Adaptive local structure learning for document co-clustering

Shudong Huang\textsuperscript{a}, Zenglin Xu\textsuperscript{a,∗}, Jiancheng Lv\textsuperscript{b}

\textsuperscript{a}SMILE Lab, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
\textsuperscript{b}College of Computer Science, Sichuan University, Chengdu 610065, China

A R T I C L E   I N F O

Article history:
Received 20 September 2017
Revised 6 February 2018
Accepted 9 February 2018
Available online 15 February 2018

Keywords:
Adaptive local structure learning
Graph regularization
Document co-clustering
Nonnegative matrix tri-factorization

A B S T R A C T

The goal of document co-clustering is to partition textual data sets into groups by utilizing the duality between documents (i.e., data points) and words (i.e., features). That is, the documents can be grouped based on their distribution on words, while words can be grouped based on their distribution on documents.

However, traditional co-clustering methods are usually sensitive to the input affinity matrix since they partition the data based on the fixed data graph. To address this limitation, in this paper, based on nonnegative matrix tri-factorization, we propose a new framework of co-clustering with adaptive local structure learning. The proposed unified learning framework performs intrinsic structure learning and tri-factorization (i.e., 3-factor factorization) simultaneously. The intrinsic structure is adaptively learned from the results of tri-factorization, and the factors are reformulated to preserve the refined local structures of the textual data. In this way, the local structure learning and factorization can be mutually improved. Furthermore, considering the duality between documents and words, the proposed framework explores not only the adaptive local structure of the data space, but also the adaptive local structure of the feature space. In order to solve the optimization problem of our method, an efficient iterative updating algorithm is proposed with guaranteed convergence. Experiments on benchmark textual data sets demonstrate the effectiveness of the proposed method.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Clustering is one of the most important unsupervised learning solutions and has been widely applied in text mining, computer vision, biology and so on. From a traditional view point, clustering aims at partitioning a dataset into groups of similar objects [1,2]. Many clustering methods, such as K-means [3], spectral clustering [4,5], spectral embedded clustering [6] and Nonnegative Matrix Factorization (NMF) [7–9], have been proposed up to now.

In recent years, co-clustering has received widespread attention in algorithm development and applications. It overcomes several limitations associated with traditional clustering methods by allowing automatic discovery of similarity based on a subset of attributes. Co-clustering methods have been studied intensively through many different theories and methodologies [10–11], including co-clustering based on Bayesian models [12–17], Nonnegative Matrix Tri-Factorization (NMTF) based co-clustering methods [18], spectral co-clustering [19–21], and so on. In particular, co-clustering methods based on graph theory have abstracted a lot of attentions [22], since intrinsic geometrical structure of data graph

have been proved to be useful in a number of machine learning methods [23–30]. However, data graphs of these methods are usually constructed by considering the K-Nearest Neighbors (KNN) which may mislead the clustering process since the nearest neighbors may belong to different clusters [31,32]. Furthermore, these methods are sensitive to the input affinity matrix since they partition data based on the fixed data graph. In other words, the similarity measurement and data clustering are often conducted in two separated steps, the learned data graph may not be the optimal one for clustering and lead to the suboptimal results.

To address this issue, we propose a new co-clustering method with adaptive local structure learning based on nonnegative matrix tri-factorization. Instead of performing similarity measurement and data clustering in two separated steps, the proposed model learns the affinity matrix and tri-factorization simultaneously to achieve the optimal clustering results. Meanwhile, both the data graph and the feature graph are constructed by selecting the adaptive and optimal neighbors for each data point and feature respectively. It is based on the assumption that the data points (or features) with a smaller distance should have a larger probability to be the optimal neighbors. We also apply the proposed method to the problem of document clustering using the benchmark textual data sets. The experimental results show that our method has several favorable features compared with previous related clustering algorithms. In

∗ Corresponding author.
E-mail address: zxu@uestc.edu.cn (Z. Xu).

https://doi.org/10.1016/j.knosys.2018.02.020
0950-7051/© 2018 Elsevier B.V. All rights reserved.
Table 1 Important notations used in this paper.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>Data matrix of size $m \times n$</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of data points</td>
</tr>
<tr>
<td>$m$</td>
<td>Number of features</td>
</tr>
<tr>
<td>$c$</td>
<td>Number of data clusters</td>
</tr>
<tr>
<td>$e$</td>
<td>Number of features clusters</td>
</tr>
<tr>
<td>$x_i$</td>
<td>$i$th column of $X$ (ith data point)</td>
</tr>
<tr>
<td>$f_i$</td>
<td>$i$th row of $X$ (ith feature)</td>
</tr>
<tr>
<td>$S'$</td>
<td>Similarity matrix of the data graph</td>
</tr>
<tr>
<td>$s'_{ij}$</td>
<td>$i$th row of $S'$</td>
</tr>
<tr>
<td>$s''_{ij}$</td>
<td>$i$th element of $s'_{ij}$</td>
</tr>
<tr>
<td>$S''$</td>
<td>Similarity matrix of the feature graph</td>
</tr>
<tr>
<td>$s''_{ij}$</td>
<td>$i$th row of $S''$</td>
</tr>
<tr>
<td>$s'''_{ij}$</td>
<td>$i$th element of $s''_{ij}$</td>
</tr>
<tr>
<td>$V$</td>
<td>Data partition matrix of size $n \times c$</td>
</tr>
<tr>
<td>$v_i$</td>
<td>$i$th row of $V$</td>
</tr>
<tr>
<td>$U$</td>
<td>Data partition matrix of size $m \times e$</td>
</tr>
<tr>
<td>$u_i$</td>
<td>$i$th row of $U$</td>
</tr>
<tr>
<td>$H$</td>
<td>Cluster centroid of size $e \times c$</td>
</tr>
<tr>
<td>$W^e$</td>
<td>Feature affinity matrix of size $m \times m$</td>
</tr>
<tr>
<td>$W'^e$</td>
<td>Data affinity matrix of size $n \times n$</td>
</tr>
<tr>
<td>$D^e$</td>
<td>Data degree matrix of size $n \times n$</td>
</tr>
<tr>
<td>$D'^e$</td>
<td>Feature degree matrix of size $m \times m$</td>
</tr>
<tr>
<td>$L^e$</td>
<td>Data graph Laplacian of size $c \times n$</td>
</tr>
<tr>
<td>$L'^e$</td>
<td>Feature graph Laplacian of size $m \times m$</td>
</tr>
</tbody>
</table>

In order to solve the optimization problem of our method, an efficient iterative updating algorithm is proposed and its convergence is also guaranteed theoretically. It is worthwhile to highlight several aspects of the proposed approach here:

- We propose a novel co-clustering with adaptive local structure learning based on nonnegative matrix tri-factorization. We also derive an efficient updating algorithm to solve the optimization problem by making use of the non-smooth based loss function, and its convergence is guaranteed theoretically.
- The unified learning framework performs intrinsic local structure learning and tri-factorization simultaneously. The intrinsic local structure is adaptively learned from the results of tri-factorization, and the factors are reformulated to preserve the refined local structures of data; In this way, the local structure learning and tri-factorization can be mutually improved.
- The adaptive local structures of both data space and feature space are explored, which is precisely essential for document co-clustering task.
- Extensive experiments on benchmark textual data sets show that the proposed method consistently outperforms the closely related clustering methods, suggesting the effectiveness of the proposed method.

The remainder of this paper is organized as follows. We introduce previous research which we are based on in Section 2. In Section 3, we illustrate our model in detail. Experimental results are presented in Section 4. The paper ends with a conclusion in Section 5.

2. Related works

Before we go into the details of our method, first we briefly review Nonnegative Matrix Factorization (NMF) [7]. For convenience of discussion, we present in Table 1 the important notations used in this paper.

NMF and its graph regularized extensions have attracted significant attention in recent years. Suppose nonnegative data matrix

$X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{m \times n}$,

where $n$ is the number of data points and $m$ is the number of features. The goal of NMF is to find two nonnegative matrices $U$ and $V$ which minimize the objective function as follows

$$J_{\text{NMF}} = \sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ij} - (UV^T)_{ij})^2 = ||X - UV^T||^2_F$$

s.t. $U \geq 0, V \geq 0$, (1)

where $|| \cdot ||_F$ is Frobenius norm. For vector $v$ and matrix $V, v \geq 0$ and $V \geq 0$ mean all the elements of $v$ and $V$ are equal to or larger than zero.

It has been proved that the objective function $J_{\text{NMF}}$ in Eq. (1) is convex in $U$ only or $V$ only. Accordingly, it is unrealistic to find the global minimum of $J_{\text{NMF}}$. To optimize the objective function, [7] proposed the following iterative updating algorithm

$$U_{ij} \leftarrow U_{ij} \frac{(XV)_{ij}}{U^TVV^T}$$

$$V_{ij} \leftarrow V_{ij} \frac{(X^TU)_{ij}}{VV^TU^T}.$$ (2)

In the clustering setting of NMF [33], $V \in \mathbb{R}^{e \times c}$ is the cluster assignment matrix where $c$ is the number of clusters. In reality, we have $c \ll n$ and $c \ll m$. That is, NMF essentially tries to find a compressed approximation of the original nonnegative data matrix.

Instead of focusing on 2-factorization as described above, Ding et al. [18] provided a systematic analysis of 3-factorization named Orthogonal Nonnegative Matrix Tri-factorization (ONMTF). The goal of ONMTF is to find three nonnegative matrices $U$, $V$ and $H \in \mathbb{R}^{e \times c}$ which minimize the objective function as follows

$$J_{\text{ONMTF}} = ||X - UHV^T||^2_F$$

s.t. $U \geq 0, V \geq 0, H \geq 0, U^TU = I, V^TV = I$, (3)

where $I$ is the identity matrix. To optimize the objective function, [18] proposed the following iterative updating algorithm

$$U_{ij} \leftarrow U_{ij} \sqrt{\frac{(XVH^T)_{ij}}{(U^TU)_{ij}}}$$

$$V_{ij} \leftarrow V_{ij} \sqrt{\frac{(X^TUH)_{ij}}{(V^TV)_{ij}}}$$

$$H_{ij} \leftarrow H_{ij} \sqrt{\frac{(U^TXV)_{ij}}{(U^TUH^TV)_{ij}}}.$$ (4)

However, NMF and ONMTF cannot deal with the data lying on manifold. To address this problem, [34] presented a graph regularized NMF (GNMF) to explore the intrinsic geometrical structure of data manifold. Instead of just considering the manifold of data space, [27] proposed a graph dual regularized NMF (DNMF) to explore the intrinsic geometrical structure of both data manifold and feature manifold. Meanwhile, [27] further presented an extension of DNMF named graph dual regularization non-negative matrix tri-factorization (DNNMTF).

ONMTF brings NMF new features by applying the orthogonality constraint. While GNMF and DNMF can improve the performance of NMF models by utilizing graph regularization. In a later section, we will show that the proposed method owns all these advantages.

3. Adaptive dual regularized co-clustering

In this section, first we describe the process of assigning the adaptive and optimal neighbors of the data points and features by exploring the adaptive local structures of data space and feature space, respectively. Then we will present a co-clustering with adaptive neighbors based on nonnegative matrix tri-factorization, followed with its optimization algorithm. Finally, we prove the convergence of the algorithm.
3.1. Adaptive local structure of data space

It is a successful strategy to preserve local manifold structure in the development of unsupervised co-clustering algorithms, especially considering the low-dimensional manifold structure which is embedded in the high dimensional ambient space [23,27]. In the past decade, a number of approaches have been proposed to explore the underlying local structure by constructing the graph Laplacian with a K-Nearest Neighbors (KNN) graph [34–36]. However, the graphs constructed with KNN may destroy the local connectivity of data, and thus lead to suboptimal neighborhoods. As a result, the manifold structure captured by such graph Laplacian would be inevitably affected.

Motivated by recent study in probabilistic neighborhood relationship [37], we focus on learning a Euclidean distance induced probabilistic neighborhood matrix instead of utilizing the graph Laplacian with the fixed neighborhood relationship. For each data point \( x_i \), all the data points \( \{x_j\}_{j=1}^{n} \) can be considered as the neighborhood of \( x_i \) with probability \( S_{ij}^{\prime} \), where \( S_{ij}^{\prime} \in \mathbb{R}^{n \times m} \) is determined by solving the problem as follows

\[
\min_{f} \sum_{i,j=1}^{n} \left( |x_i - x_j|^{2} S_{ij}^{\prime} + \gamma_{S}(S_{ij}^{\prime})^{2} \right),
\]

(5)

where \( \gamma_{S} \) is a regularization parameter controlling the contribution of the second term in Eq. (5). Note that a smaller distance \( d_{ij}^{2} = |x_i - x_j|^{2} \) will lead to a larger probability \( S_{ij}^{\prime} \). It can be observed that problem (5) suffers from a trivial solution when \( \gamma_{S} = 0 \). That is, only the nearest data point can be the neighbor of \( x_i \) and no other data points can be the neighbors of \( x_i \). When \( \gamma_{S} \) approaches \( \infty \), the optimal solution of problem (5) is that all the data points are neighbors of \( x_i \) with the same probability \( \frac{1}{k} \), which can be seen as a prior of uniform distribution [37], presented an effective method to search parameter \( \gamma_{S} \) by setting

\[
\gamma_{S} = \frac{1}{2n} \sum_{i,j=1}^{n} \left( kd_{ij}^{2} - \sum_{k=1}^{k} d_{ij}^{2} \right),
\]

(6)

where \( k \) is the data point neighborhood size. It is worth noting that the search of parameter \( \gamma_{S} \) can be better handled by searching the parameter \( k \), which has explicit meaning and easy to tune.

The matrix \( S^{\prime} \) obtained in the neighbors assignment can be seen as a weight matrix [37,38], thus we can use the following term to measure the smoothness of the data low-dimensional representation:

\[
\min_{S} \sum_{i,j=1}^{n} \left( |x_i - x_j|^{2} S_{ij}^{\prime} + \gamma_{S}(S_{ij}^{\prime})^{2} \right) + \beta \sum_{i,j=1}^{m} \left( |f_i - f_j|^{2} S_{ij}^{2} + \gamma_{f}(S_{ij}^{2})^{2} \right)
\]

(7)

where \( \beta > 0 \) is a regularization parameter balancing the smoothness of the data points and features in the first and third terms and the reconstruction error of co-clustering in the fifth term, \( \alpha > 0 \) and \( \beta \) are regularization parameters balancing the fitting error of adaptive local structure learning of data space and feature space in the second and fourth terms. Since the adaptive local structure of both data space and feature space are utilized in our model, we call Eq. (11) Adaptive Local Structure Learning for Dual Regularized Co-Clustering (ALSCLC). When letting \( \lambda = \mu = 0 \), ALSCLC degenerates to ordinary co-clustering method. To make the objective in Eq. (11) lower bounded and no longer suffers from either scale transfer problem or trivial solution [39], we use \( L_s \)-normalization on columns of \( U \) and \( V \) in the optimization, and compensate the norms of \( U \) and \( V \) to \( H \).

In the following, an efficient iterative updating algorithm is proposed to solve the optimization problem of Eq. (11). We will present an alternating scheme to optimize the problem. Specifically, we will optimize the objective with respect to one variable while fixing the other variables. This procedure repeats until convergence.

3.2. Adaptive local structure of features space

Previous studies have shown that not only the data space, but also the feature space presents low-dimensional manifold structures [23,27]. For each feature \( f_i \), all the features \( \{f_j\}_{j=1}^{m} \) can be considered as the neighborhood of \( f_i \) with probability \( S_{ij}^{\prime} \), where \( S_{ij}^{\prime} \in \mathbb{R}^{n \times m} \) is determined by solving the problem as follows

\[
\min_{f} \sum_{i,j=1}^{m} \left( ||f_i - f_j||^{2} S_{ij}^{\prime} + \gamma_{f}(S_{ij}^{\prime})^{2} \right),
\]

(8)

where \( \gamma_{f} \) is the regularization parameter. It can be observed that a smaller distance \( d_{ij}^{2} = ||f_i - f_j||^{2} \) will lead to a larger probability \( S_{ij}^{\prime} \). Similar with the computation of \( \gamma_{S} \), the problem of searching parameter \( \gamma_{f} \) can be solved by setting

\[
\gamma_{f} = \frac{1}{2m} \sum_{i=1}^{m} \left( l d_{ij}^{2} - \sum_{j=1}^{m} d_{ij}^{2} \right),
\]

(9)

where \( l \) is the feature neighborhood size.

With the estimated weight matrix \( S_{ij}^{\prime} \), we can measure the smoothness of the feature low-dimensional representation:

\[
\min_{S} \sum_{i,j=1}^{m} \left( ||f_i - f_j||^{2} S_{ij}^{\prime} + \gamma_{f}(S_{ij}^{\prime})^{2} \right)
\]

(10)

where \( D_{ij}^{S} = D_{ij}^{S} - W_{ij}^{S} \) is Laplacian matrix of the feature graph, \( W_{ij}^{S} \) is symmetric similarity matrix and \( W_{ij}^{S} = S_{ij}^{\prime} + S_{ji}^{\prime} \), and \( D_{ij}^{S} \in \mathbb{R}^{m \times m} \) is the diagonal degree matrix where the \( i \)-th diagonal element is \( (D_{ij}^{S})_{ii} = \sum_{j}(W_{ij}^{S})_{ij} \).
3.3.1. Updating $H$

Optimizing Eq. (11) w.r.t. $H$ is equivalent to minimizing the problem as follows

$$J_1 = ||X - UHV^T||_F^2$$

s.t. $H \succeq 0$.  

(12)

Let $\Phi \in \mathbb{R}^{c \times c}$ be the Lagrangian multiplier for constraint $H_{ij} \geq 0$. Then the Lagrange function is

$$L(H) = ||X - UHV^T||_F^2 + \text{Tr}(\Phi H^T).$$

(13)

The partial derivation of $L(H)$ w.r.t. $H$ is

$$\frac{\partial L(H)}{\partial H} = -2U^TXV + 2U^TVH^TV + \Phi.$$  

(14)

According to the KKT condition [40] $\Phi_iH_{ij} = 0$, we have

$$(-U^TXV + U^TVH^TV)_{ij}H_{ij} = 0.$$  

(15)

Eq. (15) leads to the following update rule

$$H_{ij} \leftarrow H_{ij} \frac{(U^TXV)_{ij}}{(U^TVH^TV)_{ij}}.$$  

(16)

3.3.2. Updating $V$

Optimizing Eq. (11) w.r.t. $V$ is equivalent to minimizing the problem as follows

$$J_2 = ||X - UHV^T||_F^2 + \lambda \text{Tr}(V^TL^S_yV)$$

s.t. $V \succeq 0$.  

(17)

Let $\Omega \in \mathbb{R}^{m \times c}$ be the Lagrangian multiplier for constraint $V_{ij} \geq 0$. Then the Lagrange function is

$$L(V) = ||X - UHV^T||_F^2 + \lambda \text{Tr}(V^TL^S_yV) + \text{Tr}(\Omega V^T).$$

(18)

The partial derivation of $L(V)$ w.r.t. $V$ is

$$\frac{\partial L(V)}{\partial V} = -2X^TUH + 2VH^TU^TV + 2\lambda L^S_y + \Omega.$$  

(19)

According to the KKT condition [40] $\Omega_iV_{ij} = 0$, we have

$$(-X^TUH + VH^TU^TV + \lambda L^S_y)_{ij}V_{ij} = 0.$$  

(20)

Introduce $L^S_y = (L^S_y)^* - (L^S_y)^-$. Then $X^TUH = (X^TUH)^+ - (X^TUH)^-$. $H^TU^TV = (H^TU^TV)^+ - (H^TU^TV)^-$ where $(L^S_y)_{ij} = (|(L^S_y)_{ij}| + (L^S_y)_{ij})/2$ and $(L^S_y)_{ij} = (|(L^S_y)_{ij}| - (L^S_y)_{ij})/2$ [23]. Thus we have

$$\left(\lambda \left((L^S_y)^* - (L^S_y)^-\right)V - (X^TUH)^+ + (X^TUH)^- \
+ V\left((H^TU^TV)^+ - (H^TU^TV)^-\right)\right)_{ij}V_{ij} = 0.$$  

(21)

According to [18], the following equation has the same fixed point condition of Eq. (21)

$$\left(\lambda \left((L^S_y)^* - (L^S_y)^-\right)V - (X^TUH)^+ + (X^TUH)^- \
+ V\left((H^TU^TV)^+ - (H^TU^TV)^-\right)\right)_{ij}V_{ij}^2 = 0.$$  

(22)

Eq. (22) leads to the following update rule

$$V_{ij} \leftarrow V_{ij} \frac{\left|\lambda \left((L^S_y)^* - (L^S_y)^-\right)V + (X^TUH)^+ + V\left((H^TU^TV)^+ - (H^TU^TV)^-\right)\right|_{ij}}{\left|\lambda \left((L^S_y)^* - (L^S_y)^-\right)V + (X^TUH)^+ + V\left((H^TU^TV)^+ - (H^TU^TV)^-\right)\right|_{ij}}.$$  

(23)

3.3.3. Updating $S^k$

Optimizing Eq. (11) w.r.t. $S^k$ is equivalent to solving the problem as follows

$$\min_{S^k} \frac{1}{n} \sum_{i,j=1}^{n} \left(\|x_i - x_j\|^2 S^k_{ij} + \gamma_a(S^k_{ij})^2 + \lambda Tr(V^TL^S_yV)\right)$$

s.t. $(S^k)^T1 = 1, 0 \leq S^k_{ij} \leq 1$.  

(24)

where $\lambda = \frac{\gamma_a}{2}$. According to Eq. (7), Eq. (24) can be rewritten as

$$\min_{S^k} \frac{1}{n} \sum_{i,j=1}^{n} \left(\|x_i - x_j\|^2 S^k_{ij} + \gamma_a(S^k_{ij})^2 + \frac{1}{2} \lambda \|V_i - V_j\|^2 S^k_{ij}\right)$$

s.t. $(S^k)^T1 = 1, 0 \leq S^k_{ij} \leq 1$.  

(25)

Since problem (25) is independent between different $i$, the problem can be solved individually for each $i$:

$$\min_{S^k} \frac{1}{n} \sum_{j=1}^{n} \left(\|x_i - x_j\|^2 S^k_{ij} + \gamma_a(S^k_{ij})^2 + \frac{1}{2} \lambda \|V_i - V_j\|^2 S^k_{ij}\right)$$

s.t. $(S^k)^T1 = 1, 0 \leq S^k_{ij} \leq 1$.  

(26)

Denote $d^a_{ij} = \|x_i - x_j\|^2$ and $d^a_{ij} = \|V_i - V_j\|^2$, and denote $p_i \in \mathbb{R}^{c \times 1}$ as a vector with the $j$th element as $p_{ij} = d^a_{ij} + \lambda d^a_{ij}$, then the problem (26) can be reformulated as

$$\min_{S^k \in [1,0]} \frac{1}{2} \|S^k + \frac{p_i}{2\gamma_a}\|^2.$$  

(27)

Let $\zeta$ and $\eta$ be the Lagrange multipliers for constraints $(S^k)^T1 = 1$ and $0 \leq S^k_{ij} \leq 1$, respectively. Thus the Lagrangian function of problem (27) is

$$L(S^k, \zeta, \eta) = \frac{1}{2} \|S^k + \frac{p_i}{2\gamma_a}\|^2 - \zeta ((S^k)^T1 - 1) - \eta \|S^k\|.  

(28)

According to the KKT condition [40], the optimal solution can be obtained by

$$S^k_{ij} = \left(\zeta - \frac{p_{ij}}{2\gamma_a}\right).$$  

(29)

Without loss of generality, suppose $p_{11}, p_{22}, \ldots, p_{mm}$ are ordered from small to large. If the optimal $S^k$ has only $k$ nonzero elements (i.e., $k$ is the data point neighborhood size), then according to Eq. (29), we know $S^k_{ir} = 0$ and $S^k_{i,k+1} = 0$. Thus we have

$$\{\zeta - \frac{p_{ij}}{2\gamma_a} > 0, \zeta - \frac{p_{ij}}{2\gamma_a} \leq 0.  

(30)

Considering the constraint $(S^k)^T1 = 1$, we have

$$\sum_{j=1}^{k} \left(\zeta - \frac{p_{ij}}{2\gamma_a}\right) = 1 \iff \zeta = 1 + \frac{1}{2\gamma_a} \sum_{j=1}^{k} p_{ij}.  

(31)

According to Eqs. (29), (31) and (6), the optimal solution can be rewritten as

$$S^k_{ij} = \left(\frac{1}{2\gamma_a} \sum_{j=1}^{k} p_{ij} + \frac{1}{k} - \frac{p_{i,k+1}}{2\gamma_a}\right).$$  

(32)

3.3.4. Updating $U$

Optimizing Eq. (11) w.r.t. $U$ is equivalent to minimizing the problem as follows

$$J_3 = ||X - UHV^T||_F^2 + \mu \text{Tr}(U^TL^S_yU)$$

s.t. $U \succeq 0$.  

(33)

Let $\Psi \in \mathbb{R}^{m \times c}$ be the Lagrangian multiplier for constraint $U_{ij} \geq 0$. Then the Lagrange function is

$$L(U) = ||X - UHV^T||_F^2 + \mu \text{Tr}(U^TL^S_yU) + \text{Tr}(\Psi U^T).  

(34)
The partial derivation of $L(U)$ w.r.t. $U$ is
\[
\frac{\partial L(U)}{\partial U} = -2XV^T + 2UHV^T + 2\mu L_2^X U + \Psi.
\] (35)
According to the KKT condition [40] $\Psi_i U_{ij} = 0$, we have
\[
\left( -XV^T + UHV^T + \mu L_2^X U \right)_{ij} U_{ij} = 0.
\] (36)
Introduce $L_0^U = \left( L_2^X \right)^+ - \left( L_2^X \right)^-$, $XV^T = \left( XV^T \right)^+ - \left( XV^T \right)^-$, $HV^T = \left( HV^T \right)^+ - \left( HV^T \right)^-$, then we have
\[
\left( \mu \left( \left( L_2^X \right)^+ - \left( L_2^X \right)^- \right) U - \left( XV^T \right)^+ + \left( XV^T \right)^- \\
+ U \left( \left( HV^T \right)^+ - \left( HV^T \right)^- \right) \right)_{ij} U_{ij} = 0.
\] (37)
According to [18], the following equation has the same fixed point condition of Eq. (37)
\[
\left( \mu \left( \left( L_2^X \right)^+ - \left( L_2^X \right)^- \right) U - \left( XV^T \right)^+ + \left( XV^T \right)^- \\
+ U \left( \left( HV^T \right)^+ - \left( HV^T \right)^- \right) \right)_{ij} U_{ij} = 0.
\] (38)
Eq. (38) leads to the following update rule
\[
U_{ij} \leftarrow U_{ij} \left[ \frac{\mu \left( \left( L_2^X \right)^+ - \left( L_2^X \right)^- \right) U - \left( XV^T \right)^+ + \left( XV^T \right)^-}{\mu \left( \left( L_2^X \right)^+ - \left( L_2^X \right)^- \right) U + \left( HV^T \right)^+ - \left( HV^T \right)^-} \right]_{ij}
\] (39)
3.3.5. Updating $S_i^j$ Optimizing Eq. (11) w.r.t. $S_i^j$ is equivalent to solving the problem as follows
\[
\min_{S_i^j} \sum_{i,j=1}^m \left( \| f_i - f_j \|^2 S_i^j + \gamma_f (S_i^j)^2 \right) + \eta \| T(U^T L_2^X U) \|
\] s.t. $S_i^j^T 1 = 1, 0 \leq S_i^j \leq 1,$
where $\eta = \frac{\| \xi \|^2}{\| \gamma_f \|^2}. $ According to Eq. (10), Eq. (40) can be rewritten as
\[
\min_{S_i^j} \sum_{i,j=1}^m \left( \| f_i - f_j \|^2 S_i^j + \gamma_f (S_i^j)^2 + \frac{1}{2} \eta \| u_i - u_j \|^2 S_i^j \right) \\
\] s.t. $S_i^j^T 1 = 1, 0 \leq S_i^j \leq 1.$
Since problem (41) is independent between different $i$, the problem can be solved individually for each $i$:
\[
\min_{S_i^j} \sum_{j=1}^m \left( \| f_i - f_j \|^2 S_i^j + \gamma_f (S_i^j)^2 + \frac{1}{2} \eta \| u_i - u_j \|^2 S_i^j \right) \\
\] s.t. $S_i^j^T 1 = 1, 0 \leq S_i^j \leq 1.$
Denote $d_i^j = \| f_i - f_j \|^2, d_i^j = \frac{1}{2} \| u_i - u_j \|^2$, and $q_i \in \mathbb{R}^{m \times 1}$ as a vector with the $j$th element as $q_{ij} = d_i^j + \eta u_i^T u_j$, then the problem (26) can be reformulated as
\[
\min_{S_i^j} \left\| S_i^j \right\|^2 + \frac{1}{2} \eta q_i \| S_i^j \|^2.
\] (43)
Let $\xi$ and $\zeta$ be the Lagrange multiplier for constraints $(S_i^j)^T 1 = 1$ and $0 \leq S_i^j \leq 1$, respectively. Then the Lagrangian function of problem (43) is
\[
L(S_i^j, \xi, \zeta_i) = \frac{1}{2} \left\| S_i^j \right\|^2 + \frac{q_{ij}}{2 \gamma_f} \left\| S_i^j \right\|^2 - \xi \left( (S_i^j)^T 1 - 1 \right) - \zeta_i S_i^j.
\] (44)
According to the KKT condition [40], the optimal solution can be obtained by
\[
S_i^j = \left( \xi - \frac{q_{ij}}{2 \gamma_f} \right)_+.
\] (45)
Without loss of generality, suppose $q_{i1}, q_{i2}, \ldots, q_{im}$ are ordered from small to large. If the optimal $S_i^j$ has only $l$ nonzero elements (i.e., $l$ is the feature neighborhood size), then according to Eq. (45), we know $S_i^j > 0$ and $S_i^j_{l+1} = 0$. Thus we have
\[
\left\{ \begin{array}{l}
\xi - \frac{q_{ij}}{2 \gamma_f} > 0, \\
\xi - \frac{q_{ij}}{2 \gamma_f} \leq 0.
\end{array} \right.
\] (46)
Considering the constraint $(S_i^j)^T 1 = 1$, we have
\[
\sum_{j=1}^l \left( \xi - \frac{q_{ij}}{2 \gamma_f} \right) = 1 \Rightarrow \xi = 1 + \frac{1}{2 \gamma_f} \sum_{j=1}^l q_{ij}.
\] (47)
According to Eqs. (47) and (9), the optimal solution can be rewritten as
\[
S_i^j = \left( \frac{1}{2 \gamma_f} \sum_{j=1}^l q_{ij} + \frac{1}{2 \gamma_f} - \frac{q_{il+1}}{2 \gamma_f} \right)_{+}.
\] (48)
Motivated by sparse representation learning model [41], the optimal solutions of both Eqs. (32) and (48) can be obtained explicitly by quick sort. According to the inference above, a co-clustering with adaptive local structure learning based on nonnegative matrix tri-factorization is designed. We summarize the algorithm procedure step by step in Algorithm 1.

Algorithm 1 Adaptive local structure learning for dual regularized co-clustering (ALSLC).

**Input:** data matrix $X$:
- the number of data clusters $c$, the number of feature clusters $e$;
- the point neighborhood size $k$, the feature neighborhood size $l$;
- regularization parameters $\alpha, \beta, \mu, \gamma$;
- initialize $S_i^j, S_i^j$ by the optimal solution to the problem (5) and problem (8), respectively;

**Output:** Partition matrix $V \in \mathbb{R}^{n \times c}$;

repeat

1. Update $H_{ij} \leftarrow H_{ij} \left[ (U^T X V_{ij})_{+} / (U^T U V_{ij}) \right]$
2. Update $U_{ij} \leftarrow U_{ij} \left[ (\mu (L_2^X)^+ U + (XV^T)^+ + (XV^T)^-)_{ij} / (\mu (L_2^X)^+ U + (HV^T)^+ - (HV^T)^-)_{ij} \right]$
3. Compute $S_i^j$ according to Eq. (32)
4. Update $V_{ij} \leftarrow V_{ij} \left[ (\mu (L_2^X)^+ V + (XU^T)^+ + (XU^T)^-)_{ij} / (\mu (L_2^X)^+ V + (UH^T)^+ - (UH^T)^-)_{ij} \right]$
5. Compute $S_i^j$ according to Eq. (48)
until Converges

3.4. Convergence analysis

In this subsection, we will investigate the convergence of the iterative updating algorithm. With the closed form solutions described in Eqs. (32) and (48), we only need to show that the objective in Eq. (11) is nonincreasing under the update rules in Eqs. (16), (23) and (39). Since the fifth term of Eq. (11) is only related to $H$, we have exactly the same updating rule for $H$ as in DNMFT [27]. Thus we can use the convergence proof of DNMFT to show that our objective is monotonically decreasing under the updating rule in Eq. (16). Please refer to [27] for more details.

Now we only need to prove that our objective is nonincreasing under the update rules in Eqs. (23) and (39). Furthermore, considering the symmetry of $U$ and $V$ in Eq. (11), we only need to prove the convergence under the updating rule for $V$ in Eq. (23).
Specifically, the auxiliary function \([7]\) is used to prove the convergence of the proposed method. The auxiliary function can be defined as follows.

**Definition 1** ([7]). \(Z(h, h')\) can be defined as an auxiliary function for \(G(h)\) if the conditions

\[
Z(h, h') \geq G(h), Z(h, h) = G(h)
\]

are satisfied.

**Lemma 2.** If \(Z\) is an auxiliary function for \(G\), then \(G\) is non-increasing under the updating formula as follows

\[
h^{t+1} = \arg \min_h Z(h, h').
\]

**Proof.** \(G(h^{t+1}) \leq Z(h^{t+1}, h') \leq Z(h', h') = G(h').\) \(\square\)

**Lemma 3** [42]. For any nonnegative matrices \(A \in \mathbb{R}^{n \times n}\), \(B \in \mathbb{R}^{k \times k}\), \(C \in \mathbb{R}^{n \times n}\), \(C' \in \mathbb{R}^{n \times n}\) and \(A, A\) are symmetric, then the following inequality holds

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} \left( \frac{(ACB)_{ij}C_{ij}}{C_{ij}} \right) \geq \text{Tr}(C' \text{ACB}).
\]

**Theorem 4. Let**

\[
J(V) = \text{Tr}(-2X'UHV^T + VH'U'UHV + \lambda V'L_S^2V).
\]

Then the following function

\[
Z(V, V') = -2 \sum_{ij} \left( X'UH \right)_{ij} V_{ij}^2 \left( 1 + \log \frac{V_{ij}}{V'_{ij}} \right)
\]

\[
+ 2 \sum_{ij} \left( X'UH \right)_{ij} \frac{V_{ij}^2 + V_{ij}^2}{2V_{ij}} + \sum_{ij} \left( \left( V(H'U'U'H) + \right) \right)_{ij} V_{ij}^2 \frac{V_{ij}^2}{V'_{ij}}
\]

\[
- \sum_{ijk} (H'U'U'H)_{jk} V_{ij} V_{ik} \left( 1 + \log \frac{V_{ij}V_{ik}}{V'_{ij}V'_{ik}} \right)
\]

\[
+ \lambda \sum_{ij} \left( (L_{ij}^S)' - V_{ij} + V_{ij}^2 \right)
\]

\[
- \sum_{ijk} \left( (L_{ij}^S)' \right)_{jk} V_{ij} V_{ik} \left( 1 + \log \frac{V_{ij}V_{ik}}{V'_{ij}V'_{ik}} \right)
\]

is an auxiliary function for \(J(V)\). Furthermore, it is a convex function in \(V\) and its global minimum is

\[
V_{ij} \leftarrow \frac{\left[ \lambda \left( L_{ij}^S \right)' - V + \left( X'UH \right) + V(H'U'U'H) \right]_{ij} + \left( X'UH \right)_{ij} \frac{V_{ij}^2}{V'_{ij}} \right]}{\left[ \lambda \left( L_{ij}^S \right)' - V + \left( X'UH \right) - V(H'U'U'H) \right]_{ij}}.
\]

**Proof.** We rewrite Eq. (51).

\[
L(V) = \text{Tr}(-2X'UHV^T + 2X'U'H')V^T
\]

\[
+ V(H'U'U'H)V - V(H'U'U'H)V^T
\]

\[
+ \lambda V(L_{ij}^S)'V - \lambda V(L_{ij}^S)'V.
\]

By applying **Lemma 3**, we have

\[
\text{Tr}(V(H'U'U'H)V) \leq \sum_{ij} \left( (L_{ij}^S)' \right)_{ij} V_{ij}^2
\]

\[
\text{Tr}(V(H'U'U'H)V) \leq \sum_{ij} \left( (V(H'U'U'H))' \right)_{ij} V_{ij}^2
\]

Moveover, by the inequality \(a \leq \frac{a^2 + b^2}{2b}\), we have

\[
\text{Tr}(V(H'U'U'H)V) \leq \sum_{ij} \left( (V(H'U'U'H))' \right)_{ij} V_{ij}^2
\]

To obtain the lower bound for the remaining terms, we use the inequality that \(c \geq 1 + \log c\), then

\[
\text{Tr}(V(L_{ij}^S)'V) \geq \sum_{ijk} \left( (L_{ij}^S)' \right)_{jk} V_{ij} V_{ik} \left( 1 + \log \frac{V_{ij}V_{ik}}{V'_{ij}V'_{ik}} \right)
\]

By summing over all the bounds, we can get \(Z(V, V')\), which obviously satisfies \(1) Z(V, V') \geq J(V); (2) Z(V, V) = J(V).\)

To find the minimum of \(Z(V, V')\), we take

\[
\frac{\partial^2 Z(V, V')}{\partial V_{ij} \partial V_{kl}} = 2\lambda \left( (L_{ij}^S)' \right)_{ij} V_{ij} \frac{V_{ij}}{V'_{ij}} - 2\lambda \left( (L_{ij}^S)' \right)_{ij} V_{ij}
\]

\[
- 2(X'U'H)_{ij} V_{ij} + 2(X'U'H)_{ij} \frac{V_{ij}}{V'_{ij}}
\]

and the Hessian matrix of \(Z(V, V')\)

\[
\frac{\partial^2 Z(V, V')}{\partial V_{ij} \partial V_{kl}} = \delta_{ik} \delta_{jl} (2\lambda \left( (L_{ij}^S)' \right)_{ij} V_{ij} \frac{V_{ij}}{V'_{ij}} + 2\lambda \left( (L_{ij}^S)' \right)_{ij} V_{ij})
\]

\[
+ 2(X'U'H)_{ij} V_{ij} + 2(X'U'H)_{ij} \frac{V_{ij}}{V'_{ij}}
\]

is a diagonal matrix with positive diagonal elements.

Thus \(Z(V, V')\) is a convex function of \(F\). Therefore, we can obtain the global minimum of \(Z(V, V')\) by setting \(\frac{\partial Z(V, V')}{\partial V_{ij}} = 0\) and solving for \(V\), from which we can get Eq. (23). \(\square\)

Now we can demonstrate the convergence of the proposed algorithm.
Table 3  Clustering results measured by ACC of the compared methods.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Kmeans</th>
<th>NMF</th>
<th>GNMF</th>
<th>FNMTF</th>
<th>ONMTF</th>
<th>DNMF</th>
<th>DNMTF</th>
<th>CFAN</th>
<th>ALSLC</th>
<th>ALSLCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>tr12</td>
<td>0.231</td>
<td>0.2104</td>
<td>0.2396</td>
<td>0.2275</td>
<td>0.1971</td>
<td>0.246</td>
<td>0.2371</td>
<td>0.23</td>
<td>0.2454</td>
<td>0.2466</td>
</tr>
<tr>
<td>tr11</td>
<td>0.2287</td>
<td>0.2082</td>
<td>0.2251</td>
<td>0.2074</td>
<td>0.1748</td>
<td>0.2227</td>
<td>0.2246</td>
<td>0.2072</td>
<td>0.2362</td>
<td>0.2411</td>
</tr>
<tr>
<td>TDT2-10</td>
<td>0.2917</td>
<td>0.3587</td>
<td>0.4074</td>
<td>0.3052</td>
<td>0.1614</td>
<td>0.404</td>
<td>0.4132</td>
<td>0.3688</td>
<td>0.5243</td>
<td>0.5311</td>
</tr>
<tr>
<td>tr31</td>
<td>0.2398</td>
<td>0.228</td>
<td>0.2386</td>
<td>0.235</td>
<td>0.1787</td>
<td>0.2421</td>
<td>0.2145</td>
<td>0.2155</td>
<td>0.2507</td>
<td>0.2529</td>
</tr>
<tr>
<td>Wap</td>
<td>0.1715</td>
<td>0.1448</td>
<td>0.1601</td>
<td>0.1623</td>
<td>0.1029</td>
<td>0.1582</td>
<td>0.1538</td>
<td>0.151</td>
<td>0.1867</td>
<td>0.1892</td>
</tr>
<tr>
<td>TDT2-20</td>
<td>0.2347</td>
<td>0.3316</td>
<td>0.4547</td>
<td>0.2369</td>
<td>0.0826</td>
<td>0.4492</td>
<td>0.4516</td>
<td>0.3293</td>
<td>0.4933</td>
<td>0.4831</td>
</tr>
<tr>
<td>Hitech</td>
<td>0.2547</td>
<td>0.233</td>
<td>0.2381</td>
<td>0.2443</td>
<td>0.1904</td>
<td>0.2389</td>
<td>0.2405</td>
<td>0.2356</td>
<td>0.2554</td>
<td>0.2501</td>
</tr>
<tr>
<td>Cacmcsi</td>
<td>0.8899</td>
<td>0.9029</td>
<td>0.9228</td>
<td>0.7655</td>
<td>0.7441</td>
<td>0.9228</td>
<td>0.9285</td>
<td>0.9262</td>
<td>0.9648</td>
<td>0.9654</td>
</tr>
<tr>
<td>Classic</td>
<td>0.6686</td>
<td>0.6177</td>
<td>0.6405</td>
<td>0.6774</td>
<td>0.3836</td>
<td>0.6417</td>
<td>0.6327</td>
<td>0.6165</td>
<td>0.7113</td>
<td>0.701</td>
</tr>
<tr>
<td>Av.</td>
<td>0.3577</td>
<td>0.3577</td>
<td>0.3919</td>
<td>0.3402</td>
<td>0.2412</td>
<td>0.3917</td>
<td>0.3885</td>
<td>0.3645</td>
<td>0.4289</td>
<td>0.4298</td>
</tr>
</tbody>
</table>

Table 4  Clustering results measured by purity of the compared methods.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Kmeans</th>
<th>NMF</th>
<th>GNMF</th>
<th>FNMTF</th>
<th>ONMTF</th>
<th>DNMF</th>
<th>DNMTF</th>
<th>CFAN</th>
<th>ALSLC</th>
<th>ALSLCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>tr12</td>
<td>0.3123</td>
<td>0.3054</td>
<td>0.3265</td>
<td>0.3148</td>
<td>0.3043</td>
<td>0.3233</td>
<td>0.3272</td>
<td>0.3169</td>
<td>0.3271</td>
<td>0.3296</td>
</tr>
<tr>
<td>tr11</td>
<td>0.3456</td>
<td>0.3366</td>
<td>0.3589</td>
<td>0.3401</td>
<td>0.3239</td>
<td>0.3614</td>
<td>0.3478</td>
<td>0.3411</td>
<td>0.3601</td>
<td>0.3591</td>
</tr>
<tr>
<td>TDT2-10</td>
<td>0.3183</td>
<td>0.3992</td>
<td>0.4334</td>
<td>0.3551</td>
<td>0.1164</td>
<td>0.4340</td>
<td>0.4297</td>
<td>0.4144</td>
<td>0.5467</td>
<td>0.5571</td>
</tr>
<tr>
<td>tr31</td>
<td>0.4019</td>
<td>0.4021</td>
<td>0.4097</td>
<td>0.3949</td>
<td>0.3601</td>
<td>0.4095</td>
<td>0.3948</td>
<td>0.3855</td>
<td>0.4122</td>
<td>0.4128</td>
</tr>
<tr>
<td>Wap</td>
<td>0.3138</td>
<td>0.3033</td>
<td>0.3155</td>
<td>0.3177</td>
<td>0.2359</td>
<td>0.3121</td>
<td>0.3056</td>
<td>0.3074</td>
<td>0.3232</td>
<td>0.3214</td>
</tr>
<tr>
<td>TDT2-20</td>
<td>0.2575</td>
<td>0.3628</td>
<td>0.5111</td>
<td>0.2809</td>
<td>0.0826</td>
<td>0.5111</td>
<td>0.5085</td>
<td>0.5721</td>
<td>0.5312</td>
<td>0.5229</td>
</tr>
<tr>
<td>Hitech</td>
<td>0.2653</td>
<td>0.2683</td>
<td>0.2753</td>
<td>0.2719</td>
<td>0.2635</td>
<td>0.2732</td>
<td>0.2727</td>
<td>0.2712</td>
<td>0.2742</td>
<td>0.2761</td>
</tr>
<tr>
<td>Cacmcsi</td>
<td>0.9067</td>
<td>0.9029</td>
<td>0.9228</td>
<td>0.8021</td>
<td>0.7950</td>
<td>0.9228</td>
<td>0.9285</td>
<td>0.9262</td>
<td>0.9648</td>
<td>0.9654</td>
</tr>
<tr>
<td>Classic</td>
<td>0.7053</td>
<td>0.6512</td>
<td>0.6737</td>
<td>0.7156</td>
<td>0.4844</td>
<td>0.6731</td>
<td>0.6647</td>
<td>0.6530</td>
<td>0.7444</td>
<td>0.7402</td>
</tr>
<tr>
<td>Av.</td>
<td>0.4252</td>
<td>0.4169</td>
<td>0.4009</td>
<td>0.4215</td>
<td>0.3318</td>
<td>0.4600</td>
<td>0.4648</td>
<td>0.4431</td>
<td>0.4981</td>
<td>0.4985</td>
</tr>
</tbody>
</table>

**Theorem 5.** Updating $V$ using Eq. (23) will monotonically decrease the value of the objective in Eq. (11), hence it converges.

**Proof.** By Lemma 2 and Theorem 4, we can get that $J(V) = Z(V^0, V^0) \geq Z(V^1, V^1) \geq \cdots \geq Z(V^t, V^t) \geq \cdots$ So $J(V)$ is monotonically decreasing. Since $J(V)$ is obviously bounded below, we prove this theorem.

3.5. Complexity analysis

In this subsection, we will introduce the computational cost of the presented method in comparison to other closely related methods. In detail, we discuss the computational cost of our method, standard NMF [7], graph regularized NMF (GNMF) [34], and graph dual regularized NMF (DNMF) [27]. As summarized in [34], the overall time cost for NMF is

$$O(tMNC),$$

(55)

where $t$ is the number of iterations, $N$ is the number of data points, $M$ is the number of features and $C$ denotes the number of clusters. In reality, we have $c \ll N$ and $c \ll M$. Since GNMF needs $O(N^2M)$ to compare the $p$-nearest data graph, the overall cost for GNMF is

$$O(t(N + q(p + 4))MC + N^2M),$$

(56)

where $p$ is the number of nearest neighbors, $q$ is the number of iterations in Conjugate Gradient (CG) and the value of $q$ can be set to 20 according to [34]. Since DNMF needs $O(N^2M)$ to construct the $p$-nearest data graph and $O(NM^2)$ to construct the $p$-nearest feature graph respectively, the overall cost for DNMF is

$$O(t(MNC + Nc + MC)).$$

(58)

where $O( NC)$ and $O( MC)$ are the computational cost of updating $S^t$ and $S^t$ in each iteration, respectively.

4. Experiments

To evaluate the performance of the presented method, we compared our method with several related nonnegative matrix (tri-factorization methods: K-means, NMF [7], ONMTF [18], GNMF [31], FNMTF [43], CFAN [38], DNMF and DNMTF [27]. In our method, when we ignore the feature space adaptive local structure learning (i.e., $\mu = 0$), the method degenerates to a simple version, denoted by ALSLC. In order to verify the importance of the feature space adaptive local structure learning, we also compare it with ALSLCC.

4.1. Data sets

The experiments are performed on several real-world textual data sets. The specific characteristics of the textual data sets are summarized as follows

- The data sets tr11, tr12, tr31 and Hitech were derived from TREC collections. The classes of tr11, tr12 and tr31 correspond to the documents that were judged relevant to particular queries. Hitech was collected from the San Jose Mercury newspaper articles, and it contained documents about electronics, technology, computers, health, medical and research.
- TDT2-10 and TDT2-20 are subset of NIST Topic Detection and Tracking (TDT2) corpus. The TDT2 Corpus has been designed to collect the material drawn on a daily basis from six sources, including two news wires (APW, NYT), two radio programs (VOA, PRI), and two television programs (CNN, ABC).
- The dataset Wap is from the WebACE project [44]. Each document corresponds to a web page listed in the subject hierarchy of Yahoo.
- Cacmcsi and Classic were obtained by combining different abstracts that were used to evaluate various information retrieval systems [45]. Classic was obtained by combining the CRANFIELD, MEDLINE, CISI and CACM (CMCC) abstracts. Cacmcisi was obtained by combining CISI and CACM abstracts.

1. ftp://ftp.cis.ohio-state.edu/pubsmart
The number of classes for these textual data sets ranges from 2 to 20, and the number of samples ranges from 313 to 7094. These data sets are available in the CLUTO toolkit\(^5\) [46]. The specific characteristics of the data sets are given in Table 2.

4.2. Parameter setting

Since many clustering methods have one or more parameters to be tuned, in order to fairly compare these methods, we preform these algorithms under different parameter settings. We independently repeat the experiments under each parameter setting for 20 times and the best average result is recorded for comparison. We set the number of clusters equal to the true number of classes for all the data sets and clustering algorithms.

For graph regularized matrix factorization algorithms (i.e., GNMF, DNMF and CFAN), the binary weighting scheme is used for constructing the data graph and the neighborhood size is set by the grid \([1, 2, \ldots, 10]\) according to [31]. And the regularization parameter is searched over the grid \([0.1, 1, 10, 100, 500, 1000]\).

For FNMTF, ONMTF and DNMTF, the number of word clusters is set to be the same as the number of document clusters according to [18]. For DNMTF, the binary weighting scheme is used for constructing both the data graph and feature graph, the neighborhood size of the data graph is set to be the same as that of the feature graph and is set by searching the grid \([1, 2, \ldots, 10]\).

For the proposed algorithm, the number of word clusters is set to be the same as the number of document clusters. The neighborhood size of the data graph is set to be the same as that of the feature graph, i.e., \(k = l\), which are determined by the grid \([1, 2, \ldots, 10]\). For simplicity, we also set regularization parameter \(\alpha = \lambda = \mu = \beta\) and \(\lambda\) is set by searching the grid \([0.1, 1, 10, 100, 500, 1000]\). So the parameters of ALSLCC is tuned roughly. Better parameter tuning would achieve better clustering performance than that reported in this paper.

Given the number of clusters, no parameter selection is needed for K-means and NMF.

4.3. Evaluation measure

Clustering Accuracy (ACC) and Purity are the standard measures widely used for clustering [31].

ACC aims to find the one-to-one relationship between classes and clusters. It is usually used to measure the extent to which each cluster contains instances from the corresponding class. The specific definition of Clustering Accuracy can be defined as follows:

\[
\text{ACC} = \frac{\sum_{i=1}^{n} \delta(\text{map}(r_i), l_i)}{n},
\]

\(^5\) http://glaros.dtc.umn.edu/gkhome/views/cluto/
where \( r_i \) is the cluster label of a data point and \( l_i \) is the corresponding true class label, \( n \) denotes the total number of the data points, \( \delta(x, y) \) is the delta function that equals 1 if \( x = y \) and equals 0 otherwise. map(r_i) denotes the permutation mapping function that maps each cluster label \( r_i \) to the equivalent label from the data set. A large ACC value indicates a better clustering result.

The cluster purity is applied to measure the extent to which each cluster contained data points from primarily one class. The purity of a clustering result can be defined as follows

\[
Purity = \frac{1}{n} \sum_{i=1}^{c} \frac{n_i}{n} P(S_i) P(S_i) = \frac{1}{n} \max(n_i^j),
\]

where \( S_i \) denotes the particular cluster of size \( n_i \), \( n_i^j \) denotes the number of data points of the \( i \)th input class that was assigned to the cluster \( \mathcal{C}_j (1 \leq j \leq c) \), \( n \) is the total number of data points while \( c \) is the number of clusters.

### 4.4. Clustering results

The clustering results of different clustering methods on all the data sets are shown in Tables 3 and 4.

It can be seen that the proposed method outperforms the other methods in terms of ACC and Purity. The superiority of our method arises in the following two aspects: (1) co-clustering the words and documents together, and the clustering of words can lead to improvement in the clustering of documents; (2) the performance of document co-clustering can be improved by making use of the adaptive local structure learning, i.e., it verifies our assumption that performs intrinsic local structure learning and tri-factorization simultaneously can mutually improve the performance of both intrinsic local structure learning and tri-factorization. Furthermore, ALSLCC outperforms ALSLC on most of the data sets. This indicates utilizing the adaptive local structure of feature space can further improve the performance of document co-clustering at most cases.

### 4.5. Parameters selection

In this subsection, we investigate the sensitivity with respect to the neighborhood size \( k (=l) \) and the regularization parameter \( \lambda (=\mu) \). When we vary the value of one parameter, we keep the other parameters fixed at the optimal value.

The clustering results with respect to the number of the neighborhood size \( k \) are plotted in Fig. 1. It is clear that ALSLCC is a little sensitive to the neighborhood size \( k \). Relatively speaking, a better clustering result may be obtained when \( k \) varies from 4 to 6.

The clustering results with respect to the regularization parameter \( \lambda \) are plotted in Fig. 2. It can be seen that ALSLCC is a little sensitive to the regularization parameter \( \lambda \). Generally speaking, it achieves good performance when \( \lambda \) varies from 1 to 100. In summary, we may set \( k = 5 \) and \( \lambda = 10 \) in application for simplicity.
4.6. Convergence study

The updating rules for minimizing the objective of ALSLCC are essentially iterative and its convergence is also guaranteed theoretically. In this subsection, we investigate how fast these rules can converge.

The convergence curves of ALSLCC on all the data sets are plotted in Fig. 3, where the x-axis denotes the iteration number and the y-axis denotes the value of objective function. It can be observed that the iterative update rules for ALSLCC converge very fast, usually within 50 iterations.

5. Conclusion

In this paper, we propose a co-clustering with adaptive local structure learning based on nonnegative matrix tri-factorization. The unified learning framework performs intrinsic structure learning and tri-factorization simultaneously. The intrinsic structure is adaptively learned from the results of tri-factorization, and the factors are reformulated to preserve the refined local structures of the textual data. In this way, the local structure learning and factorization can be mutually improved. Furthermore, considering the duality between documents and words, the proposed framework explores not only the adaptive local structure of the data space, but also the adaptive local structure of the feature space. To the best of our knowledge, this is the first work on using adaptive local structures for document co-clustering. In order to solve the optimization problem of our method, an efficient iterative updating algorithm is proposed and its convergence is also guaranteed theoretically. Experiments on benchmark textual data sets demonstrate the effectiveness of the proposed method.

Acknowledgment

This work was supported by NSF China (No. 61572111, No. G05Q09R004), a Fundamental Research Fund for the Central Universities of China (No. ZYGX2016Z003), and a 985-Project of UESTC (No.A1098531023601041).

References