Adaptive density peaks clustering: Towards exploratory EEG analysis

Tengfei Gao, Dan Chen, Yunbo Tang, Bo Du, Rajiv Ranjan, Albert Y. Zomaya, Schahram Dustdar

The School of Computer Science and the National Engineering Research Center for Multimedia Software, Wuhan University, Wuhan 430072, China
The School of Computing, Newcastle University, Newcastle upon Tyne, NE4 5TG, United Kingdom of Great Britain and Northern Ireland
The School of Information Technologies, the University of Sydney, Sydney, 2006, Australia
The Distributed Systems Group (DSG) of Information Systems Institute, Vienna University of Technology, Vienna, 1040, Austria

Abstract

Finding appropriate cluster centers and determining the scope of influence explicitly associated with each center is at the very core of a successful clustering process, which has long been particularly difficult and important when handling bio-signals such as electroencephalography (EEG). Considering exploratory EEG analysis as a typical case, this study forms an adaptive density peaks clustering (ADPC) solution to the open problem based on the Density Peaks Clustering (DPC) algorithm. First, in order to optimize the cutoff distance (key parameter previously set manually) to adapt to various clustering tasks, an optimization function was constructed with the target dataset’s uncertainty that can be solved by the extended Pattern Search Algorithm (PSA). Second, ADPC automatically constructs a set of cluster centers by jointly ranking the local density and relative distance, and then fine-tuning the set by balancing the intra-set independence and the tendency as a center against extra-set competitors from the perspective of each candidate. An exploratory EEG analysis framework was then fostered by centering on ADPC. Benchmarks on public datasets show the superiority of ADPC over its mainstream counterparts in terms of effectiveness and adaptability. The case study on epileptic EEG indicates that (1) the framework achieves averages on Precision, Recall, and F1-score of 100%, 92.46%, and 95.92%, respectively, in seizure detection involving no a priori information, and (2) the key observations revealed through clustering match the accepted conclusions well. Overall, ADPC enables automated clustering, which is well adaptive to exploratory EEG analysis.

© 2022 Elsevier B.V. All rights reserved.

1. Introduction

Clustering aims to organize data elements of a dataset into distinct clusters according to their resemblance through unsupervised learning of the intrinsic patterns (if any) of the dataset, which then enables exploration of the dynamics of the underlying (real or imagined) system on the basis of the observations [1]. Data elements in the same cluster are characterized by a similarity higher than those in different clusters [2]. Clustering is probably the most important and fundamental means of exploratory data analysis for finding intrinsic hidden information and patterns (if any) without the need for a priori knowledge. For example, to detect uncertain abnormal states from brain imaging data. Its capability of unsupervised learning has proven extremely useful in various fields such as signal processing, data mining and pattern recognition in general [3].

An exploratory analysis of electroencephalography (EEG) is a typical case. As the dominant means for examining cerebral electrical activities [4], EEG technologies have a wide range of applications in cognitive neuroscience, and the diagnosis of diseases such as epilepsy, schizophrenia, and autism [5]. Nowadays EEG-based neuroscience & engineering tasks still heavily demand onerous and highly skilled labeling. Decision making will otherwise be unreliable even with contemporary supervised and semi-supervised learning methods, such as mainstream EEG classification models [6,7]. Clustering holds the potential to explore brain dynamics especially malfunctions recorded in EEG without the need for labels or having to know all the potential problems/pathology of the subjects beforehand, that is, basically a priori knowledge about the observations (an EEG dataset) provided by experts [8].

The earliest clustering algorithms such as K-means simply rely on distance to make decision on clustering, which assign the cluster members to centers based on their minimum distances and find the most appropriate cluster centers optimization of an distance-based objective function [9]. Distance-based algorithms are the most widely applied benefiting from this simple principle,
but they cannot adapt to nonspherical clusters as data elements are assigned to the nearest center [10].

Distribution-based algorithms such as the Gaussian Mixture Model (GMM) that utilize predefined probability distribution functions to reproduce data elements for adapting to various data distributions subsequently emerge [11]. The accuracy of clustering is subject to the capability of the trial probability in representing the data. The number of distribution functions must be set in advance similar to the number of cluster centers in K-means according to a priori knowledge [12].

Clusters are more likely to have arbitrary shapes. Density-based algorithms identify clusters in a data space with a contiguous region of high point density [13]. The DBSCAN algorithm and its variants can find appropriate cluster centers, but still require manually setting key distance parameters (MinPoints and neighborhood radius) [14] to distinguish different density levels, which itself is an active research topic for selecting optimal parameters [15].

The advent of density peaks clustering (DPC) [16] marks the peak of these attempts. DPC has the advantages of both distance- and density-based methods, with the merits of identifying arbitrarily shaped clusters and their number without requiring explicit manual settings [17]. Although grand successes have been achieved with DPC and its variants [18] (see Section 2.1), challenges with adaptability still remain when handling data with no strong features or with intensive interferences such as bio-signals in practical applications. They are (1) unnecessary limits incurred by empirical setting of the key parameter (cutoff distance), and (2) difficulty in selecting the cluster centers.

However, it is not trivial even for a sophisticated clustering method to detect abnormalities and events from such bio-signals which are highly complex and nonstationary [19]. There is a pressing need for an alternative clustering method with adaptability that is significantly enhanced towards exploratory EEG analysis. This study utilizes advanced optimization theory in connection with the uncertainty of the target dataset to adaptively adjust the cutoff distance ($d_c$) as the key parameter in DPC directly influencing the accuracy of clustering. It may not necessarily fix the cluster centers at the beginning, as more evidence on decision making should be revealed in the course of clustering, especially in the case of multiple center candidates coexisting in a high-density region.

This study proposes an adaptive density peaks clustering approach (ADPC) by extending the DPC algorithm (Section 3):

1. Adaptive selection of the cutoff distance. To obtain the optimal cutoff distance for adapting to various clustering tasks, an optimization function was constructed using the Gini index to measure the uncertainty of the target dataset. Then, the extended pattern search algorithm (PSA) with global optimization capability was applied to obtain the optimal cutoff distance value.

2. Automatic determination of cluster centers. To avoid bias in the manual determination of cluster centers, ADPC automatically constructs a set of cluster centers by jointly ranking the local density and relative distance (measure $\gamma$) and then fine-tuning the set by balancing the intra-set independence and the tendency as a center against extra-set competitors from the perspective of each candidate. Once the extra-set competitor holds the tendency as a center, the corresponding data point is filled in, and the corresponding candidate with relatively low independence is filtered out.

A generic framework for exploratory EEG analysis has been fostered by centering on the ADPC approach combining discrete Fourier transform (DFT) and Bayesian factorization (Section 4). The design aims at automatic clustering in the course of exploring the target. Thus, blind exploration of brain malfunctions may be possible without the need for excessive a priori information.

Benchmark experiments on public clustering datasets were performed to evaluate the adaptive capability and the accuracy of ADPC against relevant counterparts in clustering tasks (Section 5). A case study on epileptic EEG was carried out to evaluate the proposed framework in terms of the capability of exploratory analysis (Section 6).

The main contributions of this study are as follows:

1. This study develops an adaptive clustering approach by extending the DPC algorithm that can optimize the key parameters and determine the cluster centers by itself. It enables adaptive clustering with fully automated operation based on the target dataset without the need for human intervention or empirical setting of key parameters. The resulting algorithm significantly outperforms the mainstream counterparts.

2. A clustering-based framework for exploratory EEG analysis is provided aiming at pathological EEG, and it is proved to be effective for seizure detection. It holds the potential to find abnormal neural activities from EEG without explicit a priori knowledge of the subjects under examination as the classification models do.

2. Related work

Existing clustering algorithms can be based on metrics such as distance, distribution, density, or their combinations. Extensive and comprehensive literature reviews of research exist along this direction [20,21]. This section then focuses on the family of DPC and the most salient EEG clustering methods, as they are closely related to this study.

2.1. Density peak clustering family

DPC algorithm excels in detecting arbitrary shaped clusters and determines the cluster centers in a heuristic way [16,22]. DPC assumes that cluster center is a point with a higher local density compared with its surrounding neighbors and it is located at a relatively large distance from any other points with a higher local density. Recent DPC variants largely aim to tackle the pitfalls empirically setting of the cutoff distance $d_c$ and manual determination of cluster centers [23].

Du et al. proposed the DPC-KNN algorithm incorporating k-nearest neighbors in local density computation [24], which might reduce the risk of incorrect clustering by $d_c$. The DPC-KNN still needed to know the number of the nearest neighbors beforehand. Jiang et al. have developed the GDPC algorithm based on the gravitation theory to accurately identify centroids and anomalies, which attempts to exclude the influence of $d_c$ on the clustering results [25]. Both methods left cluster centers manually determined via Decision Graph.

Wang et al. proposed to determine the number of clusters prior to clustering by finding the "knee point" as the boundary between a cluster center and a noncluster center [26], and the knee point could be derived by minimizing the root mean squared error of two fitting curves. Bie et al. have also developed a method (fuzzy-CFSFDP) to select the cluster centers aided by fuzzy theory raised from statistical analysis against local density and relative distance of cluster centers conforming to Gaussian distribution [27]. Both methods can effectively handle synthetic data but leave the situation of multiple cluster centers co-existing in a high-density region unattended.
2.2. EEG clustering

Existing EEG clustering methods focus on feature engineering tasks such as feature selection, feature extraction, and/or grouping of EEG epochs of different states. Alshebeli et al. have enabled seizure prediction based on K-means with a high performance in terms of prediction accuracy and false alarm rate [28]. Based on the statistical characteristics of amplitude, median, mean, variance and derivative of EEG, K-means could complete the task of seizure prediction with a performance comparable to multilayer perception (MLP) networks.

Bizopoulos et al. have also developed a method to detect the epileptic seizures using K-means and Ensemble Empirical Mode Decomposition (EEMD) [29]. Marginal Spectrum (MS) obtained via EEMD formed the basis for clustering by K-means without the need for training data. Paolo et al. proposed to use a fuzzy clustering method to sustain workflow for EEG event-related potentials [4]. Fuzzy clustering worked in the very core to grade the weighted feature vectors in clustering. Experimental results of emotional Go/NoGo task show the method's robustness to artifacts.

As a contrast to the above, this study is intended to bridge the gap between the most advanced clustering algorithm and the open problems complicated in exploratory analysis of bio-signals. The major concerns include (1) how to automatically optimize the key parameters and determine the cluster centers adapting to various datasets and (2) how to enable an effective method of exploratory EEG analysis.

3. Adaptive density peak clustering

This section first covers the background of the DPC method and notations referenced throughout the following discussions. The design of the adaptive DPC is then given. In particular, the basic theories of the ADPC algorithm are detailed, including: (1) adaptive selection of cutoff distance, (2) automatic determination of cluster centers and, (3) complexity analysis.

3.1. Background and notations

For a given dataset \( S = \{ x_1, x_2, \ldots, x_N \} \), let \( I_S = \{ 1, 2, \ldots, N \} \) represent its corresponding index set. A certain distance between data points \( x_i \) and \( x_j \) in \( S \), such as the Euclidean distance, is denoted by \( d_{ij} = \text{dist}(x_i, x_j) \). In the context of the DPC algorithm, the local density \( \rho_i \) of data point \( x_i \), similar to MinPoints in DBSCAN, is defined as follows:

\[
\rho_i = \sum_j \chi(d_{ij} - d_c),
\]

(1)

where \( [i, j]|i \neq j \in I_S \), and the function \( \chi(x) \) is defined as (2):

\[
\chi(x) = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0. \end{cases}
\]

(2)

When continuous data are assumed, the local density can be alternatively defined as (3):

\[
\rho_i = \sum_j e^{-d_{ij}^2/d_c^2},
\]

(3)

where \( d_c \) is the only parameter to be manually set in the DPC and defines the scope of influence of each data element/point. The cutoff distance \( d_c \) plays an important role in computing the local density \( \rho_i \). As shown in Fig. 1, \( \rho_i \) indicates the number of data points in the influence scope confined by \( d_c \). The accuracy of clustering depends heavily on the setting of \( d_c \).

The DPC algorithm empirically sets \( d_c \) to ensure that the average number of neighbors is approximately 1%–2% of the total number of data points in the dataset. This setting aims to gain adaptability, but the appropriate \( d_c \) for various problem domains may differ. Note that feature extraction or alike can alleviate the high complexity and nonstationarity embedded in the data under examination, and the direct application of DPC can still be risky, as the range of \( d_c \) remains unclear. The success of clustering in this context is subject to the appropriate, that is, adaptive to the dataset, setting of \( d_c \). The capability of adaptively setting an appropriate \( d_c \) is desired to adapt to the corresponding problem domain.

Data field depicts the interactions between objects (i.e., other data points) associated to each data point in the whole data space, mimicking the field theory in physics [30]. In this context, individual data point radiate their strengths (measurements of influence) and vice versa. The local density \( \rho_i \) is defined as the difference in focusing on each individual data point. The data field theory is then applied in designing the adaptive DPC.

For data point \( x_i \), the relative distance \( \delta_i \) is defined as (4):

\[
\delta_i = \min \left\{ \{ d_{ij} \mid \rho_j < \rho_i \}, \{ \max \{ \rho_k \} \mid j \neq i \geq \rho_i \} \right\}
\]

(4)

Relative distance \( \delta_i \) is measured by computing the minimum distance between data point \( x_i \) and any other data points with higher density. However, for the data point with the highest local density, \( \delta_i \) takes the maximum distance to all other points.

As shown in Fig. 1, data point 14 is closest to data point 12 among the first 13 points with higher density, such that \( \delta_{14} = d_{14,12} \) and the relative distance of data point 1 is \( \delta_{1,12} \), which is the maximum distance from data point 1 to the rest of the data points. Cluster centers are recognized as points for which the value of \( \rho_i \) and \( \delta_i \) are anomalously large, and the Decision Graph is constructed based on \( \rho_i \) and \( \delta_i \) to identify cluster centers. The remaining data points are then allocated to the same clusters as their nearest neighbors of higher density. As shown in Fig. 1, the areas within the blue and red dashed lines represent the scope of influence explicitly associated with centers 1 and 10, respectively, and the scope of center 10 is equal to its influence area with \( d_c \).

Moreover, for each cluster, the set of data points assigned to it but also located within a distance \( d_c \) from the data points belonging to other clusters form a border region, and the maximum local density within its border region is denoted as \( \rho_{\text{max}} \). The data points with a density higher than \( \rho_{\text{max}} \) are considered as cluster cores; otherwise, they are regarded as cluster halos, that is, noise, which are more likely.

Pattern Search Algorithm (PSA) aims to solve the optimization problem of functions that are difficult to derive [31]. This study extends this to solve the optimization function constructed by measuring the uncertainty of the target dataset (Section 3.2.1). This metric reflects the chaotic state of the potential distribution of the dataset and can be measured using the Gini index:

\[
\text{Gini} = 1 - \sum_{i=1}^{c} p_i^2,
\]

(5)

where \( p_i \) is the probability that the data points belong to the \( i \)th category, and \( c \) is the number of categories.

Let \( \Phi_i = \psi(x_i) \) denote the potential value of the data point \( x_i \). The relationship among the uncertainty, potential distribution, and Gini index is detailed in Section 3.2.1.
To automatically determine cluster centers, the metric $\gamma$ is proposed to jointly consider $\rho$ and $\delta$ [16] as defined in (6):

$$\gamma_i = \rho_i \delta_i,$$  \hspace{1cm} (6)

$\gamma_i$ is sorted in descending order to obtain the $\gamma$ sequence $\{\text{seq}(\gamma_i)_{i=1}^n\}$. Furthermore, the $\gamma$ of the cluster center point nearest to the noncluster center point in the $\gamma$ sequence is expressed as $\gamma_{BD}$. Meanwhile, the noncluster center point with the $\gamma$ value closest to $\gamma_{BD}$ is treated as a competitor. The difference between candidates in the $\gamma$ sequence can be viewed as the independence in the set of center candidates to be adjusted. The tendency of a candidate to be identified as a cluster should be measured to screen out unqualified candidates (see Section 3.2.2).

### 3.2. Design and theories of adaptive density peak clustering

This study complements the DPC method with the innovations of adaptively selecting the optimal cutoff distance and automatically determining the appropriate cluster centers. The resulted algorithm of ADPC is described in Algorithm 1.

To tackle the pitfall of the empirical setting of the cutoff distance, ADPC introduces data field theory to adaptively obtain the optimal cutoff distance $\delta_c$ by constructing an optimization function with the Gini index and then using the extended PSA to drive the function to the optimum. To enable automatic determination of appropriate cluster centers instead of resorting to experts’ visual inspection as DPC does, ADPC constructs a set of cluster centers by ranking the $\gamma$ and fine-tuning the set based on the independence and tendency of cluster center candidates. The pillar theories proposed in ADPC are described in Sections 3.2.1 and 3.2.2 respectively.

#### 3.2.1. ADPC: Selection of cutoff distance

The data field theory is introduced here to adaptively obtain the optimal cutoff distance $\delta_c$ value based on the interactions of data points by fully considering the overall distribution especially the potential value representing the interaction dynamics of objects (data points). The distribution of the data field can be described as a potential function, and the potential value of an arbitrary point $x_i \in \Omega(1 = 1, 2, \ldots, n)$ in the data field is defined as (7):

$$\psi(x_i) = \sum_{j=1}^{n} (m_j \times K(\frac{||x_i - x_j||}{\sigma})), \hspace{1cm} (7)$$

where $m_j$ ($m_j > 0$, $\sum_{j=1}^{n} m_j = 1$) is the mass of $x_j$ and it represents the strength of the data field from $x_i$, $||x_i - x_j||$ is the distance between point $x_i$ and point $x_j$, $\sigma$ is an impact factor

![Fig. 1. Data point distribution. Data points are ranked in order of decreasing density and $d_c$ is the radius of the influence area of data points. When the cutoff distance $d_c$ is too large, such as $d_c > \max(d_{ij})$, all $\rho_i$ are the same, and all the data points will be identified as one cluster. On the contrary, when the cutoff distance is too small, such as $d_c < \min(d_{ij})$, all the data points will be regarded as the cluster centers and divided into a single cluster. Data points 26, 27, and 28 have a relatively high $\delta$ and a low $\rho$, each considered as an individual cluster consisting of a single data point, i.e., an outlier.](image-url)
that controls the interaction distance between points, and $K(x)$ is a unit potential function. Clearly, the potential function directly measures the density of data distribution. The potential is strong in a data-intensive area and weak in a data-sparse area [32]. $K(x)$ is assumed to be a Gaussian kernel function as it is ubiquitous and well matches the data nature of the data field [33].

The Gaussian potential of point $x_i$ is defined as (8):

$$\varphi(x_i) = \sum_{j=1}^{n} e^{-\frac{\|x_i - x_j\|^2}{\sigma^2}}.$$  

(8)

The definition of $\rho_i$ is similar to the Gaussian potential of point $x_i$. As shown in Fig. 3, both the local density and potential match the distribution of original dataset. The distribution of the potential value is determined by impact factor $\sigma$, as the local density is determined by $d_i$. Hence, the optimal value of $d_i$ can be solved in the same manner as the optimal $\sigma$.

The Gini index measures the uncertainty of systems associated with random variables and is positively related to uncertainty. For a data field, the Gini index (defined as (9)) measures degree of uncertainty, that is, how nonstationary the dataset under examination is. When the potential values of all data points are the same, the corresponding uncertainty reaches its maximum with the greatest Gini index obtained; in contrast, the smallest Gini index denotes that the data field is the most unbalanced and clustering of the dataset should be the most straightforward.

$$Gini = 1 - \sum_{j=1}^{n} \left( \frac{\Phi_j}{\sum_{i=1}^{n} \Phi_i} \right)^2.$$  

(9)

The Gini index then manifests a nonlinear univariate function with impact factor $\sigma$, and the optimal $\sigma$ can be obtained when the Gini index is minimized. The potential distribution of the data field is then said to best match its distribution [34]. Therefore, the objective function with respect to the impact factor $\sigma$ can be denoted as (10):

$$\hat{\sigma} = \arg \min_{\sigma} Gini(\sigma) = 1 - \sum_{j=1}^{n} \left( \frac{\Phi_j(\sigma)}{\sum_{j=1}^{n} \Phi_j(\sigma)} \right)^2.$$  

(10)

The objective function of the cutoff distance $d_i$ can be obtained as (11). Clearly, the optimal value of $d_i$ is the same as that of $\sigma$.

$$\hat{d}_i = \arg \min_{d_i} Gini(d_i) = 1 - \sum_{j=1}^{n} \left( \frac{\rho_i(d_j)}{\sum_{j=1}^{n} \rho_i(d_j)} \right)^2.$$  

(11)

Obviously, the solution of (11) has a fair initialization that is to the optimal cutoff distance $\hat{d}_i$ when the initialization is set to the first 1%–2% of the $d_i$ sequence [16]. ADPC extends the PSA to derive a solution to this optimization problem, which iteratively drives the objective function to the optimum. The conventional PSA decides (in the current step $x_k$) the direction of the next search (step $x_{k+1}$) only in comparison with the value of the objective function in the current step, that is, $f(x_k)$, which may result in premature convergence. The improved algorithm then refers to $m$ ($m > 1$) steps to reduce the risk of falling in the local optimum: $(f(x_{k+1}), f(x_k), f(x_{k−m}), \ldots, f(x_0))$. Therefore, the rule of step 2 in Algorithm 2 is changed to:

$$f(y_i \pm d_{e_j}) < \max_{m \in [k−m,0]} f(y_i).$$  

(12)

When $m = 0$, it degenerates into the original comparison method.

The main purpose of introducing an extended PSA is to quickly search $\min(Gini)$ and the corresponding $\hat{d}_i$, which makes the search more efficient than the traversal method. Although some extra hyperparameters have been introduced, their effect on the final clustering results is much lower than that of $d_i$. Among them, only the acceleration and contraction coefficients can directly influence the search process speed. For the comparison

\begin{figure}[h]
  \centering
  \includegraphics[width=\textwidth]{fig2.png}
  \caption{$\gamma$ is ranked in descending order to construct set of cluster center candidates.}
\end{figure}

\begin{figure}[h]
  \centering
  \includegraphics[width=\textwidth]{fig3.png}
  \caption{Both local density and potential can match the distribution of original dataset.}
\end{figure}
data points with large need for manual operations. The DPC selects cluster centers from accurate cluster centers based on the dataset itself without the becomes difficult. ADPC then aims to automatically determine categories scenarios in which human participation does not apply or circles can be identified as cluster centers. This also compli-
cation, the direct setting of the original factor, setting it to be a smaller integer (i.e., 1,2,3) is less difficult and influential than the direct setting of the original $d$, as shown in Fig. 4, there is only one extremum of $\text{Gini}$, at which the comparison coefficient $m$ has little influence. Overall, these hyper-parameters introduced are easier to set and have a lower direct impact on the results.

3.2.2. ADPC: Determination of cluster centers

The DPC requires visual inspection of the Decision Graph to manually set the cluster centers. This is particularly risky when handling complex Decision Graphs grows complex. For example in Fig. 5, difficult to determine whether the data points in the circles can be identified as cluster centers. This also complicates scenarios in which human participation does not apply or becomes difficult. ADPC then aims to automatically determine accurate cluster centers based on the dataset itself without the need for manual operations. The DPC selects cluster centers from data points with large $\rho$ and $\delta$, that is, a large $\gamma$. Following this loose principle, the key issue is to find the boundary ($\gamma_{\text{BD}}$) between the cluster center point ($\gamma_{\text{CC}}$) and the noncluster center point ($\gamma_{\text{NC}}$).

Fig. 5. An instance of decision graph.

Ranking $\gamma_i$ of all data points in descending order, as shown in Fig. 2, most of the $\gamma_i$’s concentrate on a low value, while those of few data points are large. For noncluster center points (approximately indexed 7 onward), the average value of $\gamma_i$ is close to $\gamma_i$ in the middle, and their $\gamma_i$’s descend almost linearly and slowly. When viewing the $\gamma_i$ values of data points through the boundary point ($\gamma_{\text{BD}}$) to a cluster center point ($\gamma_{\text{CC}}$), there exists a jumping point with $\gamma_i$. This observation provides a clue for identifying $\gamma_{\text{BD}}$ by scanning the $\gamma_i$ sequence. The consecutive $Z$ (odd positive) $\gamma_i$’s are selected from $[\text{seq}(\gamma_i)]_{i=1}^N$: $[\text{seq}(\gamma_i)] = [\text{seq}(\gamma), \text{seq}(\gamma_{j+1}), \ldots, \text{seq}(\gamma_{j+z-1})]$, and the mean value of $[\text{seq}(\gamma_j)]$ can be obtained:

$$\text{seq}(\gamma_j) = \frac{\sum_{i=k+1}^{N} \text{seq}(\gamma_{j+k-1})}{Z}.$$  \hspace{1cm} (13)

The $\gamma_i$’s of noncluster center data points can be expressed as formula Eq. (14):

$$[\text{seq}(\gamma_j)] \approx [\text{seq}(\gamma_{j+(Z-1)/2})].$$  \hspace{1cm} (14)

Consequently, the following formula can be obtained to determine whether $[\text{seq}(\gamma_j)]$ satisfies the linear trend:

$$\left| [\text{seq}(\gamma_j)] - [\text{seq}(\gamma_{j+(Z-1)/2})] \right| \leq \xi,$$  \hspace{1cm} (15)

where $\xi$ is a small positive value. For the sequence $[\text{seq}(\gamma_j)]$, there are two moving methods, forward and backward, to scan $[\text{seq}(\gamma_j)]_{i=1}$ to find $\gamma_{\text{BD}}$.  

![Fig. 4. The trend of Gini with different cutoff distance in dataset Spiral.](image-url)

**Algorithm 2: Extended Pattern Search Algorithm**

**Input:** Acceleration coefficient $\kappa \geq 1$, contraction coefficient $\beta \in (0, 1)$, convergence coefficient $\varepsilon > 0$, initial step-size $d$, ($d > \varepsilon$).

**Output:** Optimal solution: $x_k$.

1: Set the initial value $x_1 \in R^n$ and set $y_1 = x_1, k = 1, j = 1$;

2: The axial search:

   if $f(y_j + d\varepsilon) < \max_{\max(k-m,0) < j \leq k} f(y_j)$

   $y_{j+1} = y_j + d\varepsilon$, turn to step 3;

   else if $f(y_j - d\varepsilon) < \max_{\max(k-m,0) < j \leq k} f(y_j)$

   $y_{j+1} = y_j - d\varepsilon$, turn to step 3;

   else set $y_{j+1} = y_j$;

3: if $(j < n)$

   $j := j + 1$, turn to step 2.

   if $(f(y_{n+1}) < f(x_k))$

                 turn to step 4;

   else turn to step 5;

4: The pattern search:

   set $x_{k+1} = y_{n+1}, y_1 = x_{k+1} + \kappa(x_{k+1} - x_k)$.

   $k := k + 1, j = 1$ and turn to step 2.

5: if $(d < \varepsilon)$

   End the search and output point $x_k$;

   else

   Set $d = \beta d, y_1 = x_k, x_{k+1} = x_k$.

   Set $k := k + 1, j = 1$ and turn to step 2.
When using the forward moving method, sequence \( \{ \text{seq}(y_j) \} \) is constructed from \( y_0 \) and moves forward one point each time. In each movement, we observe whether the sequence \( \{ \text{seq}(y_j) \} \) satisfies in formula (15). Once sequence \( \{ \text{seq}(y_j) \} \) satisfies the above inequality relationship, the movement stops and the \( y_j \) in \( \{ \text{seq}(y_j) \} \) is regard as the \( y_{90} \).

When using the backward moving method, sequence \( \{ \text{seq}(y_j) \} \) is constructed from \( y_9 \), which is in the middle of \( \{ \text{seq}(y_j) \}_{1}^{n-1} \) and moves backward by one point each time. Similarly, we observe whether the sequence \( \{ \text{seq}(y_j) \} \) satisfies the linear trend in each movement. On the contrary, it satisfies the inequality relation at the early stage of backward movement. Therefore, when the sequence \( \{ \text{seq}(y_j) \} \) does not satisfy the inequality relation for the first time, the corresponding \( y_j \) in \( \{ \text{seq}(y_j) \} \) is regard as \( y_{90} \). After finding \( y_{90} \), the data points satisfying \( y_j \geq y_{90} \) are considered as candidate cluster centers. Then the set of cluster center candidates is constructed for fine-tuning.

It is worth noting that the value of \( y_0 \) can be set to the first 50% point of \( \{ \text{seq}(y_j) \}_{1}^{n} \). The reason for this setting instead of starting from the last point is for fewer movements and comparisons. Under normal circumstances, the number of clusters in the data set is not very large, so there is a small probability to cause the omission of cluster centers. Meanwhile, this setting is not mandatory, and \( y_0 \) can also be located further back in \( \{ \text{seq}(y_j) \}_{1}^{n-1} \).

In addition, parameter \( Z \) determines the length of \( \{ \text{seq}(y_j) \} \) and can be set to an odd number above 3 (e.g. 5, 7). Different ways of moving correspond to different ways of setting. When using the forward moving method, \( Z \) can be set to slightly smaller than \( Z \) to reduce the risk of ending the movement early when \( \{ \text{seq}(y_j) \}_{1}^{n-1} \) corresponding to the dataset with many true cluster centers showing a downward linear trend before \( y_{90} \). When using the backward moving method, \( Z \) is slightly smaller. \( Z \) will be more suitable, because it can reduce the risk of a small jump at the \( y_{90} \) resulting in a small \( \{ \text{seq}(y_j) \} \) value transformation and then continue looking forward for \( y_{90} \). Even if \( Z \) is not set appropriately, which results in the inaccuracy of the candidate set, the subsequent fine-tuning process will correct this to ensure that an accurate cluster center is obtained.

Fine-tuning checks extra-set competitors against intra-set candidates in terms of independence. Given \( L \) candidates in the initial set, the difference in \( \gamma \) between the \( j \)th data point and its rightward neighbor is \( D_j \); the average value of \( D \) among the candidates is \( D_o \).

Starting from the \( L \)th \( D \), ADPC compares \( D_j \) \((j > L)\) with \( D_o \) in order: Once \( D_j \geq D_o \) is satisfied, the competitor will be filled into the candidate set and \( D_o \) updates; the number of comparisons to be made, \( T \), can be set to 5. Next, the ADPC filters out the unqualified candidates based on the tendency. According to the fundamental assumption of cluster centers regarding relative distance, their values are much larger than the cutoff distance \( d_c \), which are said to conform to the restriction:

\[
\delta_{CC} \geq d_c. \tag{16}
\]

Moreover, when there are multiple center candidates in a region with high density, they are usually very close to each other. Therefore, it is necessary to determine whether these center candidates can be identified as the final independent cluster centers, which determines whether the clusters after subsequent allocation should be merged or separated. Thus, all cluster centers complete the screening process by comparing the cutoff distance and the shortest distance \( \text{dist}_{\text{min}} \) of the center candidates. When \( \text{dist}_{\text{min}} \) is less than \( d_c \), the center candidate is filtered out. Thus, all the candidates satisfy (17).

\[
\text{dist}_{\text{min}} > d_c. \tag{17}
\]

Ultimately, the final actual cluster centers are determined accurately from the set of candidates using the fine-tuning process.

### 3.2.3. Complexity analysis

Suppose that the dataset contains \( n \) data points and let \( C \) denote the number of clusters. In the process of ADPC, there are three main parts that require storage spaces: first, the distance matrix needs space to store and the space complexity is \( O(n^2) \). Second, ADPC needs space to store \( d_i \), which is \( c_i n^2 \) entries, where \( c_i \) represents a constant. Third, each point has four attributes: \( \rho \), \( \delta \), \( Gini \), and \( \gamma \), which need \( 4n \) spaces. Thus, the overall space complexity of ADPC is \( O(n^2) \). As for DPC, the distance matrix and the two attributes (\( \rho \) and \( \delta \)) of each point need to be stored, while the value of \( d_i \) is only an entry. Furthermore, as an improved algorithm of DPC, the space required by DPC-KNN [24] includes three parts. The first two parts are consistent with DPC, and the latter part, which is used to store \( d_i \), is the same as the second part of the ADPC. Thus, the overall space complexity of the three algorithms can be written as \( O(n^2) \).

The time complexity of ADPC is mainly derived from the following five aspects: (a1) the time complexity for computing \( \rho \), \( \delta \) and \( Gini \) for each data point \( i \) are all \( O(n^2) \), and (a2) the time complexity of the extended PSA algorithm is \( O(\log(n)) \). Therefore, the time complexity of obtaining \( d_i \) is \( O(n^2\log(n)) \). (a3) The time complexity of \( d_i \) and \( \gamma \) depends on the sorting algorithm, the minimum \( O(n\log(n)) \), and the maximum \( O(n^2) \), so the total complexity of this aspect does not exceed \( O(n^3) \); (a4) the time complexity of fine-tuning in the determination of the cluster center module is \( O(n) \), and (a5) the time complexity in the data point allocation process is \( O(C \ast n) \). Therefore, the overall time complexity of ADPC is \( O(n^3\log(n)) \).

Compared with ADPC, the time complexity of DPC involves two aspects: (b1) the time complexity for computing \( \rho \) and \( \delta \), and (b2) the time complexity in the data point allocation process. In addition to (b1) and (b2), DPC-KNN still has the same time complexity (a3) for finding the k-nearest neighbors by sorting. As for determining the cluster centers, this does not apply to the manual operations for DPC and DPC-KNN. The overall time complexity for the both is \( O(n^2) \). The ADPC has an overhead in solving (11), whereas DPC and DPC-KNN are not involved, but this is obviously tolerable.

### 4. EEG exploratory analysis framework

A framework for exploratory EEG analysis was then designed based on ADPC, to find abnormal neural activities from pathological EEG without explicit a priori knowledge of the subjects under examination as the available to classification models. Fig. 6 provides an overview of the framework, which consists of three major modules: (1) time–frequency transformation with DFT, (2) EEG feature extraction with the Bayesian factorization approach (BF), and (3) clustering of EEG states.

EEG is first evenly segmented into a sequence of EEG epochs or samples in the context of machine learning; Spectrum information of the EEG samples is obtained via DFT; The resulted EEG tensor (three dimensions of sample-channel-frequency) is then factorized via BF to extract factor features; The distance matrix of the features forms the inputs for ADPC.

### 4.1. Extraction of EEG factor features

The framework first performs time–frequency transformation to EEG samples and obtains the frequency information with the Hamming window applied to avoid truncation. The EEG tensor is then formed with latent features extracted by BF, which excels with the merit of no demand for sufficient a priori knowledge of
the problem domain and the capability to shift the most informative factors of EEG. The linear model for tensor decomposition is expressed as follows:

\[ Y = X + \epsilon, \]

\[ X = \sum_{r=1}^{R} U^{(1)} \circ \cdots \circ U^{(N)}, \tag{18} \]

where \( Y \) is the \( \mathcal{N} \)-order tensor of size \( I_1 \times \ldots \times I_N \), composed of the true tensor \( X \) and the noise tensor \( \epsilon \). \( U^{(n)} \) represents the \( n \)-mode factor matrix of size \( I_n \times R \) with the \( r \)th column \( U^{(n)}_r \) and the positive integer \( R \) is referred to as the tensor’s rank. The operator \( \circ \) denotes the outer product.

BF assumes non-informative prior of the factor vector \( U^{(n)}_r \), and each element in noise tensor \( \epsilon \) both conform to i.i.d. Gaussian distribution. The probabilistic model adapts an approximate inference under the variational Bayesian framework for the posterior distribution. Then the posterior distribution of the factor matrix is obtained when the lower bound of marginal likelihood \( p(Y) \) is maximized and the factor matrices for all modes are obtained by the mean of posterior distributions [35].

### 4.2. Clustering EEG states

ADPC-based state recognition is at the core of the framework. Taking 3-order tensor (sample-channel-frequency) as an example, the most significant features obtained by the BF approach are depicted in the mode of sample, frequency and channel. The factor matrix of sample mode \( U^{(s)} \) includes the dissimilarity among the EEG data samples. Clustering the factor features of the EEG sample mode holds the power to explore and identify the pathological states of the brain’s cerebral electrophysiological activities. ADPC is applied for grouping EEG states from low-dimensional sample factors. The distance matrix of the sample features is calculated to be the input of the ADPC and the clustering results obtained can be used for exploratory analysis of pathological EEG. The detected abnormalities may then be concentrated for further examination with more sophisticated analytics in neuroscience and engineering tasks.

### 5. Performance evaluation

This study performed benchmarks of ADPC on public datasets against the mainstream counterparts. The testbed for the experiments was a PC equipped with CPU (Intel (R) i5-7500@3.4 GHz), RAM (16 GB), and OS (Windows 10).

Benchmarks were intended to test the capability of ADPC to recognize clusters of arbitrary shapes on the public datasets available at http://cs.uef.fi/sipu/datasets/. Table 1 described the details of these synthetic datasets significantly varying from each other in size and number of clusters. Four state-of-the-art clustering algorithms were checked against three types: (1) classic clustering: K-means and DBSCAN, (2) the original DPC algorithm, and (3) the improved DPC algorithm: DPC-KNN [24]. The Euclidean distance was used to calculate the distance in the clustering process for all algorithms.

The performance of these clustering algorithms was evaluated using two widely applied metrics: the adjusted rand index (ARI) and adjusted mutual information (AMI).

- **ARI** could be regarded as a modified metric of Rand index (RI) with higher discrimination. The value ranges of ARI and AMI are both \([-1,1]\), and the larger ARI-implied clustering results were more consistent with the real data distribution:

\[ ARI = \frac{RI - E(RI)}{\max(|RI|) - E(RI)}, \tag{19} \]
Themainobservationsfromtheresultswereasthefollows:

- k is computed as a percentage (p) of the number of datapoints N, so k = p × N. This parameter significantly influenced the performance of DPC-KNN, and no solution to automating the setting was available but enumeration verification.

| Dataset      | K-means | DBSCAN | DPC-KNN |
|--------------|---------|--------|---------|
| Flame        | k = 2   | eps = 1.56, MinPoints = 14 | p = 1.0% |
| Spiral       | k = 3   | eps = 1.20, MinPoints = 2 | p = 2.5% |
| R15          | k = 15  | eps = 0.40, MinPoints = 10 | p = 0.5% |
| Aggregation  | k = 7   | eps = 1.50, MinPoints = 9 | p = 1.0% |
| S1           | k = 15  | eps = 0.04, MinPoints = 26 | p = 1.0% |
| Unbalance    | k = 8   | eps = 0.03, MinPoints = 8 | p = 1.0% |

- AMI measured the degree of the consistency between data distributions based on the Mutual Information theory:

AMI = \frac{MI - E(MI)}{\max(H(U), H(V)) - E(MI)}

(20)

Suppose U and V are two distributions of N sample labels, then \( H(U) \) and \( H(V) \) are the entropies of the two distributions, respectively; \( MI \) is the mutual information between U and V. The value range of AMI is also \([-1,1]\], and a larger AMI indicated that the clustering results were more consistent with the real labels.

DPC and DPC-KNN both set \( d_s \) to 2% of \( d_{ij} \) in the dataset. ADPC sets the parameters of extended PSA in the process of clustering: acceleration coefficient \( \kappa = 2 \), contraction coefficient \( \beta = 0.5 \) and comparison coefficient \( m = 3 \). The search initialization and initial step size were set to 2% and 0.1% of \( d_{ij} \), respectively. In addition, Table 2 presents the details of the K-means, DBSCAN and DPC-KNN remaining relevant parameter settings.

Table 3 presented the benchmark results of the clustering algorithms in terms of ARI and AMI. All results were averaged across 20 iterative trials. The results indicated that:

- All algorithms performed well on datasets R15, S1, and Unbalance. This was also the case on Flame, Spiral, and Aggregation data sets (except K-means).
- ADPC, DBSCAN, and DPC-KNN outperformed DPC and K-means on Spiral dataset; ADPC’s AMI and ARI values on Flame, Spiral, and Aggregation datasets were all 1.
- ADPC outperformed DPC on all datasets except the Unbalance dataset.

The performance of DPC-KNN was close to that of ADPC, but its key parameter k should be set manually beforehand, where k is computed as a percentage (p) of the number of data points N, so k = p × N. This parameter significantly influenced the performance of DPC-KNN, and no solution to automating the setting was available but enumeration verification.

Fig. 7 presented the clustering results of ADPC in 2D scaling. The main observations from the results were as the follows:

- ADPC efficiently aggregated typical shape datasets (Flame, Spiral, and Aggregation), even though Spiral dataset brought challenges to the most clustering algorithms owing to its nonspherical shape. ADPC holds the potential to adapt to arbitrary shape clustering tasks.
- ADPC accurately identified the real clusters of all datasets, although the number of clusters was different. There were more than two clusters in all datasets except the Flame dataset, and the number of clusters in the R15 and S1 datasets were both 15. ADPC excelled in finding the correct number of clusters.
- The above datasets were of various sizes, and the performance of ADPC was not sensitive to the sizes at all.

6. Case study: Epileptic EEG exploration

The case study examined the potential of the ADPC-based framework to explore the pathological brain dynamics recorded in pathological EEG, without sufficient background information about the observations, that is, the exact problems with the subjects. It explored an epileptic EEG dataset to evaluate the effectiveness of singling out epileptic seizures, and these unpredictable and rare occurrences of electrical discharges in a focal area or the entire brain were very meaningful in the monitoring and diagnosis of epilepsy patients. The case study consisted of three stages: (1) blind exploration, (2) examination of abnormalities, and (3) verification.

This case study begins with blind exploration, which should complete the clustering-related task of blind state division to cater to the need to differentiate states of brain activities. The resulting clusters corresponding to different states were then be discussed in examination of abnormalities, where an in-depth examination was then performed on the detected abnormalities. Verification checked the credibility of the conclusions drawn in the case study against the ground truth. Note that this a priori knowledge was not involved in the first two stages but only the final verification.

The CHB-MIT scalp EEG dataset was used for this study, which was recorded simultaneously at 256 Hz with 916 h from 23 pediatric patients with severe epilepsy caused by organic lesions. This study examined the EEG data of 10 patients out of the 23 subjects as these consist of the same number of channels.

6.1. Blind exploration

EEG data were first divided into portions of 20 min each to view the evolution in a shorter duration rather than in a full time scale in hours. There was no overlap between the portions. For each portion, a sliding window with a length of 8 s (2048 data points) applied to segment it at a pace of 4 s. A 3-order EEG tensor (sample – channel – frequency) per portion was constructed after the DFT of EEG segments with a passband of 0–50 Hz, and factor matrices were constructed by means of BS as this method excelled in extracting the latent structural information. The 1-mode factor matrix for each portion (the sample factor matrix) was then clustered for blind exploration. Multiple clusters were obtained for most portions of each patient. For a small amount of them, two clusters were obtained. Through visual inspection, no informative observations were obtained for the former cases.

In sharp contrast, significant differences could be observed in the latter cases: typical clustering results for each patient are presented in Fig. 8. In these cases, most samples were concentrated in the green cluster. The samples in the red cluster were rare that is, approximately 6.5% in all the samples in a portion, compared with those in the green cluster. The highest proportion was 9% of patient chb04 and the lowest was 4.33% of patient chb07. By tracing the samples back to the time domain, it was found that the samples in the red cluster were a continuous segment with short duration. Note that the duration time of the samples in red cluster was variant, even for one patient. They formed in a similar salient pattern and showed abnormalities in the rest.
Table 3
Benchmark results on public datasets: AMI and ARI.

| Dataset     | Flame | Spiral | R15 | Aggregation | S1 | Unbalance |
|-------------|-------|--------|-----|-------------|----|-----------|
| Algorithms  | ARI   | AMI    | ARI | AMI         | ARI| AMI       |
| K-means     | 0.451 | 0.396  | −0.006 | 0.993     | 0.994 | 0.761 |
| DBSCAN      | 0.971 | 0.935  | 1.000 | 1.000       | 0.922 | 0.934 |
| DPC         | 1.000 | 1.000  | 0.703 | 0.982       | 0.986 | 0.998 |
| DPC-KNN     | 1.000 | 0.999  | 1.000 | 0.992       | 0.993 | 0.994 |
| ADPC        | 1.000 | 1.000  | 1.000 | 0.994       | 0.995 | 1.000 |

Fig. 7. Clustering results of ADPC on each dataset.

6.2. Examination of abnormalities

The samples in the red clusters were examined to determine the neurophysiological characteristics associated with the abnormalities. The key observations are as follows:

- As illustrated in Fig. 9, there is a significant difference between the samples in the two clusters. The amplitude of those in the red cluster was much larger with violent oscillations, which formed abnormalities in sharp contrast to the green cluster. In fact, this observation agrees well with the conclusions of ictal EEG samples vs. inter-ictal ones [43]. The abnormal samples in the red cluster were prone to ictal.

- Fig. 10 presents the characteristics of the samples in the red cluster in frequency and channel domains. In Fig. 10(a), there was a peak at a low frequency of approximately 2.5 Hz. In Fig. 10(b), the corresponding channel pair (F8-T8) in temporal lobe region exhibited significant neural activity. These observations provide key evidence that corresponds to epileptic seizures.

6.3. Verification

To verify the preliminary conclusions on the results obtained by the ADPC-based framework (green cluster: inter-ictal, red cluster: ictal), the labels obtained via clustering were verified against the labels (ground truth) provided by the experts with the original EEG dataset. Table 4 illustrates the confusion matrix for the sum of all patients, where Precision, Recall, F1-score, and Accuracy reached 100%, 92.49%, 96.10%, and 99.47%, respectively.

The clustering results for each patient are presented in Table 5. The Precision of each patient was 100%, indicating that all abnormal samples identified were ictal samples. The average Recall and F1-score of all patients’ clustering results reached 92.46% and
95.92%, respectively. Recall was affected by the uneven proportion of samples, and the number of ictal samples was small. The average results were calculated based on the number of patients. The results of verification indicate that the identification of the two clusters obtained in blind exploration is consistent with the preliminary conclusions in examination of abnormalities.

6.4. Comparison

Moreover, DPC-KNN was applied to complete the clustering task with the same EEG portions. However, the setting of parameter (p) affects the clustering results. As shown in Fig. 11 (a, b, c), when different p values were set, the clustering results were different and the result of p = 2% (F1-score = 81.82%) was better than those of p = 1% and 3% (F1-score = 76.19%) for chb08 patients.

In addition, the appropriate selection of cluster centers in DPC-KNN was not easy because of the manual determination via Decision Graph. As shown in Fig. 11 (d, e, f), the three points in the upper right corner all had the possibility of becoming the cluster centers, and different results were obtained by determining different cluster centers from the Decision Graph. The problem of ambiguous selection of cluster centers also emerged with patient chb03. As shown in Fig. 12, there were four close points in the upper right corner of the Decision Graph, which probably led to the selection of four cluster centers. Even assuming that the number of known clusters was 2, it remains unclear which two cluster centers to choose.
**Fig. 10.** Examination of abnormalities in frequency and channel domains.

**Fig. 11.** DPC-KNN Decision Graph and clustering results of patient chb08 and chb24.
7. Conclusions

Inspired by the urgent need in the field of clustering to automate the process of finding appropriate cluster centers and determining their scope of influence, this study fostered an adaptive solution (ADPC) based on the density peaks clustering theory. In particular, the solution aimed to tackle the challenges in the exploratory analysis of bio-signals using EEG as a case study.

ADPC enabled adaptive selection of the cutoff distance with an optimization function constructed with the Gini index to measure the uncertainty of the target dataset. An extended PSA was developed to obtain the optimal cutoff distance value. ADPC supported the automatic determination of the cluster centers. It automatically constructed a set of cluster centers by jointly ranking the local density and relative distance and then fine-tuning the set by balancing the intra-set independence and the tendency as a center against extra-set competitors.

Benchmarks on public datasets indicated ADPC’s superiority of adaptability to various datasets and effectiveness against the mainstream counterparts in terms of ARI and AMI. The case study of the ADPC-based framework on epileptic EEG indicated that (1) the framework achieved an average on Precision, Recall, and F1-score of 100%, 92.46%, and 95.92%, respectively, in seizure detection involving no priori information, and (2) the key observations revealed through clustering well match the expert’s conclusions.

ADPC enabled fully automated adaptive clustering based on the target dataset without the need for human intervention or an empirical setting of key parameters. The ADPC empowered the exploratory analysis of EEG, which was previously a well-known problem in the community of bio-signal processing. The resulting framework on its top could find abnormal neural activities from EEG without explicit a priori knowledge of the subjects under examination as the classification models do.

Overall, ADPC holds potential in exploratory analysis of bio-signals in general, as it operated independently in the course of data exploration.

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.

Acknowledgments

This work was supported in part by Scientific and Technological Innovation 2030, China (No. 2021ZD0204300), the National Natural Science Foundation of China (Nos. 62172304, 61977027), and Science & Technology Major Project of Hubei Province, China (Next-Generation AI Technologies, No. 2021BEA159). We would like to show our deepest gratitude to Ms. Siwei Chen for providing the original ideas for this work and our special thanks to Prof. Xiaoli Li, Mr. Lei Zhang, and Dr. Hengjin Ke for their assistance through all stages in this study.

References

[1] H.-P. Kriegel, P. Kröger, A. Zimek, Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering, Acm Trans. Knowl. Discov. Data (TKDD) 3 (1) (2009) 1–58.
