Discovering cluster evolution patterns with the Cluster Association-aware matrix factorization

Wathsala Anupama Mohotti1,2 · Richi Nayak1

Received: 13 March 2019 / Revised: 6 March 2021 / Accepted: 13 March 2021 / Published online: 9 April 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract
Tracking of document collections over time (or across domains) is helpful in several applications such as finding dynamics of terminologies, identifying emerging and evolving trends, and concept drift detection. We propose a novel ‘Cluster Association-aware’ Non-negative Matrix Factorization (NMF)-based method with graph-based visualization to identify the changing dynamics of text clusters over time/domains. NMF is utilized to find similar clusters in the set of clustering solutions. Based on the similarities, four major lifecycle states of clusters, namely birth, split, merge and death, are tracked to discover their emergence, growth, persistence and decay. The novel concepts of ‘cluster associations’ and term frequency-based ‘cluster density’ have been used to improve the quality of evolution patterns. The cluster evolution is visualized using a $k$-partite graph. Empirical analysis with the text data shows that the proposed method is able to produce accurate and efficient solution as compared to the state-of-the-art methods.

Keywords Cluster evolution · Text mining · Matrix factorization

1 Introduction

Text data, widespread in social media platforms and document repositories such as news broadcasting platforms and research publications, have emerged as a powerful means of communication among people and organizations [14]. Topics (or clusters or concepts) and associated terminologies in the text data change over time as well as across domains.1 For example, text repositories include the text data that are spanned across domains or/and time [2]. A social network data include users opinions expressed on diverse concepts over the time. Search engines are another popular internet medium that store (or index) a large collection.

---

1 This paper uses the terms topic, cluster and concept interchangeably.

Wathsala Anupama Mohotti
wathsalaanupama.mohotti@hdr.qut.edu.au; wathsala@dcs.ruh.ac.lk

Richi Nayak
r.nayak@qut.edu.au

1 Queensland University of Technology, 2 George Street Brisbane, Brisbane, Australia
2 Department of Computer Science, University of Ruhuna, Matara, Sri Lanka
It is useful for scholars, journalists, and practitioners of diverse disciplines to mine these data, spanned across domains/time, to find a trend with decaying, current and emerging concepts [16,17,19]. A term analysis tool such as Google Trends can track how the popularity of a term changes over time based on query log analysis [7]. With the rise in big data and the dependence amongst terms, it is appropriate to analyze the formation and evolution of topics instead of individual terms in a dynamic text corpora. Over the time, a cluster can go through the states of birth, death, split and merge indicating the persistency, growth and decay of concepts [15].

Tracking evolution across different domains provides insight on how the same concept has been used over the diverse domains. For example, consumer behavior is a well-known concept mainly used in the finance domain; however, it is also important for politics and agriculture domains. It is useful to identify how this concept is used in the agriculture domain to establish marketing strategies for businesses. Further, the trends showing through this concept dynamics in the politics domain will create opportunities for political parties and governments to mend their campaigns. Similarly, discovering cluster dynamics over the time in a specific field is useful for researchers to setup their publications, strategies and research. Further these trends provide insight for businesses and government to set up policies accordingly to succeed. Tracking of concepts over the domain/time can also provide insight to historians and social scientists to understand how a theory has evolved [26].

There exist myriad text cluster visualization tools and methods such as dendrograms, reachability plots, and different types of graphs [37]. Most of these methods show the connectedness among the cluster members and their neighbors. They are unable to represent all the cluster dynamics including birth, split, merge and death, and associated evolution patterns.

Only a handful of research studies exist that study topic/cluster evolution. Most of these methods only deal with identifying emerging or novel topics [22,23]. There are only a couple of studies that focus on identifying emerging, persistence and diminishing topics [10,15]. The method in [10] identifies some of these patterns by measuring how the term frequency changes over time; however, it is not able to track the individual state differences in topics such as split and merge. Also, it does not support visualization to illustrate the evolution. The method in [15] performs similarity calculation between clusters using overlapping terms in each consecutive time stamp, to visualize various states. However, it ignores the global evolution over the time and focuses only on adjacent time stamps to determine similarity. Other methods [28,31] assume a fixed structure of clusters over the time to identify their evolution. Consequently, they fail to consider new formations or the changes in the structure of the clusters.

This paper proposes a novel and accurate method of Cluster Association-aware matrix factorization for discovering Cluster Evolution, called CaCE. It can track four major lifecycle states of clusters, namely birth, split, merge and death, to discover their emergence, growth, persistence and decay. Input to CaCE is a set of clustering solutions where each clustering solution is generated for the dataset representing each time/domain. Using the set of clustering solutions, CaCE generates the inter-cluster and intra-cluster associations matrices with the terms in clusters. CaCE includes a novel Non-negative Matrix factorization (NMF)-based process to identify the groups of similar clusters. Specifically, inter-cluster associations—modeled with the number of overlapping terms between clusters—semantically assist matrix factorization for the cluster-group discovery. To separate less cohesive clusters from a cluster group, we introduce a novel notion of density based on uniform term frequency distribution within a group. Finally, we propose to use the bipartite graph to effectively visualize the cluster evolution as a progressive $k$-partite graph in a novel fashion across the $k$ temporal/domain
dimensions. The evolution is represented by drawing edges in a $k$-partite graph between consecutive time intervals if the clusters possess the same level of density and belong to the same group.

Empirical analyses with several document corpus, over the varying number of time stamps and the varying number of clusters, reveal that CaCE can discover cluster evolution accurately and efficiently compared to other state-of-the-art cluster/topic evolution methods.

Figure 1 shows an example of the evolution of clusters generated by CaCE. Consider a document collection in a university archive collected over four years. A clustering solution is generated for the corpus of each year. For example, a total of 3, 3, 3 and 2 clusters are identified in the datasets of Years 1, 2, 3 and 4, respectively. Each cluster in a clustering solution is represented by the top-n terms. CaCE is able to show the changes in clusters states over the years. For instance, clusters related to ‘Mathematics’ are shown as persistent over $t_1$ to $t_3$ periods (green linked). Further, ‘IT/Computer Science’ which is born in $t_2$ shows growth with a split in $t_3$ (blue linked). In contrast, ‘Archeology’ shows decay with a merge between $t_1$ and $t_2$ and shows its death at $t_2$, as no similar cluster is identified $t_3$ beyond. Timestamp $t_4$ shows emergence of a ‘Physics’-related cluster and a new cluster that varies from the existing IT/Computer Science clusters.

Similarly, in a set of clusters formed over different domains, CaCE can show evolution patterns of emergence (i.e., if a concept has not appeared in another domain), growth (i.e., spread across multiple domains), decay (i.e., merging amongst domains), and persistence (i.e., a concept is consistent in considered domains).

More specifically, this paper brings the following novel contributions to the area of cluster evolution:

- An NMF-based approach with inter-cluster and intra-cluster associations to identify the groups of similar clusters that are formed over the time or domains.
A term frequency-based notion of density to remove the loosely connected clusters in a cluster group.

- A progressive $k$-partite graph-based approach to display evolution of clusters in the cluster groups.

To the best of our knowledge, CaCE is the first method that considers the cluster association using an inter-cluster matrix built with overlapping terms for discovering cluster evolution.

The rest of the paper is organized as follows. Section 2 reviews related work and presents the motivation behind this research. Section 3 introduces the problem definition that is followed by the proposed CaCE method. Experiments are discussed in Sect. 4 with two real-world case studies. Final conclusion remarks are given in Sect. 6.

2 Related work

Text mining approaches that attempt to analyze the dynamic text over time can belong to cluster evolution discovery [15,17], topic evolution discovery [8,10,38] and event detection methods [20,30]. All these paradigms focus on tracking content shift and identifying emerging trends in dynamic text datasets, as shown in Fig. 2. These methods explore changes in cluster/topic structure over time through textual content associated with clusters/topics to characterize the evolutionary events, concepts or terms. In comparison with cluster evolution, topic evolution or event detection is done in much smaller data space (i.e., topic space). The number of extracted topics is much less in topic evolution than the entire document collection, and associated vocabulary with topic clusters in the collection is much smaller than the complete vocabulary of the collection. On the other hand, community evolution [25,28,31] tracks cluster dynamics, which considers user groups as clusters.

2.1 Clusters evolution discovery

Research in text-cluster evolution is infancy [15,17]. A survey-based research [17] was carried out to identify evolution of concepts in clusters of publications using bibliometric tools. This only considers the citation network in tracking evolution. TextLuas [15] models each cluster solution with the respective terms at each time stamp and considers similarity between consecutive clusters, as determined by the term intersections. It considers only the local
relations between two consecutive time stamps in defining evolution. In contrast, the proposed method CaCE globally identifies the cluster groups over a period and visualizes the evolution concerning all time stamps using a \(k\)-partite graph.

### 2.2 Topics evolution discovery

Topic modeling is another powerful paradigm for semantic analysis of large collections of documents [22,38]. Researchers have attempted to identify the evolution of topics in document collections using extensions of Latent Dirichlet Allocation (LDA) [6,11,18,38]. In [38], a probabilistic topic modeling approach is used to track the topic occurrence in different time dimensions with the calculated respective probabilities; however, it is found incapable of identifying topic evolution with splits and merge. Authors in [12] determine text cluster evolution based on the changes to term probability within topics. This proposal was limited by the fixed vocabulary constraint where only a general set of terms in the topics was studied and neglected the tracking of new topic formations. In [22], NMF is used to identify a set of steady topics through minimizing learning error and report emerging topics by filtering deviated topics. However, they fail to provide the complete insight of persistence, decay and growth.

Similarly, topic models have been used in understanding the topic dynamics across temporal dimensions in social media domain [8,10]. Authors in [10] extended the topic identification to track persistent and decay trends using the term frequency-based energy notion defined for each cluster solution. The density notion, used in CaCE to determine the consistent cluster groups, is inspired by the energy concept. These topic evolution methods are limited to identify the few states in cluster lifecycle. The process of identifying the complex dynamics of topics such as merge and split and detailing a complete cluster lifecycle is challenged by the lack of information in the text data due to the sparseness of text representation.

### 2.3 Event detection

Event detection methods have been applied in social media communication to find novel or trending events [20,30,40]. This stream of methods keeps track of event clusters (much smaller number than the clusters in original space) that appear across time to identify the novel events or shifts that are deviated from the existing event clusters. A novelty score is assigned to each event cluster to identify new events in a Twitter dataset considering tweet similarities [30]. Topic modeling has been first used in identifying a fixed set of events in a Twitter data across the time, and emerging events are then identified through deviations to previously existing events [40]. However, these researches are limited in tracking evolution and fail to identify growth and decay of clusters.

### 2.4 Community evolution

Researchers have studied the community evolution in social networks focusing on structural properties of communities [25,28,31]. In the area of network-based community detection, clusters consist of users instead of text as depicted in Fig. 2c. The snapshot model [39] considers different snapshots of a network at different time steps to find clusters of users and then track clusters over time in order to interpret their evolution. The majority of community detection methods assume a fixed number of communities across the time by disregarding
new formation and dissolution [31] or relying on a pre-determined community structure [28]. The temporal smoothness model [9] is used to analyze continuous stream of atomic changes to the considered network to derive communities over time. It can be considered similar to the fixed vocabulary constraint in some of the text evolution analysis methods [12].

2.5 Clustering visualization

Visualization methods such as dendrograms, reachability plots, unified distance matrices (U-Matrices), and graphs [37] are commonly used to visually encode the groupings and relations in clusters and their neighbors. Hierarchical clustering algorithms produce a tree-like structure which are visually encoded using dendrograms [13]. In dendrograms, data elements connected are perceived as part of the same cluster and show the merge and split of clusters within a clustering solution. OPTICS, a density-based clustering algorithm, sorts data points such that they can be visually encoded using reachability plots [37]. In this visualization, the formed valleys can be seen as clusters, valleys within valleys as clusters within clusters, and peaks as outliers. This technique focuses only on showing splitting of clusters and emerging patterns within outliers. A self-organizing map (SOM) is a type of neural network that is trained using unsupervised learning to produce a low-dimensional discretized representation of the input space network [34]. The output (i.e., the low-dimensional view) of a SOM can be visually encoded using a U-Matrix, which is a 2D arrangement of mono-colored hexabins. Proximity in elements can be shown by opacity color of bins where a high opacity shows dissimilarities [37]. Identifying cluster dynamics or evolution patterns is not a focus of this method. An incremental network model that can learn the important topological relations in a given set of input vectors is visualized using forced directed graphs [37]. These topologies are visually encoded in a 2D plot using a graph layout technique. Some methods visualize the evolution of time-series data to aid the understanding of anomaly detection [27].

All these cluster visualization techniques represent the links among items within a cluster or with their neighbors. They fail to track cluster dynamics within the cluster life cycle to identify evolution patterns in text data.

2.6 Challenges in text cluster evolution

Identifying similarity between clusters identified over each time/domain is fundamental to cluster evolution. Text clustering methods are challenged by the high-dimensional and sparse vector representation [1,3]. Non-negative Matrix factorization (NMF) which maps the high-dimensional data to a lower-dimensional space has been found to provide an effective solution by allowing to form clusters in the lower-dimensional space [24]. However, information loss is inevitable in this family of methods that may result in poor outcome [3].

There are a few recent studies that use additional information to assist sparse text clustering problem with additional semantic information [29,33,36]. They use word association relationships, Skip-Gram and Skip-Gram with Negative-Sampling (SGNS), similar to the notion of word embedding. The Skip Gram model is a training method for neural networks to learn neighbors or the context of a word in a corpus for word embedding [32]. In [36], the term × term association matrix modeled with SGNS is used to semantically assist the NMF in short text clustering for topic discovery. Negative sampling tries to maximize the probability of observed term pairs to be 1 and unobserved term pairs to be 0 within the term association matrix. Inheriting these notions to cluster evolution, we propose the use of SGNS to model the inter-cluster association using overlapping terms. We conjecture that by learning
Table 1  Summary of existing evolution detection methods

| Category               | Applied data domain | Major drawback                                                                 |
|------------------------|--------------------|--------------------------------------------------------------------------------|
| Cluster evolution      | Text data          | Neglect global evolution patterns and focus on only the consecutive time-stamps  |
|                        |                    | analysis [15]                                                                  |
| Topic/event evolution  | Text data          | Unable to identify complex cluster dynamics [10,22,38]                         |
|                        |                    | Focus on studying changes in the fixed set of terms and neglects new formations |
|                        |                    | [12,30,40]                                                                     |
| Community evolution    | Network data       | Focus on studying changes in the fixed set of structures and neglect new        |
|                        |                    | formations [9,28]                                                              |
|                        |                    | Assume a fixed number of communities over time [31]                            |
| Cluster visualization  | Many data types    | Only represents the link among items within a cluster or some of the links with |
|                        |                    | their neighbors [37]                                                           |

the context of terms, clusters can be accurately grouped together that share similar concepts and terms.

2.7 Summary

Table 1 summarizes the existing cluster, topic/event, community detection and visualization methods and their drawbacks in accurately identifying text cluster evolution. TextLuas [15] visualization of text cluster evolution comes closest to our proposed work. It considers cluster dynamics such as birth, death, split, and merge and represents a dynamic cluster timeline with a directed acyclic graph. However, its rendering process needs the inclusion of dummy nodes to adhere to the diagram layout, and later, they are removed before the diagram display. Secondly, TextLuas only shows connections of a cluster in a timestamp to its successive/predecessor time stamp based on Jaccard similarity between clusters. In contrast, CaCE utilizes the higher-to lower-dimensional mapping via matrix factorization to identify the cluster associations and track all of their states over the time/domains. Moreover, the proposed $k$-partite graph visualization follows a straightforward approach. Firstly, CaCE identifies cluster groups and thereby draws exactly existing cluster nodes in a particular timestamp with the connections based on the cluster groups and densities.

3 Cluster Association-aware matrix factorization method of cluster evolution (CaCE)

3.1 Preliminaries and problem definition

Consider a set of text data collections $\mathcal{D} = \{D_1, D_2, ... D_k\}$ over a time period $k$ or a set of $k$ domains. Let $\{t_1, t_2, ... t_s, ... t_k\}$ be the considered time period or a set of domains with $k$ consecutive occurrences. Let $\mathcal{C} = \{C_1, C_2, ... C_s, ... C_k\}$ be the set of respective cluster solutions where $C_i$ represents a clustering solution for the dataset $D_i \in \mathcal{D}$ for each occurrence.
Using a text clustering method of choice, a clustering solution $C_i = \{c_1, c_2, ...c_m\}$ with $m$ clusters is generated for each time-stamp/domain dataset $D_t \in D$ where $m > 1$. The value of $m$ can vary among the clustering solutions in $C$. As described in Sect. 3.4, the proposed NMF-based method determines the similarity between all $c_i \in C$ based on cluster associations and discover the latent relationships between all $c_i$ clusters. Each cluster $c_i$ is mapped to a cluster group. The resulting cluster groups show the set of clusters that belong to the same group which spans across different time/domains.

**Problem Definition:** Given a text data collection spanned across the time/domain and the set of clustering solutions $C$, CaCE aims to identify the cluster groups and show the cluster evolution over a period of time or domains.

**Definition 1** An individual cluster $c_i \in C$ at each time-stamp or domain $t_s$ holds a lifecycle state that reveals the evolution patterns. These states are:

- **Birth** if cluster $c_i$ that appears in time/domain $t_s$ does not have any similar cluster in time/domain $t_{s-1}$, it marks the birth of $c_i$;
- **Death** if cluster $c_i$ that appears in time/domain $t_s$ does not have any cluster that is similar in time/domain $t_{s+1}$, it marks the death of $c_i$;
- **Split** if cluster $c_i$ that appears in time/domain $t_s$ does have multiple similar clusters in time/domain $t_{s+1}$, it marks the split of $c_i$; and
- **Merge** if cluster $c_i$ that appears in time/domain $t_s$ does have multiple similar clusters in time/domain $t_{s-1}$, it marks the merge of $c_i$

Based on the cluster similarities between consecutive time-stamps or domains, the following evolution patterns are identified using the aforementioned Birth, Death, Split and Merge states:

- **Persistence** if cluster $c_i \in C_s$ has a similar cluster in each consecutive clustering solution until cluster solution $C_p$ where $p \leq k$, cluster $c_i$ will display a persistent evolution pattern within time/domain $s$ to $p$.
- **Growth** if cluster $c_i \in C_s$ has a gradual increase in the number of splits until the cluster solution $C_p$ where $p \leq k$, cluster $c_i$ will display a growth evolution pattern within time/domain $s$ to $p$.
- **Decay** if cluster $c_i \in C_s$ has a gradual decrease in the number of merges until the cluster solution $C_p$ where $p \leq k$, cluster $c_i$ will display a decay evolution pattern within time/domain $s$ to $p$.
- **Emerging** if cluster $c_i \in C_s$ has been born in time/domain $s$, it displays an emerging pattern in time/domain $s$.

### 3.2 Inter- and intra-cluster associations

Let the set of cluster solutions $C$ over the $k$ time-stamps or domains consist a total number of $N$ clusters $\{c_1, c_2, ..., c_N\}$ that contain a total number of $M$ terms $\{w_1, w_2, ..., w_M\}$. Let $w_{c_i} \in \{w_1, w_2, ..., w_M\}$ represent the terms in cluster $c_i$. We apply a minimal set of pre-processing steps, before applying CaCE to $C$, that include removal of stop-words as well as terms that occur very highly and lowly in $C$.

**Definition 2** **Intra-cluster Association** The term association in a cluster $c_i$, defined as intra-cluster association, is modeled by the representative terms $w_{c_i} \in c_i$ and their frequency in the space of total $M$ terms.
Let matrix $S$ represent the ‘intra-cluster association’ with term $\times$ cluster relationship modeling $N$ clusters with $M$ terms. The matrix $S$ is modeled with the traditional bag-of-words model with each entry showing the existence of a term in a cluster.

**Definition 3 Inter-cluster Association** The term association between a cluster pair $(c_i, c_j) \in C$, defined as inter-cluster association, is modeled by the common terms shared among two clusters, $w_{c_i} \cap w_{c_j}$ where $w_{c_i} \in c_i$ and $w_{c_j} \in c_j$.

Let the symmetric matrix $A$ represent ‘inter-cluster association’ with cluster $\times$ cluster relationship, with each entry showing a number of overlapping terms between pairwise clusters. Matrix $A$ is modeled with the Skip-Gram with Negative-Sampling (SGNS) [29] weighting to make the probability of presence of cluster association high.

The Skip-Gram model is a popular training approach for neural networks to learn distributed word representation [32]. The Skip-Gram model predicts neighbors or the context for a considered word in a corpus in comparison with the continuous bag-of-words model which uses context to predict the word [32]. The notion of negative sampling is used to maximize the probability of observed (word,context) pair to be 1 while minimizing the unobserved pairs to be 0 [36]. In [29], SGNS is proved to be equivalent to factorizing a (shifted) word correlation matrix. It shows that SGNS is implicitly factorizing a word-context matrix, whose cells are the point-wise mutual information of the respective word and context pairs. In CaCE, the inter-cluster association matrix modeled with SGNS semantically assists the NMF in accurately learning the context of the clusters. CaCE uses SGNS to maximizing probability $P(A = 1|c_i, c_j)$ for closely associated cluster pairs $(c_i,c_j)$ within the observed $k$ time stamps while minimizing $P(A = 0|c_i, c_j)$ for loosely associated cluster pairs $(c_i,c_j)$.

Matrix $A$ is represented as:

$$A_{c_i,c_j} = \log \left[ \frac{\# \{ w_{c_i} \cap w_{c_j} \} \times V}{\sum_{c_b \in C} \# \{ w_{c_b} \cap w_{c_i} \} \times \sum_{c_b \in C} \# \{ w_{c_b} \cap w_{c_j} \}} \right]$$

(1)

where $w_{c_i}$ and $w_{c_j}$ are a set of terms in cluster $c_i$ and $c_j$, respectively. $\# \{ w_{c_i} \cap w_{c_j} \}$ is the number of overlapping terms in cluster $c_i$ and $c_j$. $V = \sum_{(c_i,c_j) \in C} \# \{ w_{c_i} \cap w_{c_j} \}$ is the total number of overlapping terms among all the cluster pairs.

This modeling represents the inter-cluster similarity within each pair, normalized with term similarities within clusters over the time/domain. The entries of $A$ with less than 0 are converted to zero to minimize the probability of unobserved pairs after taking logarithm as in Eq. (1).

### 3.3 Overview of CaCE

CaCE includes three main phases for discovering cluster evolution as depicted by Fig. 3. Suppose each documents set (as per time/domain) in the corpus is clustered into groups using a text clustering method of choice.

1. CaCE firstly uses NMF to identify the groups of similar clusters within this set of clusters $C$ using the inter- and intra-cluster associations. The lower rank in NMF that is used for identifying factor matrices represents the number of cluster groups. We align the rank value with the number of cluster dynamics (i.e., 4, more details in Sect. 4.7: sensitivity analysis). In CaCE, intra-cluster association information assists the factorization of inter-cluster association information. Two factor matrices are learned, each representing the cluster group assignment. At the end of factorization process, both matrices are combined.
to form a single cluster group assignment matrix. Each cluster is then assigned to a unique group based on the maximum likelihood coefficient that shows in this final matrix containing cluster group information. This allows identifying similar clusters within the cluster solutions $C$.

2. Secondly, the cluster groups mapping is refined using the novel notion of \textit{density} based on uniform term distribution within a group (detail in Sect. 3.5). A cluster with insufficient density value is excluded from the group, indicating that the cluster does not share enough matching terms within the group to be a member of that group. This allows the cluster groups to be tightly cohesive based on the common terms that they share.

3. Finally, CaCE visualizes the global cluster evolution patterns of emergence, persistence, growth and decay across time using a $k$-partite graph where nodes represent clusters and edges represent relationships between clusters such as persistence, split and merge considering cluster groups. A cluster evolution with all state changes of a cluster lifecycle (i.e., birth, death, split and merge) can be tracked with this visualization as given in Fig. 1 example.
3.4 Cluster Association-aware matrix factorization

3.4.1 Matrix factorization

The aim of CaCE is to identify the clusters’ evolution showing the trends detailing how the groups of terms have evolved over the time/domain. The first step in CaCE is to identify the groups of common clusters in the ‘intra-cluster association’ matrix $S$ representing term $\times$ cluster. Due to its high-dimensional sparse nature, we propose to approximate it into a lower-dimensional space and identify common sub-groups. NMF, which takes fewer parameters and produces coherent topics compared to other popular dimensionality reduction methods such as LDA [5], is used in this approximation.

In traditional NMF [1], $S \in R^{M\times N}$ is approximated by learning $W \in R^{M\times g}$ and $H \in R^{N\times g}$ where $g$ is the number of cluster groups, as follows:

$$ S \approx WH^T $$

In CaCE, we also propose to utilize the latent information represented within the inter-cluster association matrix $A \in R^{N\times N}$ during approximation of $S$. We conjecture that by including both $S$ and $A$, we introduce co-clustering in CaCE. In other words, we find commonalities among the terms based on the clusters in which they appear as well as we find commonalities among the clusters based on the terms they share.

The symmetric NMF [21] is applied to $A$ for generating two commutative matrices, $H_C \in R^{N\times g}$ and $H \in R^{N\times g}$ where $g$ is the number of cluster groups, as follows:

$$ A \approx HH^T_C $$

3.4.2 CaCE objective function

CaCE uses both these learning processes to simultaneously discover the cluster groups, as defined in the following objective function:

$$ \min_{W, H \geq 0} \| S - WH^T \|_F + \min_{H, H_C \geq 0} \| A - HH^T_C \|_F + \alpha \| W \|_1 $$

The intra-cluster association matrix $S$ and inter-cluster association matrix $A$ are approximated with the minimum learning error. We introduce $L1$ regularization on factor matrix $W$ to promote sparsity, control over-fitting and highlighting the distinguishing terms. This can be considered as the sparse dictionary learning, which models the sparse input data representation using only a few (important) terms of the dictionary learned from the data itself [4]. Prior research on traditional NMF has found this constraint to be effective for detecting deviations or novelty in text data [23]. We conjecture that this constraint will be able to discriminate cluster groups more effectively.

3.4.3 Solving the optimization problem

We propose to use the Block Coordinate Descent (BCD) algorithm [24] to optimize the objective function in Eq. (4). BCD divides the matrix members into several disjoint subgroups and iteratively minimizes the objective function with respect to the members of each subgroup at a time. It relies on the most recent values of the members for solving sub-problems related to their updates. When solving sub-problems that depend on each other, they must be computed sequentially to make use of the most recent values for BCD.
CaCE solves these interdependent sub-problems sequentially starting from $W$. At first iteration, $W$ is initialized with random values. The BCD update rule is used for finding $W$ in the NMF optimization using $S$ and an initial randomly generated matrix $H$. Matrix $H$ is then updated using the current values of $W$ and $H_c$. Next, $H_c$ is updated using $A$ and the most recent values of $H$. This enables the factorization process to include both inter- and intra-cluster associations. This is done for each $g' \in g$. Algorithm 1 in Fig. 6 shows the factorization process of CaCE.

\[
W_{(i,g')} \leftarrow W_{(i,g')} + \frac{(SH)_{(i,g')}}{(HTH)_{(i,g')}} - \left( WHTH \right)_{(i,g')}
\]

(5)

\[
H_{(i,g')} \leftarrow H_{(i,g')} + \frac{(STW)_{(i,g')}}{(WTH)_{(i,g')}} + \left( HTWH \right)_{(i,g')} - \frac{(HHTH)_{(i,g')}}{(WTH)_{(i,g')}} + \left( HTHH \right)_{(i,g')}
\]

(6)

\[
Hc_{(i,g')} \leftarrow Hc_{(i,g')} + \frac{(AH)_{(i,g')}}{(HTH)_{(i,g')}} - \left( HcHTH \right)_{(i,g')}
\]

(7)

The factorization process generates two perspectives of cluster $\times$ group matrices $H$ and $H_c$. This lower rank approximation of higher-dimensional cluster $\times$ term matrix gives a dense representation compared to original. It is generated with high co-occurrences so that clusters that share common terms can appear in the same group. CaCE forms a final cluster group matrix $H^F$ based on the maximum pairwise coefficient of $H$ and $H_c$, overcoming the learning process of each single perspective.

\[
H^F = \max (H, H_c)
\]

(8)

The final cluster assignment vector $h^f$ is defined using the hard cluster assignment policy. A cluster group that possesses the highest coefficient within $H^F$ is used as the group for a specific cluster.

\[
h^f = \arg\max \sum_{i=1}^{g} \left( H^F_{(i,i)} \right)
\]

(9)

**Running Example:** Let us consider the example of a set of document collections in a university archive with four cluster solutions, $C_1$ - $C_4$ (Fig. 1). Assume this corpus has a vocabulary of 29 terms, and it consists of a total of 11 clusters ($c_1$-$c_{11}$) over the 4 time-periods. An initial matrix $A$ representing inter-cluster associations using the number of overlapping terms is derived as shown in Fig. 4a, b which depicts $A$ with SGNS modeling. Matrix $S$ that represents intra-cluster association can be derived as Fig. 4c. CaCE sets four as the lower-rank for matrix factorization to identifying four cluster groups aligning with four cluster dynamics. The NMF process with input matrices $A$ and $S$ generates factor matrices $H$ and $H_c$ that forms $H^F$ as given in Fig. 5a. Applying hard clustering on $H^F$, the final cluster groups are assigned as in Fig. 5b.
Discovering cluster evolution patterns...

3.5 Cohesive cluster groups based on term density

The above matrix factorization process forces each cluster in a cluster solution to be included in a cluster group. This may result in loosely connected clusters to reside within a cluster group due to the fewer terms shared with others in the group. To handle this, we propose the density notion that determines the strength between a cluster and its associated cluster group considering the term frequencies. More specifically, the density value of a cluster $c_i$ is defined as the ratio of the term frequencies within the cluster using each term $w_j \in c_i$ to
the maximum term frequency of the corresponding cluster group $g_z \in g$, as follows:

$$\text{Den}_{c_i} = \frac{\sum_{j=1}^{\mid w_{ci} \mid} \text{tf} \left( w_j \right)}{\max \left[ \forall g_x : \text{tf} \left( \sum_{j=1}^{\mid w_{cx} \mid} w_j \right) \right] \times \mid w_{ci} \mid} \quad (10)$$

Density values that fall within first quantile (‘mean–standard deviation’) within a group implies the clusters with least densities. CaCE uses this threshold to separate the loosely connected clusters ensuring uniform term distribution within a group. This allows identification of a set of cohesive cluster groups over the time. A cluster that receives the density value less than the set threshold is considered ‘inconsistent’ and its density value is set to zero. A cluster with zero density value is indicated as a new singleton cluster group within the visualization step.

**Running Example:** The cluster group labeled as 1 (i.e., IT/Computer Science cluster group) in Fig. 5b shows less cohesiveness with the inclusion of cluster 10 (c10) that contains terms such as ‘computer simulation for hypothesis’ (as shown in Fig. 1). Cluster 10 shows a low density value of 3 as compared to other clusters that show a value of 5 or 4. Accordingly, the density of cluster 10 falls within the first quantile of the density distribution of that group (i.e., <3.29).

### 3.6 Visualization of cluster evolution with a $k$-partite graph

CaCE proposes to visualize all cluster dynamics including birth, death, split and merge within a $k$-partite graph. The set of clusters within the cluster solution at time $t_s$ is represented with the respective partite $s$, and each distinct cluster group across $k$ partite is uniquely identified with a color code. Each cluster in the $s > 1$ partite in the graph is compared with each cluster in its predecessor partite to add edges between two clusters to mark them as similar if they belong to the same group.

A cluster pair in two successive partites should have an edge between them for being similar if (1) they belong to the same group and (2) either both of them possess zero density values or both of them possess nonzero density value.

In contrast, a cluster pair in two successive partites should not have an edge between them if only one of them has zero density value, even though they belong to the same group.

The color code of the cluster with nonzero density is updated with a non-existing color in the graph to separate it from the current cluster group. However, a cluster pair in unsuccessive partites is not considered to have an edge between them.

This process continues in an incremental manner to represent the cluster evolution spanned across time $t_1$ to $t_k$. The $k$-partite graph allows CaCE to identify birth, death, as well as growth and decay patterns in clusters, within multi-step periods through colors and edges.

Application of Definition 1 on the drawn edges identifies the corresponding patterns:

- A cluster that appears in time $t_s$ ($s \leq k$) that does not have any edge to a cluster in time $t_{s-1}$ marks the birth of that cluster which represents an emerging pattern.
- A cluster that appears in time $t_s$ ($s < k$) that does not have any edge to a cluster in time $t_{s+1}$ marks the death of that cluster.
- A cluster that appears in time $t_s$ ($s < k$) with multiple edges to clusters in time $t_{s+1}$ marks the split of that cluster showing a growth pattern.
- A cluster that appears in time $t_s$ ($s \leq k$) with multiple edges to clusters in time $t_{s-1}$ marks the merge of that cluster showing a decay pattern.
Discovering cluster evolution patterns…

Fig. 6 Algorithms of CaCE

A cluster born in time $t_s (s < k)$ and continues across the time with a single edge to succeeding time stamp $t_{s+1}$ shows a *persistent* pattern,

This is further assisted by the colors to uniquely identify the similar clusters that belong to the same group. Algorithm 2 in Fig. 6 shows the visualization of clusters evolution in CaCE using $k$-partite graph.

**Running Example:** Figure 1 illustrates the clusters evolution in example dataset. It shows clusters in the same group with unique color. Also it shows all the cluster dynamics including birth, death, split, and merge. Thereby, it shows the persistent evolution pattern of Mathematics, growth of IT/Computer Science, decay of Archaeology, and emergence of Physics. Additionally, it depicts emergence of the Computer Simulation cluster with a different color which shows a density variation with the included terms in IT/Computer Science cluster.
4 Empirical analysis

4.1 Experimental objectives

We evaluate three phases of CaCE to show its effectiveness. (1) The quantitative comparison against baselines using ground-truths evaluates the 1st phase of CaCE, measuring how effective is CaCE with inter-cluster association for identification of cluster groups. (2) The impact of 2nd phase with ‘density’ in obtaining cohesive cluster groups is evaluated with and without using the density notion quantitatively. (3) The 3rd phase, which shows the evolution patterns of clusters through edges on the $k$-partite graph visualization, is compared against baseline methods that are able to visualize the cluster evolution qualitatively.

Further, we compare the time efficiency and computational complexity of CaCE against different cluster groups identification methods as detailed in Sect. 4.6. Other different notions used in the proposed method together with the parameters/thresholds are analyzed in the sensitivity analysis section.

Finally, we conduct two case studies to qualitatively interpret the power of CaCE in identifying cluster evolution in real-time data with the large number of clusters that span across a larger period of time.

4.2 Datasets

We use two types of datasets with medium length text vectors (containing < 150 terms on an average, i.e., DS1 and DS2) and short length text vectors (containing < 50 characters on an average, i.e., DS3 and DS4). As shown in Table 2, for each dataset, a few categories (or domains) spanned across the time have been selected/created to have the ground-truth information, in terms of the number of clustering solutions and the number of clusters in each clustering solution.

- For the 20News group dataset (DS1), we selected four categories and spread them across three time periods.
- For the Patent abstract dataset (DS2), four categories of documents were collected during the three months of 2017.
- For the health-related tweets (DS3), media posts sent to six disease-specific Twitter groups within a four-year period (2014–2017) were selected.
- For the sport-related tweets (DS4), media posts sent to four sports-specific Twitter groups within a five-year period (2010–2014) were selected.

These clusters were placed in such a way as to show emerging, persistent, growth and decay patterns over time as in Table 2. We have made these datasets available to researcher.2

Additionally, we have conducted two case studies with (1) the DBLP-ACM research publication data and (2) the job posting data obtained from the Kaggle website. The dataset of the first case study spans across 10 years from 1994 to 2003, and each cluster solution is clustered into 3 clusters to identify the evolution patterns. The second case study data span across the year 2004–2006, and each cluster solution is clustered into 10 clusters to find the evolution.

2 https://drive.google.com/drive/u/1/folders/1gHoEm-R9S2Okh9LRVNk3JLVeGpRdWXn.
Table 2  Summary of datasets used for experiments

| Name                | # of clusters for each time-stamp cluster solution | Ground-truth evolution |
|---------------------|---------------------------------------------------|------------------------|
| 20Newsgroup (DS1)   | \(t_0: 3, t_1: 5, t_2: 4\)                        |                        |
| Patent (DS2)        | \(t_0: 5, t_1: 5, t_2: 5\)                        |                        |
| Health (DS3)        | \(t_0: 5, t_1: 5, t_2: 5, t_3: 5\)                |                        |
| Sports (DS4)        | \(t_0: 4, t_1: 3, t_2: 2, t_3: 4, t_4: 4\)       |                        |
| DBLP-ACM Years 1994–2003 | - (Case study 1)                              |                        |
| Job posting Years 2004–2006 | - (Case study 2)                              |                        |

4.3 Baselines

Several benchmarking methods were used to evaluate the accuracy of cluster groups identification: (1) general NMF [24] on intra-cluster association matrix \(S\); (2) the state-of-the-art clustering evolution method TextLuas [15] which uses Jaccard coefficient to determine the cluster similarity within cluster pairs in consecutive timestamps; and (3) a variation of CaCE (named as CaCE-CS) that uses cosine similarity for an inter-cluster association matrix instead of SGNS representation based on the number of overlapping terms. Additionally, the topic evolution method proposed in [10] for social media with short text is used to compare with CaCE in identifying the evolution patterns. Experiments were done using python 3.5 on 1.2 GHz–64-bit processor with 16 GB Memory.

4.4 Evaluation measures

The standard pairwise harmonic average of the precision and recall (F1-score) and Normalized Mutual Information (NMI) were used as the evaluation measures to identify the quality of cluster groups [35]. Evolution patterns of clusters including emerging, persistent, decay and
Table 3  Accuracy performance comparison for identifying cluster groups with different datasets, methods, and metrics

| Dataset | F1-score  | NMI       |
|---------|-----------|-----------|
|         | CaCE      | CaCE-CS   | NMF | TextLuas |
|         | CaCE      | CaCE-CS   | NMF | TextLuas |
| DS1     | 0.84      | 0.75      | 0.60 | 0.60 | 0.82 | 0.75 | 0.48 | 0.67 |
| DS2     | 0.68      | 0.68      | 0.65 | 0.56 | 0.68 | 0.68 | 0.37 | 0.61 |
| DS3     | 0.58      | 0.57      | 0.34 | 0.58 | 0.57 | 0.57 | 0.17 | 0.51 |
| DS4     | 0.74      | 0.66      | 0.51 | 0.53 | 0.65 | 0.54 | 0.06 | 0.46 |
| Average | 0.71      | 0.67      | 0.53 | 0.57 | 0.68 | 0.64 | 0.27 | 0.56 |

The performance of CaCE is indicated with bold values.

growth indicated through states changes are automatically identified within the visualization using top-frequent terms in each cluster.

4.5 Accuracy analysis

4.5.1 Quantitative interpretation

Results in Table 3 show that CaCE is able to produce higher accuracy in cluster groups identification compared to all other methods due to the use of inter-cluster association information in the matrix factorization using the number of common terms with SGNS. The modified version of CaCE, CaCE-CS, produces the next best performance. It uses cosine similarity to identify the inter-cluster association using representative terms. It fails to maximize the probability of closely associated clusters as original CaCE does with using the SGNS encoding. Cosine similarity, which measures the cosine angle between vectors that represent the clusters, is found inferior in modeling the inter-cluster association. TextLuas, which employs Jaccard similarity coefficient based on the number of common terms in clusters, links the clusters in consecutive time stamps if the similarity is higher than a threshold (set as 0.5). However, this approach is inferior in identifying global evolution over a period.

The proposed NMF with intra- and inter-cluster associations used in CaCE is able to accurately learn the latent relationships between each cluster based on the terms that they share and identify the cluster groups within the entire period, whereas the traditional NMF applied on term × cluster is not able to capture the cluster groups in the projected lower-dimensional space accurately. The result is worse when the number of clusters varies significantly within the cluster solutions across the corpus as in DS4 (Table 2). As shown by results, CaCE is capable of handling varying cluster numbers and the uniformly distributed clustering solutions, over the multiple time stamps.

Figure 7a shows the effect of applying regularization to the objective function in Eq. (4). $L_1$ regularization on the factor matrix $W$ (term × cluster groups) promotes sparsity in $W$ and enables the factorization process to focus on the distinct set of terms that links to a cluster group. This has been shown to be more effective for identifying distinct cluster groups for all the datasets based on the representative terms as depicted by higher F1-score and NMI in Fig. 7a.

We also analyze the effectiveness of using the density notion in CaCE. Density defined as in Eq. (10) based on the term frequencies is capable of filtering out loosely attached clusters to a group. CaCE assigns the zero density value to a cluster that does not share many
common terms with the rest of the group and make it a new singleton cluster group. This ensures a uniform term distribution within a group. Figure 7b confirms that this density-based filtering allows CaCE to identify a set of cohesive cluster groups over the time as shown by the improved performance. DS4, which supports less common terms in clusters, the density-based filtering, results in slightly poorer performance.

4.5.2 Qualitative interpretation: CaCE versus TextLuas

Figures 8, 9, 10, 11a show insight on evolution patterns obtained by CaCE, which show similar clusters in a group with a unique color. We label each cluster with its top-6 frequent terms to represent the included concept. Figures 8, 9, 10, 11b show the visualization of cluster evolution given by TextLuas. Table 2 represents the cluster groups and an expected cluster evolution patterns (ground-truths) for these datasets.

According to the derived evolution patterns in Fig. 8a by CaCE for DS1, (1) a persistent cluster related to ‘computer technology’ appears in blue color, (2) there is decay in information related to ‘games’ as revealed by merging of clusters from $t_1$ to $t_2$ (green color) and (3) it shows both split and merge of clusters by the cluster group which is a mix with ‘religion’ and ‘war’ (yellow color). This reveals a growth pattern within $t_0$ to $t_1$ through the split while showing a decay pattern within $t_1$ to $t_2$ through the merge. It should be noted from results in Table 3 that though CaCE achieves highest accuracy, however, it is not 100%. It can be confirmed as CaCE misses to identify linking of a cluster (marked as red) to the cluster group ‘game’ (marked as green) that seem to be highly similar according to top-frequent terms. However, an investigation of the cluster vector shows that this (red) cluster includes many other terms that are not part of the (green colored) cluster and only these few terms are shared among the two.

On the other hand, the evolution patterns on DS1 as shown in Fig. 8b show that TextLuas fails to identify the persistence pattern of ‘computer technology’ and the decay pattern of ‘games’ as identified by CaCE. TextLuas, based on local evolution patterns between cluster pairs in consecutive time stamps, is not capable of identifying these cluster dynamics accurately.
Figure 9a represents the CaCE cluster evolution identified in the Patent dataset (DS2). (1) It shows that CaCE is able to capture the growth of a ‘block chain’-related cluster group (yellow color) as revealed by its splits between $t_1$ and $t_2$. (2) It identifies the ‘computer vision’-related cluster group (green color) as a persistent pattern within $t_0$ to $t_2$; however, it should have been shown as decay as per ground-truth. The top-terms within non-linked clusters show the evidence for this deviated pattern as they show slight variations. (3) CaCE correctly identifies the birth of a cluster (gray color) showing an emerging pattern, which is under the ‘Microbiota’ cluster group according to ground-truth. (4) Finally, several related clusters are shown as distinct groups. A close investigation reveals that these clusters are related, but contain several unrelated terms. Therefore, CaCE identifies them as new groups with unique colors.
Figure 9b shows the evolution of clusters in DS2 obtained by TextLuas. It identifies the decay pattern of ‘computer vision’ correctly. It could not identify the growth pattern of the cluster group ‘block chain’ or birth of ‘Microbiota’. A closer investigation on these patterns reveals that TextLuas is not able to distinguish the clusters.

Table 3 shows that CaCE performs poorly on DS3 in identifying cluster groups as compared to other corpus. Visualization of the patterns obtained by CaCE, as shown in Fig. 10a, confirms that. Several evolution patterns are incorrectly drawn. (1) The growth of ‘mental’ health-related clusters (blue color) has been correctly identified (as per ground-truth in Table 2) as shown through the splits. (2) It identifies birth of the ‘diabetes’ cluster (pink color) in $t_3$ as an emerging pattern. However, it is incorrectly identifies as per the ground truth. (3) CaCE shows a mixed group with different types of clusters (in yellow color) as a decay pattern over
t₀ to t₂ through the merges. A closer investigation of top frequent terms reveals that common terms are shared by many diseases with high frequency, and this misleads CaCE to recognize them in different cluster groups separately.

Figure 11a shows the evolution of clusters in DS4 displayed by CaCE. (1) It correctly identifies the ‘soccer’ cluster group (blue color) as persistent over the time through its continuous appearance in each consecutive time stamp. (2) The growth pattern of ‘cricket’-related clusters (green color) within t₃ to t₄ and the decay pattern of ‘cycling’-related clusters (yellow color) within t₀ and t₁ are also partially identified as per the ground truth in Table 2. Deviated patterns from the ground-truth cluster evolution in Table 2 are resultant of some terms that are different in clusters and that contributes to identifying these clusters as unmatched patterns.
Figures 10, 11b show the cluster evolution pattern given by TextLuas for DS3 and DS4 and fail to identify many useful patterns. Both of them clearly show the mix of cluster groups as compared to CaCE. This ascertains: (1) the global cluster evolution patterns (as obtained by CaCE) cannot be accurately identified through the local connection analysis as done in TextLuas; and (2) Jaccard similarity, which relies on intra-cluster similarity, is not sufficient in cluster groups identifications.

**Benchmarking with other techniques** We compare the evolution patterns of CaCE with a recent method of emerging topic detection for short text [10]. This method uses a set of heuristics such as energy based on term frequencies to identify terms that have become important in the current time period and then creates a directed term-correlation graph and identifies the topics from the previous time window that persist in the current time window. Table 4 shows the emerging topics identified by the method in [10]. It identifies: (1) the ‘talk’-related topic as an emerging topic in DS1 with the terms people, war and fbi; (2) ‘blockchain’ in DS2; (3)
Table 4: Emerging topics obtained by the graph theoretic temporal topic model [10]

| Dataset | Emerging topics |
|---------|-----------------|
| DS1     | (1) people war fbi |
| DS2     | (1) invention blockchain relates |
| DS3     | (1) today sign reach (2) cardiac (3) womenshearts |
| DS4     | (1) Clarke (2) Grella (3) career (4) Mr (5) mrcricket (6) Veteran BREAKING 2014 |

Fig. 12: Time taken by each method for identifying the evolution of clusters

‘heart’-related topics in DS3; and (4) ‘cricket’ and ‘soccer’-related topics in DS4. However, the graph theoretic temporal topic model [10] can only show the identified topics as emerging. In contrast, CaCE is able to identify emergent, growing, persistent and decaying concepts.

Discussion: As detailed in Table 3, CaCE shows a higher performance as compared to benchmarking methods in identifying correct cluster groups. This confirms the superiority of CaCE in identifying evolution patterns (i.e., persistence, growth and decay) globally, which rely on accurate cluster group identification over the considered period. As revealed by the results in Table 3 and Figs. 8, 9, 10, 11, CaCE misses some evolution patterns due to some common terms appearing in many cluster groups and subgroups within cluster groups. Having said that, CaCE is the first method that details the comprehensive global evolution patterns with high accuracy and informs the lifecycle of the main clusters inherent in a corpus, which can be displayed through time/domain.

4.6 Efficiency and complexity analysis

Figure 12 shows the time taken by CaCE and its comparative methods. CaCE consumes slightly more time than the traditional NMF due to the inclusion of additional inter-cluster association matrix. The modified version of CaCE-CS consumes much higher time due to the additional step of cosine similarity calculation between clusters. The naive approach of calculating Jaccard coefficient considering the term intersection in TextLuas also consumes lesser time on average. The higher performance with 152% and 21% increase of average NMI in CaCE as per Table 3 compared to NMF and TextLuas, respectively, well justifies the 2–6 times higher consumption in time.
The computational complexity of CaCE, which is based on NMF, is $O(n^2)$ where $n$ is number of clusters. Similarly, CaCE-CS also processes the same computational complexity. However, the time complexities vary according to the additional matrices and steps included in the approaches. TextLuas has a linear computational complexity of $O(rm)$ where $r$ is the number of time stamps and $m \leq n$ is the number of clusters in a generic time-stamp.

### 4.7 Sensitivity analysis

One of the strengths in CaCE is modeling the inter-cluster association matrix using SGNS. Empirically, we validate this as in Fig. 13 by modeling matrix $A$ (1) with using the number of overlapping terms between clusters, and (2) with SGNS. As depicted by the results, the cluster associations modeled with SGNS, which is able to predict the neighbors correctly, assists the sparse term $\times$ cluster matrix factorization process in forming a lower-dimensional cluster $\times$ group matrix accurately. The inter-cluster association given with the cluster $\times$ cluster association matrix using the number of overlapping terms (without any weighting) between clusters gives lower performance as it fails to boost the distinction between clusters. As an exception, there is not much gain in DS4 by using SGNS where the number of clusters considerably varies within time stamps. We conjecture that in this case, maximizing probability of close cluster pairs is not making much difference due to the heterogeneous nature in the inter-cluster association matrix that formed with the number of overlapping terms.

The term frequency-based density uses a threshold value to determine the cohesiveness within a cluster group. Figure 14a shows that the density value less than ‘mean–standard deviation’ gives the best result in all other datasets than DS3. The density value of a cluster within the first quantile (mean–standard deviation) range of a group implies the cluster to be less cohesive in the group. Due to the higher occurrence of terms that are common in many clusters (i.e., noise), median shows the highest performance in DS3. CaCE uses ‘mean–standard deviation’ as the default threshold value for determining uniform density in a cluster group.

CaCE focuses on identifying four major cluster dynamics (i.e., birth, death, split and merge). It is natural to identify the 4 groups of clusters aligning with them, so the evolution patterns within the groups can be studied. However, in order to empirically verify this number of cluster groups, we experimented with different numbers of cluster groups. As shown in
4.8 Case studies: research and job trend analysis

4.8.1 Case study I

We conducted a case study using the DBLP-ACM publication data\(^3\) to confirm the capability of CaCE to accurately detect evolution patterns over a considerably large period of time (10 time stamps). We consider the DBLP-ACM bibliographic titles related to data science within the period of 1994–2003. The purpose of this case study was to display the effectiveness of cluster evolution given by CaCE without considering the clustering method used for the primary clustering solutions over a large time period. We use traditional NMF for generating the primary cluster solutions with three clusters per each time stamp. In the visualization, each cluster is shown with its top-3 terms. The cluster evolution patterns may change with the change in primary clustering solutions in each timestamp/domain.

Figure 15 shows that the research on database query and language technologies was at its peak in 1994 to 1995 (yellow color clusters with the growth pattern revealed by splits). There exist variations of database and management within this period such as data replication (pink color) and emerging pattern of large object databases (green color) shown with a new born cluster. Later from 1995 to 1996, this group of clusters (which showed growth earlier) shows a decay through merges. Remarkably, it shows the re-emergence of this concept in 2001 in the form of XML and web semantic languages.

For the period 1997–1998, the database technologies with commercial applications, such as multidimensional databases and related querying algorithms (green color), show decay as revealed by merging of those concepts. An emerging concept of transaction management information (orange color) is born in 1997.

Data mining emerged in 1998, which grows into distributed database architecture and data warehouse concepts within 1999, is depicted through the split (red color clusters). These concepts, in combination with query processing, form a separate cluster group (blue color) that shows the decay of clusters as shown by merges within 2000–2001. In 2003,

\(^3\) https://www.openicpsr.org/openicpsr/project/100843/version/V2/view,
CaCE captures the birth of a new cluster on query optimization deviating from the rest of the concepts, as identified by the emerging concept.

Generally, this case study shows the footprint of data science that moves from simple database management to web/xml base databases through data mining and warehousing over 1994 - 2003.

4.8.2 Case study II

The aim of this case study is to show the capability of CaCE in handling the larger number of clusters within a time stamp for the identification of accurate cluster evolution. The study uses the online job posting data in Kaggle website\textsuperscript{4} posted through the Armenian human resource portal ‘CareerCenter’. We consider a subset of job postings which span across 2004–2006.

The primary cluster solution per each time stamp is obtained using NMF, with setting the number of clusters to 10. Figure 16 depicts the interesting evolution patterns revealed by CaCE for this dataset, where clusters are represented with top-5 frequent terms. The study is able to reveal the demand and changes in certain professions over the years and shows evolution of necessary skills that are most frequently required by employers.

In general, it shows how the demand for administrative, coordination, sales and software-related job positions evolves over these years. The ‘administrative/director positions’ cluster group (yellow color) indicates the changes in the scope and skills of the position across the time. In 2004–2005, it shows a growth pattern with splits. Director positions are posted for accounting/finance skills and program implementation skills separately in 2004. In contrast, director positions require both of these skills in 2005 as qualifications. This again changed to different job positions in 2006, as shown by the splits (i.e., consultant, finance officers, director supervision, program coordinator, etc.).

The cluster group depicted by the green color shows a mix of ‘administrative positions’ and ‘software developer’ positions in 2004. It is obvious for CaCE to fail in separating them due to terms used by both these groups such as ‘design’ and ‘implementation’. Thus, it shows as a decay pattern over the years 2004–2005 with the merge of clusters. The post ‘software

\textsuperscript{4} https://www.kaggle.com/madhab/jobposts,
developer’ is persistent within 2005–2006. Furthermore, CaCE identifies a persistent pattern attached with jobs related to area-specific programs over 2004–2005 (red color). In 2005, it marks the death of those positions.

CaCE identifies the demand for ‘customer care’ positions in 2004 as marked by the birth of a (pink color) cluster. However, ‘customer sales’ or ‘product sales’-related positions appear in 2005 with slight variations to the skills required as a new position (i.e., emerging pattern showing in the blue color cluster group). Over the years 2005–2006, this shows a decay pattern with a change to necessary skills (i.e., ability to handle social and international activities). Furthermore, CaCE discloses two emerging positions in 2005 with the birth of ‘community coordination’ and ‘rural program supervision’-related clusters (olive and orange colors, respectively).

5 Discussion

The Cluster Association-aware matrix factorization for discovering cluster evolution tracks four major lifecycle states to discover cluster emergence, persistence, growth, and decay patterns that show over time or different domains. We summarize the interesting observations of this CaCE as follows:

- CaCE can accurately project the high-dimensional term × cluster representation into a lower-dimensional space for identifying global cluster groups through the assistance given by inter-cluster association information. Also, CaCE uses $L_1$ regularization on the factor matrix $W$ to promote sparsity and highlighting the distinguishing terms. It discriminates cluster groups effectively.

- CaCE models the inter-cluster association information for the matrix using the number of common terms enforcing SGNS to accurately learning the context of clusters. SGNS enforces through factorizing a (shifted) cluster correlation matrix semantically assists the NMF in learning the context of the clusters.

- CaCE introduces the notion of density to determine the strength between a cluster and its associated cluster group considering the ratio between term frequencies within the cluster and maximum term frequency within the corresponding cluster group. Clusters with a less density than a set threshold are identified as singleton clusters. This separation
Discovering cluster evolution patterns… 1425

of loosely connected clusters from a group results in cohesive cluster groups, improving the accuracy of cluster group identification.

– The visualization of links between clusters through a $k$-partite graph in CaCE visualizes all cluster dynamics including birth, death, split and merge. Each cluster from the first partite in the graph (within two consecutive partites) is compared with each cluster in its predecessor partite to add edges between two clusters to mark them as similar if they belong to the same group and, either both of them possess zero density values or both of them possess nonzero density value.

– The progressive $k$-partite graph-based approach to display the evolution of clusters in the cluster groups in CaCE can show persistent, growth, decay, and emerging patterns. Specifically, the birth of a cluster represents an emerging pattern while the split of a cluster showing a growth pattern and the merge of that cluster showing a decay pattern. A cluster that connects to a succeeding timestamp with a single link can show a persistent pattern.

– While comparing CaCE with baselines, CaCE-CS, a variation of CaCE that uses cosine similarity to identify the inter-cluster association using representative terms, measures the cosine angle between vectors that represent the clusters. It is inferior to modeling inter-cluster association with the cardinality of term set intersection between clusters as in CaCE. Thus, it fails to maximize the probability of closely associated clusters as original CaCE.

– The original NMF that projects the high-dimensional term $\times$ cluster representation into a lower-dimensional space without using cluster association information as in CaCE produce inferior results in both data categories that are having medium and short length text. TextLuas is the main state-of-the-art text cluster evolution visualization method, and it uses Jaccard similarity coefficient to link the clusters in consecutive time stamps if this goes beyond a threshold. Due to the consideration of only consecutive timestamps in drawing the links, it lacks in identifying global evolution over the considered period as in CaCE.

6 Conclusion

The proposed Cluster Association-aware matrix factorization for discovering cluster evolution (CaCE) approach is an effective Non-negative matrix factorization method to discover the evolution of clusters across the time-domain due to its underlying techniques. It can track four major lifecycle states of birth, death, split, and merge to discover their emergence, persistence, growth, and decay patterns generated over time or different domains. CaCE provides an assistance to the matrix factorization process of a sparse term $\times$ cluster matrix with inter-cluster associations. The inter-cluster association matrix is built with overlapping terms between clusters modeled with Skip-Gram with Negative-Sampling (SGNS). Further, we conjecture that term frequency-based density of clusters can be used to identify the inconsistent clusters in these cluster groups, and thereby, we form tight cluster groups. The evolution of clusters is visualized using a $k$-partite graph over the time considering the important cluster dynamics of birth, death, split, and merge through the identified cluster groups.

An extensive experimental study has been conducted with both qualitative and quantitative evaluation. News data and Patent abstracts which include medium size text vectors and tweets data with short-text vectors are used for experiments. Empirical results conducted on medium and short length datasets, benchmarked with relevant methods, show that CaCE
discovers emerging, persistence, growth, and decay of clusters with considerably higher accuracy performance. The use of inter-cluster association via the term set intersection between clusters and enforcement of SGNS concept in modeling inter-cluster association allow providing superior results than the baselines. Moreover, CaCE keeps track of the global cluster evolution over the considered period.

Currently, CaCE focuses on identifying four cluster groups that equal the number of cluster dynamics and sets that number as the lower rank for the matrix factorization process. This is an experimental-based threshold introduced to the CaCE aligning with the cluster dynamics. This parameter may depend on the datasets. Additionally, the density-based cluster group refinement phase identifies isolated clusters that are excluded from the existing groups. The evolution of these isolated clusters could be studied further. Extending this approach to make it independent on a clustering method is focus of our future investigation.

References

1. Aggarwal CC, Zhai C (2012) Mining text data. Springer, Berlin
2. Amado A, Cortez P, Rita P, Moro S (2018) Research trends on big data in marketing: a text mining and topic modeling based literature analysis. Eur Res Manage Bus Econo 24(1):1–7
3. Anastasius DC, Tagarelli A, Karypis G (2013) Document clustering: the next frontier.
4. Bao C, Ji H, Quan Y, Shen Z (2016) Dictionary learning for sparse coding: algorithms and convergence analysis. IEEE Trans Pattern Anal Mach Intell 38(7):1356–1369
5. Belford M, Mac Namee B, Greene D (2018) Stability of topic modeling via matrix factorization. Expert Syst Appl 91:159–169
6. Bolelli L, Ertekin Ş, Giles CL (2009) Topic and trend detection in text collections using latent dirichlet allocation. In: European conference on information retrieval, pp 776–780. Springer, Berlin
7. Carneiro HA, Mylonakis E (2009) Google trends: a web-based tool for real-time surveillance of disease outbreaks. Clin Infect Dis 49(10):1557–1564
8. Cataldi M, Di Caro L, Schifanella C (2010) Emerging topic detection on twitter based on temporal and social terms evaluation. In: Proceedings of the tenth international workshop on multimedia data mining, p 4. ACM
9. Chi Y, Song X, Zhou D, Hino K, Tseng BL (2007) Evolutionary spectral clustering by incorporating temporal smoothness. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 153–162. ACM
10. Churchill R, Singh L, Kirov C (2018) A temporal topic model for noisy mediums. In: Pacific-Asia conference on knowledge discovery and data mining, pp 42–53. Springer, Berlin
11. Du L, Buntine W, Jin H, Chen C (2012) Sequential latent dirichlet allocation. Knowl Inf Syst 31(3):475–503
12. Du N, Farajtabar M, Ahmed A, Smola AJ, Song, L (2015) Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 219–228. ACM
13. Gallili T (2015) dendextend: an r package for visualizing, adjusting and comparing trees of hierarchical clustering. Bioinformatics 31(22):3718–3720
14. Gandomi A, Haider M (2015) Beyond the hype: Big data concepts, methods, and analytics. Int J Inf Manage 35(2):137–144
15. Greene D, Archambault D, BelÉd V, Cunningham P (2014) Textlua: Tracking and visualizing document and term clusters in dynamic text data. arXiv preprint arXiv:1502.04609
16. Greene D, Cross JP (2017) Exploring the political agenda of the european parliament using a dynamic topic modeling approach. Polit Anal 25(1):77–94
17. Hervas-Oliver JL, Gonzalez G, Caja P, SemperE-Ripoll F (2015) Clusters and industrial districts: Where is the literature going? identifying emerging sub-fields of research. Eur Plan Stud 23(9):1827–1872
18. Hoffman M, Bach FR, Blei DM (2010) Online learning for latent dirichlet allocation. In: advances in neural information processing systems, pp 856–864
19. Hong L, Davison BD (2010) Empirical study of topic modeling in twitter. In: Proceedings of the first workshop on social media analytics, pp 80–88. ACM
20. Huang G, He J, Zhang Y, Zhou W, Liu H, Zhang P, Ding Z, You Y, Cao J (2015) Mining streams of short text for analysis of world-wide event evolutions. World wide web 18(5):1201–1217

21. Huang K, Sidiropoulos ND, Swami A (2014) Non-negative matrix factorization revisited: uniqueness and algorithm for symmetric decomposition. IEEE Trans Signal Process 62(1):211–224

22. Kasiviswanathan SP, Melville P, Banerjee A, Sindhwani V (2011) Emerging topic detection using dictionary learning. In: Proceedings of the 20th ACM international conference on Information and knowledge management, pp 745–754. ACM

23. Kasiviswanathan SP, Wang H, Banerjee A, Melville P (2012) Online II-dictionary learning with application to novel document detection. In: Advances in neural information processing systems, pp. 2258–2266

24. Kim J, He Y, Park H (2014) Algorithms for nonnegative matrix and tensor factorizations: a unified view based on block coordinate descent framework. J Global Optim 58(2):285–319

25. Kim MS, Han J (2009) A particle-and-density based evolutionary clustering method for dynamic networks. Proc VLDB Endow 2(1):622–633

26. Kutty S, Nayak R, Turnbull P, Chernich R, Kennedy G, Raymond K (2019) Paperminer—a real-time spatiotemporal visualization for newspaper articles. Digital Scholarship in the Humanities

27. Landauer M, Wurzenberger M, Skopik F, Settanni G, Filzmoser P (2018) Dynamic log file analysis: an unsupervised cluster evolution approach for anomaly detection. Comput Secur 79:94–116

28. Lee P, Lakshmanan LV, Milios EE (2014) Incremental cluster evolution tracking from highly dynamic network data. In: 2014 IEEE 30th international conference on data engineering (ICDE), pp 3–14. IEEE

29. Levy O, Goldberg Y (2014) Neural word embedding as implicit matrix factorization. In: Advances in neural information processing systems, pp 2177–2185

30. Li Q, Nourbakhsh A, Shah S, Liu X (2017) Real-time novel event detection from social media. In: Data Engineering (ICDE), 2017 IEEE 33rd international conference on, pp 1129–1139. IEEE

31. Lin YR, Chi Y, Zhu S, Sundaram H, Tseng BL (2008) Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In: Proceedings of the 17th international conference on World Wide Web, pp 685–694. ACM

32. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119

33. Mohotti WA, Nayak R (2018) Corpus-based augmented media posts with density-based clustering for community detection. In: 2018 IEEE 30th International conference on tools with artificial intelligence (ICTAI), pp 379–386. IEEE

34. Olszewski D (2014) Fraud detection using self-organizing map visualizing the user profiles. Knowl-Based Syst 70:324–334

35. Schütze H, Manning CD, Raghavan P (2008) Introduction to information retrieval, vol 39. Cambridge University Press, Cambridge

36. Shi T, Kang K, Choo J, Reddy CK (2018) Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web, pp 1105–1114. International World Wide Web Conferences Steering Committee

37. Ventocilla E, Riveiro M (2020) A comparative user study of visualization techniques for cluster analysis of multidimensional data sets. Inf Vis 19(4):318–338

38. Wang X, McCallum A (2006) Topics over time: a non-markov continuous-time model of topical trends. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 424–433. ACM

39. Yang T, Chi Y, Zhu S, Gong Y, Jin R (2011) Detecting communities and their evolutions in dynamic social networks—a bayesian approach. Mach Learn 82(2):157–189

40. You Y, Huang G, Cao J, Chen E, He J, Zhang Y, Hu L (2013) Geam: A general and event-related aspects model for twitter event detection. In: International conference on web information systems engineering, pp 319–332. Springer, Berlin

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
Wathsala Anupama Mohotti is currently a Senior Lecturer at the University of Ruhuna, Sri Lanka. She received a B.Sc. (Hons) in Information Technology and M.sc in Information Technology from the University of Moratuwa, Sri Lanka. She completed her Ph.D. at Queensland University of Technology (QUT), Australia. Her research interests include data mining, machine learning, text mining, and clustering and outlier detection.

Richi Nayak is currently Associate Professor of Computer Science in the School of Computer Science, Queensland University of Technology (QUT), Australia. She is an internationally recognized expert in data mining, text mining and web intelligence. She has combined knowledge in these areas very successfully with diverse disciplines such as Social Science, Science, and Engineering for Technology transfer to real-world problems to change their practices and methodologies. Her particular research interests are machine learning, text mining, personalization, and social network analysis.