k-Means clustering in EEG (brain waves) timeseries

Nadezda Sukhorukova and Liam Kelly

Abstract Clustering is a process of assigning data points into groups (clusters). The goal is to assign "similar" points to the same cluster, while "different" points belong to distinct clusters. k-means and its modification form a very important and commonly used group of clustering methods. k-means is fast, simple and can be easily implemented in any programming package. Classical k-means leads to a local minimum of a function that represents the sum of the squares of the Euclidean distances between all the points and the centres of the clusters where the points were assigned. Therefore, the mathematical interpretation of k-means is straigthforward. In this paper, we use k-means methods and their modifications to produce clusters in timeseries data and use these clusters to create classification rules for distinguishing between the classes. The results of numerical experiments demonstrate that these fast and simple approaches provide high classification accuracy.

1 Introduction

Clustering is a process of grouping similar points. This technique allows one to identify the structure of data. In signal processing, there are two main approaches for representing signals (timeseries): by treating each time moment as an independent dimension or by constructing a prototype curve for each group of similar timeseries. In this study the former approach is used: each timeseries of n time moments is simply a point in $\mathbf{R}^n$. At first glance, this approach completely ignores the fact that the timeseries values at the nearby time moment strongly relate to each other, but

Nadezda Sukhorukova
Swinburne University of Technology, John Street, Hawthorn, Victoria, Australia 3122, e-mail: nsukhorukova@swin.edu.au

Liam Kelly
Swinburne University of Technology, John Street, Hawthorn, Victoria, Australia 3122, e-mail: lkelly@student.swin.edu.au

the corresponding optimisation problems are simple and can be handled even for high dimensional timeseries. Our numerical experiments demonstrate that, despite its simplicity, this approach is very efficient for the data we use in this study.

The k-means algorithm is a fast and well-studied clustering method developed for grouping points in $\mathbf{R}^n$. The name "k-means" was first proposed in 1967 by James MacQueen [6]. Mathematically, this method is based on the minimisation of the so-called total dissimilarity function, which is the sum of squares of the Euclidean distances between the points and the cluster centres to which these points are assigned. There are a number of modifications of k-means: k-medians, where the dissimilarity is computed in the Manhattan-distance formulation, some combinations of k-means and k-medians. For theoretical and computational properties of this group of methods and their applicability to data analysis and signal processing, we refer to [2, 10].

Another clustering method that we use in this study is k-medoids. The k-medoids problems are known to be NP-complete [8]. A comprehensive review of such methods can be found in [4].

In general, the k-medoids methods are slower than k-means, but in some cases they may be more suitable. For example, when the dissimilarity measure is not the square of the Euclidean distance and/or it is beneficial to assign one of the sample point from the data to the cluster centres. Examples of such problems can be found in [3].

2 Mathematical background of k-means

2.1 k-means and k-means with L_1 norm

The classical k-means method contains two steps.

1. Assign each point to the cluster with the nearest centre. The initial location of the cluster centres can be chosen in a number of ways. One of the most common ways is to take k random points from the data.
2. For each cluster, we recompute the centres by minimising the total sum of dissimilarities within each cluster.

We repeat these two steps until none of the points move from one cluster to another. In the case when the square of the Euclidean distance is chosen as the dissimilarity function, the minimal total dissimilarity within each cluster is simply the barycenter of the cluster and therefore the algorithm is very fast. This implementation of k-means terminates at a local minimum of the function

$$\min \sum_{i=1}^{N} \min_{c_1,\ldots,c_k} \|x_i - c_j\|, \tag{1}$$

where N is the total number of points in the dataset and $c_1, \ldots, c_k$ are the cluster centres (each point is assigned to the cluster with the nearest centre).

k-medians has a similar structure, but the dissimilarity function is based on L_1 norm (step 1) and at step 2 one computes the median point rather than the barycentre. Finally, *k*-means with L_1 is a combination of *k*-means and *k*-medians: step 1 is the same as in *k*-medians, while the updated centre is the barycentre.

2.1.1 *k*-medoids

As mentioned in section 1, the *k*-medoids problem (in its classical formulation) is NP-complete [8], but there are a number of approaches for *k*-medoids that deal with some reformulations of the classical *k*-medoids problem, for example, the Partitioning Around Medoids (PAM) algorithm [5]. PAM relies on iterative clustering rather than optimisation and uses a number of heuristics.

In its classical form, the *k*-medoids problem can be formulated as an integer linear programming problem (ILPP) [7]. Assume that there are n data points in total and the distance matrix

$$\mathbf{D} = \{d_{ij}\},\ i = 1,\dots,n,\ j = 1,\dots,n.$$

The goal is to select k points as centres. The decision variables are binary: $x_{ij} \in \{0,1\},\ i = 1,\dots,n,\ j = 1,\dots,n$ and $y_i \in \{0,1\},\ i = 1,\dots,n$. Variable y_i is 1 if point i is treated as a centre, otherwise, this variable is zero. Variable x_{ij} is 1 if point i was assigned to centre j. The corresponding optimisation problem is as follows:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{2}$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1,\ i = 1,\dots,n; \tag{3}$$

$$x_{ij} \le y_i,\ i,j = 1,\dots,n; \tag{4}$$

$$\sum_{i=1}^{n} y_i = k; \tag{5}$$

$$x_{ij},\ y_i \in \{0,1\},\ i,j = 1,\dots,n. \tag{6}$$

Coefficients d_{ij} are the distances between points i and j (based on chosen similarity). Constraints (3) ensure that each point is assigned to a single centre. Constraints (4) ensure that point i can only be assigned to point j if this point is also a centre. Finally, constraint (5) ensures that exactly k points are selected as centres. It is clear that problem (2)-(6) is an ILPP. It can be solved, for example, by using a Branch and Bound method.

3 Numerical experiments

In our study, we use EEG signals (brain waves), from a publicly available dataset, collected by the epileptic center at the University of Bonn, Germany [1]. This dataset consists of five classes: 1, 2, 3, 4 and 5. Each class contains 100 signal segments recorded during 23.6 seconds with a sampling frequency of 173.61 Hz, which means that every signal has 4097 recordings.

The same dataset (class 1 and class 2 only) was used in [9], where the authors applied a more advanced and time-consuming optimisation based approach for data classification. The models in [9] are more accurate (for separating class 1 and class 2), but our approach is simple and fast. Moreover, it provides a reasonable level of classification accuracy for separating the remaining classes within this dataset and this topic was not touched at all in [9].

Classes 1 and 2 were recorded from five healthy volunteers. Volunteers were relaxed and awake with eyes open (class 1) and eyes closed (class 2). Classes 3 and 4 correspond to the patients who are seizure free during the recording, but had a seizure in the past and it occurred in the opposite hemisphere of the brain (class 3) or in the same (class 4). Finally, class 5 segments were recorded during an active stage of seizure.

In our experiments, we take 80% of the points of each class for the training set, while the remaining 20% are the test set. All the experiments were performed on all possible pairs of classes (pair-wise separation). We produced up to six clusters in each class (training sets), assigned the points from the test sets to the clusters with the nearest centres and calculated the percentage of correctly classified points (test set classification accuracy). We use four types of clustering: k-means, k-means with L_1, k-medoids with squared Euclidean distance as the similarity measure (we call it k-medoids for simplicity) and k-medoids with L_1 as the similarity measure.

The k-means group of clustering is very sensitive to the choice of the initial centres. To be able to compare results obtained for different clustering settings, we always choose the first k points from the data as initial centres rather than choosing them randomly.

Tables 1-4 represent the test set classification results. In these tables, we only present the highest classification accuracy for each pair-wise combination of classes. The conclusions are as follows.

1. The k-means group of methods is significantly faster and more accurate than the k-medoids group.
2. Theoretically, the increase in the number of clusters leads to more accurate approximations of classes and higher classification accuracy. In practice, this is not always the case. One possible explanation for this observation is that the increase in the number of clusters leads to the increase in the dimension of the corresponding optimisation problems. Since we can only guarantee local optimality, the approximation quality for a higher number of clusters may be questionable.
3. Despite their simplicity, methods from the k-means group demonstrated their classification efficiency even for clustering very complex and long timeseries.

Table 1 *k*-means clustering and classification results

Classes to separate	Optimal number of clusters per class (highest test accuracy)	Test classification accuracy	Time (seconds)
1,2	2,3	52.5%	1.016071
1,3	5,5	82.5%	0.961105
1,4	6,6	72.5%	1.019396
1,5	6,6	62.5%	0.99491
2,3	4,4	77.5%	1.01654
2,4	5,6	77.5%	1.050314
2,5	5,6	65%	0.974254
3,4	4,6	72.5%	1.051843
3,5	1,2	65%	0.943939
4,5	6,6	65%	1.01614

Table 2 *k*-means with L_1 clustering and classification results

Classes to separate	Optimal number of clusters per class (highest test accuracy)	Test classification accuracy	Time (seconds)
1,2	5,6	57.5%	1.496592
1,3	4,4	80%	1.617608
1,4	5,5	77.5%	1.428125
1,5	1,3	72.5%	1.354719
2,3	2,2	72.5%	1.714098
2,4	5,5	77.5%	1.341092
2,5	4,6	77.5%	1.338823
3,4	1,1	72.5%	1.606779
3,5	1,2	70%	1.623985
4,5	1,1	67.5%	1.29161

Table 3 *k*-medoids clustering and classification results

Classes to separate	Optimal number of clusters per class (highest test accuracy)	Test classification accuracy	Time (seconds)
1,2	1,2	55%	33.510218
1,3	2,6	50%	33.318993
1,4	1,1	52.5%	33.941688
1,5	1,2	52.5%	33.284459
2,3	2,1	57.5%	33.552932
2,4	2,1	55%	33.776406
2,5	1,2	52.5%	33.459145
3,4	5,1	60%	33.954362
3,5	1,2	52.5%	33.355816
4,5	1,2	52.5%	33.840175

Table 4 k-mediods with L_1 clustering and classification results

Classes to separate	Optimal number of clusters per class (highest test accuracy)	Test classification accuracy	Time (seconds)
1,2	1,3	57.5%	31.80871
1,3	1,3	52.5%	31.541411
1,4	2,5	62.5%	31.948276
1,5	1,2	52.5%	31.609486
2,3	3,1	60%	31.762258
2,4	1,1	52.5%	31.880342
2,5	1,2	52.5%	31.729709
3,4	3,5	70%	31.871092
3,5	1,2	52.5%	31.58272
4,5	1,2	52.5%	31.682192

4 Conclusions and further research directions

Our numerical experiments demonstrated that k-means group of methods can be used for distinguishing classes of timeseries recorded from epileptic patients. For pair-wise separation of classes, the test set classification accuracy is around 70%-80%.

For our future studies, we identify two main directions.

1. The k-means group of methods is very sensitive to the initial choice of cluster centres and therefore the study on the choice of efficient initial centres is needed.
2. The k-medoids group of methods did not perform well in our experiments. The only potential direction here is the separation between class 3 and class 4, which essentially means the prediction which brain hemisphere one can expect a seizure. Despite being slow and not very accurate, this group of methods is less sensitive to the initial choice of cluster centres and therefore still has some potential if its efficiency is improved.

Acknowledgements This research was supported by the Australian Research Council (ARC), Solving hard Chebyshev approximation problems through nonsmooth analysis (Discovery Project DP180100602).

References

1. Ralph G Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6):061907, 2001.
2. Adil Bagirov and Julien Ugon. Nonsmooth dc programming approach to clusterwise linear regression: optimality conditions and algorithms. *Optimization methods and software*, 33(1):194–219, 2018.

3. Behrooz Bodaghi and Nadezda Sukhorukova. Solving resource allocation problems in disaster management using linear programming relaxations. *Communications in Optimization Theory*, 2022:1–11, 2022.
4. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. Springer Series in Satistics, 2008.
5. L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: an introduction to cluster analysis*. Wiley, 1990.
6. James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967.
7. Hrishikesh D. Vinod Mathematica. Integer programming and the theory of grouping. *Journal of the American Statistical Association*, 64:506–519, 1969.
8. C. Papadimitriou. Worst-case and probabilistic analysis of a geometric location problem. *SIAM Journal on Computing*, 10(3):542–557, 1981.
9. V. Peiris, N. Sharon, N. Sukhorukova, and J. Ugon. Generalised rational approximation and its application to improve deep learning classifiers. *Applied Mathematics and Computation*, 389:125560, 2021.
10. H. Späth. *Cluster Analysis Algorithms for Data Reduction and Classification of Objects*. Ellis Horwood Limited, Chichester, 1980.