QDC: Quick Density Clustering of Geo-located Data

Katarzyna Kopczewska

Abstract—**This paper develops the Quick Density Clustering (QDC) method which fills the gap in the toolbox of density clustering of spatially geo-located points. It uses a K-means algorithm which is run on two normalized spatial variables: fixed-radius nearest neighbours (NN) and a sum of distances to k nearest neighbours (NN) to find diverse densities of points in 2D. Clusters detected by QDC classify all (x,y) geo-points to high/mid/low-density clusters. QDC uses a standard clustering method on transformed data, unlike many other sophisticated methods that are run on 2D geo-coordinates. It is a quick, efficient, semi-autonomous and big-data tool applicable to static and streaming data. A major parameter in QDC, the number of K clusters to detect, is interpretation-driven, while the other two: the radius for counting NN and the number of NN to sum the distances are of secondary importance and in a minor way impact the outcome. Classification for new points (prediction) is quicker than a typical kNN algorithm by using thresholds of spatial variables. The approach is suitable for tracking human activity as traffic or crowd detection from spatially geo-located mobile data – it finds the high-density points independently of phenomenon intensity and works well with streaming data.**

Index Terms— **spatial density clustering; autonomous algorithm; static and streaming clustering, geospatial analysis, spatial resolution, pattern clustering, clustering algorithm**

I. Introduction

DEVELOPMENTS around DBSCAN (Density-based spatial clustering of applications with noise) and DPC (Density Peak Clustering) channelled density clustering methods towards discovering irregular non-spatially continuous shapes in 2D [1]. Despite many efforts to upgrade those algorithms (overviewed in [2-3]), those solutions fail in the case of big data, still require expert knowledge in setting parameters, their outcomes are highly dependent on parameters and fail in distinguishing diverse density patterns for spatially-continuous data (as discussed in Section II). DBSCAN classifies points as densely located or as noise, similar to core and halo points in DPC. This means that some points, dependently on the parameters of the algorithm, are classified as high-density, while the rest are labelled as non-dense. Even if DBSCAN is great at finding irregular shapes like groups of points separated with blank spaces, it fails in classifying if a given point belongs to one of the high/mid/low-density groups. DPC outcomes are similar to DBSCAN, only richer by indicating a point that is the core of the high-density group [4].

However, density analyses get increasing attention. The inflow of human activity data like traffic, crowd, mobility, infections (e.g. COVID-19), business activity etc. requires detecting geo-located points that are highly concentrated in space [5]. The major difference is that human activity data are much more continuous in space, thus finding irregular patterns is of lower interest. More important is distinguishing between high-, mid- and low-density clusters. Importantly, human mobility or epidemiological events have also three features: first, they may be unpredictable and appear quickly and unexpectedly; secondly, these are usually massive volumes of data; third, their spatial scale and density are unknown – due to phenomena itself or data availability. Therefore there is a need for an algorithm that detects high-density clusters quickly and (semi)autonomously, i.e. without prior information on parameters of spatially dense distribution, even in big data [6]. The expectation from such an algorithm is to classify each point into clusters of different spatial densities.

The methods that deal with human activity data should fulfil some criteria: a) work quickly, b) do not involve deep pre-studies to get parameters (best, when their outcome weakly depends on parameters set), c) can set the high/low-density benchmark autonomously or use reference given by the user, d) are suitable for big data, e) can easily work with stream data, f) have the self-calibrating or at least self-noticing mechanism giving an alert if previously calibrated model stops be valid due to structural change in new data. This kind of (semi)autonomous (self-service) method is desired by users due to low computational cost and high analytical gain. The proposed QDC algorithm fulfils those criteria.

The core point in working with density-clustering algorithms is the definition of density. The most intuitive definitions use aggregates - count the number of units on a given territory and find the count per area unit, i.e. inhabitants per km^2. The wider the territory the less informative the measure as it erases the information on local differences. DBSCAN and DPC make a ring of a given radius around each point and check if the number of units within this circle is higher than the threshold – in high-density clusters circles include more points than the required threshold. However, both parameters, radius and count thresholds are to be set by the user and the outcome is highly dependent on their values. Additionally, one gets only binary classification – high or low-density points benchmarked around a given threshold and radius. This paper defines spatial density differently – it uses two spatial variables: fixed-radius nearest neighbours (NN) and a sum of distances to k nearest neighbours (NN). First, for each observation, it counts the number of points within a given

This paper is a part of a project titled “Modelling and forecasting business location in context of economies of density. Theoretical, methodological and empirical approach using spatial econometrics and spatial machine learning” financed by National Science Center, Poland (Krakow, Poland) [OPUS 21 call, grant number UMO-2021/41/B/HS4/00285].

Katarzyna Kopczewska is associate professor at University of Warsaw, Faculty of Economic Sciences, ul.Długa 44/50, 00-241 Warszawa, Poland (e-mail: kkopczewska@wne.uw.edu.pl)

radius (but it does not compare it to any threshold) – the higher the density, the more points around. Secondly, for each observation it selects (fixed) k nearest neighbours, calculates the distance to each of them and sums them up – the higher the density, the closer to k nearest neighbours (and the lower the total distance).

Setting parameters of algorithms is a separate problem of machine learning. In supervised learning they can be optimized to get the best outcome – often using random or grid search to minimize error. Unsupervised learning balances between validation to guarantee cluster consistency and interpretation of outcomes. Parameters in DBSCAN and DPC are arbitrarily decided, which in consequence generates very diverse outcomes. In this approach, spatial variables are normalized. This automatically benchmarks the clustering process to the general situation over the territory and causes those parameters do not affect the outcome and work as well in a subsample. This limits guessing the proper parameters and also reacts to the volume and spatial range of analysed data. The number of K clusters is to support meaningful interpretation, not only technical optimization.

The paper shows that spatial density can be detected using a K-means algorithm run on spatial variables: fixed-radius nearest neighbours (NN) and a sum of distances to k nearest neighbours[1]. The algorithm classifies points according to autonomously obtained thresholds into K clusters of high/mid/low density. The major advantage of QDC is its suitability for big data due to many developments in K-means, kNN and fixed radius NN and usability in streaming data, which makes it a real machine learning solution. QDC works differently than typical methods: instead of using sophisticated methods on simple data – geo-coordinates [7], it applies a standard algorithm to transformed data. The paper shows it is a very efficient approach.

The remainder of the paper is as follows: Section 2 presents the current state of density clustering algorithms; Section 3 describes the novel solution - QDC; Section 4 presents the prediction mechanism for new data, and Section 5 gives the conclusions.

II. Existing methods of the density detection

Let's start with the illustration of the analytical problem – Fig.1 presents the spatial point distribution of the population over the territory as a full set of 65K points (Fig.1a) and a subset of 5K points (Fig.1b). The goal of the analysis is to divide spatially located points into K density clusters. Even if both patterns are similar, operating on a subsample diminishes the number of points in the neighbourhood compared with a full sample.

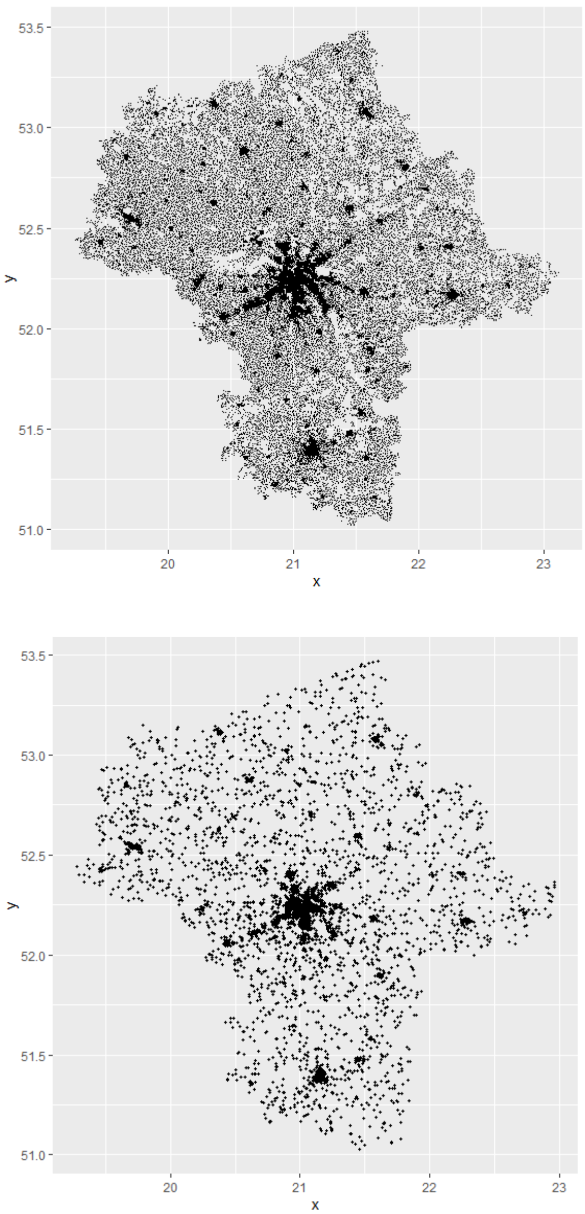


Fig. 1. Point pattern of human settlement (xy points as longitude/latitude): a) full sample of ca. 65K points, b) subset of randomly sampled 5K points

The most popular method for density-clustering, DBSCAN (Density-based spatial clustering of applications with noise) classifies each point as core, border or noise [8]. It generates a circle of radius *eps* around each point, counts the number of units within this circle and compares with some threshold *minPts*. Core points have at least the threshold number of points within the radius *eps*. Border points do not have enough points around, but belong to the radius of the core point. Noise points are all other points (Fig.2). In high-density clusters circles include more points than the required threshold. Beyond many advantages of this algorithm, using DBSCAN may be problematic. First, it is not easy to guess properly the radius *eps* and threshold *minPts* which distinguish between high and low density. Secondly, one cannot use the same set

[1] There is coincidence of two symbols 'k' in analysis - they are distinguished as K for K-means and k for k nearest neighbours.

of parameters in a full dataset and a subsample. Subsamples include automatically fewer points in each radius, thus by setting the same *minPts* and *eps* one looks for a very different spatial structure.

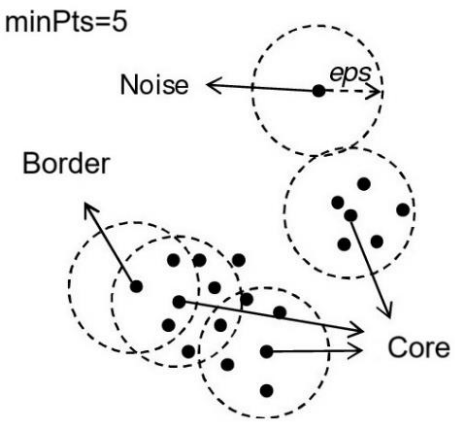
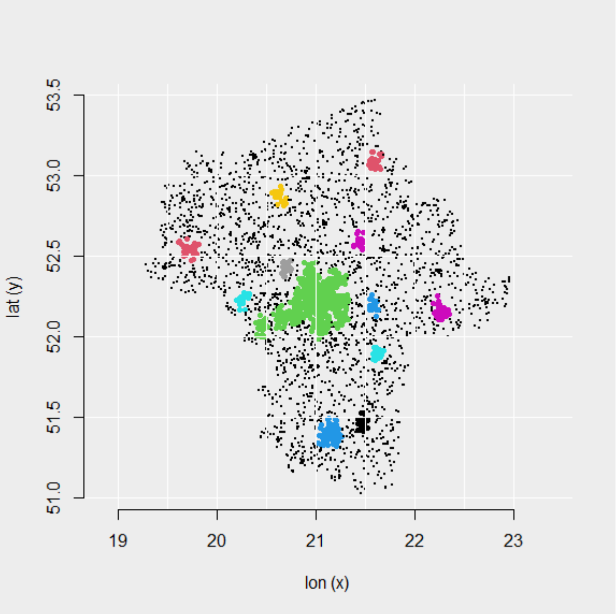


Fig. 2. DBSCAN algorithm

DBSCAN clustering (Fig.3) can detect well high-density clusters without specifying how many of them are expected. However, the results are highly dependent on parameters defined by the user [9]. In Fig.3a DBSCAN with radius *eps*=0.05 and threshold *minPts*=25 detected 13 clusters, while the ratio of noise is 47%. In Fig.3b DBSCAN with radius *eps*=0.15 and threshold *minPts*=50 detected 9 clusters, while the ratio of noise is 9%. As visible from Fig.3, guessing proper parameters is a challenging task and together limited scalability makes it an unattractive solution for this problem. These findings were confirmed with overview studies on clustering [6] and underlined the low quality of DBSCAN clusters and low efficiency in big data.



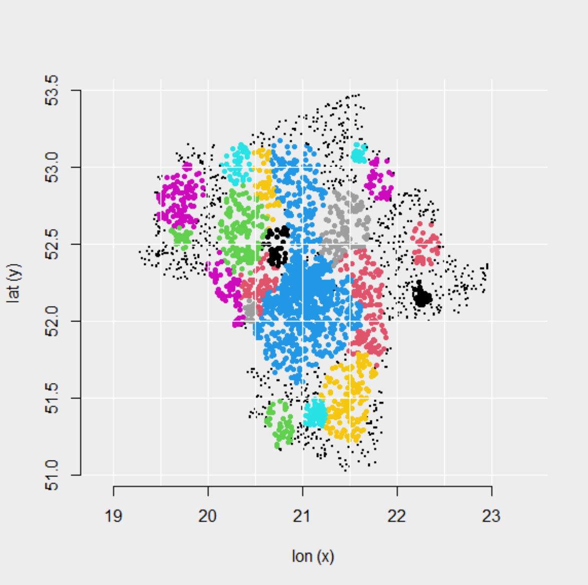


Fig. 3. DBSCAN clustering of subsample a) radius *eps*=0.05 and threshold *minPts*=25; b) radius *eps*=0.15 and threshold *minPts*=50

The spatial clustering literature usually claims that the K-means algorithm detects only irregular tile-like shapes [10] – that is true when input data are geo-coordinates (x,y). Fig.4 presents this kind of clustering. The outcome is useful for deriving catchment areas e.g. schools, postal offices or sales representatives, but also e.g. stratified sampling, spatial cross-validation. This clustering does not refer to density explicitly – clusters include points of high and low density. However, point density is referred to in the optimization of the location of cluster centroids due to the mass of data in a given location.

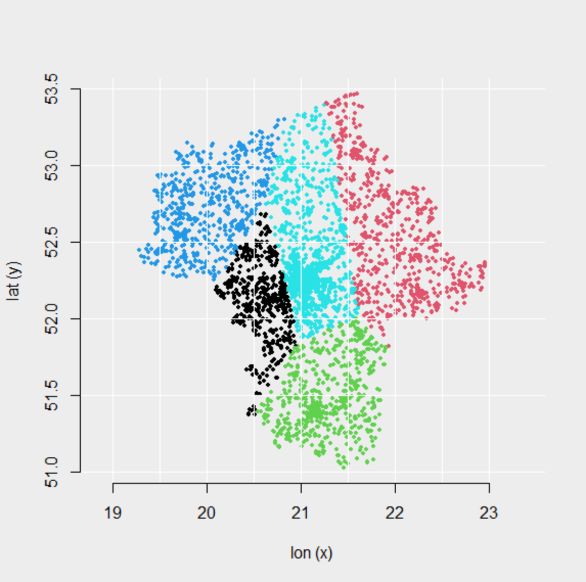


Fig. 4. K-means clustering of geo-coordinates (K=5)

This shows that none of these approaches is suitable to solve the problem of density clustering for spatially continuous data as in Fig.1.

III. Quick Density Clustering (QDC): K-means density clustering using spatial variables

Density clustering methods available in the literature were mostly directed towards the detection of irregular patterns [1], while human mobility analyses were abandoned or focused on other than density aspects [5, 11-12]. Human activity data, such as traffic, crowd movement, mobility patterns, infections (like COVID-19), or business activities are continuous in space and the focus is primarily on distinguishing between high, mid, and low-density clusters. It is essential to note that human mobility or epidemiological events possess three distinct characteristics that should be addressed by quantitative solutions. Firstly, they can be unpredictable and occur unexpectedly and rapidly, therefore methods should be flexible and quick in implementation. Secondly, they usually are big data having thousands or millions of observations, therefore methods should work well on representative subsamples. Third, their spatial and density scale vary due to available data or the process itself, therefore methods should be robust to the spatial scale.

Consequently, there is a pressing need for an algorithm that can swiftly and (semi)autonomously detect high-density clusters, i.e., without requiring prior information on parameters of spatially dense distribution and work independently of subsample size. The main objective of such an algorithm is to classify each data point into clusters based on different levels of spatial density. Policy actions can be therefore targeted to the specific territory that exhibits a higher or lower density of human activity than the "standard one". As shown in the previous section, existing algorithms such as DBSCAN or DPC are insufficient in density classification as they distinguish density in binary mode: high- and low-density, they are not autonomous / self-service, and the decision on values of parameters is crucial for the outcome and they poorly deal with big data and/or subsets. Even if there exist autonomous clustering algorithms as the ADP algorithm [6], they are not suitable for spatial density detection. What is needed, is a mechanism which (semi)autonomously determines what the "standard density" is (on a given territory in a given time) and then classifies observations due to this benchmark. It should be flexible to allow for a specific number of clusters.

A. Spatial variables

The innovation of this paper lies in redefining the approach to measure the density around the point. Exiting algorithms (e.g. DBSCAN, DPC) struggle to measure it with a single criterion as a dummy if the number of points in the specified radius exceeds the assumed threshold. QDC method uses two criteria (two spatial variables): fixed-radius nearest neighbours (NN) and a sum of distances to k nearest neighbours (kNN) (Fig.5). Both variables need setting hyperparameters, which are of secondary importance due to the normalization procedure and should fit the context of social data.

The first spatial variable, a sum of distances to *k* nearest neighbours (kNN), selects kNN for each point, calculates the distance to each of them and sums them up. This is natural that the higher the point density, the closer to kNN and the lower the total distance. It can take values from almost 0 (in the case of a very close neighbourhood) up to an undefined value, depending on the spatial range of data and measurement scale. The number of k for social data should be around 15-30.

The second spatial variable, fixed-radius nearest neighbours (*frNN*), counts the number of points within the given radius for each observation. However, it does not compare it to any threshold (as DBSCAN) but stores the count of observations – the higher the density, the more points around. It can take values from 0 to *n-1*, where *n* is the number of observations in the dataset. The radius, fixed for all *n* points analysed to assure comparability, should be around 5-15 km (0.05°-0.15°). There were many developments for quick neighbour search (overviewed in [13]), often implemented in clustering algorithms, e.g. grid search for density peak clustering [14], which makes this procedure computationally efficient.

Fig.5 evidences the idea of spatial variables: for triangle point (▲) which has a low-density location, the number of neighbours in a fixed radius is low, while the sum of distances to k=5 nearest neighbours is high; for square point (▪) which has a high-density location, the number of neighbours in a fixed radius is high, while the sum of distances to k=5 nearest neighbours is low.

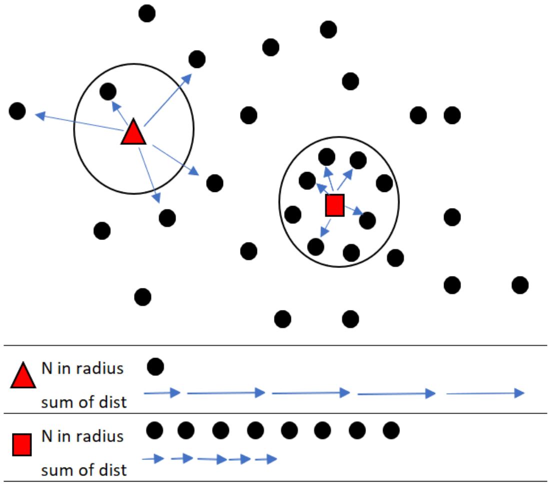


Fig. 5. Concept of spatial variables for a given spatial distribution of points

B. Normalisation

An important element of the algorithm is the normalisation of spatial variables according to the formula:

$$z_i = \frac{x_i - \bar{x}}{s} \tag{1}$$

where x_i and z_i are the values of the variable before and after normalization, $\bar{x}$ is the average value of a raw variable (before normalisation) and s is its standard deviation. After normalization values become relative, with the average of z_i values equal to 0 and the standard deviation equal to 1. It automatically determines typical (around the average) and untypical (far from the average) values. This approach makes the method much more objective than those that are based on arbitrarily set absolute parameters (e.g. DBSCAN, DPC). What is more, normalization can be executed with different parameters ($\bar{x}$ and s). They can be taken from the sample – especially when the density clustering is to detect the current situation in the data. However, one can also use "typical" (if known) parameters – especially in a situation of rapid changes. This will enable quick detection of changes above / below the typical density level. Third, normalization allows for the use of subsets of different sizes. Making a subsample (e.g. using 10% of data, see Fig.1) changes the parameters of distributions, especially the number of points in the neighbourhood. Therefore, methods that are based on nominal thresholds (e.g. DBSCAN) give poorly comparable results between datasets of different sizes. Normalisation in QDC eliminates this issue as spatial variables are centred around the average value and become relative.

An important feature of this concept is the relative independence of normalized spatial variables from parameters such as the number of *k* nearest neighbours or the size of the radius. Fig.6 presents statistical distributions of normalised (scaled) spatial variables for data from Fig.1b in two settings: counting neighbours in radius *r1*=0.05 (5 km) and *r2*=0.15 (15 km) and counting the sum of distances for *k1*=10 and *k2*=30 nearest neighbours. Such hyperparameters are appropriate for social interactions that are by nature narrow over space. Radius is expressed in geographical degrees (1°≈111 km) as the data are geo-projected locations. The densities of the variables mostly overlap – the statistical distribution of the sum of distances to *k1* or *k2* are similar; the same is true about the statistical distribution of the number of neighbours in radius *r1* or *r2*. This shows that parameters of spatial variables are of minor importance for the result as its outcome is weakly dependent on the parameters set. The method does not involve deep pre-studies to get parameters, as almost any "reasonable" parameters allow getting proper results. Therefore it can be considered semi-autonomous. The second conclusion from that figure is that information included in each spatial variable is different, what justifies using them both in the algorithm.

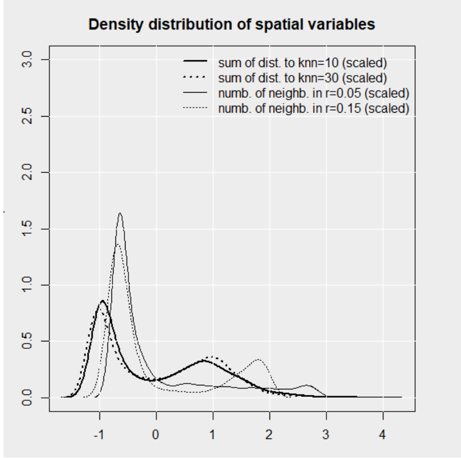


Fig. 6. Statistical distributions of normalised (scaled) spatial variables with different parameters (for data from Fig.1b)

C. QDC construction

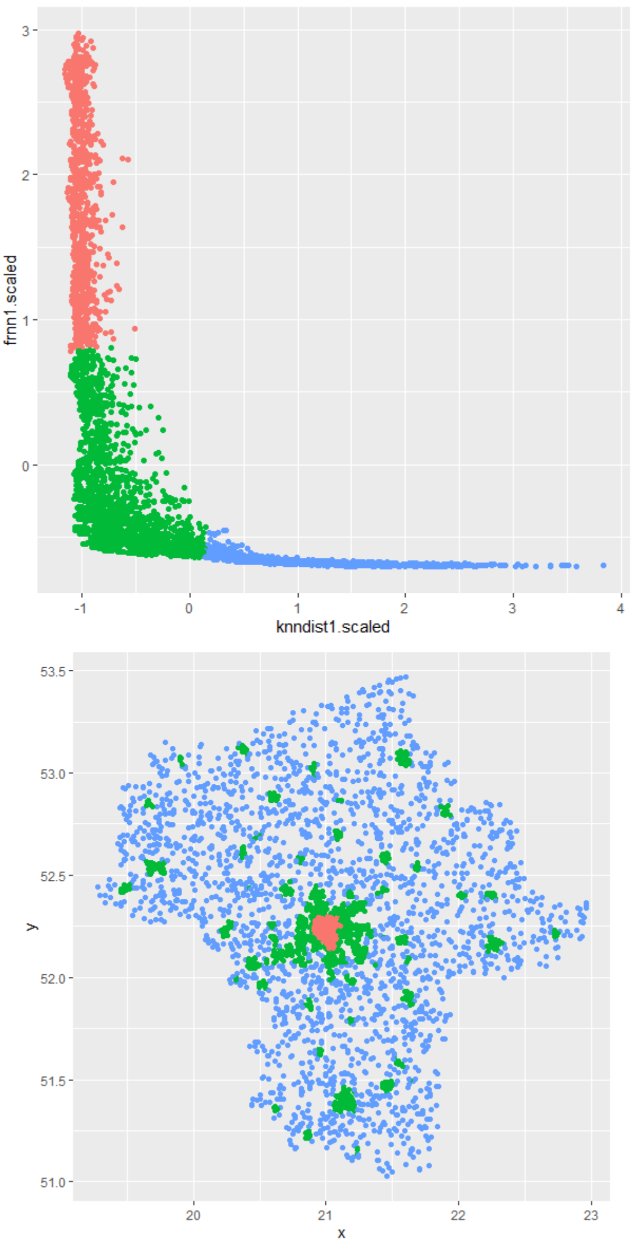
The Quick Density Clustering (QDC) method is a K-means clustering of two normalised spatial variables. Each of the K-means clusters represents a separate density group. The algorithm requires specifying three hyperparameters: two for spatial variables (already discussed number of kNN and radius *r* for *frNN*), and a number of K clusters. K should be interpretation-driven, e.g. two clusters for binary division (high/low density), three clusters for benchmarked division (low/standard/high density), and five clusters to follow a Likert-like scale (very low/low/standard/high/very high density), but can also be cross-checked with silhouette statistics. The algorithm for QDC is as follows.

Algorithm Quick Density Clustering (QDC)

```
CLUSTERING
    k=hyper-parameter, e.g.30
    K=hyper-parameter, e.g. 3
    r=hyper-parameter, e.g. 0.15
    spat.var1 ← ∑dist(knn=k)
    spat.var2 ← frnn(r)
    spat.var1.s ← (spat.var1-mean(spat.var1))/sd(spat.var1)
    spat.var2.s ← (spat.var2-mean(spat.var2))/sd(spat.var2)
    data ← (spat.var1.s, spat.var1.s)
    kmeans(data, K)

CLASSIFYING
    t1←max(min(spat.var1|clust1), …., min(spat.var1|clustK))
    t1←max(min(spat.var2|clust1), …., min(spat.var2|clustK))
    low-density ←spat.var1>t1
    high-density← spat.var2>t2
```

Developments to K-means in terms of its speed and scalability [15] make it an attractive algorithm. Fig.7 illustrates the QDC for population data (5'000 obs.) using *kNN*=10 and *r*=0.05. Fig.7a shows the relation between both spatial variables - the sum of distances to kNN on the x-axis and the number of neighbours in fixed radius on the y-axis, while colour indicates the division into three clusters. Red points represent high-density clusters – they have many points in radius (y) and low total distance to kNN (x). Blue points, oppositely, represent low-density clusters - they have few points in radius (y) and a high total distance to kNN (x). Green points represent a mid-density cluster – both spatial variables are on its average level. The relation between spatial variables is not linear, therefore applying e.g. Pearson correlation is useless. Fig.7b illustrates the geographical distribution of high/mid/low-density clusters. It is highly coherent with "optical inspection" from Fig.1. In the analysed case, the red cluster is a metropolitan area, the green cluster are smaller cities, and the blue cluster are peripheral/rural areas. Fig.7c justifies interpretation-driven division into three clusters – as a typical cluster validation measure, silhouette, is the highest. However, this tool has only a supportive role. The analyst can decide about the desired number of clusters if needed, while silhouette should be considered as general guidance. Fig.7d illustrates the feature importance of the K-means algorithm. It shows that both spatial variables matter for the miss-classification rate: ca. 0.43 in the case of the sum of distances and ca. 0.30 in the case of fixed-radius neighbours, so both should be included in K-means clustering.



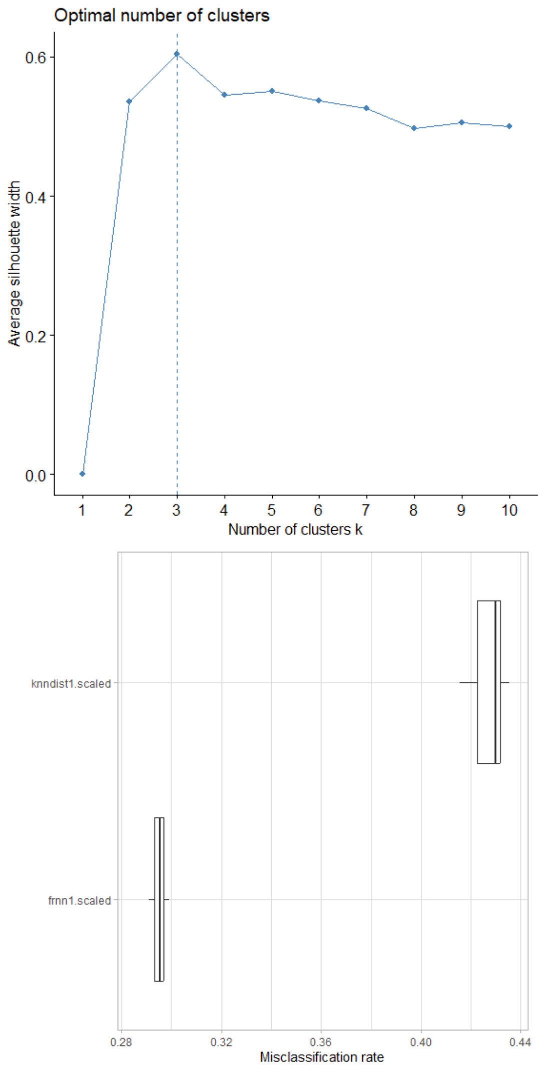


Fig. 7. Quick Density Clustering algorithm for kNN=10 and radius=0.05: a) mutual relation between normalized spatial variables; b) geo-location of density clusters of the population; c) silhouette statistic as a cross-check of the optimal number of clusters; d) feature importance for K-means clustering

The core issue of QDC is its relative independence from the parameters of spatial variables. Fig.8 presents the alternative division into clusters by using different parameters: kNN=30 and radius=0.15. What is visible, the mutual relation between spatial variables is more diversified (Fig.8a) – this is a natural phenomenon as increasing the spatial range of a neighbourhood increases diversity. However, the optimal number of clusters was also three and feature importance reached similar values as in the previous case. The division into density clusters is very similar to the previous case – the Rand Index comparing both partitioning is 0.92. This low sensitivity to values of parameters makes QDC an intuitive and easy-to-use robust algorithm regardless of initial parameter choice.

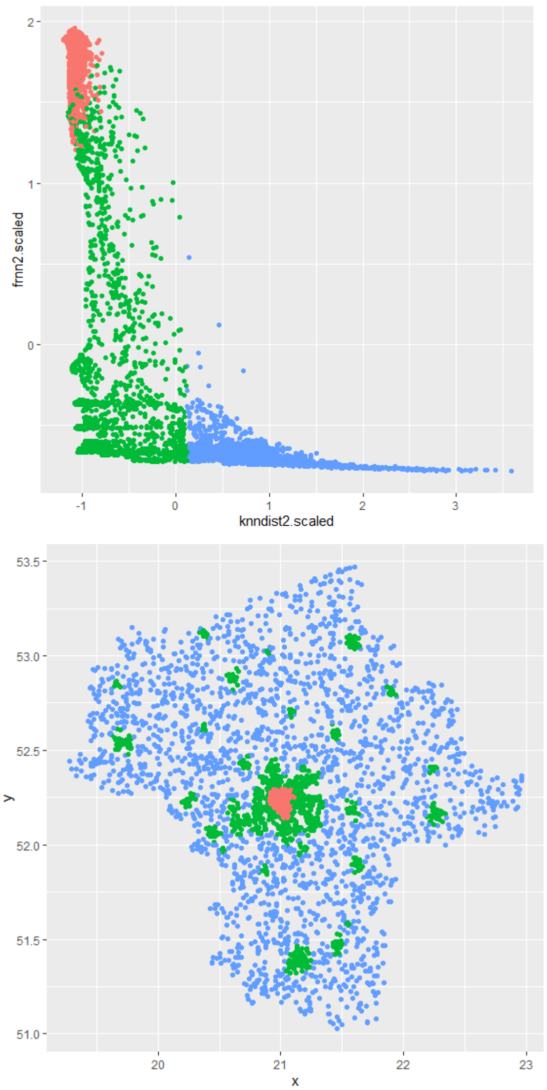


Fig. 8. Quick Density Clustering algorithm for kNN=30 and radius=0.15: a) mutual relation between normalized spatial variables; b) geo-location of density clusters of population

IV. Machine learning predictions for new data

A crucial issue of machine learning models is their ability to run cluster classification quickly and precisely for new points. There are two issues discussed: a) mechanism of classification of new points to clusters; and b) automatic flagging of model validity in case of streaming data. Those issues are interconnected and should be considered jointly.

QDC algorithm is based on two normalised spatial variables. New observation – point to be classified to one of the density clusters – usually appears as (x,y) geo-location. QDC requires as input the information on the neighbourhood – values of spatial variables normalised with the same average and standard deviation values as the baseline model. Those values are compared with the two-dimensional characteristic of clusters (as in Fig.8a) – classification to one of the clusters is straightforward, according to the threshold values. The only exception are border points between cluster regimes – here the classification might be random/fuzzy. In the case of division into three clusters (as in Fig.7,8), there are two thresholds t, calculated as the maximum of within-cluster minima:

$$t_{knndist} = \max\left(\min(knn.dist.s|cluster = 1),\ \min(knn.dist.s|cluster = 2),\ \min(knn.dist.s|cluster = 3)\right) \quad (2)$$

$$t_{frnn} = \max\left(\min(frnn.s|cluster = 1),\ \min(frnn.s|cluster = 2),\ \min(frnn.s|cluster = 3)\right) \quad (3)$$

where *knn.dist.s* is a normalised spatial variable being a sum of distances to kNN neighbours, *frnn.s* is a normalized spatial variable being a number of neighbours in a fixed radius, and *cluster* is a division into K-means clusters. Point belongs to the low-density cluster when its *knn.dist.s* is higher than $t_{knndist}$, which is understood that the nearest *k* points are located relatively far. On the other hand, a point belongs to a high-density cluster when its *frnn.s* is higher than t_{frnn}, what is understood that the number of points in a fixed radius is relatively high. Other points belong to medium-density cluster.

There is also a possibility of using a typical for K-means approach – assigning a new point to the cluster of its nearest neighbour. However, this is a much more computationally demanding procedure, as each new point must be compared with all existing points to find the nearest neighbour. Multiple search procedures are inefficient in big data and the clusters' thresholds approach is much simpler and quicker.

A huge challenge of all quantitative methods is their validity in case of new data. Once the model is calibrated for static data and the thresholds of normalised spatial variables of density groups are set, one needs an automated solution that alerts when data structurally changes. Let's imagine two scenarios for the population. In the first one, new persons are coming randomly over space. Each new observation changes local density and the number of neighbours – however, as long it is random, it does not impact the spatial distribution significantly. In the second scenario, new persons are accumulating in one place (like a traffic jam, holiday destination, big concert etc.). Then it changes the spatial pattern and prior parameters – the average and standard deviation of spatial variables may be not valid anymore. Depending on the type of baseline spatial distributions, each new point may change its characteristics (conditional density of point pattern, see [16]), and therefore automated streaming monitoring of parameters is of high importance.

QDC can be robust to structural changes by using streaming parameters. In the literature one can find a streaming data approach to detect outliers in new data in ML research [18-22]. According to Welford's online algorithm [17] one can easily derive the value of the mean and standard deviation of the increased dataset, by updating the existing parameters with new data:

$$\bar{x}_n = \frac{(n-1)\bar{x}_{n-1} + x_n}{n} = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n} \quad (4)$$

$$s_n^2 = \frac{(n-2)}{(n-1)} s_{n-1}^2 + \frac{(x_n - \bar{x}_{n-1})^2}{n} \text{ for n>1} \quad (5)$$

where $\bar{x}_{n-1}$ is "old" average, $\bar{x}_n$ is "new" average, x_n is a value of "new" observation, n is the "new" number of observations, s_n^2 and s_{n-1}^2 are "new" and "old" unbiased sample variance. With each new point appearing for classification, one can derive the new normalisation parameters of spatial variables (mean and standard deviation). They can be easily tested if they are still equal to the baseline parameters – using a two-sample t-test for means and an F-test for variances. As long as both parameters are statistically the same, thresholds of density clusters can be used for assigning new points into density clusters. In case the parameters decalibrate significantly, one needs to re-run density clustering. This automatic alert makes QDC a robust model to be applied responsibly.

V. Conclusion

This paper introduces the Quick Density Clustering (QDC) algorithm – a new approach to grouping spatially located points into density clusters. It is based on two normalized spatial variables: fixed-radius nearest neighbours (NN) and a sum of distances to k nearest neighbours which are clustered using the K-means approach.

This method is highly useful in dealing with human activity data and it fulfils the criteria of good index. First, it works quickly. Secondly, it does not involve deep pre-studies to get hyperparameters - a number of clusters is interpretation-driven and normalization of both spatial variables makes them sample-size independent (they become relative) and causes they have highly similar statistical distributions (so their parameters do not affect the outcome significantly). Third, it sets the high/low-density benchmark autonomously through the clustering process, which limits the need for user intervention. Fourth, it deals well with big data, while at the same time, it yields equivalent results for a subsample. Fifth, it can easily work with stream data due to the low-cost mechanism of classification of points to density groups. Sixth, it has a self-noticing mechanism that gives an alert if the previously calibrated model stops being valid due to structural change in new data. This solution was implemented in R software at Code Ocean [23].

REFERENCES

[01] Khan, K., Rehman, S. U., Aziz, K., Fong, S., & Sarasvady, S. (2014, February). DBSCAN: Past, present and future. In The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014) (pp. 232-238). IEEE.

[02] Ahmed, K. N., & Razak, T. A. (2016). An overview of various improvements of DBSCAN algorithm in clustering spatial databases. International Journal of Advanced Research in Computer and Communication Engineering, 5(2), 360-363.

[03] Bushra, A. A., & Yi, G. (2021). Comparative analysis review of pioneering DBSCAN and successive density-based clustering algorithms. IEEE Access, 9, 87918-87935.

[04] Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. Science, 344(6191), 1492-1496.

[05] Zhao, K., Tarkoma, S., Liu, S., & Vo, H. (2016, December). Urban human mobility data mining: An overview. In 2016 IEEE International Conference on Big Data (Big Data) (pp. 1911-1920). IEEE.

[06] Gu, X., Angelov, P. P., & Príncipe, J. C. (2018). A method for autonomous data partitioning. Information sciences, 460, 65-82.

[07] Xue, W., Yang, R. L., Hong, X. Y., Zhao, N., & Ren, S. G. (2017, May). A novel k-Means based on spatial density similarity measurement. In 2017 29th Chinese Control And Decision Conference (CCDC) (pp. 7782-7784). IEEE.

[08] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).

[09] Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. D. F., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. PloS one, 14(1), e0210236.

[10] Mumtaz, K., & Duraiswamy, K. (2010). A novel density based improved k-means clustering algorithm–Dbkmeans. International Journal on computer science and Engineering, 2(2), 213-218.

[11] Hu, T., Wang, S., She, B., Zhang, M., Huang, X., Cui, Y., ... & Li, Z. (2021). Human mobility data in the COVID-19 pandemic: characteristics, applications, and challenges. International Journal of Digital Earth, 14(9), 1126-1147.

[12] Alessandretti, L. (2022). What human mobility data tell us about COVID-19 spread. Nature Reviews Physics, 4(1), 12-13.

[13] Ukey, N., Yang, Z., Li, B., Zhang, G., Hu, Y., & Zhang, W. (2023). Survey on exact knn queries over high-dimensional data space. Sensors, 23(2), 629.

[14] Luan, S., Lu, C., Bai, L., & Wang, H. (2019, November). Density peaks spatial clustering by grid neighborhood search. In 2019 2nd International Conference on Safety Produce Informatization (IICSPI) (pp. 331-336). IEEE.

[15] Bai, L., Cheng, X., Liang, J., Shen, H., & Guo, Y. (2017). Fast density clustering strategies based on the k-means algorithm. Pattern Recognition, 71, 375-386.

[16] Baddeley, A., Rubak, E., & Turner, R. (2015). Spatial point patterns: methodology and applications with R. CRC press.

[17] Welford, B. P. (1962). Note on a method for calculating corrected sums of squares and products. Technometrics, 4(3), 419-420.

[18] Heigl, M., Weigelt, E., Fiala, D., & Schramm, M. (2021a). Unsupervised feature selection for outlier detection on streaming data to enhance network security. Applied Sciences, 11(24), 12073.

[19] Krleža, D., Vrdoljak, B., & Brčić, M. (2021). Statistical hierarchical clustering algorithm for outlier detection in evolving data streams. Machine Learning, 110, 139-184.

[20] Carnein, M., & Trautmann, H. (2019). Customer segmentation based on transactional data using stream clustering. In Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, April 14-17, 2019, Proceedings, Part I 23 (pp. 280-292). Springer International Publishing.

[21] Heigl, M., Weigelt, E., Urmann, A., Fiala, D., & Schramm, M. (2021b). Exploiting the outcome of outlier detection for novel attack pattern recognition on streaming data. Electronics, 10(17), 2160.

[22] Oprea, S. V., Bâra, A., Diaconita, V., Preotescu, D., & Tör, O. B. (2019, October). Big Data solutions-data ingestion and stream processing for demand response management. In 2019 23rd International Conference on System Theory, Control and Computing (ICSTCC) (pp. 697-702). IEEE.

[23] Kopczewska K., 2023, "Quick Density Clustering (QDC) - R code for classifying geo-located points into a few spatial density groups" Code Ocean, doi: 10.24433/CO.8391813.v1 https://codeocean.com/capsule/9163124/tree

Katarzyna Kopczewska received a Ph.D. degree in economics from the University of Warsaw, Poland in 2007 and a habilitation in economics with specialization in computer science and econometrics in 2017. Since 2019 she is employed as an associate professor at the Faculty of Economic Sciences, University of Warsaw, Poland. Her research focuses on spatial quantitative methods, including spatial statistics, spatial econometrics and spatial machine learning. Prof. Kopczewska is a member of European Regional Science Association.