Analytica Chimica Acta xxx (xxxx) xxx

Contents lists available at ScienceDirect

Analytica Chimica Acta

journal homepage: www.elsevier.com/locate/aca

A graph-based clustering method with special focus on hyperspectral imaging

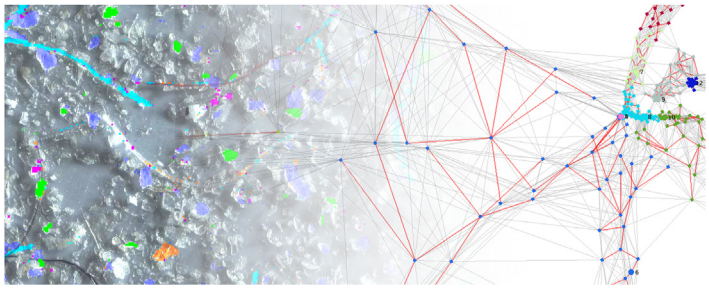
Benedikt Hufnagl*, Hans Lohninger

Institute of Chemical Technologies and Analytics, Vienna University of Technology, Austria

HIGHLIGHTS

- Sensitive to small variations in data density.
- Complementary clustering approach to K-Means or HCA.
- Applicable to complex problems where a large number of clusters is expected.

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 30 May 2019
Received in revised form
9 September 2019
Accepted 7 October 2019
Available online xxx

Keywords:
Graph-based clustering
Exploratory analysis
Hyperspectral imaging
Density estimation
Nearest neighbors
Digraph

ABSTRACT

A common trait of the more established clustering algorithms such as *K*-Means and HCA is their tendency to focus mainly on the bulk features of the data which causes minor features to be attributed to larger clusters. For hyperspectral imaging this has the consequence that substances which are covered by only a few pixels tend to be overlooked and thus cannot be separated. If small lateral features such as particles are the research objective this might be the reason why cluster analysis fails. Therefore we propose a novel graph-based clustering algorithm dubbed GBCC which is sensitive to small variations in data density and scales its clusters according to the underlying structures. The analysis of the proposed method covers a comparison to *K*-Means, DBSCAN and *K*NSC using a 2D artificial dataset. Further the method is evaluated on a multisensor image of atmospheric particulate matter composed of Raman and EDX data as well as an FTIR image of microplastics.

1. Introduction

Graph-based clustering comprises a family of unsupervised classification algorithms that are designed to cluster the vertices and edges of a graph instead of objects in a feature space. A typical application field of these methods is the Data Mining of online social networks or the Web graph [1]. Usually the vertices of social graphs are only sparsely connected. For clustering this has the consequence that the similarity between objects (or vertices) can no longer be established by simply measuring the distance between two data points as they do not necessarily have to be connected by an edge. Instead paths that lead over other vertices have to be found to determine their relation.

In the context of matrix-based spectroscopic data, graphs can be used to derive an abstraction of the dataset that reflects only the

* Corresponding author. Institute of Chemical Technologies and Analytics, Vienna University of Technology, Getreidemarkt 9, 1060, Wien, Austria.
E-mail addresses: benedikt.hufnagl@tuwien.ac.at (B. Hufnagl), johann.lohninger@tuwien.ac.at (H. Lohninger).

https://doi.org/10.1016/j.aca.2019.10.071
0003-2670/

local neighborhood relations between objects. This has the advantage that less weight is given to bulk features of the data whereas smaller structures gain more importance. Therefore graph-based clustering algorithms are more flexible with regard to non-hyperspherical shapes and some of them can cope with clusters that differ greatly in terms of density and extent spatially speaking.

In this paper we propose a novel graph-based algorithm for the exploratory cluster analysis of hyperspectral images dubbed *Graph-Based Competitive Clustering* (GBCC). The rest of the paper is organized as follows: Section 2 will give a short overview of graph-based clustering algorithms. In section 3 we will discuss common separability problems that arise in clustering and describe the fundamental concept behind GBCC. Algorithmic aspects are covered in section 4 followed by an experimental assessment in section 5. The paper concludes with section 6 which offers some final remarks about when GBCC might be applied.

2. Fundamentals and related work

Let $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d})$ denote an object of d measurements and $\mathscr{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ a dataset of n objects. Then a graph $\mathscr{G} = (V, E)$ can be derived from $\mathscr{X}$ such that the relations between objects are represented by the presence or absence of connecting edges. The objects $\mathbf{x}_i, \mathbf{x}_j \in \mathscr{X}$ thus become vertices $v_i, v_j \in V$. The edge that connects vertices v_i and v_j is denoted as $e_{ij} \in E$ with an associated weight $w_{ij} \geq 0$ describing the similarity between the two vertices. Using matrix notation a graph can be described by assembling the edge weights as the weighted adjacency matrix $\mathbf{W} = (w_{ij})$. If v_i and v_j are not connected then the corresponding entry has the value $w_{ij} = 0$.

Some algorithms require a certain type of graph whereas others leave it to the user to decide which graph representation is best suited for the given problem. A straightforward technique for clustering graphs is the removal of 'inconsistent edges'. Here one tries to create disconnected subgraphs by removing edges with weights that differ significantly from others. Among many other graph types the Minimum Spanning Tree (MST) and its variants [2,3] are often used for this approach. Zahn [4] derived some basic criteria how such edges can be detected.

A global approach is to determine the average edge length μ and variance σ of the graph and remove all edges with weights exceeding, e.g., $\mu + 3\sigma$. A different approach would be to limit the edges used for the determination of μ and σ to, e.g., the 2nd -order neighbors of the two vertices which are connected by the edge under consideration.

Zhong et al. [5] argued that MST-based clustering uses too little information to determine inconsistent edges referring to the sparsity of such graphs. In order to make the clustering more robust they proposed to enhance the connectivity in the MST by a additional MST that is constructed from those edges not included in the first one. Whether a cut that produces two disconnected subgraphs is valid is then determined by a score that is based on the intersecting sets of the removed edges and the 1st - and 2nd -round MST respectively.

Liu et al. [6] proposed using global and local edge constraints to remove inconsistent edges from the Delaunay triangulation. The remaining graph is then clustered by an algorithm based on spatial reachability criteria which are in some sense comparable to those used in DBSCAN [7].

Another approach to graph-based clustering is spectral clustering. It can be defined as a family of algorithms that use standard clustering methods such as *K*-Means to cluster the eigenvectors of the Laplacian matrix $\mathbf{L}$. It can be derived from the symmetric weighted adjacency matrix $\mathbf{W}$ and the diagonal degree matrix $\mathbf{D}_{ii} = \sum_{j=1}^{n} w_{ij}$ through $\mathbf{L} := \mathbf{D} - \mathbf{W}$. By computing the first K eigenvectors of the generalized linear eigenproblem $\mathbf{Lu} = \lambda\mathbf{Du}$ we can construct the matrix $\mathbf{U} \in \mathbb{R}^{n \times K}$ which contains the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_K$ as columns. The final clustering step is performed on the vectors $\mathbf{y}_i \in \mathbb{R}^K$ which are the vectors corresponding to the i^{th} row of $\mathbf{U}$. The resulting labels are then mapped onto the original vertices or objects respectively in order to come to the result.

The above eigenproblem was proposed by Shi and Malik [8] as a bipartitioning algorithm and was later reused for *K*-way partitioning in *K*NSC [9] which clusters the eigenvectors following the above description. For an introductory tutorial to spectral clustering see Von Luxburg [10] and Nascimento and De Carvalho [9] for a survey on this research field. Recently proposed methods that are based on spectral clustering are the ensemble method by Zhong et al. [11] and a robust path-based method by Chang and Yeung [12].

3. Motivation and concept

Clustering is not a domain-independent discipline. In any field where clustering is applied there are different goals which one wants to achieve. As a result there are different problem definitions for cluster separation which may or may not play a role in the respective field or the dataset under consideration. Zahn [4] and Handl and Knowles [13] give a rough classification of common problems and separability criteria which are summarized in Fig. 1 (In this illustration we assume that the manually assigned cluster labels constitute a meaningful separation).

The most trivial problem is the case of *well-separated* clusters. Here any pair of points within one cluster is closer together than a pair of points taken from two different clusters. A *distance-separated* problem arises if two clusters are so close together that a separation can only be achieved because the distance between the two closest points is still significantly larger than the distance to their respective neighbors within each cluster. This would be the case for cluster pairs 1–5, 2–4 and 4–5. Contrary to such cluster pairs the pair 1–2 is defined as a *touching* problem because the two clusters are connected by a 'neck'. A more advanced problem is the cluster pair 2–3 which is referred to as *density-separated*. Here a separation might be achieved by measuring the jump in data density if one moves from cluster 2 to cluster 3. Cluster 5 is called a *connected* cluster because the relationship between the objects can be established by connecting all objects that are closer than a certain threshold. In some applications noise rejection capabilities also play an important role. A noise cluster is usually formed by

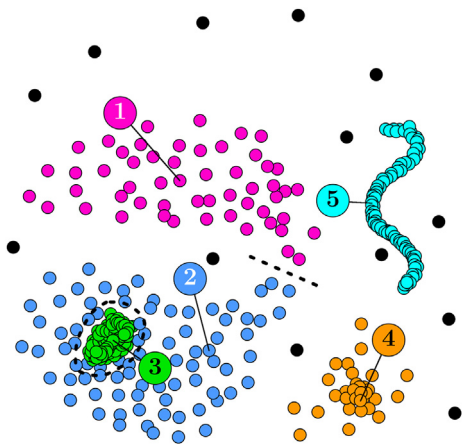


Fig. 1. Common clustering problems such as distance-separated (pairs 1–5, 2–4, 4–5), density-separated (pair 2–3), touching (pair 1–2), and connected (cluster 5) clusters.

Please cite this article as: B. Hufnagl, H. Lohninger, A graph-based clustering method with special focus on hyperspectral imaging, Analytica Chimica Acta, https://doi.org/10.1016/j.aca.2019.10.071

those objects which have not been assigned to any cluster. In Fig. 1 the noise cluster is indicated by the black dots.

The above separability problems are antithetic and therefore very difficult to address conjointly. As each individual problem may or may not play a role in certain application fields of clustering some algorithms are specifically designed for the associated tasks. A common approach to assess and compare the characteristics of a clustering algorithm with respect to others is to apply it to an artificial 2D dataset such as the one given. In the hopes that the obtained results also translate to higher dimensions we can then make assumptions about which algorithm might be better suited to solve our task.

Seen from the perspective of hyperspectral imaging we believe that an important clustering problem is missing here. The data structure of hyperspectral images (HSIs) differs in many aspects from the cluster structures given in Fig. 1. Spectra of the same chemical compound tend to form lobe-like clusters which protrude from the origin and extend into the d-dimensional feature space. There can be different reasons why these lobes occur such as varying concentration or thickness of the constituent. Another source for varying intensity values can be the limited lateral resolution which causes pixels to contain information from both the background and the constituent.

Depending on the measured sample, mixture effects between the chemical compounds may occur that connect these lobes through veils of spectra which can be viewed as linear combinations of the pure components. As a consequence the assumption that there exist clear cluster boundaries between the chemical compounds may not be justified. Further marginal differences in the spectra may be indicative of different species which means that a clustering algorithm has to be very sensitive to local variations in density.

As will be shown later the lobe-like cluster structures are best approached by using a more suitable distance measure such as the *cosine* similarity. However the differences in spatial extent and the above mentioned mixture effects remain. Fig. 2 illustrates this case as a two-dimensional artificial dataset. At first glance there are two clusters at positions 1 and 2 which show no clear boundary between them. If one takes a closer look at position 2 the cluster is yet another combination of two minor clusters. One might argue that this constitutes a density-separable problem but on the other hand this would require some notion of a boundary where the density changes significantly. One approach to solve this problem is to use the density gradient between the three clusters in order to separate them. The authors will denote clustering problems which are separable using data gradients as *gradient-separable*.

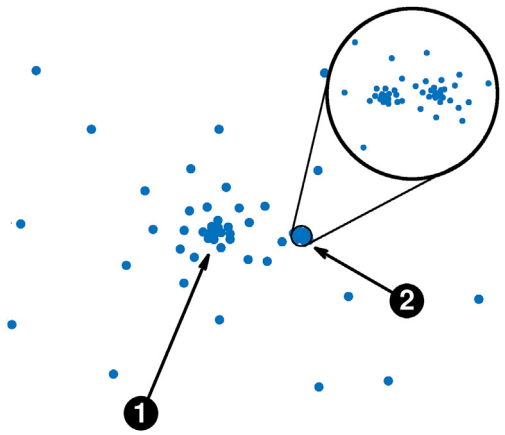


Fig. 2. A gradient-separable clustering problem combining three clusters. This kind of clustering problem can be solved by using the changing density gradient between the clusters as a separation criterion.

The concept behind GBCC is to use the distance gradient between consecutive pairs of data points to establish a decision boundary for cluster separation. Fig. 3 illustrates this idea for two gradient-separable clusters. Let $\mathscr{P}$ be a path whose endpoints lie in the densest areas of the two clusters and has the following property: If one starts a walk along the edges at vertex A the distance between each consecutive pair of vertices increases monotonically until one reaches vertex G. From there the distances decrease monotonically until one reaches vertex L. An obvious candidate for separating the two clusters is edge k as it is the longest edge along the path.

The goal is to label A to F and G to L respectively with distinct cluster IDs. We can achieve this by using a simple set of conditions: First a cluster can allocate the next vertex along path $\mathscr{P}$ if the weight of the edge that leads to this vertex is greater than the weight to the previous vertex. Second a cluster can conquer an already allocated vertex if the first condition holds and the weight of the edge following the next vertex is greater than the weight of the edge between the current and the next vertex.

If we choose vertex A as our starting position and apply the first condition we can thus label the vertices A to G with a cluster ID. At vertex G however we have to stop as $k > l$. If we now start at vertex L we can proceed until vertex H as G is already allocated to the first cluster. Using the second condition we can now conquer vertex G and overwrite its label as $m < l$ and $l < k$. Since the second condition is not satisfied to also conquer F as $j < k$ the growth of the second cluster is finished and we have arrived at the desired labeling.

This concept is of course oversimplified but nonetheless it describes the basics needed to transform this one-dimensional path problem to a general graph. In the following sections we will deal with how dense areas in graphs can be detected in order to use them as starting positions and how this simple procedure can be generalized in order to work on arbitrary graphs.

4. Graph-Based Competitive Clustering

4.1. Center detection

As the central idea of GBCC is that clusters can only expand along paths of increasing edge weights the starting positions

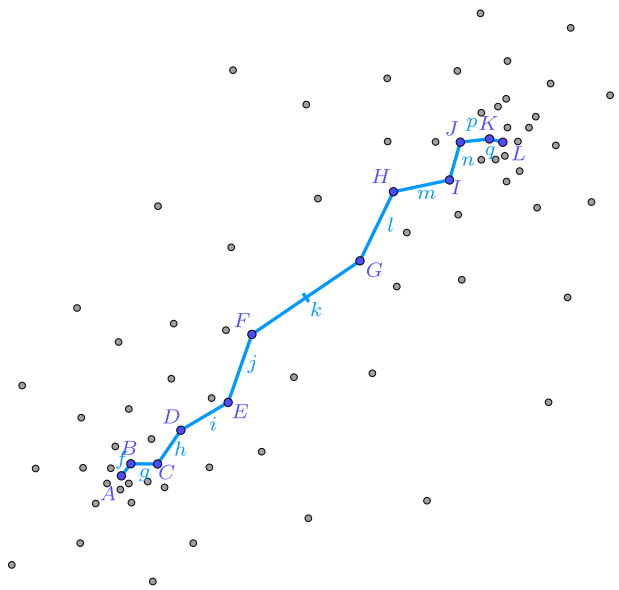


Fig. 3. Fundamental Concept behind GBCC. The key idea of the algorithm is illustrated by means of this one-dimensional path problem where the longest edge k separates the two clusters.

Please cite this article as: B. Hufnagl, H. Lohninger, A graph-based clustering method with special focus on hyperspectral imaging, Analytica Chimica Acta, https://doi.org/10.1016/j.aca.2019.10.071

should be situated in dense areas of the dataset. In the following we will denote these positions as *centers*. With respect to the clustering problem given in Fig. 2 we require a detection technique that is invariant to large differences in density and spatial extent of the clusters so that minor features can also be detected.

Let $\mathbf{W} = (w_{ij})$ be a weighted adjacency matrix and w_{ij} the weight of the edge e_{ij} that connects vertices v_i and v_j. Further let $N_{i,k}$ be the set of neighbors of v_i limited to a maximum of k-nearest neighbors. Then the vertex weight is defined as

$$v_i^{\mathrm{W}} := \frac{1}{|N_{i,k}|}\sum_j w_{ij} \quad \forall j : v_j \in N_{i,k}. \tag{1}$$

Limiting $N_{i,k}$ to the k-nearest neighbors arises from the simple necessity that a meaningful average weight has to be limited to the closer neighborhood. By computing v_i^{W} for every vertex $v_i \in V$ we can thus compare the values of neighboring vertices with each other to determine a minimum. Let N_i^q be the q^{th} -order neighborhood of v_i which are all vertices that are reachable from v_i over q edges. We can then define the order of the center as

$$v_i^{\mathrm{o}} := \max\left\{q : v_i^{\mathrm{W}} < v_j^{\mathrm{W}} \quad \forall v_j \in N_i^q \backslash v_i\right\}. \tag{2}$$

v_i has to be excluded from N_i^q as it is its own higher order neighbor. Put in other words a vertex i is a center of order q if it is the minimum of the vertex weight within an q^{th} -order neighborhood. Fig. 4 illustrates this concept for a one-dimensional graph.

As can be expected the computation of v_i^{o} is time-intensive and as the global maximum is of infinite order we have to define an upper boundary u at which the calculation for a specific vertex stops. As will be shown in section 5.2.2 it suffices to compute v_i^{o} up to a value of $u := 10$ to extract relevant centers. Since the number of vertices for which $v_i^{\mathrm{o}} \geq 1$ is usually very large we can limit the number of centers used for the clustering to

$$v_i^{\mathrm{c}} := \begin{cases} \text{true} & \text{if } v_i^{\mathrm{o}} \geq p \\ \text{false} & \text{otherwise} \end{cases}, \tag{3}$$

where p is a user-defined value that is tuned until the number of centers comes close to the number of desired clusters.

The above methodology for detecting the centers is a form of density estimation. The notion of detecting clusters as dense regions in feature space is a common approach in clustering. For *K*-Means this notion can be used to initialize the centroids in dense regions of the feature space which may result in an improved convergence rate [14,15]. Alternative algorithms for such tasks which are perhaps less sensitive than the above method but certainly faster are DDDE proposed by Fränti and Sieranoja [16] and the initial step of the density peaks algorithm by Rodriguez and Laio [17].

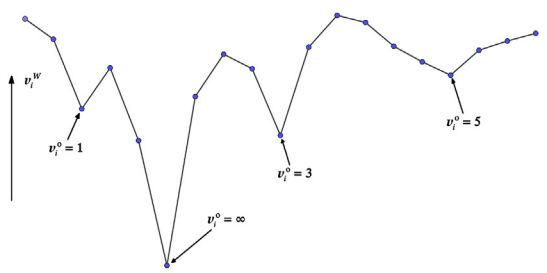


Fig. 4. Illustration of the vertex weight v_i^{W} and the respective order v_i^{o} of the detected centers for a one-dimensional graph.

4.2. Clustering

The conceptual problem in Fig. 3 assumed that a path $\mathscr{P}$ satisfying the above criteria is already given. For an arbitrary graph and a set of centers however there will be a multitude of paths that connect the centers. Therefore the two conditions for splitting the path at edge k still need to be generalized. One obstacle that has to be overcome here is that with respect to a certain vertex v_i there is no obvious candidate for the 'previous vertex' or 'previous edge' respectively which can be used to determine the distance gradient. This issue is addressed by introducing the vertex property v_i^{a} denoted as the *allocation weight* which will take the place of the previous vertex. How this value is determined will be explained in due course. As multiple vertices will have to be processed at the same time we further require different states a vertex can assume which will be denoted by the vertex property v_i^{s}. The possible states are free, pending, active and passive which are illustrated in Fig. 5. The cluster ID will be denoted by v_i^{ID}.

At the start of the algorithm all vertices are initialized with $v_i^{\mathrm{a}} := 0$. The centers v_i^{c} assume the state active and each receives a unique cluster ID. The remaining vertices are initialized as free.

Definition 1. Let v_j be a vertex of the neighborhood of v_i. Then v_j fulfills the *gradient condition* with respect to v_i if

$$v_i^{\mathrm{a}} < w_{ij}. \tag{4}$$

Definition 2. The term *allocation* is used to denote the process under which an active vertex v_i assigns a free vertex v_j of the neighborhood to its cluster providing that v_j satisfies the *gradient condition*. The properties of v_j then change to $v_j^{\mathrm{a}} := w_{ij}$, $v_j^{\mathrm{ID}} := v_i^{\mathrm{ID}}$ and $v_j^{\mathrm{s}} :=$ pending.

Definition 3. Let v_i be an active vertex and v_j an already allocated vertex of its neighborhood. Then v_j fulfills the *update condition* with respect to v_i if it satisfies the *gradient condition* and further satisfies

$$w_{ij} \leq v_j^{\mathrm{a}}. \tag{5}$$

Definition 4. The term *update* refers to the process under which an active vertex v_i overwrites the allocation weight of a neighboring vertex v_j if the *update condition* is fulfilled. The properties of v_j then change to $v_j^{\mathrm{a}} := w_{ij}$ and $v_j^{\mathrm{s}} :=$ pending. The process of *conquering* a neighboring vertex that has been allocated by a different cluster further induces the change $v_j^{\mathrm{ID}} := v_i^{\mathrm{ID}}$.

Using the above definitions we can now formulate a pseudo code version of GBCC which is given in Algorithm 1. Lines 1 to 10 describe the initialization of the vertices. The actual clustering happens in the while-loop of lines 11 to 30 which repeats until all active vertices have been processed. The first step is always to

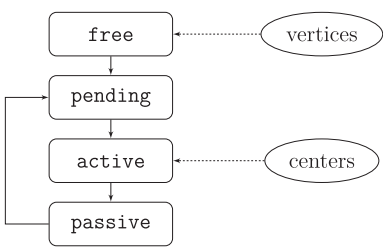


Fig. 5. Flow chart of the vertex states. At the start of the clustering algorithm all centers are initialized as active while all other vertices are initialized as free. The algorithm terminates once all vertices remain in the state passive.

Please cite this article as: B. Hufnagl, H. Lohninger, A graph-based clustering method with special focus on hyperspectral imaging, Analytica Chimica Acta, https://doi.org/10.1016/j.aca.2019.10.071

determine the set of currently active vertices A. For each respective vertex of that set the neighborhood N is determined which is then processed using definitions 1 to 4. From a computational point of view it does not make sense to distinguish between *updating* and *conquering* as the only difference is that the cluster ID of the respective vertex needs to be changed. Therefore the property v_j^{ID} is always overwritten.

Lines 20, 24 and 28 describe the changes in the vertex states as illustrated in Fig. 5. Once the currently active vertices have checked their neighbors they fall back into the state passive. All vertices of the respective neighborhoods that undergo the process of *allocation*, *updating* or *conquering* are set to pending. At the end of the while-loop they become active and thus form the next generation that tests its neighbors. In general a vertex will go through the process of being set to pending by updating or conquering multiple times since the initial allocation weight v_j^{a} is usually not the lowest possible value. The algorithm finally terminates once there are no more active vertices left.

Fig. 6 shows a clustering result where GBCC was applied to a 30-nearest neighbor graph derived from an artificial dataset of three mixing clusters. By using the detection method which was described in section 4.1 the centers have been placed into the three densest regions. The red lines that overlay the underlying graph structure indicate the allocation weights. Starting from the centers we can thus follow a path along the red lines along which the distances between each consecutive pair of vertices increases until we reach the border vertex. This is the two-dimensional version of the conceptual idea illustrated in Fig. 3.

5. Experimental

The following sections are devoted to evaluating the characteristics of GBCC both with respect to other clustering algorithms as well as low and high-dimensional spectroscopic data. In general the evaluation and comparison of clustering algorithms is a problematic subject. Von Luxburg et al. [18] severely criticized the common approaches such as applying cluster validity indices [19] or using labeled benchmark classification data as being insufficient and sometimes completely misleading. They also pointed out that the evaluations often fall short of actually measuring the *usefulness* of the method in question. In light of these arguments we want to stress that this evaluation is focused on hyperspectral images and introduces assumptions about the structural situation in higher dimensions that might not be applicable to other kinds of data at all. We further want to highlight when the additional overhead of using GBCC may be justified and what the limitations of the method are.

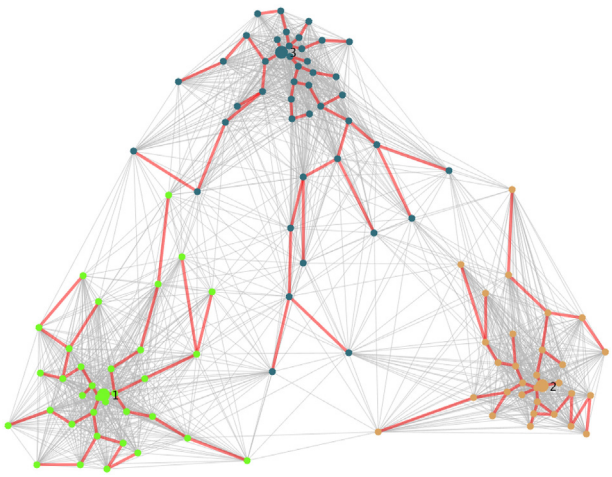


Fig. 6. An example of applying GBCC to a 30-nearest neighbor graph of three mixing clusters.

The following computations were done using MATLAB R2016b (The MathWorks, Inc.; mathworks.com). Result assessment and visualizations regarding hyperspectral data were done using Epina ImageLab (Epina GmbH; imagelab.at) by using its import routines and scripting language ILabPascal.

5.1. 2D artificial dataset

The following clustering experiment offers a comparison of GBCC, *K*-Means, DBSCAN [7] and *K*NSC [9]. The underlying artificial dataset [dataset] [20] was designed manually and draws its inspiration from structural aspects that can be seen in principal component plots of hyperspectral data. One such characteristic is the occurrence of the lobe-like structures which have already been discussed in section 3. Further distance-separated, density-separated, gradient-separated as well as connected clusters have been placed into the dataset. Following the notion that clusters may vary significantly with respect to their spatial extent the respective separability problems are scaled at different levels and only become visible by magnifying certain parts of the dataset. Another special aspect of this dataset is that cluster borders have been kept rather ambiguous. We believe that this better resembles the situation in spectroscopic data since the commonly used 2D shape datasets [4,12,21–24] overemphasize the separability.

Fig. 7 summarizes the results. The unclustered data is given in Fig. 7a. GBCC was applied to a 15-nearest neighbor graph using the Euclidean distance as a similarity metric. The vertex weight v_i^{W} was computed as defined in (1) using $k = 2$ neighbors. The centers v_i^{c} were extracted as defined in (3) for a value of $p = 1$ which resulted in $K = 16$ clusters. Fig. 7b and c shows the results where the former also includes the underlying graph. The red lines indicate the allocation weights which were used to determine the distance gradient. *K*-Means was computed for $K = 20$ clusters and is given in Fig. 7d. The parameters of DBSCAN were set to MinPts $= 2$ and Eps $= 2.2$ and produced the clustering in Fig. 7e. The result of *K*NSC is given in Fig. 7f. Here we created an undirected 15-NN graph by setting all $w_{ij} := w_{ji}$ if $w_{ij} = 0$ and $w_{ji} \neq 0$. The clustering was then performed by following the procedure described in section 2 by setting $K = 16$.

A quick comparison of the results in Fig. 7c–f shows that all tested algorithms seem to have problems with identifying the lobe-like structures as these are partitioned into smaller clusters. Seen from the perspective of hyperspectral imaging this tells us that the Euclidean metric might not be a suitable similarity measure for these kinds of clusters.

Another distinguishing aspect between the four algorithms is that *K*-Means and DBSCAN focused on the bulk features of the data, whereas GBCC and *K*NSC created clusters that were adapted to the scale of the underlying structures. This characteristic becomes more evident if one compares the results of Figs. 7 to 8. Here close-up views of the embedded microstructures are given for GBCC in Fig. 8b and *K*NSC in Fig. 8c. Positions 1 and 2 indicate two density-separable clusters. Another such example is the cluster at position 5 where the difference in density is even greater. Positions 3 and 4 indicate two clusters which are gradient-separable. Here both GBCC and *K*NSC were able to separate them.

5.2. Hyperspectral images

While the above experiment gives a good impression of how GBCC compares to other algorithms when exposed to a multi-challenge artificial dataset it tells little about how it will behave

Please cite this article as: B. Hufnagl, H. Lohninger, A graph-based clustering method with special focus on hyperspectral imaging, Analytica Chimica Acta, https://doi.org/10.1016/j.aca.2019.10.071

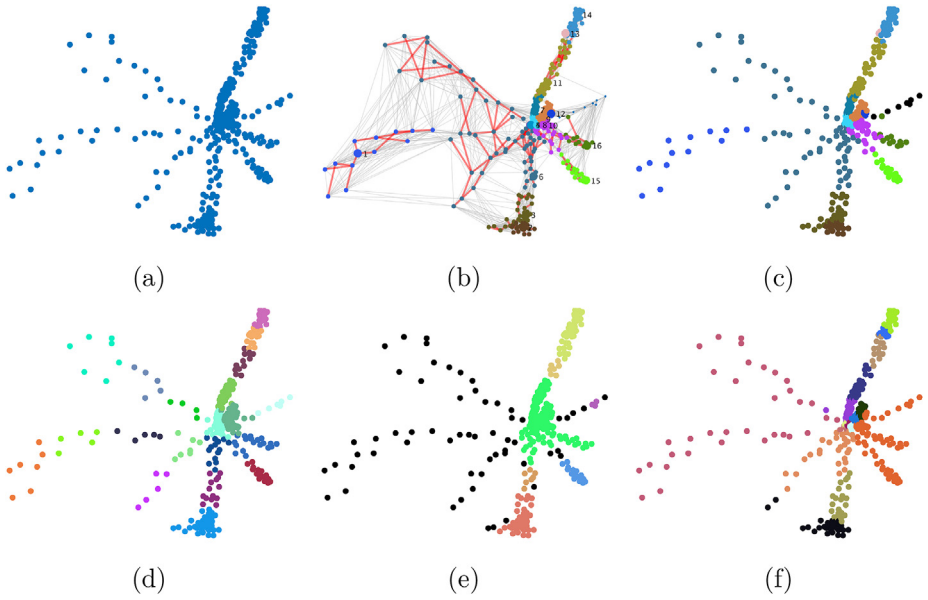


Fig. 7. Clustering of an artificial dataset [dataset] [20]. (a) unclustered data, (b) GBCC with underlying 15-NN graph and allocation weights highlighted in red, (c) GBCC, (d) *K*-Means, (e) DBSCAN, (f) *K*NSC. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

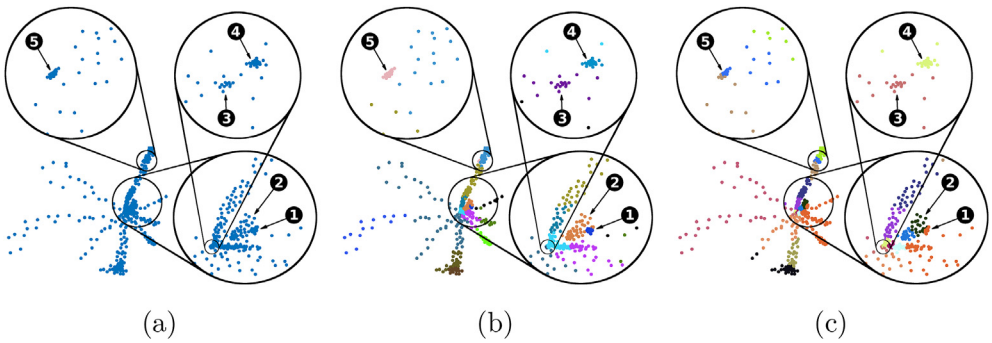


Fig. 8. Zoomed in views of the clustering for (a) unclustered data, (b) GBCC and (c) *K*NSC.

when the clustering is performed on HSIs.

In the following sections we will evaluate GBCC on two well-studied HSIs from the literature. The first test case is a multi-sensor image [dataset] [26] of precipitated atmospheric particulate matter combining energy dispersive electron probe X-ray (EDX) and Raman microspectroscopy (RMS). The hypercubes of the EDX and RMS measurements have been coregistered so that lateral positions coincide and allow for a joint analysis. The image has a size of 101 × 101 pixels with 13 element specific bands in the EDX cube and 1024 bands in the RMS cube which were measured from 202.7 cm^{-1} to 3335.1 cm^{-1}. As this HSI has been thoroughly analyzed by Ofner et al. [25] using principal component analysis (PCA), hierarchical clustering of the loadings (PCA-HCA), *K*-Means and vertex component analysis (VCA) we consider this dataset a good choice for comparing GBCC to other chemometric approaches.

The second test case is taken from a recent study conducted by Hufnagl et al. [28] where a classifier for detecting microplastics in environmental fresh water samples has been developed. The particular HSI [dataset] [27] shows a membrane filter on which a spiked plankton sample has been concentrated for the purpose of creating training data for supervised classification. The original sample was spiked with a mixture of polymer particles consisting of polyethylene (PE), polypropylene (PP), polystyrene (PS), polyacrylonitrile (PAN) and poly(methyl methacrylate) (PMMA). The image was measured using an FPA-based micro-FTIR microscope in the range of 1249.6 cm^{-1} to 3594.5 cm^{-1} and has a size of 276 × 295 pixels. Details on the sample pretreatment and the measurement setup can be found in Löder et al. [29,30]. The choice for selecting this dataset was influenced by the idea that the results of that study allow for a comparison between the clustering obtained by applying GBCC and the classification obtained from a random decision forest [31] classifier.

An important conclusion that can be drawn from the experiment on the 2D artificial dataset is that the Euclidean metric seems to be ill-suited for separating the lobe-like clusters that are expected to play a dominant role in the high-dimensional data structure. Therefore the graphs used in the following experiments are all based on the cosine similarity which reflects the angular relation between the observations. As both the center detection and the clustering step are rather complex algorithms we will further evaluate each process separately.

5.2.1. Dimensionality reduction

The studies conducted by Ofner et al. [25] and Hufnagl et al. [28] have in common that spectral descriptors [32,33] (SPDCs) were used as a means of dimensionality reduction. The idea behind this approach is to project the spectral features into a descriptor space of reduced dimensionality by means of some simple predefined

mathematical functions. These can be as trivial as the baseline corrected area of a peak or the ratio between two spectral raw intensities. On the other hand more complex characteristic vibrational band patterns may be mapped by computing correlation coefficients between a template pattern and a specific spectral range.

In unsupervised learning approaches such as cluster analysis or PCA the design of a set of SPDCs is done manually and thus allows to incorporate chemical expert knowledge. The outcome is a descriptor space of reduced noise and improved data structure which can boost the performance of chemometric techniques. For the sake of brevity we will not describe this process in more detail but provide the SPDC definitions as a supplement to this paper [dataset] [34]. The interested reader may find a more in-depth introduction and references in Hufnagl et al. [28].

The experiments which were conducted on the particulate matter dataset were all done on unprocessed raw data in the original feature space yet it should be kept in mind that Ofner et al. [25] used SPDCs before applying PCA, PCA-HCA and *K*-Means.

In the case of the much more complex microplastic dataset a set of SPDCs [dataset] [34] was used to reduce the original dimensionality of the hypercube from 609 spectral features to 30 descriptor variables. As most of the added microplastics are in the size range of 10 μm−200 μm the spectra exhibit severe baseline distortions due to Mie scattering. These effects make a direct analysis of the raw data very difficult for which reason we decided that such a data pretreatment was necessary for the purpose of this evaluation.

5.2.2. Center detection

To assess the performance of the center detection with respect to its exploratory capabilities we first applied it to the raw data of the particulate matter dataset (in this case EDX and RMS). Since Ofner et al. used VCA on the raw data for confirming the findings of the other chemometric methods this allows for a direct comparison in the same feature space. VCA is an unsupervised unmixing method proposed by Nascimento and Dias [35] which assumes that an HSI is a linear mixture of so-called spectral endmembers. The algorithm fits a *d*-simplex into the spectral raw data were each vertex represents a pure chemical component.

In order to compare GBCC to VCA two 30-NN graphs were computed for the EDX and RMS hypercubes. The vertex weight v_i^W was calculated using a value of $k = 3$. The computation of the order of the centers v_i^o as defined in (2) was limited to an upper value of $u := 10$. For the EDX spectra the minimum order was set to $p = 2$ which resulted in $K = 9$ centers. The corresponding spectra were identified as iron (Fe), silicon (Si), sodium chloride (NaCl) as well as sodium nitrate ($NaNO_3$) and are depiced in Fig. 9. The remaining spectra were identified as background and were therefore not included in the illustration. In the case of RMS the minimum order was set to $p = 1$ which resulted in $K = 11$ centers. These were identified as soot, Si, $NaNO_3$, secondary organic aerosol (SOA), calcium sulfate ($CaSO_4$), a mixture of $CaSO_4$ and SOA as well as titanium dioxide (TiO_2) and are depicted in Fig. 10. The remaining spectra were identified as background. The results of this analysis are summarized in Table 1 where a comparison to the results obtained by Ofner et al. is made.

The microplastic dataset which has been preprocessed as described in section 5.2.1 was analyzed using a 30-NN graph. The vertex weight was calculated using $k = 10$ neighbors. The centers were extracted by setting the minimum order to $p = 2$ which resulted in 122 clusters. By assessing the corresponding spectra of each center the five polymers could be identified. The polymers PE and PP were each detected by one center. PS, PAN and PMMA on the other hand were detected by two, three and five centers respectively. The corresponding spectra are visualized in Fig. 11 where spectra of the same polymer type have been superimposed.

5.2.3. Clustering

The clustering results of the particulate matter dataset are given in Fig. 12. Here the associated clusters of the identified centers overlay the SEM image of the atmospheric sample. Clusters which have been identified as noise or background are not shown. For a comparison to *K*-Means and PCA-HCA see Table 1. A more detailed comparison is possible using the stack images given in Ofner et al. [25].

The results of the microplastic dataset are given in Figs. 13 and 14. Since the underlying data structure of that HSI is quite complex Fig. 13 shows a PCA analysis in combination with the detected clusters. As discussed in section 3 the spatial extent of the clusters varies by orders of magnitude. This is also the reason why the 122 clusters are not visible in Fig. 13a. In Fig. 13d a close-up 3D view reveals that many clusters amass close to the origin. As an alternative visualization we also provide a spherical projection of the first three principal components in Fig. 13b by which the angular relationship between the clusters can be seen.

By assessing the spectral information of the center detection the 122 clusters were merged to form five polymer clusters and an additional cluster for the remaining matrix and filter disc spectra. The results of that combination can be seen in Fig. 13c as well as Fig. 14d where the cluster map is superimposed with the optical image of the filter disc. Of the 81420 pixels of this HSI 320 pixels (0.39% of the data) were clustered as PE, 257 (0.31%) as PP, 812 (1.0%) as PS, 2172 (2.7%) as PAN and 3181 (3.9%) as PMMA.

5.3. Discussion

A key aspect of the proposed algorithm is its ability to detect clusters independently of their relative data density and spatial extent. This quality could be proven by applying GBCC to the artificial dataset in Figs. 7 and 8 were it is quite plain to see that only *K*NSC scales its clusters in a comparable manner. These findings already show that GBCC can detect micro and macro structures at the same time which clearly distinguishes this approach from the well-established *K*-Means algorithm.

Another distinct quality is the spectroscopic interpretability of the detected cluster centers which allows to identify chemical compounds. Figs. 9 and 10 in conjunction with Table 1 show that the center detection provides comparable results with respect to VCA when applied to the raw data of the EDX and RMS hypercubes. In the case of the more complex microplastic dataset the spectral information allowed for a quick identification and merging of the 12 polymer clusters as well as the 110 matrix and filter disc spectra. This translated the seemingly meaningless result in Fig. 14c into an easily interpretable cluster image in Fig. 14d.

A typical characteristic of spectroscopic data is its high dimensionality which can have a significant negative impact on many chemometric techniques. Here the conducted experiment on the RMS hypercube shows that the center detection can still perform very well in a 1024-dimensional feature space. However when comparing Fig. 12a and b we note that the $CaSO_4$ cluster and some of the other clusters are considerably more noisy than in the EDX hypercube and contain some false assignments. Please note that Si is an impurity of the aluminum foil and as such is correctly detected.

With respect to the comparison of the results which is shown in Table 1 we conclude that GBCC performed comparable to VCA and thus may be applied as an alternative strategy for detecting pure chemical compounds. Not surprisingly the results that were obtained by applying SPDCs as a dimensionality reduction technique performed significantly better, especially if one considers the PCA-HCA approach. This clearly shows the advantages of combining multisensor data and chemical expert knowledge, but hampers the comparability between the respective approaches due to the

Please cite this article as: B. Hufnagl, H. Lohninger, A graph-based clustering method with special focus on hyperspectral imaging, Analytica Chimica Acta, https://doi.org/10.1016/j.aca.2019.10.071

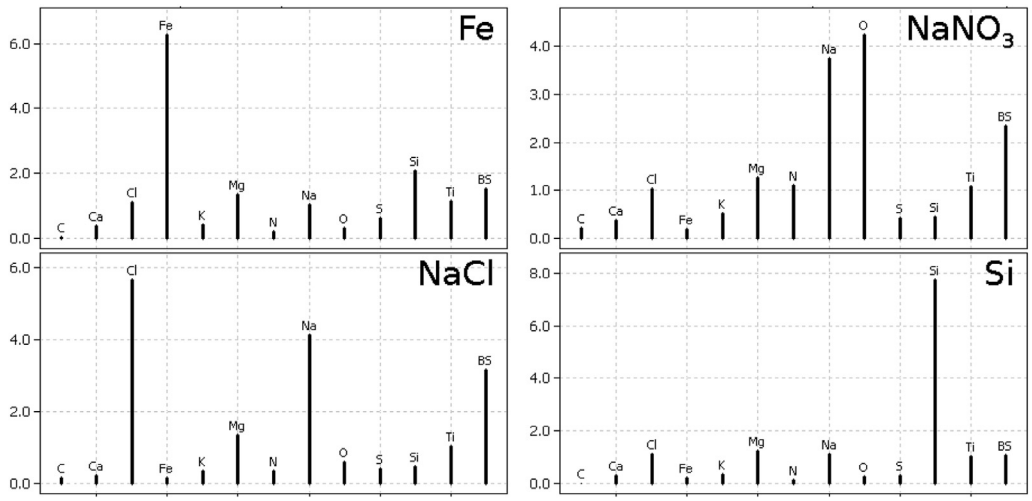


Fig. 9. A selection of centers detected in the EDX hypercube. The centers were identified as iron (Fe), sodium nitrate ($NaNO_3$), sodium chloride (NaCl) and silicon (Si). BS denotes the back scattering signal.

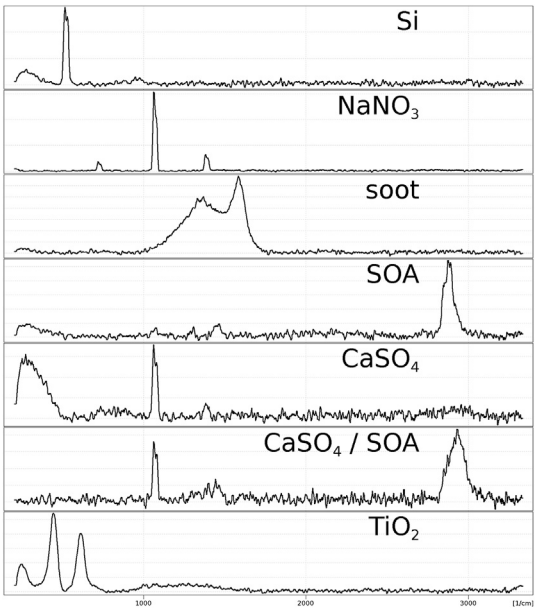


Fig. 10. A selection of detected centers in the RMS hypercube. The centers were identified as silicon (Si), sodium nitrate ($NaNO_3$), soot, secondary organic aerosol (SOA), calcium sulfate ($CaSO_4$), a mixture of $CaSO_4$ and SOA, and titanium dioxide (TiO_2).

Table 1
GBCC in comparison with the results obtained by Ofner et al. [25]. This comparison includes the original feature spaces of energy dispersive electron probe X-ray (EDX) and Raman microspectroscopy (RMS) as well as the results obtained by applying spectral descriptors (SPDCs).

method	feature space	Fe	Si	NaCl	$NaNO_3$	soot	SOA	$CaSO_4$	TiO_2
PCA	SPDC	n	n	n	n	y	n	n	y
PCA-HCA	SPDC	y	y	y	y	y	y	y	y
K-Means	SPDC	n	y	y	y	y	y	n	y
VCA	EDX	n	y	y	y	n	n	y	y
VCA	RMS	n	y	n	y	y	y	n	y
GBCC	EDX	y	y	y	y	n	n	n	n
GBCC	RMS	n	n	n	y	y	y	y	y

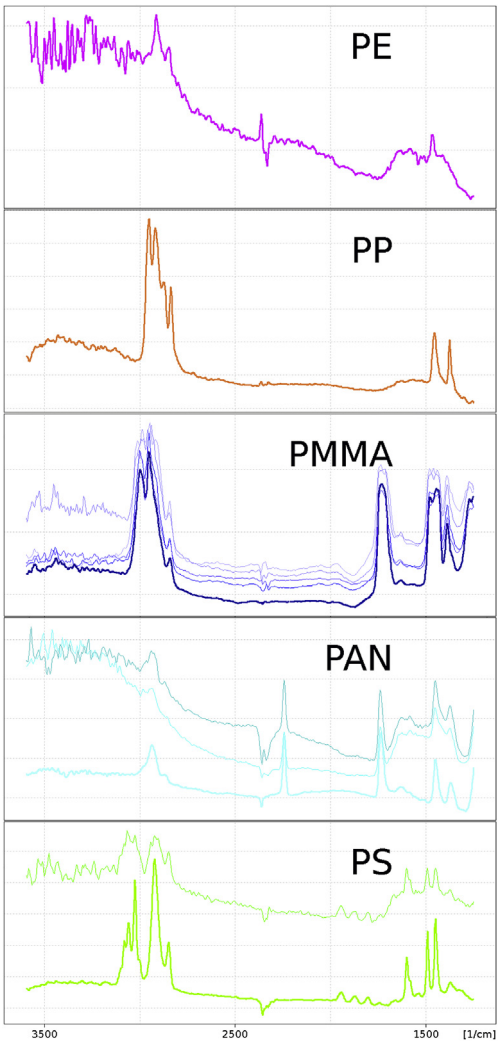


Fig. 11. A selection of detected polymer centers in the Microplastic dataset. The 12 centers were identified as polyethylene (PE), polypropylene (PP), poly(methyl methacrylate) (PMMA), polyacrylonitrile (PAN) and polystyrene (PS).

Please cite this article as: B. Hufnagl, H. Lohninger, A graph-based clustering method with special focus on hyperspectral imaging, Analytica Chimica Acta, https://doi.org/10.1016/j.aca.2019.10.071

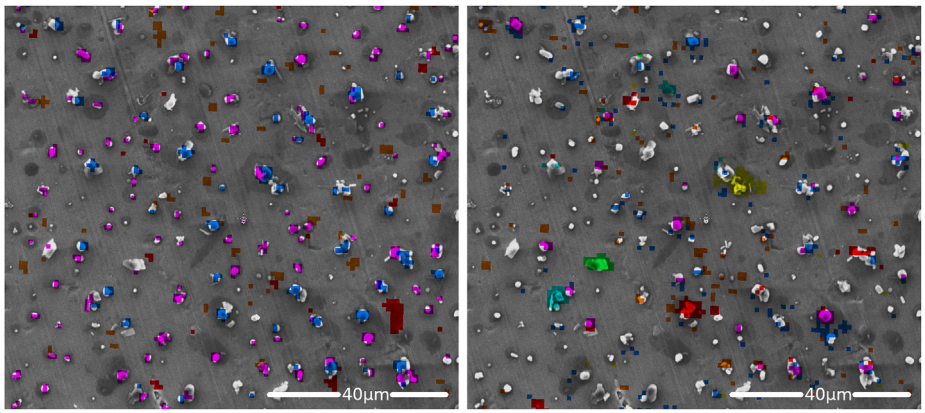


Fig. 12. SEM image of atmospheric particulate matter [dataset] [25,26] overlayed with the clustering result. (a) EDX; pink, NaCl; orange, Si; red, Fe; blue, $NaNO_3$. (b) RMS; red, soot; blue $CaSO_4$, yellow, TiO_2; green, $CaSO_4$/SOA; cyan, SOA; pink, $NaNO_3$; orange, Si. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

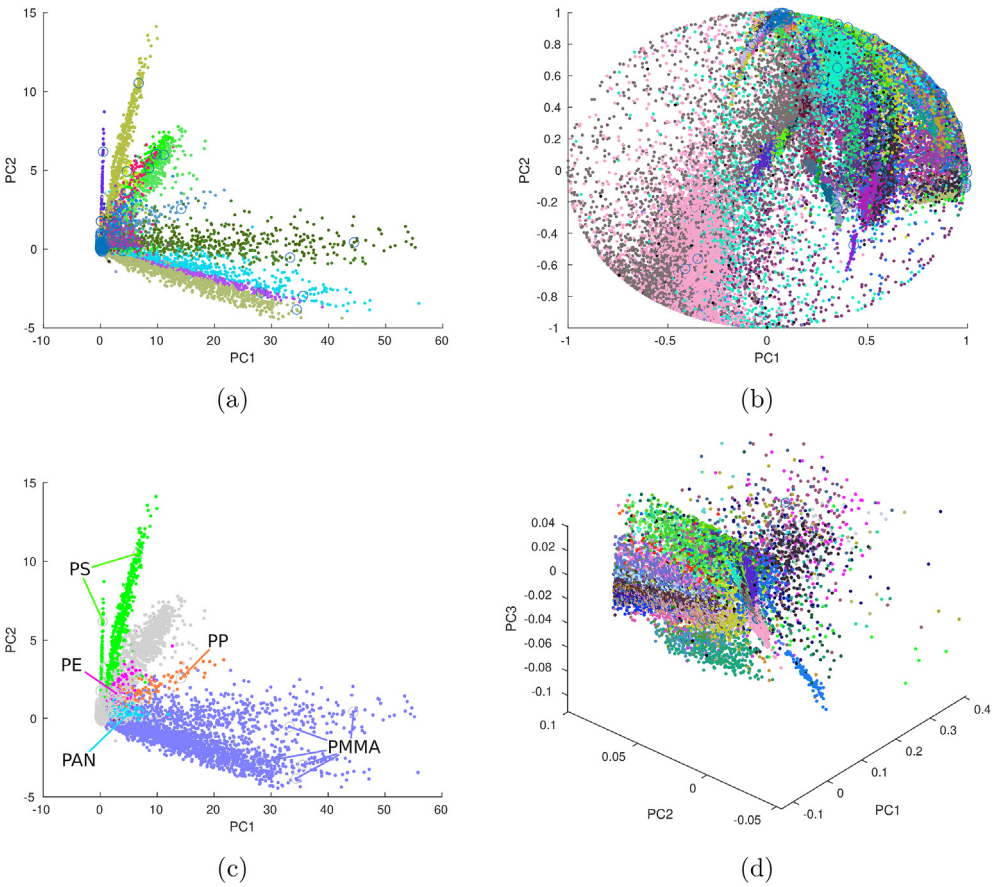


Fig. 13. Cluster analysis of the microplastic dataset [dataset] [27]. (a) PC plot of the 1st and 2nd component; (b) scores of the three largest components projected onto a sphere; (c) highlighted polymer clusters; (d) close-up view of the origin in a 3D PCA plot. The small circles indicate a detected center.

different feature spaces.

In the case of the microplastic dataset which was also pre-processed using the SPDC approach the results obtained by GBCC are in good agreement with the classification result of the random decision forest as can be seen in Fig. 14a and b. If one closely compares the identified microplastics it is interesting to note that the clustered particles are in many cases slightly larger than their classified counterparts. This effect can be explained by considering the mathematical differences between the algorithms. In supervised learning the decision rules are inferred from training data and are as such fixed once the classification model is applied to the target data. Contrary to that unsupervised learning approaches such as clustering deduce the decision rules based on structural aspects of the data. This is the reason why GBCC is more flexible with respect to the spectra at the particle edges than the random decision forest.

Please cite this article as: B. Hufnagl, H. Lohninger, A graph-based clustering method with special focus on hyperspectral imaging, Analytica Chimica Acta, https://doi.org/10.1016/j.aca.2019.10.071

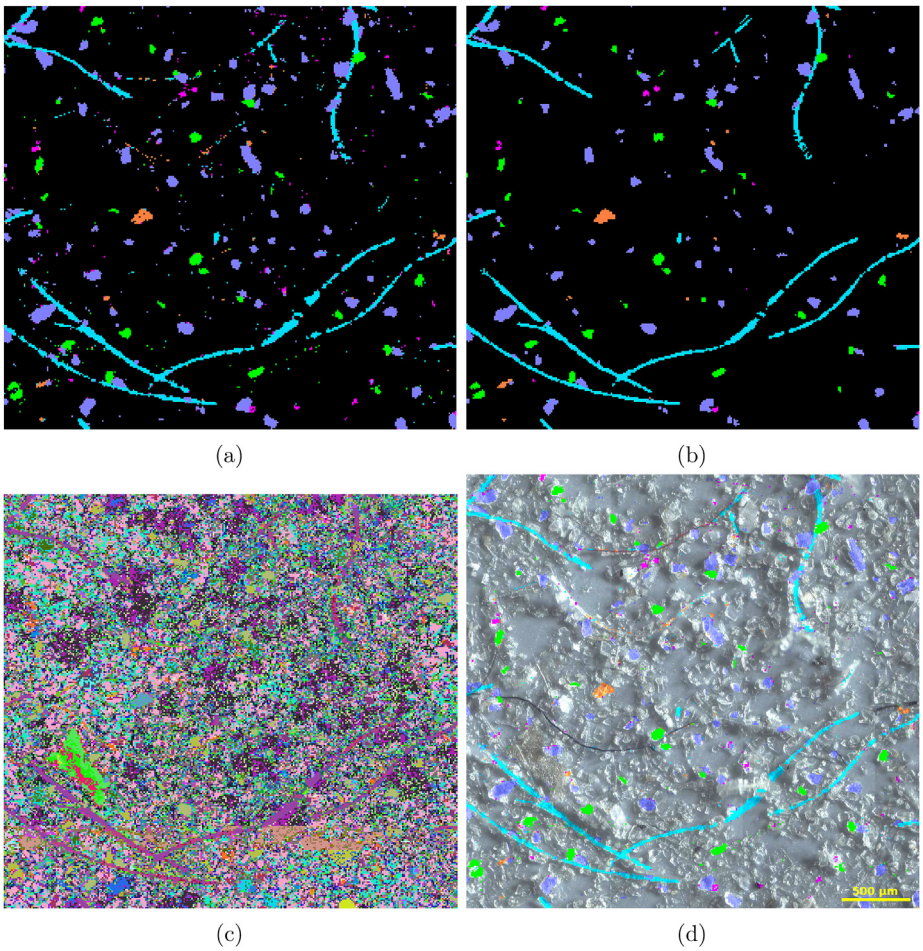


Fig. 14. Clustering results of the microplastic dataset [dataset] [27]. (a) GBCC's merged clusters based on the detected polymer centers; (b) random decision forest classification result as published by Hufnagl et al. [28] under a Creative Commons Attribution 4.0 International License (https://doi.org/10.1039/c9ay00252a); (c) unprocessed output showing the original 122 clusters; (d) merged clusters overlayed with the optical image of the filter disc; Pink, PE; orange, PP; purple, PMMA; cyan, PAN; green, PS; black/transparent, matrix and filter surface. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

An important aspect when choosing algorithms for exploratory analysis is the computational cost which has to be weighted against the benefits of the respective approach. For the purpose of evaluating GBCC with respect to *k*NN graphs we conducted complete searches for the k nearest neighbors. One should keep in mind that such computations are very costly and are of order $O(dn^2)$ where d denotes the dimension of the feature space and n the number of objects. As the determination of the *k*NN is an important problem in most data retrieval systems a lot of research is devoted to specialized hardware and algorithms that lower the computation time by finding approximate nearest neighbors. See for example Chen et al. [36] for a fast algorithm in high-dimensional spaces and Sismanis et al. [37] for a parallel GPU implementation. As the computation of the *k*NN graph forms the bottleneck of the method we consider the costs of applying GBCC for the purpose of exploratory analysis acceptable yet also conclude that the application in the context of routine analysis will likely be too time consuming. On the other hand alternative settings such as computing approximate nearest neighbors or subsampling of the data as well as determining the effect of lowering the number of neighbors k where not explored in this study which can have a significant impact on computation times.

6. Conclusion

In this paper we proposed a novel clustering method dubbed GBCC which can handle clusters independently of their relative differences in data density and spatial extent. In the view of the authors this constitutes the most important finding of this study as only *K*NSC, which is also a graph-based method, showed a similar behaviour. Contrary to that *K*-Means which is a predominant clustering algorithm in chemometrics focused mainly on the bulk features of the data.

With respect to the dilemma of choosing a clustering algorithm [38] for a dataset we therefore conclude that it is less a question of selecting the 'right' or the 'best' one but rather to choose two antithetic algorithms. With this in mind a combined cluster analysis using GBCC and e.g. *K*-Means might give more insights than a combination of methods that behave very similarly.

As for the *usefulness* [18] of the method we conclude the following: GBCC is a sensitive method that can detect small variations in data density and thus allows an analysis of minor features which might be overlooked by other clustering algorithms. If this trait is of little importance in your field of study or the dataset is known to be of low structural complexity then GBCC might not be the right way to go, especially if one considers the increased computational costs of graph-based clustering approaches. On the other hand this assumption could be the pitfall that causes you to miss an important detail.

Algorithm 1. Pseudocode of GBCC.

Please cite this article as: B. Hufnagl, H. Lohninger, A graph-based clustering method with special focus on hyperspectral imaging, Analytica Chimica Acta, https://doi.org/10.1016/j.aca.2019.10.071

Algorithm 1: Pseudocode of GBCC

```
for i := 1 to n do
    if v_i^c = true then
        v_i^s := active;
        v_i^ID := getUniqueClusterID();
        v_i^a := 0;
    else
        v_i^s := free;
        v_i^ID := 0;
    end
end
while ∃v_i ∈ V : v_i^s = active do
    A := {v_i ∈ V | v_i^s = active};
    foreach v_i ∈ A do
        N := {v_j ∈ V | w_ij > 0};
        foreach v_j ∈ N do
            if updateCondition(v_i, v_j) ∨ (v_j^s = free ∧ gradientCondition(v_i, v_j))
             then
                v_j^ID := v_i^ID;
                v_j^a := w_ij;
                if v_j^s = free ∨ passive then
                    v_j^s := pending;
                end
            end
        end
        v_i^s := passive;
    end
    P := {v_i ∈ V | v_i^s = pending};
    foreach v_i ∈ P do
        v_i^s := active;
    end
end
```

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We wish to thank Martin G. J. Löder and Christian Laforsch of the University of Bayreuth for providing the Microplastic dataset for our research. Further many thanks also go to the anonymous reviewers for providing additional literature references and their comments which helped us in the improvement of the paper.

References

[1] G. Malewicz, M.H. Austern, A.J. Bik, J.C. Dehnert, I. Horn, N. Leiser, G. Czajkowski, Pregel: a system for large-scale graph processing, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, 2010, pp. 135–146.
[2] N. Päivinen, Pattern Recognit. Lett. 26 (2005) 921–930.
[3] C. Zhong, M. Malinen, D. Miao, P. Fränti, Inf. Sci. 295 (2015) 1–17.
[4] C.T. Zahn, IEEE Trans. Comput. 100 (1971) 68–86.
[5] C. Zhong, D. Miao, R. Wang, Pattern Recognit. 43 (2010) 752–766.
[6] Q. Liu, M. Deng, Y. Shi, Wang, J. Comput. Geosci. 46 (2012) 296–309.
[7] others, et al., A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Kdd, 1996, pp. 226–231.
[8] J. Shi, J. Malik, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 888–905.
[9] M.C. Nascimento, A.C. De Carvalho, Eur. J. Oper. Res. 211 (2011) 221–231.
[10] U. Von Luxburg, Stat. Comput. 17 (2007) 395–416.
[11] C. Zhong, X. Yue, Z. Zhang, Lei, J. Pattern Recognit. 48 (2015) 2699–2709.
[12] H. Chang, D.-Y. Yeung, Pattern Recognit. 41 (2008) 191–203.
[13] J. Handl, J. Knowles, IEEE Trans. Evol. Comput. 11 (2007) 56–76.
[14] K.M. Kumar, A.R.M. Reddy, Inf. Sci. 418 (2017) 286–301.
[15] L. Galluccio, O. Michel, P. Comon, A.O. Hero III, Signal Process. 92 (2012) 1970–1984.
[16] P. Fränti, S. Sieranoja, Dimensionally distributed density estimation, in: International Conference on Artificial Intelligence and Soft Computing, 2018, pp. 343–353.
[17] A. Rodriguez, A. Laio, Science 344 (2014) 1492–1496.
[18] U. Von Luxburg, R.C. Williamson, I. Guyon, Clustering: Science or Art? Proceedings of ICML Workshop on Unsupervised and Transfer Learning, 2012, pp. 65–79.
[19] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J.M. Pérez, I. Perona, Pattern Recognit. 46 (2013) 243–256.
[20] B. Hufnagl, H. Lohninger, A Multi-Challenge Clustering Benchmark Dataset Embedding Large Differences in Spatial Extent. Zenodo, 2019, https://doi.org/10.5281/zenodo.2583762 (supplementary clustering benchmark dataset).
[21] A. Gionis, H. Mannila, P. Tsaparas, ACM Trans. Knowl. Discov. Data 1 (2007) 4.
[22] L. Fu, E. Medico, BMC Bioinf. 8 (2007) 3.
[23] C.J. Veenman, M.J.T. Reinders, E. Backer, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 1273–1280.
[24] A.K. Jain, M.H. Law, Data clustering: a user's dilemma, in: International Conference on Pattern Recognition and Machine Intelligence, 2005, pp. 1–10.
[25] J. Ofner, K.A. Kamilli, E. Eitenberger, G. Friedbacher, B. Lendl, A. Held, H. Lohninger, Anal. Chem. 87 (2015) 9413–9420.
[26] J. Ofner, H. Lohninger, Atmospheric Particulate Matter (DS005), 2015. http://www.imagelab.at/data/atm_part_matter.zip. last visited on March 5, 2019.
[27] B. Hufnagl, D. Steiner, E. Renner, M.G.J. Löder, C. Laforsch, H. Lohninger,

Microplastic, Zenodo, https://doi.org/10.5281/zenodo.2555732, 2019 (supplementary hyperspectral image dataset).
[28] B. Hufnagl, D. Steiner, E. Renner, M.G.J. Löder, C. Laforsch, H. Lohninger, Anal. Methods 11 (2019) 2277–2285.
[29] M.G.J. Löder, M. Kuczera, S. Mintenig, C. Lorenz, G. Gerdts, Environ. Chem. 12 (2015) 563–581.
[30] M.G.J. Löder, H.K. Imhof, M. Ladehoff, L.A. Löschel, C. Lorenz, S. Mintenig, S. Piehl, S. Primpke, I. Schrank, C. Laforsch, G. Gerdts, Environ. Sci. Technol. 51 (2017) 14283–14292.
[31] L. Breiman, Mach. Learn. 45 (2001) 5–32.
[32] H. Lohninger, J. Ofner, Spectrosc. Eur. 26 (2014) 6–10.
[33] J. Ofner, F. Brenner, K. Wieland, E. Eitenberger, J. Kirschner, C. Eisenmenger-Sittner, S. Török, B. Döme, T. Konegger, A. Kasper-Giebl, H. Hutter, G. Friedbacher, B. Lendl, H. Lohninger, Sci. Rep. 7 (2017) 6832.
[34] B. Hufnagl, H. Lohninger, A Collection of Spectral Descriptors for the Detection of Five Polymer Types, Zenodo, 2019, https://doi.org/10.5281/zenodo.3377095 (supplementary collection of spectral descriptors).
[35] J.M. Nascimento, J.M. Dias, IEEE Trans. Geosci. Remote Sens. 43 (2005) 898–910.
[36] J. Chen, H.-r. Fang, Y. Saad, J. Mach. Learn. Res. 10 (2009) 1989–2012.
[37] N. Sismanis, N. Pitsianis, X. Sun, Parallel search of k-nearest neighbors with synchronous operations, in: 2012 IEEE Conference on High Performance Extreme Computing, 2012, pp. 1–6.
[38] C. Hennig, Pattern Recognit. Lett. 64 (2015) 53–62.

Please cite this article as: B. Hufnagl, H. Lohninger, A graph-based clustering method with special focus on hyperspectral imaging, Analytica Chimica Acta, https://doi.org/10.1016/j.aca.2019.10.071