The online Expectation–Maximization (EM) algorithm extends the popular EM algorithm to this setting, via a stochastic approximation approach. We show that an online version of the Minorization–Maximization (MM) algorithm, which includes the online EM algorithm as a special case, can also be constructed in a similar manner. We demonstrate our approach via an application to the logistic regression problem and compare it to existing methods.

**Keywords:** expectation-maximization, minorization-maximization, parameter estimation, online algorithms, stochastic approximation

## 1 Introduction

Expectation–Maximization (EM) [6, 17] and Minorization–Maximization (MM) algorithms [15] are important classes of optimization procedures that allow for the construction of estimation routines for many data analytic models, including

Hien Duy Nguyen (✉)
School of Mathematics and Physics, University of Queensland, St. Lucia, 4067 QLD, Australia,
e-mail: h.nguyen7@uq.edu.au

Florence Forbes
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000, Grenoble, France,
e-mail: florence.forbes@inria.fr

Gersende Fort
Institut de Mathématiques de Toulouse, CNRS, Toulouse, France,
e-mail: gersende.fort@math.univ-toulouse.fr,

Olivier Cappé
ENS Paris, Universite PSL, CNRS, INRIA, France, e-mail: Olivier.Cappe@cnrs.fr



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_29

many finite mixture models. The benefit of such algorithms comes from the use of computationally simple surrogates in place of difficult optimization objectives.

Driven by high volume of data and streamed nature of data acquisition, there has been a rapid development of online and mini-batch algorithms that can be used to estimate models without requiring data to be accessed all at once. Online and mini-batch versions of EM algorithms can be constructed via the classic Stochastic Approximation framework (see, e.g., [2, 13]) and examples of such algorithms include those of [3, 7, 8, 10, 11, 12, 19]. Via numerical assessments, many of the algorithms above have been demonstrated to be effective in mixture model estimation problems. Online and mini-batch versions of MM algorithms on the other hand have largely been constructed following convex optimizations methods (see, e.g., [9, 14, 23]) and examples of such algorithms include those of [4, 16, 18, 22].

In this work, we provide a stochastic approximation construction of an online MM algorithm using the framework of [3]. The main advantage of our approach is that we do not make convexity assumptions and instead replace them with oracle assumptions regarding the surrogates. Compared to the online EM algorithm of [3] that this work is based upon, the `Online MM` algorithm extends the approach to allow for surrogate functions that do not require latent variable stochastic representations, which is especially useful for constructing estimation algorithms for mixture of experts (MoE) models (see, e.g. [20]). We demonstrate the `Online MM` algorithm via an application to the MoE-related logistic regression problem and compare it to competing methods.

**Notation.** By convention, vectors are column vectors. For a matrix $A$, $A^\top$ denotes its transpose. The Euclidean scalar product is denoted by $\langle a, b\rangle$. For a continuously differentiable function $\theta \mapsto h(\theta)$ (resp. twice continuously differentiable), $\nabla_\theta h$ (or simply $\nabla$ when there is no confusion) is its gradient (resp. $\nabla^2_{\theta\theta}$ is its Hessian). We denote the vectorization operator that converts matrices to column vectors by vec.

## 2 The Online MM Algorithm

Consider the optimization problem

$$\arg\max_{\theta\in\mathbb{T}} \mathbb{E}\left[f\left(\theta; X\right)\right], \tag{1}$$

where $\mathbb{T}$ is a measurable open subset of $\mathbb{R}^p$, $\mathbb{X}$ is a topological space endowed with its Borel sigma-field, $f : \mathbb{T}\times\mathbb{X} \to \mathbb{R}$ is a measurable function and $X$ is a $\mathbb{X}$-valued random variable on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. In this paper, we are interested in the setting when the expectation $\mathbb{E}\left[f\left(\theta; X\right)\right]$ has no closed form, and the optimization problem is solved by an MM-based algorithm.

Following the terminology of [15], we say that $g : \mathbb{T}\times\mathbb{X}\times\mathbb{T}, (\theta, x, \tau) \mapsto g\left(\theta, x; \tau\right)$, is a *minorizer of* $f$, if for any $\tau \in \mathbb{T}$ and for any $(\theta, x) \in \mathbb{T}\times\mathbb{X}$, it holds that

$$f(\theta; x) - f(\tau; x) \geq g(\theta, x; \tau) - g(\tau, x; \tau). \tag{2}$$

In our work, we consider the case when the minorizer function $g$ has the following structure:

A1 The minorizer surrogate $g$ is of the form:

$$g(\theta, x; \tau) = -\psi(\theta) + \left\langle \bar{S}(\tau; x), \phi(\theta) \right\rangle, \tag{3}$$

where $\psi : \mathbb{T} \to \mathbb{R}$, $\phi : \mathbb{T} \to \mathbb{R}^d$ and $\bar{S} : \mathbb{T} \times \mathbb{X} \to \mathbb{R}^d$ are measurable functions. In addition, $\phi$ and $\psi$ are continuously differentiable on $\mathbb{T}$.

We also make the following assumptions:

A2 There exists a measurable open and convex set $\mathbb{S} \subseteq \mathbb{R}^d$ such that for any $s \in \mathbb{S}$, $\gamma \in [0, 1)$ and any $(\tau, x) \in \mathbb{T} \times \mathbb{X}$:

$$s + \gamma \left\{ \bar{S}(\tau; x) - s \right\} \in \mathbb{S}.$$

A3 The expectation $\mathbb{E}[\bar{S}(\theta; X)]$ exists, is in $\mathbb{S}$, and is finite whatever $\theta \in \mathbb{T}$ but it may have no closed form. Online independent oracles $\{X_n, n \geq 0\}$, with the same distribution as $X$, are available.

A4 For any $s \in \mathbb{S}$, there exists a unique root to $\theta \mapsto -\nabla\psi(\theta) + \nabla\phi(\theta)^\top s$, which is the unique maximum on $\mathbb{T}$ of the function $\theta \mapsto -\psi(\theta) + \langle s, \phi(\theta) \rangle$. This root is denoted by $\bar{\theta}(s)$.

Seen as a function of $\theta$, $g(\cdot, x; \tau)$ is the sum of two functions: $-\psi$ and a linear combination of the components of $\phi = (\phi_1, \ldots, \phi_d)$. Assumption A1 implies that the minorizer surrogate is in a functional space spanned by these $(d + 1)$ functions. By (2) and A1–A3, it follows that

$$\mathbb{E}[f(\theta; X)] - \mathbb{E}[f(\tau; X)] \geq \psi(\tau) - \psi(\theta) + \left\langle \mathbb{E}\left[\bar{S}(\tau; X)\right], \phi(\theta) - \phi(\tau) \right\rangle, \tag{4}$$

thus providing a minorizer function for the objective function $\theta \mapsto \mathbb{E}[f(\theta; X)]$. By A4, the usual MM algorithm would define iteratively the sequence $\theta_{n+1} = \bar{\theta}\left(\mathbb{E}\left[\bar{S}(\theta_n; X)\right]\right)$. Since the expectation may not have closed form but infinite datasets are available (see A3), we propose a novel `Online MM` algorithm. It defines the sequence $\{s_n, n \geq 0\}$ as follows: given positive step sizes $\{\gamma_{n+1}, n \geq 1\}$ in $(0, 1)$ and an initial value $s_0 \in \mathbb{S}$, set for $n \geq 0$:

$$s_{n+1} = s_n + \gamma_{n+1} \left\{ \bar{S}\left(\bar{\theta}(s_n); X_{n+1}\right) - s_n \right\}. \tag{5}$$

The update mechanism (5) is a Stochastic Approximation iteration, which defines an $\mathbb{S}$-valued sequence (see A2). It consists of the construction of a sequence of minorizer functions through the definition of their *parameter* $s_n$ in the functional space spanned by $-\psi, \phi_1, \ldots, \phi_d$.

If our algorithm (5) converges, any limiting point $s_\star$ satisfies $\mathbb{E}\left[\bar{S}(\bar{\theta}(s_\star); X)\right] = s_\star$. Hence, our algorithm is designed to approximate the intractable expectation, evaluated at $\bar{\theta}(s_\star)$, where $s_\star$ satisfies a fixed point equation. The following lemma establishes the relation between the limiting points of (5) and the optimization problem (1) at hand. Namely, it implies that any limiting value $s_\star$ provides a stationary

point $\theta_\star := \bar{\theta}(s_\star)$ of the objective function $\mathbb{E}\left[f\left(\theta; X\right)\right]$ (i.e., $\theta_\star$ is a root of the derivative of the objective function). The proof follows the technique of [3]. Set

$$\mathsf{h}(s) := \mathbb{E}\left[\bar{S}\left(\bar{\theta}\left(s\right); X\right)\right] - s, \qquad \Gamma := \{s \in \mathbb{S} : \mathsf{h}(s) = 0\}.$$

**Lemma 1** *Assume that $\theta \mapsto \mathbb{E}\left[f(\theta; X)\right]$ is continuously differentiable on $\mathbb{T}$ and denote by $\mathcal{L}$ the set of its stationary points. If $s_\star \in \Gamma$, then $\bar{\theta}(s_\star) \in \mathcal{L}$. Conversely, if $\theta_\star \in \mathcal{L}$, then $s_\star := \mathbb{E}\left[\bar{S}\left(\theta_\star; X\right)\right] \in \Gamma$.*

***Proof*** A4 implies that

$$-\nabla\psi(\bar{\theta}(s)) + \nabla\phi(\bar{\theta}(s))^\top s = 0, \qquad s \in \mathbb{S}. \tag{6}$$

Use (2) and A1, and apply the expectation w.r.t. $X$ (under A3). This yields (4), which is available for any $\theta, \tau \in \mathbb{T}$. This inequality provides a minorizer function for $\theta \mapsto \mathbb{E}\left[f(\theta; X)\right]$: the difference is nonnegative and minimal (i.e. equal to zero) at $\theta = \tau$. Under the assumptions and A1, this yields

$$\nabla\mathbb{E}\left[f(\cdot; X)\right]|_{\theta=\tau} + \nabla\psi(\tau) - \nabla\phi(\tau)^\top\mathbb{E}\left[\bar{S}(\tau; X)\right] = 0. \tag{7}$$

Let $s_\star \in \Gamma$ and apply (7) with $\tau \leftarrow \bar{\theta}(s_\star)$. It then follows that

$$\nabla\mathbb{E}\left[f(\cdot; X)\right]|_{\theta=\bar{\theta}(s_\star)} + \nabla\psi(\bar{\theta}(s_\star)) - \nabla\phi(\bar{\theta}(s_\star))^\top s_\star = 0,$$

which implies $\bar{\theta}(s_\star) \in \mathcal{L}$ by (6). Conversely, if $\theta_\star \in \mathcal{L}$, then by (7), we have

$$\nabla\psi(\theta_\star) - \nabla\phi(\theta_\star)^\top\mathbb{E}\left[\bar{S}(\theta_\star; X)\right] = 0,$$

which, by A3 and A4, implies that $\theta_\star = \bar{\theta}\left(\mathbb{E}\left[\bar{S}(\theta_\star; X)\right]\right) = \bar{\theta}(s_\star)$. By definition of $s_\star$, this yields $s_\star = \mathbb{E}\left[\bar{S}\left(\bar{\theta}(s_\star); X\right)\right]$; i.e. $s_\star \in \Gamma$. □

By applying the results of [5] regarding the asymptotic convergence of Stochastic Approximation algorithms, additional regularity assumptions on $\phi, \psi, \bar{\theta}$ imply that the algorithm (5) possesses a continuously differentiable Lyapunov function $V$ defined on $\mathbb{S}$ and given by $V : s \mapsto \mathbb{E}\left[f(\bar{\theta}(s); X)\right]$, satisfying $\langle\nabla V(s), \mathsf{h}(s)\rangle \leq 0$, where the inequality is strict outside the set $\Gamma$ (see [3, Prop. 2]). In addition to Lemma 1, assumptions on the distribution of $X$ and on the stability of the sequence $\{s_n, n \geq 0\}$ are provided in [5, Thm. 2 and Lem. 1], which, combined with the usual conditions on the step sizes: $\sum_n \gamma_n = +\infty$ and $\sum_n \gamma_n^2 < \infty$, yields the almost-sure convergence of the sequence $\{s_n, n \geq 0\}$ to the set $\Gamma$, and the almost-sure convergence of the sequence $\{\bar{\theta}(s_n), n \geq 0\}$ to the set $\mathcal{L}$ of the stationary points of the objective function $\theta \mapsto \mathbb{E}\left[f(\theta; X)\right]$. Due to the limited space, the exact statement of these convergence results for our `Online MM` framework is omitted.

## 3 Example Application

As an example, we consider the logistic regression problem, where we solve (1) with

$$f(\theta; x) := yw^\top\theta - \log\left\{1 + \exp\left(w^\top\theta\right)\right\}, \qquad x := (y, w),$$

where $y \in \{0, 1\}$, $w \in \mathbb{R}^p$, and $\theta \in \mathbb{T} := \mathbb{R}^p$. Here, we assume that $X = (Y, W)$ is a random variable such that $\mathbb{E}[f(\theta; X)]$ exists for each $\theta$.

Denote by $\lambda$ the standard logistic function $\lambda(\cdot) := \exp\{\cdot\}/(1+\exp\{\cdot\})$. Following [1], (2) and A1 are verified by taking

$$\psi(\theta) := 0, \qquad \phi(\theta) := \begin{bmatrix} \theta \\ \mathrm{vec}\left(\theta\theta^\top\right) \end{bmatrix}, \qquad \bar{S}(\tau; x) = \begin{bmatrix} \bar{s}_1(\tau; x) \\ \mathrm{vec}\left(\bar{S}_2(\tau; x)\right) \end{bmatrix}$$

where

$$\bar{s}_1(\tau; x) := \left\{y - \lambda\left(\tau^\top w\right)\right\} w + \frac{1}{4} ww^\top\tau, \quad \bar{S}_2(\tau; x) = -\frac{1}{8} ww^\top.$$

With $\mathbb{S} := \{(s_1, \mathrm{vec}(S_2)) : s_1 \in \mathbb{R}^p$ and $S_2 \in \mathbb{R}^{p\times p}$ is symmetric positive definite$\}$, it follows that $\bar{\theta}(s) := -(2S_2)^{-1}s_1$.

**Online MM.** Let $s_n = \left(s_{1,n}, S_{2,n}\right) \in \mathbb{S}$. The corresponding `Online MM` recursion is then

$$s_{1,n+1} = s_{1,n} + \gamma_{n+1}\left(Y_{n+1} - \lambda\left(\bar{\theta}(s_n)^\top W_{n+1}\right) W_{n+1} + \frac{1}{4} W_{n+1}W_{n+1}^\top\bar{\theta}(s_n) - s_{1,n}\right) \quad (8)$$

$$S_{2,n+1} = S_{2,n} + \gamma_{n+1}\left(-\frac{1}{8} W_{n+1}W_{n+1}^\top - S_{2,n}\right), \quad (9)$$

where $\{(Y_{n+1}, W_{n+1}), n \geq 0\}$ are i.i.d. pairs with the same distribution as $X = (Y, W)$. Parameter estimates can then be deduced by setting $\theta_{n+1} := \bar{\theta}(s_{n+1})$.

For comparison, we also consider two Stochastic Approximation schemes directly on $\theta$ in the parameter-space: a stochastic gradient (`SG`) algorithm and a Stochastic Newton Raphson (`SNR`) algorithm.

**Stochastic gradient.** `SG` requires the gradient of $f(\theta; x)$ with respect to $\theta$: $\nabla f(\theta; x) = \{y - \lambda(\theta^\top w)\} w$, which leads to the recursion

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \gamma_{n+1}\left\{Y_{n+1} - \lambda(\hat{\theta}_n^\top W_{n+1})\right\} W_{n+1}. \quad (10)$$

**Stochastic Newton-Raphson.** In addition `SNR` requires the Hessian with respect to $\theta$, given by $\nabla^2_{\theta\theta} f(\theta; x) = -\lambda(\theta^\top w)\{1 - \lambda(\theta^\top w)\} ww^\top$. The `SNR` recursion is then

$$\hat{A}_{n+1} = \hat{A}_n + \gamma_{n+1}\left\{\nabla^2_{\theta\theta} f(\hat{\theta}_n; X_{n+1}) - \hat{A}_n\right\} \quad (11)$$

$$G_{n+1} = -\hat{A}_{n+1}^{-1} \quad (12)$$

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \gamma_{n+1} G_{n+1}\left\{Y_{n+1} - \lambda(\hat{\theta}_n^\top W_{n+1})\right\} W_{n+1}\ . \quad (13)$$

Equation (12) assumes that $\hat{A}_{n+1}$ is invertible. In this logistic example, we can guarantee this by choosing $\hat{A}_0$ to be invertible. Otherwise $\hat{A}_n$ is invertible after some $n$ sufficiently large, with probability one. Again in the logistic case, observe that, from the structure of $\nabla^2_{\theta\theta} f$ and from the Woodbury matrix identity, Equations (11–12) can be replaced by

$$G_{n+1} = \frac{G_n}{1-\gamma_{n+1}} - \frac{\gamma_{n+1}}{1-\gamma_{n+1}} \frac{a_{n+1} G_n W_{n+1} W_{n+1}^\top G_n}{\left\{(1-\gamma_{n+1}) + \gamma_{n+1} a_{n+1} W_{n+1}^\top G_n W_{n+1}\right\}} .$$

where $a_{n+1} := \lambda(\hat{\theta}_n^\top W_{n+1}) \left\{1 - \lambda(\hat{\theta}_n^\top W_{n+1})\right\}$,

It appears that the `Online MM` recursion in the $s$-space defined by (8) and (9) is equivalent to the `SNR` recursion above (i.e., (11)–(13)) when the Hessian $\nabla^2_{\theta\theta} f(\theta; x)$ is replaced by the lower bound $-\frac{1}{4} w w^\top$. This observation holds whenever $g$ is quadratic in $(\theta - \tau)$.

**Polyak averaging.** In practice, for `Online MM`, `SG`, and `SNR` recursions, it is common to consider Polyak averaging [21], starting from some iteration $n_0$, chosen such as to avoid the initial highly volatile estimates. Set $\hat{\theta}^A_{n_0} := 0$, and for $n \geq n_0$,

$$\hat{\theta}^A_{n+1} = \hat{\theta}^A_n + \alpha_{n-n_0+1}(\hat{\theta}_n - \hat{\theta}^A_n), \tag{14}$$

where $\alpha_n$ is usually set to $\alpha_n := n^{-1}$.

**Numerical illustration.** We now demonstrate the performance of the `Online MM` algorithm for logistic regression – defined by (5) and the derivations above. To do so, a sequence $\{X_i = (Y_i, W_i)\,, i \in \{1, \ldots, n_{\max}\}\}$ of $n_{\max} = 10^5$ i.i.d. replicates of $X = (Y, W)$ is simulated: $W = (1, U)$, where $U \sim \mathrm{N}(0, 1)$ and $[Y|W = w] \sim \mathrm{Ber}\left(\lambda\left(\theta_0^\top w\right)\right)$, where $\theta_0 = (3, -3)$. `Online MM` is run using the learning rate $\gamma_n = n^{-0.6}$, as suggested in [3]. The algorithm is initialized with $\hat{\theta}_0 = (0, 0)$ and $s_0 = \sum_{i=1}^2 \bar{S}\left(\hat{\theta}_0; X_i\right)/2$.

For comparison, we also show, on Figure 1, the `SG`, `SNR` estimates and their Polyak averaged values in $\theta$-space. As is usually recommended with Stochastic Approximation, the first few volatile estimations are discarded. Similarly, for Polyak averaging, we set $n_0 = 10^3$. As expected, we observe that the `Online MM` and the `SNR` recursions are very close but with the `SNR` showing more variability. Their comparison after Polyak averaging shows very close trajectories while the `SG` trajectory is clearly different and shows more bias. Final estimates [Polyak averaged estimates] of $\theta_0$ from the `SG`, `SNR`, and `Online MM` algorithms are respectively: $(2.67, -2.66)$ $[(2.51, -2.48)]$, $(3.03, -3.03)$ $[(2.99, -3.03)]$, and $(3.01, -3.03)$ $[(2.98, -3.02)]$, which we can compare to the batch maximum likelihood estimate $(3.00, -3.05)$ (obtained via the `glm` function in `R`). Notice the remarkable closeness between the `online MM` and batch estimates.

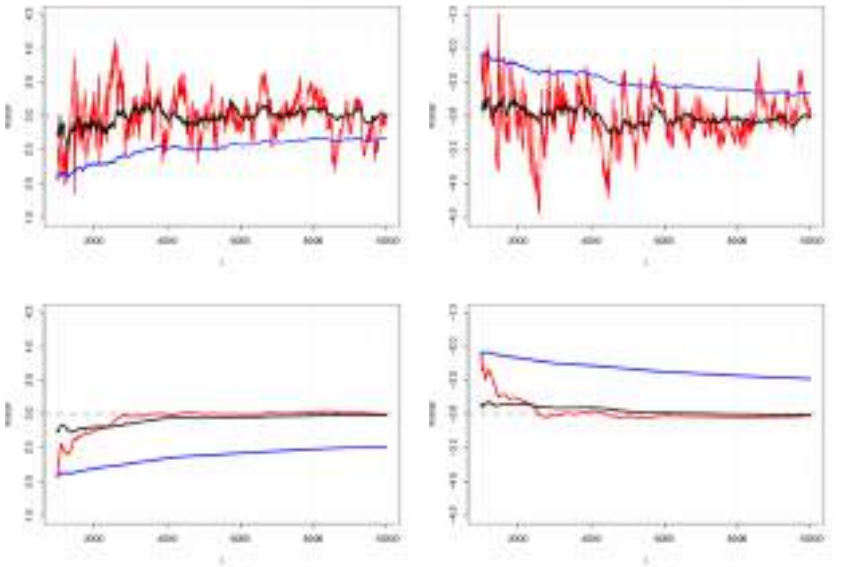

**Fig. 1** Logistic regression example: the first row shows `Online MM` (black), `SG` (blue), and `SNR` (red) recursions. The second row shows the respective Polyak averaging recursions. The estimates of the first $\theta$ (first column) and the second (second column) components of $\theta$ are plotted started from $n = 10^3$ for readability.

## 4 Final Remarks

*Remark 1* For a parametric statistical model indexed by $\theta$, let $f(\theta; x)$ be the log-density of a random variable $X$ with stochastic representation $f(\theta; x) = \log \int_{\mathbb{Y}} p_\theta(x, y)\, \mu(\mathrm{d}y)$, where $p_\theta(x, y)$ is the joint density of $(X, Y)$ with respect to the positive measure $\mu$ for some latent variable $Y \in \mathbb{Y}$. Then, via [15, Sec. 4.2], we recover the `Online EM` algorithm by using the minorizer function $g$:

$$g(\theta, x; \tau) := \int_{\mathbb{Y}} \log p_\theta(x, y)\; p_\tau(x, y) \exp(-f(\tau; x))\, \mu(\mathrm{d}y).$$

*Remark 2* Via the minorization approach of [1] (as used in Section 3) and the mixture representation from [19], we can construct an `Online MM` algorithm for MoE models, analogous to the MM algorithm of [20]. We shall provide exposition on such an algorithm in future work.

**Acknowledgements** Part of the work by G. Fort is funded by the *Fondation Simone et Cino Del Duca, Institut de France*. H. Nguyen is funded by ARC Grant DP180101192. The work is supported by Inria project LANDER.

## References


1. Bohning, D.: Multinomial logistic regression algorithm. Ann. Inst. Stat. Math. (1992)
2. Borkar, V.S.: Stochastic approximation: A dynamical systems viewpoint. Springer (2009)
3. Cappé, O., Moulines, E.: On-line expectation-maximization algorithm for latent data models. J. Roy. Stat. Soc. B Stat. Meth. **71**, 593–613 (2009)
4. Cui, Y., Pang, J.: Modern nonconvex nondifferentiable optimization. SIAM, Philadelphia (2022)
5. Delyon, B., Lavielle, M., Moulines, E.: Convergence of a stochastic approximation version of the EM algorithm. Ann. Stat. **27**, 94–128 (1999)
6. Dempster, A. P., Laird, N. M., Rubin, D. B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. B Stat. Meth. **39**, 1–38 (1977)
7. Fort, G., Gach, P., Moulines, E.: Fast incremental expectation maximization for finite-sum optimization: nonasymptotic convergence. Stat. Comput. **31**, 1–24 (2021)
8. Fort, G., Moulines, E., Wai, H. T.: A stochastic path-integrated differential estimator expectation maximization algorithm. In: Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS) (2020)
9. Hazan, E.: Introduction to online convex optimization. Foundations and Trends in Optimization. **2** (2015)
10. Karimi, B., Miasojedow, B., Moulines, E., Wai, H. T.: Non-asymptotic analysis of biased stochastic approximation scheme. Proceedings of Machine Learning Research. **99**, 1–31 (2019)
11. Karimi, B., Wai, H. T., Moulines, R., Lavielle, M.: On the global convergence of (fast) incremental expectation maximization methods. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS) (2019)
12. Kuhn, E., Matias, C., Rebafka, T.: Properties of the stochastic approximation EM alpgorithm with mini-batch sampling. Stat. Comput. **30**, 1725–1739 (2020)
13. Kushner, H. J., Yin, G. G.: Stochastic Approximation and Recursive Algorithms and Applications. Springer, New York (2003)
14. Lan, G.: First-order and Stochastic Optimization Methods for Machine Learning. Springer, Cham (2020)
15. Lange, K.: MM Optimization Algorithms. SIAM, Philadelphia (2016)
16. Mairal, J.: Stochastic majorization-minimization algorithm for large-scale optimization. In: Advances in Neural Information Processing Systems, pp. 2283–2291 (2013)
17. McLachlan, G. J., Krishnan, T.: The EM Algorithm And Extensions. Wiley, New York (2008)
18. Mokhtari, A., Koppel, A.: High-dimensional nonconvex stochastic optimization by doubly stochastic successive convex approximation. IEEE Trans. Signal Process. **68**, 6287–6302 (2020)
19. Nguyen, H.D., Forbes, F., McLachlan, G. J.: Mini-batch learning of exponential family finite mixture models. Stat. Comput. **30**, 731–748 (2020)
20. Nguyen, H. D., McLachlan, G. J.: Laplace mixture of linear experts. Comput. Stat. Data Anal. **93**, 177–191 (2016)
21. Polyak, B. T., Juditsky, A. B.: Acceleration of stochastic approximation by averaging. SIAM J. Contr. Optim. **30**, 838–855 (1992)
22. Razaviyayn, M., Sanjabi, M., Luo, Z.: A stochastic successive minimization method for nonsmooth nonconvex optimization with applications to transceiver design in wireless communication networks. Math. Program. Series B. 515–545 (2016)
23. Shalev-Shwartz, S.: Online learning and online convex optimization. Foundations and Trends in Machine Learning. **4**, 107–194 (2011)

# Detecting Differences in Italian Regional Health Services During Two Covid-19 Waves

Lucio Palazzo and Riccardo Ievoli

**Abstract** During the first two waves of Covid-19 pandemic, territorial healthcare systems have been severely stressed in many countries. The availability (and complexity) of data requires proper comparisons for understanding differences in performance of health services. We apply a three-steps approach to compare the performance of Italian healthcare system at territorial level (NUTS 2 regions), considering daily time series regarding both intensive care units and ordinary hospitalizations of Covid-19 patients. Changes between the two waves at a regional level emerge from the main results, allowing to map the pressure on territorial health services.

**Keywords:** regional healthcare, time series, multidimensional scaling, cluster analysis, trimmed $k$-means

## 1 Introduction

During the Covid-19 pandemic, the evaluation of similarities and differences between territorial health services [23] is relevant for decision makers and should guide the governance of countries [15] through the so-called "waves". This type of analysis becomes even more crucial in countries where the National healthcare system is regionally-based, which is the case of Italy (or Spain) among others. Italy is one of the countries in Europe which has been mostly affected by the pandemic, and the pressure on Regional Health Services (RHS) has been producing dramatic effects also in the economic [2] and the social [3] spheres. Regional Covid-19-related health

Lucio Palazzo (✉)
Department of Political Sciences, University of Naples Federico II, via Leopoldo Rodinò 22 - 80138 Napoli, Italy, e-mail: lucio.palazzo@unina.it

Riccardo Ievoli
Department of Chemical, Pharmaceutical and Agricultural Sciences, University of Ferrara, via Luigi Borsari 46 - 44121 Ferrara, Italy, e-mail: riccardo.ievoli@unife.it



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_30

indicators are extremely relevant for monitoring the pandemic territorial widespread [21], and to impose (or relax) restrictions in accordance with the level of health risk.

The aim of this work is to exploit the potential of Multidimensional Scaling (MDS) to detect the main imbalances occurred in the RHS, observing the hospital admission dynamics of patients with Covid-19 disease. Both daily time series regarding patients treated in Intensive Care (IC) units and individuals hospitalized in other hospital wards are used to evaluate and compare the reaction to healthcare pressure in 21 geographical areas (NUTS 2 Italian regions), considering the first two waves [4] of pandemic. Indeed, territorial imbalances in terms of RHS' performance [24] should be firstly driven by the geographical propagation flows of the virus (first wave). Then, different reactions to pandemic shock may be provided by RHSs, and changes of imbalances can be observed in the second wave.

Our proposal consists of three subsequent steps. Firstly, a matrix of distances between regional time series through a dissimilarity metric [29] is obtained. Therefore, we apply a (weighted) MDS [19, 22] to map similarity patterns in a reduced space, adding also a weighting scheme considering the number of neighbouring regions. Finally, we perform a cluster analysis to identify groups according to RHS performance in the two waves.

The paper is organized as follows: Section 2 describes the methodological approach used to compare and cluster time series, while Section 3 introduces data and descriptive analysis. Results regarding RHSs are depicted and discussed in Section 4, while Section 5 concludes with some remarks and possible advances.

## 2 Time Series Clustering

Given a matrix $T \times n$, where $T$ represents the days and $n$ the number of regions, our methodological approach consists of three subsequently steps:

*Step 1.* Compute a dissimilarity matrix $D$ based on a given measure;
*Step 2.* Apply a weighted multidimensional scaling (wMDS) procedure, storing the coordinates of the first two components;
*Step 3.* Perform cluster analysis on the MDS reduced space to identify groups between the $n$ regions.

In the first step, a dissimilarity measure is computed for each pair of regional time series. The objective is to obtain a dissimilarity matrix $D$ (with elements $d_{i,j}$) for estimating synthetic measures of the differences between regions. There are different alternatives to compare time series, some comprehensive overviews are in [29, 13].

A reasonable choice is the the Fourier dissimilarity $d_F(\mathbf{x}, \mathbf{y})$, which applies the $n$-point Discrete Fourier Transform [1] on two time series, allowing to compare the similarity between two time sequences after converting them into a combination of structural elements, such as trend and/or cycle.

In the second step, we implement a multidimensional scaling [31]. Due to its flexibility, MDS has been introduced also in time series analysis [25] and recently applied to different topics [30, 9, 16].

Since our aim is to take into account the degree of proximity between regions, we also employ a weighted multidimensional scaling technique (wMDS) [17, 14]. The $\mathcal{L}_2$ norm is multiplied by a set of weights $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ such that high weights have a stronger influence on the result than low weights.

The reduced space generated by MDS can be used as starting point for subsequent analyses. Then, a cluster algorithm can be performed on the coordinates (of the reduced space) of MDS [18]. Different procedures should be suitable to perform a cluster analysis on the wMDS coordinates map. For an overview of modern clustering techniques in time series, see e.g. [26].

In our case, both the geographical spread of the pandemic and population density can determine remarkable differences in terms of hospitalization rates [12]. To mitigate the risk of regional outliers in the data, generating potential *spurious* clusters, we employ the trimmed $k$-means algorithm [8, 11]. A relevant topic in cluster analysis is related to the choice of the $k$ number of groups. Our strategy is purely data-driven and it is based on the minimization of the within-cluster variance.

## 3 Data and Descriptive Statistics

Daily regional time series reporting a) the number of patients treated in IC units and b) the number of patients admitted in the other hospital wards are retrieved through the official website of Italian Civil Protection[1]. All patients were positive for the Covid-19 test (nasal and oropharyngeal swab). To take into account the different sizes in terms of inhabitants, both a) and b) are normalized according to the population of each territorial unit (estimated at 2020/01/01). The rates of patients treated in IC units and hospitalized (HO) patients in other hospital wards, are then multiplied by 100,000.

The whole dataset contains two identified waves[2] of Covid-19, as follows:

Wave 1 (W1): $T = 109$ days from February 24 to June 11, 2020
Wave 2 (W2): $T = 109$ days from September 14 to December 31, 2020

The date/trend may also depend on external factors, such as the implementation of restrictive measures introduced by the Italian Government [27, 6], which influenced the observed differences between W1 and W2. We have to remark that a full national lockdown was held between March 9th and May 18th 2020.

Figure 1 shows the time series for HO and IC (rows), according to the two waves of Covid-19 (columns). The anomaly of the small Italian region (Valle D'Aosta) emerges both in the first (in particular concerning IC) and second waves (also for

[1] Source: `www.dati-covid.italia.it`

[2] Refer to [7] for further details.

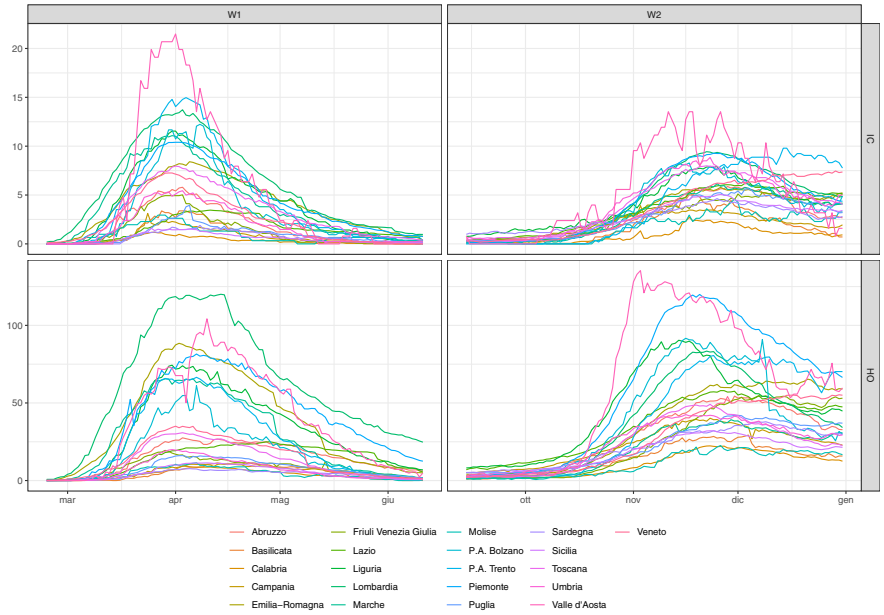


**Fig. 1** Time series distributions of Italian regions.

HO), while Lombardia, which is the largest and most populous region, dominates other territories especially when considering HO of W1. The upper panel of Figure 1 helps to understand differences between the two waves in terms of admission to intensive cares: while regions with high, medium and low IC rate can be directly identified through the eyeball of the series during W1, in W2 more homogeneity is observed. Furthermore, with the exception of Valle D'Aosta, the IC rate remains always less than 10 for all considered observations.

For what concerns HO rate, (lower panels of Figure 1), Lombardia reaches values greater than 100 in W1 (especially in April), while during W2 this threshold had exceeding by Valle D'Aosta and Piemonte (both in November). Again, if W1 opposes regions with high and (moderately) low HO rates, in W2 the following situation arises: a) Valle D'Aosta and Piemonte reach values over 100, b) four regions (Liguria, Lazio, P.A. Trento and P.A. Bolzano) present values over 75, and c) the majority of territories share similar trends with peaks always lower than 75.

## 4 Grouping Regions by Clustering and Discussion

In order to confirm and deepen the descriptive results of Section 3, we perform a cluster analysis following the scheme proposed in Section 2. We compute wMDS

equipped with the Fourier distance[3], using a set of weights $\omega$ proportional to the number of neighbourhoods for each region, ensuring a spatial feature into the model.

Figure 2 displays the main results of wMDS, distinguishing between four levels of critical issues experienced by the RHS. Outlying performances are coloured in **Violet**. A first cluster (in **Red**) includes "critical" regions while a group depicted in **Orange** contains territories with high pressure in their RHS. Regions involved in the **Green** cluster experimented a moderate pressure on RHS, while colour **Blue** indicates territories suffering from a low pressure. These clusters may also be interpreted as a ranking of the health service risk.

As regards the HO during W1, leaving apart the two outliers (Lombardia and P.A. Bolzano) the "red" cluster is composed by three Northern regions (Piemonte, Valle d'Aosta and Emilia-Romagna). The group of high pressure is composed by Liguria, Marche and P.A. Trento, while the green cluster involves Lazio, Abruzzo and Toscana (from the centre of Italy) and Veneto. The last group includes nine regions, 7 of which are located in the southern Italy. In W2 the clustering procedure Piemonte and Valle d'Aosta are identified as outliers, while the high-pressure group is composed by two autonomous provinces (Trendo and Bolzano), Lombardia and Liguria. The "orange" group is constituted by regions located in the North-East (Friuli-Venezia Giulia, Emilia-Romagna and Veneto), along with Abruzzo and Lazio. Southern regions are allocated in the "green" coloured group (together with Umbria, Toscana and Marche), while Molise, Calabria and Basilicata remain in the low-pressure cluster.

Regarding IC rates, during W1 Lombardia and Valle d'Aosta are considered as outliers while the "red" cluster is composed by four northern Italian regions (Emilia-Romagna, P.A. of Trento, Piemonte and Liguria), and Marche (located in the centre). The "orange" cluster contains Toscana, Veneto and P.A. Bolzano, while the moderate-pressure cluster involves three areas of centre Italy (Lazio and Umbria), among with the Friuli-Venezia Giulia (from the north-east) and Abruzzo. The last cluster includes only regions from the south. According to the bottom right panel of Figure 2, apart from Valle D'Aosta, the procedure identifies Calabria as an outlier. The "red" group acquires two observations from the Centre of Italy such as Toscana and Umbria, while the majority of regions are classified in the moderately pressured group. Only three Southern Italian areas are allocated in the last group (in green).

If the geography of the disease appears fundamental in W1, especially regarding adjoining territories of Lombardia, in W2 this effect is less evident. Thus, regions improving (e.g. Emilia-Romagna) or worsening (such as Lazio and Abruzzo) their clustering "ranking" can be easily observed. As mentioned, the differences of restrictive measures imposed by the Government in the two periods may have a role on these results.

[3] We remark that other distance measures have been applied. Moreover, a) the Fourier one shows better performance in terms of goodness of fit; b) the results are not sensitive with respect to the choice of distance.

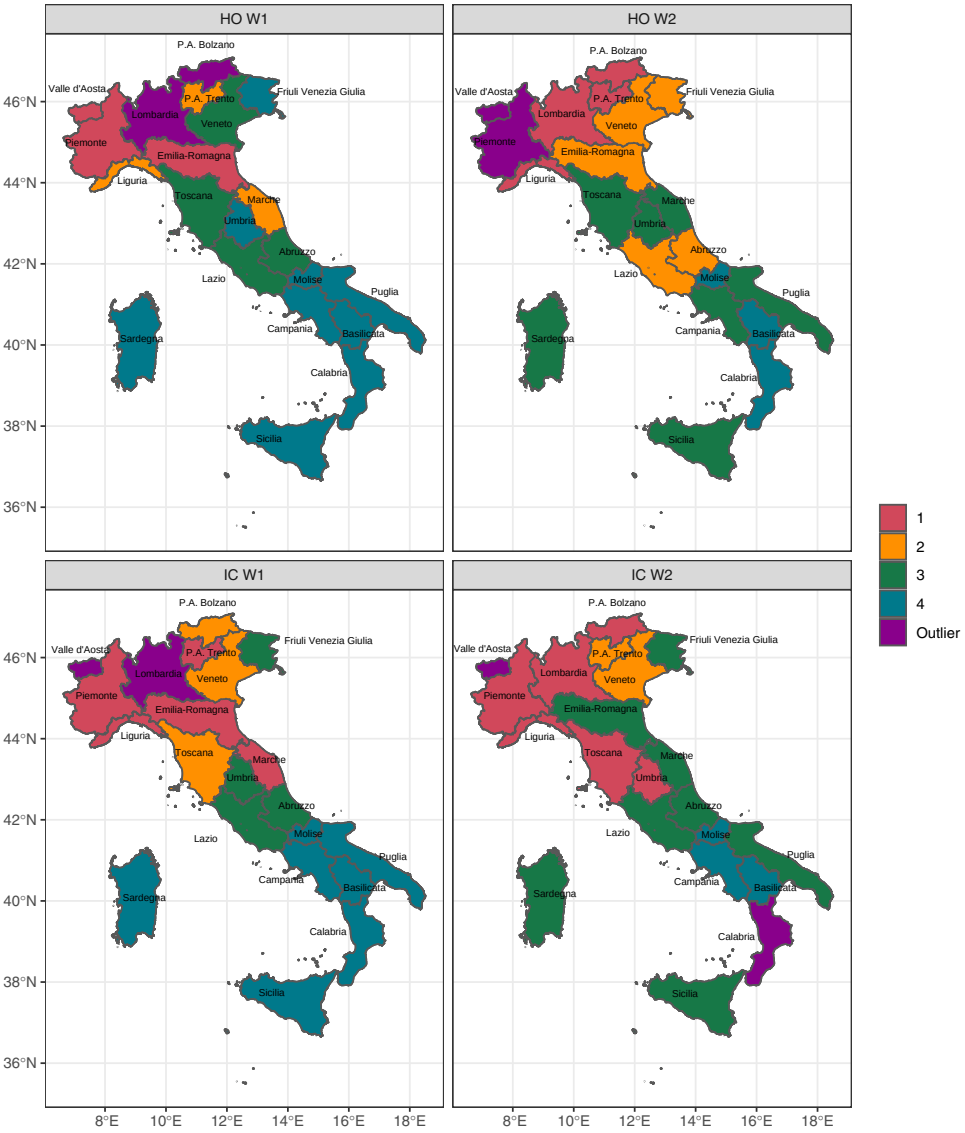


**Fig. 2** Map of the identified regional clusters.

## 5 Concluding Remarks

The Covid-19 pandemic has put a strain on the Italian healthcare system. The reactions of RHS play a relevant role to mitigate the health crisis at territorial level and to guarantee an equitable access to healthcare.

This work helps to understand similarities and divergences between the Italian regions in relation to the health pressure of the first two waves of the virus. Considering crucial measures such as HO and IC rates, the comparison between two waves allows to understand differences in the reactions to pandemic shocks of RHS. Although the northern Italy represented the epicentre of the Covid-19 spread in the first wave, some regions (e.g. Veneto and Friuli-Venezia Giulia) seem to have succeeded in avoiding hospitals overcrowding, while Southern regions (and Islands) definitively suffered from less pressure. Furthermore, in the second wave, the difference appears slightly smoothed and the cluster sizes seem more homogeneous. Moreover, there are some exceptions, such as the Emilia-Romagna, which seems to have been less affected by the second wave, compared to the other regions. The detection of clusters represents a starting point for the improvement of health governance and can be used to monitor potential imbalances in future unfortunate waves.

Further analysis may employ other dedicated indicators coming, for instance, from the Italian National Institute of Statistics[4], or using different proposals for combining wMDS with dissimilarity measures and clustering [28]. Following a different methodological approach, the recent method proposed in [10] should be applied on those data to include more complex spatial relationships between territories.

## References

1. Agrawal, R., Faloutsos, C., Swami, A.: Efficient similarity search in sequence databases. In: International Conference on Foundations of Data Organization and Algorithms, pp. 69-84. Springer, Berlin (1993)
2. Ascani, A., Faggian, A., Montresor, S.: The geography of COVID-19 and the structure of local economies: The case of Italy. Journal of Regional Science, **61**(2), 407-441 (2021)
3. Beria, P., Lunkar, V.: Presence and mobility of the population during the first wave of Covid-19 outbreak and lockdown in Italy. Sustainable Cities and Society, **65**, 102616 (2021)
4. Bontempi, E.; The Europe second wave of COVID-19 infection and the Italy "strange" situation. Environmental Research, **193**, 110476 (2021)
5. Capolongo, S., Gola, M., Brambilla, A., Morganti, A., Mosca, E. I., Barach, P.: COVID-19 and Healthcare facilities: A decalogue of design strategies for resilient hospitals. Acta Bio Medica: Atenei Parmensis, **91**(9-S), 50 (2020)
6. Chirico, F., Sacco, A., Nucera, G., Magnavita, N.: Coronavirus disease 2019: the second wave in Italy. Journal of Health Research (2021).
7. Cicchetti, A., Damiani, G., Specchia, M. L., Basile, M., Di Bidino, R., Di Brino, E., Tattoli, A.: Analisi dei modelli organizzativi di risposta al Covid-19. ALTEMS (2020). link: `https://altems.unicatt.it/altems-report47.pdf`
8. Cuesta-Albertos, J. A., Gordaliza, A., Matrán, C.: Trimmed $k$-means: An attempt to robustify quantizers. The Annals of Statistics, **25**(2), 553-576 (1997).

[4] see for example the BES indicators of the domain "Health" and "Quality of services" `https://www.istat.it/it/files//2021/03/BES\_2020.pdf`

9. Di Iorio, F., Triacca, U.: Distance between VARMA models and its application to spatial differences analysis in the relationship GDP-unemployment growth rate in Europe. In: International Work-Conference on Time Series Analysis, pp. 203-215. Springer, Cham (2017)
10. D'Urso, P., De Giovanni, L., Disegna, M., Massari, R.: Fuzzy clustering with spatial-temporal information. Spatial Statistics, **30**, 71-102 (2019)
11. Garcia-Escudero, L. A., Gordaliza, A.: Robustness properties of $k$-means and trimmed $k$-means. Journal of the American Statistical Association, **94**(447), 956–969 (1999) doi: 10.2307/2670010
12. Giuliani, D., Dickson, M. M., Espa, G., Santi, F.: Modelling and predicting the spatio-temporal spread of COVID-19 in Italy. BMC infectious diseases, **20**(1), 1-10 (2020)
13. Górecki, T., Piasecki, P.: A comprehensive comparison of distance measures for time series classification. In: Steland, A., Rafajłowicz, E., Okhrin, O. (Eds.) Workshop on Stochastic Models, Statistics and their Application, pp. 409-428. Springer, Nature (2019)
14. Greenacre, M.: Weighted metric multidimensional scaling. In: New developments in Classification and Data Analysis, pp. 141-149. Springer, Berlin, Heidelberg (2005)
15. Han, E., Tan, M. M. J., Turk, E., Sridhar, D., Leung, G. M., Shibuya, K., Legido-Quigley, H.: Lessons learnt from easing COVID-19 restrictions: an analysis of countries and regions in Asia Pacific and Europe. The Lancet, **396**(10261), 1525–1534 (2020)
16. He, J., Shang, P., Xiong, H.: Multidimensional scaling analysis of financial time series based on modified cross-sample entropy methods. Physica A: Statistical Mechanics and its Applications, **500**, 210-221 (2018)
17. Kent, J. T., Bibby, J., Mardia, K. V.: Multivariate Analysis. Amsterdam: Academic Press (1979)
18. Kruskal, J.: The relationship between multidimensional scaling and clustering. In: Classification and Clustering, pp. 17-44. Academic Press (1977)
19. Kruskal, J. B.: Multidimensional Scaling (No. 11). Sage (1978)
20. Mardia, K. V.: Some properties of classical multi-dimensional scaling. Communications in Statistics-Theory and Methods, **7**(13), 1233-1241 (1978)
21. Marziano, V., Guzzetta, G., Rondinone, B. M., Boccuni, F., Riccardo, F., Bella, A., Merler, S.: Retrospective analysis of the Italian exit strategy from COVID-19 lockdown. Proceedings of the National Academy of Sciences, **118**(4) (2021)
22. Mead, A.: Review of the development of multidimensional scaling methods. Journal of the Royal Statistical Society: Series D (The Statistician), **41**(1), 27-39 (1992)
23. Pecoraro, F., Luzi, D., Clemente, F.: Analysis of the different approaches adopted in the Italian regions to care for patients affected by COVID-19. International Journal of Environmental Research and Public Health, **18**(3), 848 (2021)
24. Pecoraro, F., Clemente, F., Luzi, D.: The efficiency in the ordinary hospital bed management in Italy: An in-depth analysis of intensive care unit in the areas affected by COVID-19 before the outbreak. PLoS One, **15**(9), e0239249 (2020)
25. Piccolo, D.: Una rappresentazione multidimensionale per modelli statistici dinamici. In: Atti della XXXII Riunione Scientifica della SIS, **2**, pp. 149-160 (1984)
26. Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., Lin, C. T.: A review of clustering techniques and developments. Neurocomputing, **267**, 664-681 (2017)
27. Sebastiani, G., Massa, M., Riboli, E.: Covid-19 epidemic in Italy: evolution, projections and impact of government measures. European Journal of Epidemiology, **35**(4), 341-345 (2020)
28. Shang, D., Shang, P., Liu, L.: Multidimensional scaling method for complex time series feature classification based on generalized complexity-invariant distance. Nonlinear Dynamics, **95**(4), 2875-2892 (2019)
29. Studer, M., Ritschard, G.: What matters in differences between life trajectories: A comparative review of sequence dissimilarity measures. Journal of the Royal Statistical Society: Series A (Statistics in Society), **179**(2), 481-511 (2016)
30. Tenreiro Machado, J. A., Lopes, A. M., Galhano, A. M.: Multidimensional scaling visualization using parametric similarity indices. Entropy, **17**(4), 1775-1794 (2015)
31. Torgerson, W. S.: Multidimensional scaling: I. Theory and method. Psychometrika, **17**(4), 401-419 (1952)

# Political and Religion Attitudes in Greece: Behavioral Discourses

Georgia Panagiotidou and Theodore Chadjipadelis

**Abstract** The research presented in this paper attempts to explore the relationship between religious and political attitudes. More specifically we investigate how religious behavior, in terms of belief intensity and practice frequency, is related to specific patterns of political behavior such as ideology, understanding democracy and his set of moral values. The analysis is based on the use of multivariable methods and more specifically Hierarchical Cluster Analysis and Multiple Correspondence Analysis in two steps. The findings are based on a survey implemented in 2019 on a sample of 506 respondents in the wider area of Thessaloniki, Greece. The aim of the research is to highlight the role of people's religious practice intensity in shaping their political views by displaying the profiles resulting from the analysis and linking individual religious and political characteristics as measured with various variables. The final output of the analysis is a map where all variable categories are visualized, bringing forward models of political behavior as associated together with other factors such as religion, moral values and democratic attitudes.

**Keywords:** political behavior, religion, democracy, multivariate methods, data analysis

## 1 Introduction

In this research we present the analysis results of a survey, which was implemented in April 2019 to 506 respondents in Thessaloniki, focusing on their religious profile as well as their political attitudes, their moral profile and the way they comprehend democracy. The aim of the analysis is to investigate and highlight the role of religious practice in shaping political behavior. In the political behavior analysis field, religion

Georgia Panagiotidou (✉)
Aristotle University of Thessaloniki, Greece, e-mail: gvpanag@polsci.auth.gr

Theodore Chadjipadelis
Aristotle University of Thessaloniki, Greece, e-mail: chadji@polsci.auth.gr



P. Brito et al. (eds.), *Classification and Data Science in the Digital Age*,
Studies in Classification, Data Analysis, and Knowledge Organization,
https://doi.org/10.1007/978-3-031-09034-9_31

and more specifically church practice has emerged as one of the main pillars that form the political attitudes of voters. Religious habits seem to have a decisive influence on electoral choices, as derives from Lazarsfeld's research at Columbia University in 1944 [3], followed by the work of Butler and Stokes in 1969 [1] and the research of Michelat and Simon in France [6]. More specifically in the comparative study of Rose in 1974 [9], it turns out that the more religious voters appear to be more conservative by choosing to place themselves on the right side of the ideological "left-right" axis, while the non-religious voters opt for the left political parties. The research and analysis of Michelat and Simon [6] brings to the surface two opposing cultural models: on the one hand we have the deeply religious voters, who belong to the middle and upper classes, residing in the cities or in the countryside, while on the other hand we have the non-religious left voters with working class characteristics. The first framework is articulated around religion and those who belong to it identifying themselves as religious people, is inspired by a conservative value system, put before the value of the individual, the family, the ancestral heritage and tradition. The second cultural context is articulated around class rivalries and socio-economic realities; those who belong to this context identify themselves as "us workers towards others". They believe in the values of collective action, vote for left-wing parties, participate actively in unions and defend the interests of the working class. To measure the influence of religious practice on political behavior, applied research uses measurement scales about the intensity of religious beliefs and the frequency of church service practice as an indicator of the level of one's religious integration.

To measure religious intensity level, variables are used such as how often they go to the service, how much do they believe in the existence of God, of afterlife, in the dogmas of the church and so on. Since the 90's there is a rapid decline in the frequency with which the population attends church service or self-identifies strongly in terms of religiousness. Nevertheless, the strong correlation between electoral preference and religious practice remains strong [5]. The most significant change for non-religious people is that the left is losing its universal influence as many of these voters expand also to the center. Strongly religious people continue to support the right more and, in some cases, strengthen the far right. In this paper, apart from attempting to explore and verify the existing literature over the effect of religion on political behavior, focusing on the Greek case, the approach exploits methods used to achieve the visualization of all existing relationships between different sets of variables. To link together numerous variables and their categories to construct a model of religious and political behavior, multiple applications of Hierarchical Cluster analysis (HCA) are being made followed by Multiple Correspondence Analysis (MCA) for the emerging clusters. In this way, a semantic map is constructed [7], which visualizes discourses of political and religious behavior and the inner antagonisms between the behavioral profiles.

## 2 Methodology

For the implementation of the research a poll was conducted on a random sample of 506 people in the greater area of Thessaloniki in Greece, during April 2019. A questionnaire was used as a research tool which was distributed with an on-site approach of the random respondents. The questionnaire consisted of three sections: a) the first section included seven questions for demographic data of the respondent such as gender, age, educational level, marital status, household income, occupation and social class to which the respondent considers belonging; b) the second part contained seven questions, ordinal variables, related to the religious practice and beliefs of the respondent: i) how often does one go to church? ii) how often does one pray? iii) how close does one feel to God, Virgin Mary (or to another seven religious concepts) during church service? iv) how strongly does one have seven different feelings during church service? v) does one believe or not in the saints, miracles, prophecies (and another six religious concepts)? Two more questions investigating their profile in terms of what is taught in the Christian dogma were included vi) one asking if one can progress only by being an ethical person and vii) another one asking if they agree on the pain/righteousness scheme, that is if one suffers in this life will be rewarded later or in the afterlife; c) questions concerning the political profile of the respondent are developed in the third part of the questionnaire: i) one's self-positioning on the ideological left-right axis, ii) a set of nine ordinal variables requiring one's agreement or disagreement level on sentences that reflect the dimensions of liberalism-authoritarianism and left-right iii) this last section also includes two different sets of pictures, used as symbolic representation for the "democratic self" and the "moral self" [4]. The first set of twelve pictures represent various conceptualizations of democracy, and one is asked to select three pictures that represent democracy. The second set of pictures represent moral values in life, and one is asked to choose three pictures that represent one's set of personal values. Variables are ordinal, using a five-point Likert scale, apart from the question regarding whether one believes or not in prophecies magic etc. and the two last questions with the pictures, where we are using a binary scale of yes-no or zero-one where zero is for a non selected picture and one is for a selected picture.

Data analysis was implemented with the use of M.A.D software (Méthodes d'Analyse des Données), developed by Professor Dimitris Karapistolis (more about M.A.D software at `www.pylimad.gr`). Firstly, Hierarchical Cluster Analysis (HCA) using chi-quare distance and Ward's linkage, assigns subjects into distinct groups based on their response patterns. This first step produces a cluster membership variable, assigning each subject into a group. In addition to this, the behavior typology of each group is examined, seeing the connection of each variable level to each cluster using two proportion $z$ test (significance level set at 0.05) between respondents belonging to cluster $i$ and those who do not belong in cluster $i$ for a variable level. The number of clusters is determined by using the empirical criterion of the change in the ratio of between-cluster inertia to total inertia, when moving from a partition with $r$ clusters to a partition with $r - 1$ clusters [8]. In the second step of the analysis, the cluster membership variable is analyzed together with the existing variables using

MCA on the Burt table [2]. All associations among the variable categories are given on a set of orthogonal axes, with the least possible loss of the original information of the original Burt table. Next, we apply HCA for the coordinates of variable categories on the total number of dimensions of the reduced space resulting from the MCA. In this way we cluster the variable, as previously we clustered the subjects. By clustering the variable response categories, we detect the various discourses of behavior, where each cluster of categories stands as a behavioral profile linked with a set of responses and characteristics. To produce the final output, the semantic map, we created a table including the output variables of the questionnaire, including demographics and variables for political behavior. Using the same two-step procedure using HCA and MCA for this final table, the semantic map is constructed, positioning the variable categories on a bi plot created by the two first dimensions of MCA.

## 3 Results

In the first step of the analysis, we apply HCA for each set of variables in each question. In the question: "How close do you feel during the service 1-To God, 2-To the Virgin, 3-To Christ, 4-To some Saint, Angel, 5-To the other churchgoers, 6-To Paradise, 7-To Hell, 8-To the divine service, 9-To his preaching priest", we get four clusters (Figure 1).

| Cluster | Responses related to the cluster | % |
|---|---|---|
| e19837 | "not at all" in everything | 7,9% |
| e19882 | "enough" in 1,2,3 / "little" or "not at all" in 5,6,9 | 55,1% |
| e19883 | "a little" in 1,2,3 / "not at all" in 4,5,6,8,9 | 19,5% |
| e19884 | "absolutely" in everything and "enough" in 5,6,9 | 17,5% |

**Fig. 1** Four clusters on how close the respondents feel during church service.

For the question: "How strongly you feel after the end of the service 1-The Grace of God in me, 2-Power of the soul, 3-Forgiveness for those who have hurt me, 4-Forgiveness for my sins, 5-Peace, 6-Relief it is over", we get six clusters (Figure 2).

| Cluster | Responses related to the cluster | % |
|---|---|---|
| e21902 | in everything "absolutely" | 9,0% |
| e21904 | "absolutely" peace, strength of soul / "not at all" forgiveness, relief | 23,4% |
| e21905 | in all "absolutely" / "not at all" relief | 11,8% |
| e21906 | "quite" relief / in all others "a little" | 16,8% |
| e21907 | in everything "not at all" | 5,9% |
| e21908 | in all "enough" | 33,0% |

**Fig. 2** Six clusters on how the respondents feel at the end of church service.

Five clusters (Figure 3) for the question: “Do you believe in 1-Bad (magic influence) 2-Magic? 3- Destiny? 4-Miracles? 5-Prophecies of the Saints? 6- Do you have pictures of holy figures in your house? 7-in your workplace? 8-Do you have a family Saint?”.

| Cluster | Responses related to the cluster | % |
|---|---|---|
| e22877 | yes to miracles and images | 23,8% |
| e22872 | yes to miracles, prophecies and pictures | 12,0% |
| e22874 | not at all | 8,4% |
| e22875 | yes in bad influence, magic, miracles, prophecies and pictures | 17,4% |
| e22879 | yes to all | 37,8% |

**Fig. 3** Five clusters on the beliefs of the respondents on various aspects of the Christian faith.

Six clusters are detected (Figure 4) for the question: “How do you feel when you come face to face with a religious image 1-Peace, 2-Awe, 3-The presence of God, 4-Emotion, 5-The need to pray, 6-Contact with the person in the picture”.

| Cluster | Responses related to the cluster | % |
|---|---|---|
| e23886 | in everything "not at all" | 5,1% |
| e23887 | in all other "moderately" (a little in awe, emotion / enough in prayer) | 16,9% |
| e23890 | "not at all" in prayer and person in the picture / in everything else "a little" | 9,8% |
| e23892 | in everything "absolutely" | 15,3% |
| e23893 | "not at all" in awe / in everything else "a little" | 12,4% |
| e23894 | in everything "enough" | 40,4% |

**Fig. 4** Six clusters on how the respondents feel when facing a religious image.

We proceed with the clustering of the replies on political views and we get seven clusters of political profiles (Figure 5).

| Cluster | Responses related to the cluster | % |
|---|---|---|
| e29881 | "strongly agrees" with drachma, individualism, anti-immigrant, anti-EU, welfare state, not leader | 7,8% |
| e29885 | "agrees" with welfare state agrees, "disagrees" with all the rest | 8,2% |
| e29886 | "agrees" with strong leader, tax cuts | 27,6% |
| e29887 | "disagrees" with the right to violence, "agrees" with all the rest | 8,9% |
| e29889 | "agrees" with drachma, individualism, anti-immigrants, welfare state, not leader (difference with 881, here simply "agrees" and not interested in EU) | 14,0% |
| e29890 | "agrees" with drachma, "disagrees" with all the rest | 11,4% |
| e29891 | "agrees" with tax cuts, drachma, anti-immigrant, anti-EU, individualism, strong leader | 22,0% |

**Fig. 5** Seven clusters according to the political views- profile of the respondents.

For the symbolic representation of the democratic self, when choosing three pictures that represent democracy for the respondent, we find eight clusters (Figure 6), and eight clusters for the symbolic representation of the moral self for the respondents, as show in Figure 7.

| Cluster | Responses related to the cluster | % |
|---|---|---|
| e31892 | direct democracy, money, revolution, riot | 5,4% |
| e31893 | parliament, money | 2,4% |
| e31914 | direct democracy | 11,6% |
| e31916 | parliament, council, church | 10,9% |
| e31918 | protest, revolution | 10,7% |
| e31920 | e-gov | 14,2% |
| e31921 | protest, council, revolution | 13,3% |
| e31924 | protest, ancient Greece, parliament, volunteering, church | 31,5% |

**Fig. 6** Eight clusters on how the respondents understand democracy.

| Cluster | Responses related to the cluster | % |
|---|---|---|
| e30970 | Christ, intimacy, volunteering, family | 24,9% |
| e30953 | fun, intimacy, meditation, win, rebellion | 2,2% |
| e30958 | Christ, family, army | 13,7% |
| e30960 | meditation, win | 7,6% |
| e30961 | fun, career, intimacy, money | 7,4% |
| e30972 | career, win, fun, career | 17,2% |
| e30966 | career, peace, family | 9,4% |
| e30968 | Christ, peace, family | 17,6% |

**Fig. 7** Eight clusters on the different sets of moral values of the respondents.

In the second step of the analysis, we jointly process the cluster membership variables. MCA produces the coefficients of each variable category which are now positioned in a two-dimensional map as seen in Figure 9. HCA is then applied again to the coefficients of the items, which bring forward three main clusters, modeling political and religious behavior. In Figure 8, Cluster 77 is connected to centre and moderate religious behaviour, cluster 78 reflects the voters of the right, with strong religious habits and beliefs, individualistic attitudes and more authoritarian and nationalistic political views, whereas cluster 79 represents the leftists, non-religious voters, closer to revolutionary political views and collective goods. Examining the antagonisms on the behavioral map (Figure 9), the first horizontal axis which explains 22.8% of the total inertia, is created by the antithesis between right political ideology - strong religious behavior and left political ideology-no religious behavior (cluster 78 opposite to cluster 79). The second axis (vertical) accounts for 7% of the inertia, and is explained as the opposition between the center (moderate religious behavior) against the left and right (cluster 77 opposite to both clusters 78 and 79).

| Variables | 77 | 78 | 79 |
|---|---|---|---|
| Ethical person | Enough, a little | Absolutely | Not at all |
| Pain / Righteousness | A little / moderately | Enough / Very / Absolutely | Not at all |
| Ideology | Centre | Right | Left |
| Praying | | I pray often | I pray sometimes / I never pray |
| Church service | | I go to church often | I rarely go to church |
| Political attitudes | [pro-drachma, individualism, anti-immigrant, anti-EU, welfare state, not leader (strongly agrees)] [strong leader, tax cuts (agrees)] [tax cuts, pro- drachma, against immigrants, against EU, person first, strong leader] | [in all others agrees / disagrees on the right to violence] [agrees with drachma, individualism, against immigrants, welfare state, not leader] [difference with 881, here simply agrees and there is no EU)] [better with drachma, everything else disagrees] | [welfare state agrees, all the rest disagrees] |
| Democratic self | [parliament, money] [direct democracy] [e-gov] | [parliament, council, church] [protest, ancient Greece, parliament, volunteering, church] | [direct democracy, money, revolution, riot] [protest, revolution] [protest, council, revolution] |
| Moral self | [Meditation, win] [Fun, Career, intimacy, Money] [Career, peace, family] [Christ, Peace, family] | [Christ, Family, Army] [Christ, intimacy, volunteering, family] | [fun, intimacy, meditation, win, rebellion] [career, win, Fun, Career] |

**Fig. 8** Three main behavioral discourses linking all variable categories together.

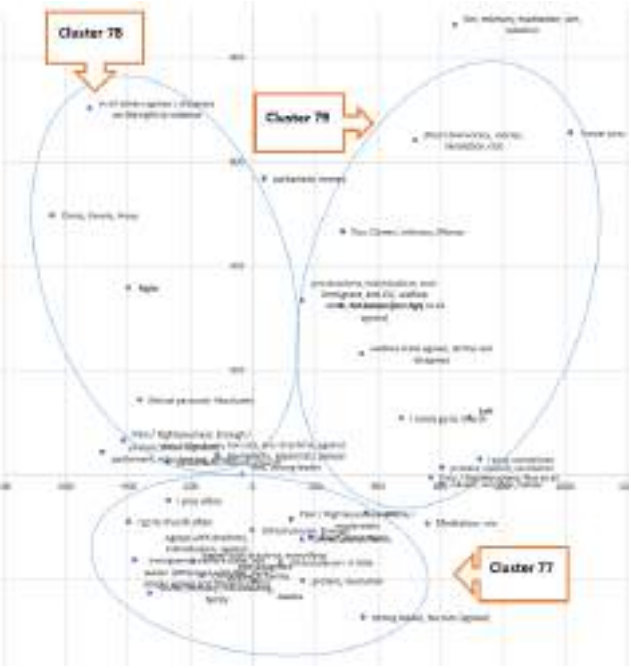


**Fig. 9** The semantic map visualizing the behavioral profiles of voters, and the inner antagonisms.

## 4 Discussion

The analysis uncovers the strong existing relationship between religious habits and political views, for the Greek case. The semantic map indicates two main antagonistic cultural discourses, including both religious, political and moral characteristics: The first discourse (cluster 77) is described as moderately religious practice and beliefs, connected to the ideological center. These voters have political attitudes that belong to the space between the center-left and the center-right. They understand democracy as a connection to money, direct democracy and electronic democracy. Their moral set of values is naturalistic and individualistic. The next behavioral discourse (cluster 78) describes the voters of right ideology, with strong religious beliefs andfrequent religious practice. They appear as very ethical and believe in the concept of pain and righteousness. Regarding their political attitudes these more religious voters are against violence, have more authoritarian and nationalistic positions. They view democracy as parliamentary, representative, ancient Greece but also as church, while their moral set of values appear clearly naturalistic, Christian and nationalistic.

Cluster 79 reflects the exact opposite discourse compared to 78. These voters belong to the left ideology and are non-religious. They do not adopt the ideas of the ethical person, or the scheme of pain and righteousness as mentioned in the Christian dogma. In terms of political attitudes, they are pro-welfare state. These non-religious and left voters understand democracy as direct with the need for revolution, protest and riot and support collective goods. Interpreting further the antagonisms as visualized on the semantic map, the main competition exists between the "right political ideology - strong religious behavior individualism" discourse and the "left political ideology-no religious behavior collectivism" discourse. A secondary opposition is found between the "center ideology- moderate religious behavior" discourse against the left and right extreme positions.

## References

1. Butler, D., Stokes, D.: Political Change in Britain. Macmillan, London (1969)
2. Greenacre, M.: Correspondence Analysis in Practice. Chapman and Hall/CRC Press, Boca Raton (2007)
3. Lazarsfeld, P. F., Berelson, B., Gaudet, H.: The People's Choice. Columbia University Press (1944)
4. Marangudakis, M., Chadjipadelis, T.: The Greek Crisis and its Cultural Origins. Palgrave-Macmillan, New York (2019)
5. Mayer, N.: Les Modèles Explicatifs du Vote. L'Harmatan, Paris (1997)
6. Michelat, G., Simon, M.: Classe, Religion et Comportement Politique. PFNSP-Editions Sociales, Paris (1977)
7. Panagiotidou, G., Chadjipadelis, T.: First-time voters in Greece: views and attitudes of youth on Europe and democracy. In T. Chadjipadelis, B. Lausen, A. Markos, T. R. Lee, A. Montanari and R. Nugent (Eds.), Data Analysis and Rationality in a Complex World, Studies in Classification, Data Analysis and Knowledge Organization, pp. 415-429. Springer (2020)
8. Papadimitriou, G., Florou, G.: Contribution of the Euclidean and chi-square metrics to determining the most ideal clustering in ascending hierarchy (in Greek). In Annals in Honor of Professor I. Liakis, 546-581. University of Macedonia, Thessaloniki (1996)
9. Rose, R.: Electoral Behavior: a Comparative Handbook. Free Press, New York (1974)