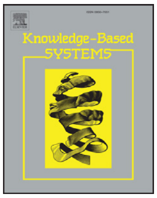
Knowledge-Based Systems 135 (2017) 147–158

Contents lists available at ScienceDirect

Knowledge-Based Systems

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



A novel regularized concept factorization for document clustering

Wei Yan[a], Bob Zhang[a,*], Sihan Ma[b], Zuyuan Yang[c]

[a] Department of Computer and Information Science, University of Macau, Macau, China
[b] Department of Computer and Information Science, Wuhan University, Wuhan, China
[c] School of Automation, Guangdong University of Technology, Guangzhou, China

ARTICLE INFO

Article history:
Received 27 March 2017
Revised 25 July 2017
Accepted 9 August 2017
Available online 10 August 2017

Keywords:
Document clustering
Concept factorization
Manifold regularization

ABSTRACT

Document clustering is an important tool for text mining with its goal in grouping similar documents into a single cluster. As typical clustering methods, Concept Factorization (CF) and its variants have gained attention in recent studies. To improve the clustering performance, most of the CF methods use additional supervisory information to guide the clustering process. When the amount of supervisory information is scarce, the improved performance of CF methods will be limited. To overcome this limitation, this paper proposes a novel regularized concept factorization (RCF) algorithm with dual connected constraints, which focuses on whether two documents belong to the same class (must-connected constraint) or different classes (cannot-connected constraint). RCF propagates the limited constraint information from constrained samples to unconstrained samples, allowing the collection of constraint information from the entire data set. This information is used to construct a new data similarity matrix that concentrates on the local discriminative structure of data. The similarity matrix is incorporated as a regularization term in the CF objective function. By doing so, RCF is able to make full use of the supervisory information to preserve the local structure of the data set. Thus, the clustering performance will be improved significantly. Our experiments on standard document databases demonstrate the effectiveness of the proposed method.

1. Introduction

Document clustering has received considerable attention as an important tool for efficient organization, navigation, retrieval, and summarization of a large text collection. With a good document clustering method, computers can automatically organize a document corpus into several hierarchies of semantic clusters. As a result, users can browse and navigate documents efficiently [1]. Recently, matrix factorization-based approaches have been applied to document clustering with impressive outcomes. In particular, Nonnegative Matrix Factorization (NMF) [2–4] and Concept Factorization (CF) [5–7].

In general, NMF aims to find two nonnegative matrix factors whose product is a good approximation to the original data matrix. The nonnegative constraints in NMF yield parts-based representation of text documents because they allow only additive, not subtractive, combinations [8,9]. Some researches have shown that there is physiological evidence for part-based representation in the human brain [10]. The major limitation of NMF is that it can only be performed in the original feature space of the data points. It is unclear how to effectively apply NMF in the transformed data space, e.g., reproducing kernel Hilbert space (RKHS) [7]. To address this limitation, Xu and Gong proposed the CF method [6], in which the data matrix $\mathbf{X}$ is decomposed into three nonnegative matrices $\mathbf{X}$, $\mathbf{U}$, and $\mathbf{V}$, such that $\mathbf{X} \approx \mathbf{XUV}^T$. CF preserves all the strengths of NMF, but also can be performed in any data representation space [11]. When using the CF method in document clustering, each document is represented by a linear combination of the cluster centers, and each cluster center is expressed by a linear combination of the data points. Here, document clustering can be accomplished by computing two sets of coefficients, i.e., $\mathbf{U}$ and $\mathbf{V}$. Since the linear coefficients have explicit semantic meanings, the label of each document can be easily derived from these coefficients. To further improve the clustering performance of CF, Locally Consistent Concept Factorization (LCCF) [7] preserves the local geometrical structure information of the data set by integrating the nearest-neighbor graph regularization into the CF model.

It should be noted that the basic CF algorithm can barely make use of the supervisory information. Recent researches have shown that the use of extra supervisory information can improve the learning performance in machine learning [12–14]. To this end, many CF variations have been proposed by using supervisory in-

[*] Corresponding author.
E-mail addresses: helloyanwei@163.com (W. Yan), bobzhang@umac.mo (B. Zhang), sihanma@whu.edu.cn (S. Ma), yangzuyuan@gdut.edu.cn (Z. Yang).

http://dx.doi.org/10.1016/j.knosys.2017.08.010
0950-7051/© 2017 Elsevier B.V. All rights reserved.

formation to guide the learning process [5,15–17]. The supervisory information is either incorporated into the basic CF model or the graph structure.

Among these CF variants, Constrained Concept Factorization (CCF) [15] used the exact label information as additional hard constraints. This model ensures that data points with the same class label must be strictly mapped to share the same representation in the new representation space. The limitation of CCF is that it ignores the local geometrical structure information of the data set. To address this issue, Local Regularization Constrained Concept Factorization (LRCCF) [16] incorporates both the local geometrical structure information and label information into the CF model. Different from LRCCF, Graph-based Discriminative Concept Factorization (GDCF) [5] utilizes the label information of a fraction of the data in a regularization term. Unfortunately, Both LRCCF and GDCF impose no restriction on data points coming from different classes to be mapped sufficiently farther in the new representation space, thus their discriminant power will be limited. Semi-supervised LCCF [7] use label information to modify the data's graph. Specifically, if two data points come from the same class, a large weight can be given to the edge connecting them. If not, the weight is set to zero. If two data points do not have supervisory information, the corresponding value in nearest-neighbor graph can be utilized as the weight. Constrained Neighborhood Preserving Concept Factorization (CNPCF) [17] uses supervisory information in the form of must-link constraint to modify the graph. However, when the supervisory information is limited, only using this information to modify the graph is not efficient to keep the geometry structure of the data set. Therefore, it is necessary to propose a novel framework for CF, which can take advantage of the unconstrained data points and keep the local geometry structure of the data set simultaneously.

To address the aforementioned drawbacks, a novel regularized concept factorization method (RCF) is proposed in this paper. The main advantage of the proposed method is that it can efficiently use the limited supervisory information to preserve the local structure of the data space. To accomplish this, we first propagate dual connected constraints to the entire data set and construct a new weight matrix. This type of propagation approach has achieved impressive results in cross-modal retrieval [18,19]. The weight matrix will then be added into the CF objective function as a regularization term. The main contributions of our paper are:

- The proposed regularized CF method achieves impressive document clustering results through utilizing the constraint propagation approach. It propagates the limited dual connected constraints from the constrained data points to the unconstrained data points, allowing us to collect more supervisory information on the data set. This practice can dramatically improve the performance of CF in document clustering.
- Our approach can obtain data representations with more discriminating power. As our method allocates large values in the weight matrix for data points belonging to the same class, it in turn assigns smaller values for data points from different classes. The weight matrix will be treated as a penalty term in the objective function, where the different penalty values will make the data points from the same class much more closer and the different classes much farther in the new representation space.
- This method also works when label information is available. Since the information of dual connected constraints is a weaker type of supervisory information, it can be readily inferred from the explicit label information.

The rest of the paper is organized as follows. In Section 2, we briefly provide the background of document representation and CF. Section 3 presents the details of regularized concept factorization (RCF) along with its optimization methods and proof of convergence. We describe the performed experiments on real data sets in Section 4. Finally, in Section 5, we draw our conclusions and suggestions for future work.

2. A brief overview of document representation and concept factorization

2.1. Document representation

In document representation each document is expressed as a weighted term-frequency vector [6]. Let $\mathcal{W} = \{f_1, f_2, \ldots, f_m\}$ be the complete vocabulary set of the document corpus after the stop-words removal and words stemming operations. The term-frequency vector X_i of document d_i is defined as:

$$X_i = \{x_{1i}, x_{2i}, \ldots, x_{mi}\}^T \tag{1}$$

$$x_{ji} = t_{ji} \cdot \log\left(\frac{n}{idf_j}\right) \tag{2}$$

where t_{ji} is the term frequency of word $f_j \in \mathcal{W}$ in document d_i, idf_j denotes the number of documents containing word f_j, n represents the total number of documents in the corpus, and m is the cardinal number of $\mathcal{W}$. In addition, X_i is normalized to be of unit Euclidean length. These term-frequency vectors are used to construct the $m \times n$ term-document matrix $\mathbf{X}$. This matrix will be utilized to conduct concept factorization, and the document clustering result will be directly obtained from the factorization result.

2.2. A brief review of CF

CF is an useful matrix decomposition technique. It has been widely used in pattern recognition and data mining [11,20–22]. Xu and Gong [6] modeled the document clustering problem by using two representations. In the first representation, each underlying concept/cluster can be characterized by associating the data points that share similar concepts. This can be expressed by a linear combination of the entire data points. Let X_i be the m-dimensional term-frequency vector denoting the data point i, where $i = 1, \ldots n$, and R_c is the center of concept c, where $c = 1, \ldots, k$. The first representation can be defined as:

$$R_c = \sum_i u_{ic} X_i \tag{3}$$

where u_{ic} is a nonnegative association weight indicating the degree of data point i relating to concept c. In the second representation, each data point can be approximated by a linear combination of all the concepts, where each linear coefficient measures the degree of overlap between the corresponding data point and the concept. Translating the above statement into mathematics, we have

$$X_i = \sum_c v_{ic} R_c \tag{4}$$

where v_{ic} is a nonnegative number indicating the projection value of X_i onto the concept center R_c. We construct the data matrix $\mathbf{X} = [X_1, X_2, \ldots, X_n] \in \mathbf{R}_+{}^{m \times n}$ using the feature vector of data point i as the ith column. From (3) and (4) we have

$$\mathbf{X} \approx \mathbf{XUV}^T \tag{5}$$

where $\mathbf{U} = [u_{jk}] \in \mathbf{R}_+{}^{m \times k}$ and $\mathbf{V} = [v_{jk}] \in \mathbf{R}_+{}^{n \times k}$. As $k \ll m$ and $k \ll n$, concept factorization leads to low-dimensional representation of the original matrix. The Euclidean distance based objective function can be defined as:

$$O = \left\| \mathbf{X} - \mathbf{XUV}^T \right\|_F^2 \tag{6}$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm of a matrix. With the above formulation, we turn the data clustering problem into the computation of **U** and **V** that minimizes the objective function O.

To minimize the above objective function, the multiplicative updating rules are introduced as [6]:

$$u_{jk} \leftarrow u_{jk} \frac{(\mathbf{KV})_{jk}}{(\mathbf{KUV}^T\mathbf{V})_{jk}} \tag{7}$$

$$v_{jk} \leftarrow v_{jk} \frac{(\mathbf{KU})_{jk}}{(\mathbf{VU}^T\mathbf{KU})_{jk}}. \tag{8}$$

where $\mathbf{K} = \mathbf{X}^T\mathbf{X}$. It is natural to construct a kernel matrix by utilizing a kernel function. Therefore, CF can effectively be kernelized to enhance its performance in some applications. Please see [6] for more details.

3. Regularized concept factorization (RCF)

As we have mentioned above, traditional CF can hardly take advantage of any supervised information when it is available. In the following, a novel regularized CF method which fully makes use of supervisory information will be proposed in detail.

Regarding the supervisory information, the explicit label information may be hard to collect. In practice, we found another type of supervisory information that is relatively easy to obtain. We call this the dual connected constraint information. Specifically, a pair of the data points with the same label are expressed in terms of a must-connect constraint and otherwise cannot-connect constraint. However, the inverse process may not be true, i.e., it is hard to directly derive the explicit labels of the data points from only dual connected constraints in general. This indicates that dual connected constraints provide one kind of supervisory information, which is inherently weaker and thus more general than the actual labels of the data points. Therefore, dual connected constraints are used as prior information in the following proposed RCF approach.

Usually, available supervisory information is very limited as the labeling process requires a lot of human input. If CF methods only utilize the scarce constraint information, its performance will not improve dramatically. To overcome this limitation, we first propagate these constraints to unconstrained data points so that we can obtain the constraint information on the entire data set. Then, the obtained constraint information is added to the CF objective function to construct the RCF model.

3.1. Constraint propagation

This subsection presents the constraint propagation in detail. The propagating process can be viewed as a two-class classification problem [18,23]. The relationship of a pair of constrained data points is encoded as “+1” or “−1”. The propagation process determines the constraint relationship of two unconstraint data points. This process is completely equivalent to classifying the relationship of a class labeled “+1” or a class labeled “−1”.

Let $\mathbf{X} = \{(x_i, l_i)\}_{i=1}^n \cup \{x_i\}_{i=n+1}^N$ be a data set, where n represents the number of labeled samples, $l_i \in \{1, \dots, C\}$, where C represents the number of classes and $N - n$ denotes the number of unlabeled samples. The must-connect constraints and cannot-connect constraints can be denoted as $MC = \{(x_i, x_j) : l_i = l_j, 1 \le i, j \le N\}$ and $CC = \{(x_i, x_j) : l_i \neq l_j, 1 \le i, j \le N\}$, respectively. The initial dual connected constraint matrix $\mathbf{Z} = \{z_{ij}\}_{N\times N}$ can be defined as follows:

$$z_{ij} = \begin{cases} +1, & \text{if } (x_i, x_j) \in MC \\ -1, & \text{if } (x_i, x_j) \in CC \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

For convenience, the candidate set of the propagated dual connected constraints matrix is defined as $\mathbf{F} = \{f_{ij}\}_{N\times N}$, where the value of $|f_{ij}|$ denotes the dual connected constraint score of (x_i, x_j). More concretely, if $f_{ij} > 0$, f_{ij} is the similarity between x_i and x_j, where the larger the value of f_{ij} is, the more x_i is similar to x_j, thus it is a must-connect. If $f_{ij} < 0$, and the smaller the value of f_{ij} is, the more x_i is dissimilar to x_j, this is known as a cannot-connect. The optimal **F** can be found through the following algorithm.

1. Construct a k-NN graph by defining its weight matrix $\mathbf{W} = \{w_{ij}\}_{N\times N}$ as follows:

$$w_{ij} = \begin{cases} \frac{a(x_i,x_j)}{\sqrt{a(x_i,x_i)}\sqrt{a(x_j,x_j)}}, & \text{if } x_i \in \mathcal{N}_p(x_j) \text{ or } x_j \in \mathcal{N}_p(x_i) \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

   where $\mathcal{N}_p(x_i)$ denotes the p-nearest neighbor set of the data point x_j. $\mathbf{A} = a(x_i, x_j)_{N\times N}$ is the kernel matrix defined on the data **X**. In particular, $a(x_i, x_j) = \exp(-\|x_i - x_j\|^2/t)$, and t denotes the bandwidth parameter. Set $\mathbf{W} = (\mathbf{W} + \mathbf{W}^T)/2$ to ensure **W** is symmetric.
2. Compute the normalized graph Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, where $\mathbf{D} = [d_{ii}]_{N\times N}$ is a diagonal matrix with $d_{ii} = \sum_j \mathbf{W}_{ij}$.
3. Iterate $\mathbf{F}_v(t+1) = \alpha\mathbf{L}\mathbf{F}_v(t) + (1-\alpha)\mathbf{Z}$ for the vertical constraint propagation until it converges, where α is a parameter ranging from 0 to 1. The proof of the convergence can be found in [24].
4. Iterate $\mathbf{F}_h(t+1) = \alpha\mathbf{F}_h(t)\mathbf{L} + (1-\alpha)\mathbf{F}_v^*$ for the horizontal constraint propagation until it converges, where $\mathbf{F}_v^*$ is the limit of $\{\mathbf{F}_v(t)\}$.
5. $\mathbf{F}^* = \mathbf{F}_h^*$ is the final representation of the propagated dual connected constraints, where $\mathbf{F}_h^*$ is the limit of $\{\mathbf{F}_h(t)\}$.
6. Construct a new weight matrix $\widetilde{\mathbf{W}} = \{\tilde{w}_{ij}\}_{N\times N}$ by using $\mathbf{F}^*$ and the original weight matrix **W**

$$\tilde{w}_{ij} = \begin{cases} 1 - \left(1 - f_{ij}^*\right)\left(1 - w_{ij}\right), & \text{if } f_{ij}^* \ge 0 \\ \left(1 + f_{ij}^*\right)w_{ij}, & \text{if } f_{ij}^* < 0. \end{cases} \tag{11}$$

   The new weight matrix $\widetilde{\mathbf{W}}$ has the following properties.

1. $\tilde{W}$ is nonnegative and symmetric, $\tilde{w} \in [0, 1]$. This means $\tilde{W}$ can be applied as a normalized weight matrix.
2. $\tilde{w} \ge w_{ij} (or < w_{ij})$ if $f_{ij}^* \ge 0 (or < 0)$. This shows the novel weight matrix $\widetilde{\mathbf{W}}$ is actually inferred from the original weight matrix **W** by increasing w_{ij} for the must-connected constraints with $f_{ij}^* > 0$ and decreasing for the cannot-connected constraints with $f_{ij}^* < 0$.
3. $(\partial\tilde{w}_{ij}/\partial w_{ij}) = 1 - |f_{ij}^*|$. This property shows the must-connected and cannot-connected constraints play the same important role (with the same derivatives) in weight adjustment if they share the same dual connected constraint values.
4. $\tilde{w}_{ij} \ge f_{ij}^* (or \le 1 + f_{ij}^*)$ if $f_{ij}^* \ge 0 (or < 0)$. This property ensures $\tilde{w}_{ij}$ to be large when f_{ij}^* takes a large positive value even if w_{ij} is initially very small and $\tilde{w}_{ij}$ to be small when f_{ij}^* takes a large negative value even if w_{ij} is initially very large.

By using the above method, more constraint information of the data set is collected. Dual connected constraint information is used to reconstruct the weight matrix. That is, if a pair of data points belong to the same class, the corresponding weight of the edge connecting them is given a much larger value, otherwise the weight is assigned a much smaller value.

3.2. The objective function of RCF

We provide in this subsection details in the construction of RCF by embedding the weight matrix to the CF objective function. The following term is applied to preserve the supervisory information of the data set:

$$\phi(\mathbf{V}) = \frac{1}{2}\sum_{i,j=1}^{N} \|v_i - v_j\|^2 \tilde{w}_{ij}. \tag{12}$$

When x_i and x_j share the same class number, x_i should appear together with x_j in the original geometric space and f^*_{ij} will be assigned a large positive value. According to the definition of $\widetilde{\mathbf{W}}$, the value of $\tilde{w}_{ij}$ will also be large. To minimize $\phi(\mathbf{V})$, v_i and v_j should have a minor Euclidean distance, such that v_i and v_j should also be close to each other in the low-dimensional representation space. On the contrary, when x_i and x_j are from different classes, x_i and x_j should be far away from each other in the geometric space and f^*_{ij} will get a large negative value, thus the value of $\tilde{w}_{ij}$ will be small. Therefore, v_i and v_j will be much farther in contrast with the result when v_i and v_j share the same class. According to the above analysis, we found that $\phi(\mathbf{V})$ can produce points with the same label appear together and points from different classes occur much farther to each other in the low-dimensional representation space. This is consistent with the points' intrinsic geometric relationships in the original space. By minimizing (12), the limited available dual connected constraint information can be used to preserve the original geometry of the data set efficiently. If the Euclidean distance is selected to measure the quality of the approximation, the objection function of RCF is

$$\mathcal{O} = \left\|\mathbf{X} - \mathbf{XUV}^T\right\|_F^2 + \lambda\phi(\mathbf{V}). \tag{13}$$

In (13), the first part can be regarded as the reconstruction term, which can guarantee the effect of the approximation. The second part can be considered as a constraint term, aimed to preserve the original geometrical structure of the data space. The regularization parameter $\lambda \geq 0$ controls the proportions of these two parts in the RCF objective function.

Although the proposed objective function is similar to that of LCCF, the significance of the two methods is different. LCCF constructs a nearest-neighbor graph to keep the geometrical structure of the data space, whereas RCF makes an attempt to preserve the supervisory information of the data set in the new representation space.

3.3. Updating rules for RCF

The objective function in (13) can be rewritten as

$$\begin{aligned}\mathcal{O} &= \left\|\mathbf{X} - \mathbf{XUV}^T\right\|_F^2 + \lambda\phi(\mathbf{V})\\ &= Tr(\mathbf{K}) - 2Tr(\mathbf{VU}^T\mathbf{K}) + Tr(\mathbf{VU}^T\mathbf{KUV}^T)\\ &\quad +\lambda Tr(\mathbf{V}^T\widetilde{\mathbf{D}}\mathbf{V}) - \lambda Tr(\mathbf{V}^T\widetilde{\mathbf{W}}\mathbf{V})\end{aligned}$$

where tr(·) is the trace of a matrix and $\widetilde{\mathbf{D}}$ denotes a diagonal matrix whose entries are column sums of $\widetilde{\mathbf{W}}$, $\tilde{d}_{ii} = \sum_j \widetilde{\mathbf{W}}_{ij}$. Denote $\widetilde{\mathbf{L}} = \widetilde{\mathbf{D}} - \widetilde{\mathbf{W}}$.

We assign two Lagrange multipliers ψ_{ij} and ϕ_{ij} for $u_{ij} > 0$ and $v_{ij} > 0$, respectively. Let $\mathbf{\Psi} = [\psi_{ij}]$ and $\mathbf{\Phi} = [\phi_{ij}]$ be matrixes that includes the Lagrange multipliers. Then, the objective function can be rewritten as

$$\begin{aligned}\mathcal{L} &= Tr(\mathbf{K}) - 2Tr(\mathbf{VU}^T\mathbf{K}) + Tr(\mathbf{VU}^T\mathbf{KUV}^T)\\ &\quad +\lambda Tr(\mathbf{V}^T\widetilde{\mathbf{D}}\mathbf{V}) - \lambda Tr(\mathbf{V}^T\widetilde{\mathbf{W}}\mathbf{V})\\ &\quad +Tr(\mathbf{\Psi}\mathbf{U}^T) + Tr(\mathbf{\Phi}\mathbf{U}^T)\end{aligned} \tag{14}$$

The partial derivatives of (14) with respect to **U** and **V** are

$$\frac{\partial\mathcal{L}}{\partial\mathbf{U}} = -2\mathbf{KV} + 2\mathbf{KUV}^T\mathbf{V}^T + \mathbf{\Psi}, \tag{15}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{V}} = -2\mathbf{KU} + 2\mathbf{VU}^T\mathbf{KU} + \mathbf{\Phi}. \tag{16}$$

Using the Karush-Kuhn-Tucker (KKT) conditions $\psi_{ij}u_{ij} = 0$ and $\phi_{ij}v_{ij} = 0$, we have the following equations for u_{ij} and v_{ij}.

$$-(\mathbf{KV})_{jk}u_{jk} + (\mathbf{KUV}^T\mathbf{V})_{ij}u_{jk} = 0 \tag{17}$$

$$-(\mathbf{KU})_{jk}v_{jk} + (\mathbf{VU}^T\mathbf{KW})_{jk}v_{jk} + \lambda(\mathbf{LV})_{jk}v_{jk} = 0. \tag{18}$$

$$\mathbf{u}_{jk} = \mathbf{u}_{jk}\frac{(\mathbf{KV})_{jk}}{(\mathbf{KUV}^T\mathbf{V})_{jk}} \tag{19}$$

$$\mathbf{v}_{jk} = \mathbf{v}_{jk}\frac{(\mathbf{KU} + \lambda\widetilde{\mathbf{W}}\mathbf{V})_{jk}}{(\mathbf{VU}^T\mathbf{KU}\lambda\widetilde{\mathbf{D}}\mathbf{V})_{jk}} \tag{20}$$

The above solutions to minimize the objective function $\mathcal{O}$ are not unique. It is easy to check that if **U** and **V** are the solution to $\mathcal{O}$, **UQ** and $\mathbf{VQ}^{-1}$ will also form a solution for any positive diagonal matrix **Q**. Thus, we further normalize the solution to make it unique. Let $\mathbf{u}_c$ be the column vector of **U** [6], with the constraints $\|\mathbf{R}_c\| = \|\mathbf{u}_c^T\mathbf{K}\mathbf{u}_c\| = 1$ and $\mathbf{UV}^T$ does not change, **U** and **V** should be updated with normalization:

$$\mathbf{V} = \mathbf{V}[\text{diag}(\mathbf{U}^T\mathbf{KU})]^{1/2} \tag{21}$$

$$\mathbf{U} = \mathbf{U}[\text{diag}(\mathbf{U}^T\mathbf{KU})]^{-1/2} \tag{22}$$

3.4. Convergence proof

A proof of the convergence of our algorithm is made in this subsection. We have the following theorem regarding the iterative updating rules Eqs. (19) and (20).

Theorem 1. *This objective function $\mathcal{O}$ in (13) is nonincreasing under the updating rules in (19) and (20). The objective function is invariant under these updates if and only if* **U** *and* **V** *are at a stationary point.*

To prove Theorem 1, we have to show that $\mathcal{O}$ is nonincreasing under the update rules in (19) and (20). Due to the fact that the second term of $\mathcal{O}$ is only about **V**, the update step for **U** in RCF is exactly the same as the original CF. Thus, we can make use of the convergence proof of CF to show that $\mathcal{O}$ is nonincreasing under the update rule in (19). Please refer to [6,25] for more details.

Now, we only need to prove that $\mathcal{O}$ is nonincreasing under the update step in (20). Our procedure is similar with that proposed in [25]. Our proof needs an auxiliary function which is used in the Expectation-Maximization algorithm [26], and it is defined as:

Definition 1. $\mathbf{G}(v, v')$ is an auxiliary function for $\mathbf{F}(v)$ if the conditions

$$\mathbf{G}(v, v') \geq \mathbf{F}(v), \mathbf{G}(v, v) = \mathbf{F}(v)$$

are satisfied.

The auxiliary function is very useful because of the following lemma:

Lemma 1. *If* **G** *is an auxiliary function of* **F***, then,* **F** *is nonincreasing under the update*

$$v^{(t+1)} = \arg\min_{v} \mathbf{G}(v, v^{(t)}). \tag{23}$$

Proof.

$$\mathbf{F}(v^{(t+1)}) \leq \mathbf{G}(v^{(t+1)}, v^{(t)}) \leq \mathbf{G}(v^{(t)}, v^{(t)}) = \mathbf{F}(v^{(t)}).$$

Next, we will show that the update rule for **V** in (20) is exactly the update in (23) with a proper auxiliary function.

Considering any element v_{ab} in **V**, $\mathbf{F}_{ab}$ is used to represent the part of $\mathcal{O}$ that is only relevant to v_{ab}. It is easy to get

$$\mathbf{F}'_{ab} = \left(\frac{\partial\mathcal{O}}{\partial\mathbf{V}}\right)_{ab} = (-2\mathbf{KU} + 2\mathbf{VU}^T\mathbf{KU} + 2\lambda\widetilde{\mathbf{L}}\mathbf{V})_{ab}, \tag{24}$$

$$\mathbf{F}''_{ab} = 2(\mathbf{U}^T\mathbf{K}\mathbf{U})_{bb} + 2\lambda\widetilde{\mathbf{L}}_{aa}. \tag{25}$$

Since our update is essentially element-wise, it is sufficient to show that each $\mathbf{F}_{ab}$ is nonincreasing under the update step of (20).

Lemma 2. *Function*

$$\begin{aligned}\mathbf{G}(v, v_{ab}^{(t)}) &= \mathbf{F}_{ab}(v_{ab}^{(t)}) + \mathbf{F}'_{ab}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) \\ &\quad + \frac{(\mathbf{V}\mathbf{U}^T\mathbf{K}\mathbf{U})_{ab} + \lambda(\widetilde{\mathbf{D}}\mathbf{V})_{ab}}{v_{ab}^{(t)}}(v - v_{ab}^{(t)})^2\end{aligned} \tag{26}$$

is an auxiliary function for $\mathbf{F}_{ab}$, *the part of* $\mathcal{O}$ *which is only relevant to* v_{ab}.

Proof. Given that $\mathbf{G}(v, v) = \mathbf{F}_{ab}(v)$ is obvious, we need only to show that $\mathbf{G}(v, v_{ab}^{(t)}) \geq \mathbf{F}_{ab}(v)$. To do this, we compare the Taylor series expansion of $\mathbf{F}_{ab}(v)$

$$\begin{aligned}\mathbf{F}_{ab}(v) &= \mathbf{F}_{ab}(v_{ab}^{(t)}) + \mathbf{F}'_{ab}\left(v_{ab}^{(t)}\right)\left(v - v_{ab}^{(t)}\right) \\ &\quad + \left[\left(\mathbf{U}^T\mathbf{K}\mathbf{U}\right)_{bb} + \lambda\widetilde{\mathbf{L}}_{aa}\right]\left(v - v_{ab}^{(t)}\right)^2,\end{aligned} \tag{27}$$

with (26) to find that $\mathbf{G}(v, v_{ab}^{(K)}) \geq \mathbf{F}_{ab}(v)$ is equivalent to

$$\frac{(\mathbf{V}\mathbf{U}^T\mathbf{K}\mathbf{U})_{ab} + \lambda(\widetilde{\mathbf{D}}\mathbf{V})_{ab}}{vab^{(t)}} \geq (\mathbf{U}^T\mathbf{K}\mathbf{U})_{bb} + \lambda\widetilde{\mathbf{L}}_{aa}. \tag{28}$$

We have

$$(\mathbf{V}\mathbf{U}^T\mathbf{K}\mathbf{U})_{ab} = \sum_{l=1}^{k} v_{al}^{(t)}(\mathbf{U}^T\mathbf{K}\mathbf{U})_{lb} \geq v_{ab}^{(t)}(\mathbf{U}^T\mathbf{K}\mathbf{U})_{bb}, \tag{29}$$

and

$$\begin{aligned}\lambda(\widetilde{\mathbf{D}}\mathbf{V})_{ab} &= \lambda\sum_{j=1}^{M}\widetilde{\mathbf{D}}_{aj}v_{jb}^{(t)} \geq \lambda\widetilde{\mathbf{D}}_{aa}v_{ab}^{(t)} \\ &\geq \lambda(\widetilde{\mathbf{D}} - \widetilde{\mathbf{W}})_{aa}v_{ab}^{(t)} = \lambda\widetilde{\mathbf{L}}_{aa}v_{ab}^{(t)}.\end{aligned} \tag{30}$$

Thus, (28) holds and $\mathbf{G}(v, v_{ab}^{(t)}) \geq \mathbf{F}_{ab}(v)$.

We can now demonstrate the convergence of Theorem 1.

Proof of Theorem 1. Replacing $\mathbf{G}(v, v_{ab}^{(t)})$ in (23) by (26) results in the update rule:

$$\begin{aligned}v_{ab}^{(t+1)} &= v_{ab}^{(t)} - v_{ab}^{(t)}\frac{\mathbf{F}'_{ab}(v_{ab}^{(K)})}{2(\mathbf{V}\mathbf{U}^T\mathbf{K}\mathbf{U})_{ab} + 2\lambda(\widetilde{\mathbf{D}}\mathbf{V})_{ab}} \\ &= v_{ab}^{(t)}\frac{(\mathbf{K}\mathbf{U} + \lambda\widetilde{\mathbf{W}}\mathbf{V})_{ab}}{(\mathbf{V}\mathbf{U}^T\mathbf{K}\mathbf{U} + \lambda\widetilde{\mathbf{D}}\mathbf{V})_{ab}}\end{aligned} \tag{31}$$

Since (26) is an auxiliary function, $\mathbf{F}'_{ab}$ is non-increasing under this updating rule.

3.5. Computational complexity analysis

In this subsection, we discuss the computational cost of the proposed RCF comparing to the standard CF and LCCF.

Based on multiplicative updating, the overall cost for RCF, CF and LCCF is $O(n^2k)$ in each iteration. Furthermore, these three methods need to construct the kernel matrix K, which requires $O(n^2m)$ operations. For graph-based methods, RCF and LCCF need to construct the p-nearest neighbor graph, which requires $O(n^2m + n^2p)$ operations. Moreover, RCF requires $O(pn^2)$ to propagate the constraints, where p is the number of nearest neighbors.

Suppose the multiplicative updates stops after t iterations, the overall cost for RCF, CF and LCCF are $O(tn^2k + n^2m + n^2p + pn^2)$, $O(tn^2k + n^2m)$ and $O(tn^2k + n^2m + n^2p)$, respectively. Since we have $p \ll n$, RCF, CF and LCCF will have the same computational complexity when dealing with high-dimensional data.

4. Experiments and analysis of the results

To demonstrate how the document clustering performance can be improved by our method, we compared RCF with KMeans, and other six CF-based related algorithms, i.e. CF [6], LCCF [7], SemiLCCF [7] CCF [15], CNPCF [17] and GDCF [5]. KMeans is the traditional k-means clustering method. CCF, SemiLCCF, CNPCF, GDCF and the proposed RCF take advantage of supervisory information to guide learning process.

4.1. Evaluation metrics

The clustering results were evaluated by comparing the achieved cluster label of each document with the original label provided by the data sets. Two standard metrics, the accuracy (AC) and the normalized mutual information metric (NMI), were selected to measure the clustering performance [7]. The accuracy is applied to compute the percentage of correctly predicted labels, and is defined as:

$$AC = \frac{\sum_{i=1}^{n}\delta(r_i, map(l_i))}{n} \tag{32}$$

where $\delta(x,\ y)$ is assigned to be 1 if $x = y$ and 0 otherwise, and $map(l_i)$ represents the permutation mapping function that maps each cluster label l_i to the corresponding label from the data set. We utilize the Kuhn-Munkres algorithm to find the best map [27]. Clearly, the larger the value of AC, the superior the clustering performance.

The normalized mutual information matrix is employed to measure the similarity of two clusters. Given two clusters C and C', the corresponding mutual information matrix is calculated as:

$$MI(C, C') = \sum_{c_i \in C, c_j{}' \in C'} p(c_i, c_j{}') \cdot log\frac{p(c_i, c_j{}')}{p(c_i) \cdot p(c_j{}')}, \tag{33}$$

where $p(c_i)$ and $p(c'_j)$ represent the probabilities that a randomly selected data belongs to the clusters c_i and c_j, respectively, and $p(c_i, c'_j)$ denotes the joint probability that the selected point belongs to both clusters at the same time. In our experiments, we used the normalized metric NMI(C, C'), which takes values ranging from 0 to 1.

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))}, \tag{34}$$

where $H(C)$ and $H(C')$ denote the entropies of C and C'. NMI takes value 1 when two selected samples are the same, and its value equals to 0 when these two sets are completely independent. A larger value of NMI indicates accurate clustering performance.

4.2. Data sets

Three data sets TDT2,[1] Reuters-21578[2] and 20 Newsgroups[3] were used in our experiments. These data sets have been widely used by researchers in the text clustering area.

The TDT2 corpus consists of 100 document clusters. In total it has 11,201 on-topic documents which are classified into 96 semantic categories. We removed those documents appearing in two or more categories and used the largest 30 categories, thus leaving us with 9394 documents in total.

Reuters-21578 corpus contains 21,578 documents in 135 categories. Those documents with multiple category labels are discarded, and the categories with more than 50 documents are kept. Therefore, we have a total of 7495 documents in 13 categories.

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html.

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html.

[3] http://labic.icmc.usp.br/.

Table 1
Details of the datasets.

Datasets	Size	Dimentions	Classes
TDT2	9394	36,771	30
Reuters-21578	8293	18,933	65
20 Newsgroups	18,808	45,434	20

The 20 Newsgroups data set consists of 20 clusters. The total number of samples is 18,808 and the feature dimension is 45,434 [28]. The preprocessed 20 Newsgroups document set proposed by Cai[4] et al. [7] was used in our experiment. We used approximately 100 documents for each individual.

The statistics of these three data sets are summarised in Table 1.

4.3. Document clustering results

4.3.1. Experimental setting

Evaluations were conducted with different cluster numbers k across a wide range. For a given k, we randomly choose k categories from the data set and mix the images of these k categories as the collection X for clustering. The corresponding labels of the selected data points were only used for evaluation not for clustering. We performed the methods for two cases: the amount of constraint information was small (e.g., $t = 2\%$) and big (e.g., $t = 20\%$), where t was the percentage of constraint information. For CCF, SemiLCCF, CNPCF, GDCF and the proposed RCF, we randomly gave t constraint information for each class in the input data. Once we obtain the new data representations, we applied k-means using cosine distance to get the clustering results. The experiment was repeated 10 times for each cluster number k, and we computed the average clustering performance in terms of AC and NMI.

Some parameters have to be predefined. LCCF and SemiLCCF have two parameters, the number of nearest neighbors p and the regularization parameter λ. We empirically set p to 5 and λ to 100. CNPCF has three parameters, the number of nearest neighbors p, and two regularization parameters λ_H and λ_S. We set p to 3, λ_H to 1 and λ_S to 10, respectively. GDCF has two trade-off parameters β_1 and β_2. We set $\beta_1 = 0.5$ and $\beta_2 = 0.1$. The max iteration number is set to 400 for all these methods. The above parameters were well set to the best values for LCCF, SemiLCCF, GDCF and CNPCF. Our proposed method has three parameters, and its analysis will be introduced later.

4.3.2. Experiments on TDT2

Tables 2–5 show the detailed clustering performance of all methods on TDT2 for the case of $t = 2\%$ and $t = 20\%$ along with its mean values and standard deviations. The last row of the table shows the average AC and NMI over the number of clusters k.

From Tables 2 and 3, we can see that when $t = 2\%$, our proposed RCF outperforms all other algorithms when the average is calculated. On average, RCF achieves 8.56% and 8.96% improvements in AC and NMI, respectively, over SemiLCCF. Also, our proposed RCF obtains 5.05% and 6.12% improvements in AC and NMI, respectively, over the second best method CNPCF.

From Tables 4 and 5, we can see that when $t = 20\%$, our proposed method achieves 3.62% and 6.12% improvements in AC and NMI, respectively, over the best algorithm other than the proposed method RCF (i.e., CNPCF).

4.3.3. Experiments on Reuters-21578

The clustering results of all methods on Reuters-21578 for the cases of $t = 2\%$ and $t = 20\%$ are shown in Tables 6–9. Once again the last row of each table shows the average AC and NMI over the number of clusters k.

From Tables 6 and 7 we can see that when $t = 2\%$, the proposed method RCF achieves a 6.04% improvement in AC and a 10.49% improvement in NMI, compared to GDCF. It also achieves 3.55% and 7.82% improvement in AC and NMI, respectively, over the best method other than RCF, i.e., CNPCF.

From Tables 8 and 9, we can see when $t = 20\%$, our proposed RCF achieves 6.33% and 17.29% improvement in AC and NMI, respectively, over the second best algorithm, i.e., CNPCF.

4.3.4. Experiments on 20Newsgroup

The detailed clustering performance on 20Newsgroup is listed in Tables 10–13 for the cases of $t = 2\%$ and $t = 20\%$. The last row of each table shows the average AC and NMI over the number of clusters k.

From Tables 10–13, we can see that two constrained methods, GDCF and CNPCF, perform worse on 20Newsgroup than on the other two data sets when the amount of supervisory information is small ($t = 2\%$). In contrast, the proposed RCF performs consistently well for all the cases. When $t = 2\%$, the proposed RCF achieves a 7.27% improvement in AC and a 12.10% improvement in NMI, compared to the second best method other than RCF (i.e. CNPCF). When $t = 20\%$, RCF achieves a 12.10% improvement in AC and a 10.99% improvement in NMI, compared to the second best method CNPCF.

4.3.5. Summary on clustering performance

The following points can be surmised from these experiments.

1. RCF always obtains the best performance on these three data sets. This indicates that the proposed method can obtain more supervisory information from the data set by propagating limited dual connected constraints and can effectively use the supervisory information to improve the performance of CF.
2. GDCF and CNPCF perform well on these three data sets. When the amount of supervisory information is small, the improved performance of these two methods are lower than that of the proposed method. This is because both of them do not have any constraints on the unconstrained data points. In addition, GDCF imposes no restriction on data points from the different classes to be mapped sufficiently far away from each other in the new representation space.
3. SemiLCCF achieves a good performance when the amount of supervisory information is large. When the amount of supervisory information is small, the performance degrades to the level of LCCF. In contrast, the proposed method obtains a satisfactory improvement compared to LCCF. The reason for the lower performance of SemiLCCF compared to the proposed method is that it does not impose restriction on data points from the different classes to be mapped sufficiently far away from each other in the new representation space. Moreover, it does not utilize the supervisory information as efficiently as RCF does.

4.3.6. Performance related to the different number of dual connected constraints

To evaluate the influence of different numbers of dual connected constraints for the performance of the proposed method, another three experiments were further conducted.

We randomly selected three different numbers of dual connected constraints, i.e., $t = 5\%, 10\%,$ and 25%, respectively. The methods were run ten times on all the clusters and the average results from these ten runs were recorded as the final performance. These results are shown in Figs. 1 and 2.

From the above experiments, our findings are summarized as follows:

[4] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html.

Table 2
Clustering AC performance on TDT2 when t = 2%.

k	KMeans	CF	LCCF	CCF	SemiLCCF	CNPCF	GDCF	RCF
3	93.10 ± 13.72	92.50 ± 11.55	95.13 ± 13.24	94.26 ± 12.60	95.31 ± 6.53	97.82 ± 17.60	**95.44 ± 14.71**	95.27 ± 0.38
6	88.18 ± 11.61	87.50 ± 16.19	92.32 ± 14.12	87.45 ± 8.25	91.71 ± 4.35	95.26 ± 6.89	92.06 ± 7.56	**98.00 ± 0.88**
9	82.38 ± 21.47	85.58 ± 8.40	86.62 ± 9.05	83.12 ± 4.07	88.69 ± 5.98	93.55 ± 6.97	85.26 ± 5.59	**92.89 ± 8.47**
12	78.91 ± 8.29	81.55 ± 9.23	78.96 ± 6.22	77.78 ± 20.79	80.26 ± 3.63	87.78 ± 5.20	86.57 ± 4.11	**91.65 ± 0.77**
15	76.18 ± 6.01	78.67 ± 6.95	75.24 ± 4.28	73.53 ± 9.83	78.37 ± 5.30	85.63 ± 5.47	81.27 ± 5.50	**90.15 ± 7.33**
18	70.38 ± 5.37	75.01 ± 7.72	76.55 ± 5.91	72.26 ± 5.29	75.93 ± 2.19	81.64 ± 3.87	79.98 ± 4.21	**89.20 ± 2.88**
21	66.29 ± 6.61	70.33 ± 8.77	76.57 ± 8.48	68.31 ± 4.72	77.89 ± 3.46	80.37 ± 4.57	75.05 ± 2.77	**86.48 ± 5.72**
24	65.51 ± 6.09	69.85 ± 5.71	74.45 ± 5.33	66.59 ± 6.36	75.28 ± 6.09	77.19 ± 3.95	73.33 ± 3.41	**84.88 ± 2.76**
27	63.60 ± 7.51	66.24 ± 6.96	73.66 ± 4.27	64.37 ± 4.59	74.86 ± 8.91	75.38 ± 7.01	72.85 ± 8.24	**85.22 ± 3.79**
30	62.41 ± 5.78	63.60 ± 5.04	72.09 ± 5.31	63.97 ± 6.97	72.47 ± 5.78	71.25 ± 4.64	70.67 ± 5.29	**82.63 ± 1.56**
Avg.	74.69	77.08	80.16	75.16	81.08	84.59	81.25	**89.64**

Table 3
Clustering NMI performance on TDT2 when t = 2%.

k	KMeans	CF	LCCF	CCF	SemiLCCF	CNPCF	GDCF	RCF
3	88.14 ± 12.29	91.66 ± 13.96	93.19 ± 11.89	92.34 ± 15.64	94.54 ± 1.86	96.47 ± 15.34	**95.29 ± 10.67**	90.54 ± 1.72
6	83.91 ± 10.09	85.38 ± 13.88	92.77 ± 11.86	86.76 ± 11.05	93.38 ± 6.15	94.29 ± 7.61	93.17 ± 8.29	**95.06 ± 2.34**
9	80.82 ± 20.07	84.60 ± 21.67	80.37 ± 21.45	80.16 ± 22.78	87.87 ± 10.78	90.68 ± 6.33	91.46 ± 4.58	**92.86 ± 4.08**
12	82.16 ± 6.12	85.17 ± 7.77	80.22 ± 8.10	78.49 ± 6.74	82.57 ± 2.14	89.47 ± 3.15	88.19 ± 4.53	**93.71 ± 0.32**
15	77.18 ± 3.75	76.75 ± 4.41	78.26 ± 4.77	80.54 ± 3.39	83.06 ± 3.51	87.97 ± 7.63	85.64 ± 7.31	**92.93 ± 4.51**
18	70.13 ± 3.39	75.56 ± 7.30	79.62 ± 4.02	71.78 ± 3.66	81.21 ± 2.47	84.53 ± 7.95	79.78 ± 7.25	**93.30 ± 2.01**
21	71.47 ± 3.05	75.16 ± 4.37	80.79 ± 2.85	69.95 ± 3.50	80.49 ± 2.63	82.76 ± 6.84	77.65 ± 7.60	**92.45 ± 2.96**
24	69.76 ± 3.90	71.39 ± 4.02	79.55 ± 3.86	69.93 ± 3.25	80.55 ± 2.57	78.26 ± 7.85	76.43 ± 2.07	**91.46 ± 1.15**
27	65.31 ± 5.57	70.65 ± 2.85	78.68 ± 4.59	67.05 ± 4.17	77.38 ± 3.34	80.87 ± 8.91	77.64 ± 9.04	**92.37 ± 1.44**
30	63.84 ± 3.50	68.23 ± 3.86	78.32 ± 6.97	65.67 ± 4.26	75.42 ± 3.19	79.52 ± 8.86	76.37 ± 5.45	**91.36 ± 1.78**
Avg.	75.27	78.46	82.18	76.27	83.65	86.48	84.16	**92.60**

Table 4
Clustering AC performance on TDT2 when t = 20%.

k	KMeans	CF	LCCF	CCF	SemiLCCF	CNPCF	GDCF	RCF
3	93.10 ± 13.72	92.50 ± 11.55	95.13 ± 13.24	95.26 ± 12.28	96.43 ± 1.22	99.38 ± 4.64	99.30 ± 6.27	**99.60 ± 2.43**
6	88.18 ± 11.61	87.50 ± 16.19	92.32 ± 14.12	93.45 ± 8.37	95.91 ± 4.56	96.66 ± 6.64	97.42 ± 8.86	**99.33 ± 1.67**
9	82.38 ± 21.47	85.58 ± 8.40	86.62 ± 9.05	86.12 ± 5.13	91.28 ± 5.24	95.65 ± 10.27	94.56 ± 9.46	**95.13 ± 5.13**
12	78.91 ± 8.29	81.55 ± 9.23	78.96 ± 6.22	86.78 ± 18.24	86.48 ± 3.19	90.21 ± 7.29	89.68 ± 9.26	**91.83 ± 7.08**
15	76.18 ± 6.01	78.67 ± 6.95	75.24 ± 4.28	83.53 ± 7.27	84.26 ± 5.31	85.63 ± 7.67	84.47 ± 8.53	**88.57 ± 5.85**
18	70.38 ± 5.37	75.01 ± 7.72	76.55 ± 5.91	79.26 ± 5.31	80.94 ± 2.28	83.64 ± 9.95	81.18 ± 6.67	**89.02 ± 5.26**
21	66.29 ± 6.61	70.33 ± 8.77	76.57 ± 8.48	75.31 ± 5.02	78.49 ± 3.07	81.17 ± 11.64	79.75 ± 12.82	**86.91 ± 7.54**
24	65.51 ± 6.09	69.85 ± 5.71	74.45 ± 5.33	75.99 ± 6.13	76.73 ± 5.56	82.37 ± 12.67	77.83 ± 8.82	**86.23 ± 0.82**
27	63.60 ± 7.51	66.24 ± 6.96	73.66 ± 4.27	74.37 ± 5.27	75.96 ± 8.31	79.25 ± 9.01	73.05 ± 10.37	**88.57 ± 1.26**
30	62.41 ± 5.78	63.60 ± 5.04	72.09 ± 5.31	70.78 ± 6.18	74.68 ± 5.23	78.37 ± 13.67	76.69 ± 6.65	**83.28 ± 1.84**
Avg.	74.69	77.08	80.16	82.09	84.12	87.23	85.49	**90.85**

Table 5
Clustering NMI performance on TDT2 when t = 20%.

k	KMeans	CF	LCCF	CCF	SemiLCCF	CNPCF	GDCF	RCF
3	88.14 ± 12.29	91.66 ± 13.96	93.19 ± 11.89	95.77 ± 15.64	95.76 ± 1.54	97.68 ± 5.68	96.47 ± 8.86	**98.21 ± 6.43**
6	83.91 ± 10.09	85.38 ± 13.88	92.77 ± 11.86	92.63 ± 11.05	94.33 ± 5.59	96.89 ± 9.02	96.07 ± 5.67	**98.35 ± 6.09**
9	80.82 ± 20.07	84.60 ± 21.67	80.37 ± 21.45	88.87 ± 22.78	92.26 ± 7.34	94.67 ± 12.58	94.35 ± 10.41	**96.47 ± 3.62**
12	82.16 ± 6.12	85.17 ± 7.77	80.22 ± 8.10	85.07 ± 6.74	87.88 ± 2.07	92.49 ± 7.56	90.93 ± 5.45	**95.16 ± 4.29**
15	77.18 ± 3.75	76.75 ± 4.41	78.26 ± 4.77	83.83 ± 3.39	85.56 ± 3.66	92.97 ± 7.25	84.23 ± 9.59	**93.46 ± 3.32**
18	70.13 ± 3.39	75.56 ± 7.30	79.62 ± 4.02	80.89 ± 3.66	83.69 ± 3.54	89.73 ± 8.29	82.82 ± 7.75	**94.05 ± 2.60**
21	71.47 ± 3.05	75.16 ± 4.37	80.79 ± 2.85	79.91 ± 3.50	82.99 ± 2.78	87.86 ± 10.58	82.06 ± 6.78	**93.30 ± 3.65**
24	69.76 ± 3.90	71.39 ± 4.02	79.55 ± 3.86	78.32 ± 3.25	80.57 ± 3.38	84.06 ± 9.89	81.24 ± 5.85	**93.49 ± 0.49**
27	65.31 ± 5.57	70.65 ± 2.85	78.68 ± 4.59	76.72 ± 4.17	79.78 ± 4.12	83.87 ± 7.63	80.88 ± 7.35	**94.27 ± 1.07**
30	63.84 ± 3.50	68.23 ± 3.86	78.32 ± 6.97	73.29 ± 4.26	78.82 ± 2.74	80.52 ± 11.28	81.73 ± 8.55	**92.55 ± 1.89**
Avg.	75.27	78.46	82.18	83.53	86.16	90.07	87.08	**94.93**

When the amount of prior information increases, the clustering performance of CF methods such as: CCF, GDCF and RCF improve in general. It should be noted that the proposed method can consistently obtain more reliable performances than the other methods. This means RCF is stable.

When the amount of prior information decreases, the clustering performance of CCF and GDCF degrade. Particularly in the case where the number of dual connected constraints is less than five percent of the total number of dual connected constraints, the clustering performance of CCF degrades dramatically. This is made possibly because CCF only utilizes the limited label information to guide the learning process, but ignores the local geometric information of the entire data in the process of clustering. Meanwhile, RCF utilizes both a prior knowledge and neighborhood information of the entire data to guide its clustering. Thus, it achieves a superior performance. GDCF suffers from performance degradation because there is not enough supervisory information or the data sets are not compact. Regardless of the data sets, our proposed method

Table 6
Clustering AC performance on Reuters when t = 2%.

k	KMeans	CF	LCCF	CCF	SemiLCCF	CNPCF	GDCF	RCF
2	79.02 ± 10.69	89.47 ± 18.15	87.64 ± 14.00	90.47 ± 12.36	91.95 ± 11.67	94.24 ± 10.28	93.59 ± 13.57	**97.20 ± 0.58**
3	66.53 ± 8.49	79.15 ± 14.00	84.64 ± 22.33	80.26 ± 15.62	86.47 ± 12.32	90.33 ± 9.64	84.14 ± 10.54	**85.73 ± 1.02**
4	65.76 ± 6.54	78.14 ± 11.93	79.94 ± 5.51	76.34 ± 5.44	80.06 ± 6.49	85.59 ± 7.34	79.04 ± 6.68	**80.75 ± 3.11**
5	62.70 ± 15.42	70.38 ± 15.98	74.26 ± 9.09	73.45 ± 10.50	74.05 ± 8.26	80.83 ± 6.36	78.67 ± 11.25	**91.56 ± 1.02**
6	56.74 ± 2.69	61.20 ± 16.45	73.25 ± 6.62	71.69 ± 2.37	71.15 ± 7.07	77.93 ± 9.94	77.15 ± 3.68	**82.57 ± 13.10**
7	56.89 ± 7.35	62.66 ± 15.31	70.25 ± 6.57	65.35 ± 7.58	72.15 ± 10.34	77.04 ± 12.61	75.51 ± 6.64	**77.14 ± 5.05**
8	50.37 ± 16.38	58.35 ± 14.43	67.33 ± 5.86	60.96 ± 9.96	64.91 ± 17.64	66.76 ± 12.08	66.55 ± 9.06	**75.00 ± 9.97**
9	50.67 ± 6.96	57.13 ± 3.91	64.64 ± 3.79	62.28 ± 11.37	63.71 ± 3.86	65.64 ± 5.23	63.94 ± 6.64	**74.47 ± 7.94**
10	52.15 ± 3.41	57.14 ± 3.01	61.96 ± 3.22	56.76 ± 4.62	60.79 ± 4.10	64.87 ± 5.28	62.29 ± 7.01	**70.82 ± 1.02**
Avg.	60.09	68.18	73.77	70.84	73.92	78.14	75.65	**81.69**

Table 7
Clustering NMI performance on Reuters when t = 2%.

k	KMeans	CF	LCCF	CCF	SemiLCCF	CNPCF	GDCF	RCF
2	52.14 ± 26.25	53.46 ± 29.42	58.59 ± 25.39	60.59 ± 21.37	62.28 ± 19.68	73.46 ± 14 34	75.99 ± 11.69	**87.42 ± 4.07**
3	52.21 ± 3.11	54.78 ± 10.08	64.84 ± 25.14	58.49 ± 10.25	66.97 ± 9.69	72.66 ± 5.90	72.09 ± 13.25	**74.15 ± 4.04**
4	58.82 ± 11.05	59.14 ± 6.12	63.47 ± 16.10	61.29 ± 15.36	65.31 ± 10.28	70.67 ± 6.56	70.24 ± 3.05	**71.97 ± 6.67**
5	49.16 ± 18.05	49.67 ± 14.06	54.67 ± 10.06	55.67 ± 12.67	63.58 ± 7.61	71.53 ± 7.26	68.41 ± 2.76	**81.28 ± 4.44**
6	53.18 ± 3.14	54.75 ± 16.03	59.46 ± 0.83	52.94 ± 10.46	61.97 ± 8.23	68.23 ± 7.68	66.79 ± 8.36	**75.27 ± 8.33**
7	50.13 ± 8.10	51.88 ± 14.11	55.82 ± 6.39	53.44 ± 9.66	58.44 ± 9.20	65.96 ± 3.64	60.35 ± 7.75	**71.21 ± 0.82**
8	45.47 ± 15.12	46.16 ± 11.04	51.49 ± 7.86	50.19 ± 12.34	54.38 ± 10.67	63.79 ± 5.68	57.29 ± 11.39	**71.79 ± 7.89**
9	43.76 ± 2.04	43.39 ± 3.01	49.67 ± 3.81	46.32 ± 5.29	50.85 ± 5.29	58.88 ± 6.35	54.43 ± 2.86	**71.11 ± 3.81**
10	51.31 ± 3.11	52.65 ± 5.16	54.86 ± 0.34	50.72 ± 3.33	52.22 ± 4.25	58.07 ± 2.67	53.61 ± 9.67	**69.47 ± 0.45**
Avg.	50.69	51.76	56.99	54.41	59.56	67.03	64.36	**74.85**

Table 8
Clustering AC performance on Reuters when t = 20%.

k	KMeans	CF	LCCF	CCF	SemiLCCF	CNPCF	GDCF	RCF
2	79.02 ± 10.69	89.47 ± 18.15	87.64 ± 14.00	96.85 ± 13.09	97.75 ± 8.64	99.82 ± 10.63	99.78 ± 7.19	**99.95 ± 2.28**
3	66.53 ± 8.49	79.15 ± 14.00	84.64 ± 22.33	95.13 ± 20.37	96.68 ± 15.52	98.67 ± 12.09	98.44 ± 2.39	**99.00 ± 4.65**
4	65.76 ± 6.54	78.14 ± 11.93	79.94 ± 5.51	91.74 ± 6.23	94.94 ± 3.07	96.06 ± 10.08	95.87 ± 2.56	**97.40 ± 2.73**
5	62.70 ± 15.42	70.38 ± 15.98	74.26 ± 9.09	80.93 ± 11.13	90.36 ± 9.61	95.39 ± 5.28	92.68 ± 6.99	**98.44 ± 1.00**
6	56.74 ± 2.69	61.20 ± 16.45	73.25 ± 6.62	79.65 ± 3.84	82.24 ± 11.22	90.61 ± 7.43	81.47 ± 1.58	**98.00 ± 0.19**
7	56.89 ± 7.35	62.66 ± 15.31	70.25 ± 6.57	75.79 ± 8.26	77.59 ± 8.81	87.43 ± 16.56	79.18 ± 7.47	**93.03 ± 8.99**
8	50.37 ± 16.38	58.35 ± 14.43	67.33 ± 5.86	74.61 ± 10.24	74.88 ± 9.82	80.96 ± 9.07	76.75 ± 4.10	**94.83 ± 0.76**
9	50.67 ± 6.96	57.13 ± 3.91	64.64 ± 3.79	73.88 ± 5.56	72.76 ± 11.37	77.96 ± 3.24	72.83 ± 5.56	**91.69 ± 1.22**
10	52.15 ± 3.41	57.14 ± 3.01	61.96 ± 3.22	74.67 ± 9.08	71.87 ± 4.43	75.31 ± 8.97	73.05 ± 12.65	**86.74 ± 6.35**
Avg.	60.09	68.18	73.77	82.58	84.34	89.13	85.56	**95.46**

Table 9
Clustering NMI performance on Reuters when t = 20%.

k	KMeans	CF	LCCF	CCF	SemiLCCF	CNPCF	GDCF	RCF
2	52.14 ± 26.25	53.46 ± 29.42	58.59 ± 25.39	80.33 ± 28.67	76.86 ± 20.37	90.99 ± 10.66	79.99 ± 18.31	**99.89 ± 3.39**
3	52.21 ± 3.11	54.78 ± 10.08	64.84 ± 25.14	72.13 ± 20.67	77.57 ± 15.07	87.48 ± 12.67	78.28 ± 9.89	**96.63 ± 1.55**
4	58.82 ± 11.05	59.14 ± 6.12	63.47 ± 16.10	71.78 ± 14.02	75.24 ± 7.26	83.67 ± 6.68	75.24 ± 14.37	**93.01 ± 1.02**
5	49.16 ± 18.05	49.67 ± 14.06	54.67 ± 10.06	63.43 ± 10.32	69.86 ± 11.37	80.56 ± 13.21	70.27 ± 12.38	**96.59 ± 2.37**
6	53.18 ± 3.14	54.75 ± 16.03	59.46 ± 0.83	61.84 ± 5.84	63.73 ± 3.77	78.24 ± 10.27	69.99 ± 13.27	**95.64 ± 0.53**
7	50.13 ± 8.10	51.88 ± 14.11	55.82 ± 6.39	58.43 ± 9.47	58.77 ± 12.28	77.29 ± 8.07	65.73 ± 5.27	**92.90 ± 5.68**
8	45.47 ± 15.12	46.16 ± 11.04	51.49 ± 7.86	55.89 ± 8.19	56.89 ± 10.04	66.28 ± 11.16	61.94 ± 4.91	**93.39 ± 1.38**
9	43.76 ± 2.04	43.39 ± 3.01	49.67 ± 3.81	53.37 ± 6.21	55.89 ± 3.76	64.79 ± 5.23	55.68 ± 7.43	**90.93 ± 2.20**
10	51.31 ± 3.11	52.65 ± 5.16	54.86 ± 0.34	54.99 ± 4.89	55.36 ± 8.46	62.86 ± 6.39	56.61 ± 4.97	**88.70 ± 4.14**
Avg.	50.69	51.76	56.99	63.08	65.57	76.91	68.19	**94.20**

Table 10
Clustering AC performance on 20Newsgroup when t = 2%.

k	KMeans	CF	LCCF	CCF	SemiLCCF	CNPCF	GDCF	RCF
2	83.80 ± 12.06	92.50 ± 12.57	96.25 ± 2.35	93.33 ± 3.67	97.57 ± 8.58	97.89 ± 5.67	98.50 ± 1.37	**99.00 ± 0.67**
3	70.63 ± 15.05	79.07 ± 17.64	85.20 ± 14.77	82.97 ± 12.19	89.41 ± 9.59	92.34 ± 8.79	90.46 ± 7.13	**94.70 ± 11.16**
4	58.88 ± 10.63	59.43 ± 17.03	70.23 ± 15.27	62.83 ± 10.87	78.93 ± 13.66	82.95 ± 15.20	80.75 ± 10.34	**89.78 ± 10.18**
5	55.82 ± 7.63	59.92 ± 7.02	70.72 ± 9.31	60.38 ± 8.71	75.67 ± 10.43	78.67 ± 12.28	75.40 ± 11.26	**87.66 ± 9.99**
6	47.95 ± 9.47	54.95 ± 11.16	61.17 ± 8.60	56.17 ± 8.46	64.83 ± 10.37	75.60 ± 6.75	64.33 ± 9.68	**81.25 ± 8.68**
7	53.57 ± 7.83	52.76 ± 4.56	58.67 ± 9.29	55.00 ± 11.09	60.77 ± 7.66	67.00 ± 7.49	60.27 ± 12.64	**72.89 ± 11.72**
8	46.55 ± 12.20	54.76 ± 9.59	58.78 ± 9.09	55.41 ± 9.89	59.63 ± 8.35	65.60 ± 9.45	58.63 ± 10.84	**80.63 ± 7.12**
9	44.20 ± 5.63	47.17 ± 8.14	54.89 ± 6.86	50.30 ± 12.28	56.43 ± 10.31	63.57 ± 8.67	60.20 ± 9.79	**74.06 ± 10.34**
10	42.40 ± 4.04	45.15 ± 4.63	49.50 ± 5.89	47.90 ± 8.64	52.04 ± 11.47	63.20 ± 8.24	62.00 ± 11.52	**72.27 ± 5.80**
Avg.	55.98	60.63	67.27	62.70	70.59	76.31	72.28	**83.58**

Table 11
Clustering NMI performance on 20Newsgroup when t = 2%.

k	KMeans	CF	LCCF	CCF	SemiLCCF	CNPCF	GDCF	RCF
2	46.16 ± 24.98	73.12 ± 23.79	78.28 ± 10.91	57.10 ± 18.64	80.33 ± 12.62	81.05 ± 9.46	80.25 ± 6.73	**92.84 ± 4.30**
3	42.96 ± 15.96	65.32 ± 19.15	68.43 ± 14.73	58.81 ± 16.43	71.88 ± 13.40	75.50 ± 11.24	70.63 ± 7.42	**88.80 ± 11.25**
4	32.64 ± 13.43	46.70 ± 19.08	56.73 ± 12.36	57.20 ± 11.64	58.24 ± 10.29	73.80 ± 8.49	68.24 ± 13.37	**82.01 ± 10.50**
5	34.13 ± 9.06	49.65 ± 10.41	61.01 ± 8.65	56.96 ± 9.86	62.19 ± 10.67	65.76 ± 14.17	63.85 ± 16.22	**82.01 ± 9.89**
6	29.33 ± 8.92	49.63 ± 11.07	53.27 ± 6.54	57.34 ± 6.71	58.78 ± 7.88	65.88 ± 9.38	60.21 ± 11.34	**75.97 ± 7.07**
7	37.46 ± 7.98	48.97 ± 5.82	53.50 ± 8.90	59.08 ± 7.45	56.41 ± 8.13	65.50 ± 11.27	57.02 ± 8.07	**71.46 ± 10.72**
8	33.35 ± 12.98	50.49 ± 10.47	53.54 ± 8.38	55.28 ± 8.16	55.33 ± 7.78	64.48 ± 10.46	56.10 ± 7.41	**78.79 ± 6.41**
9	34.32 ± 5.45	47.18 ± 7.98	51.47 ± 6.33	55.73 ± 6.32	53.51 ± 6.01	60.67 ± 9.66	56.01 ± 5.88	**74.34 ± 9.48**
10	32.07 ± 4.81	45.41 ± 4.07	48.69 ± 4.05	50.22 ± 5.67	51.91 ± 5.42	56.76 ± 10.13	53.88 ± 4.12	**72.07 ± 3.44**
Avg.	35.82	52.94	58.32	56.41	60.95	67.71	62.91	**79.81**

Table 12
Clustering AC performance on 20Newsgroup when t = 20%.

k	KMeans	CF	LCCF	CCF	SemiLCCF	CNPCF	GDCF	RCF
2	83.80 ± 12.06	92.50 ± 12.57	96.25 ± 2.35	97.05 ± 11.10	98.57 ± 9.41	99.89 ± 8.40	99.50 ± 3.16	**99.00 ± 0.67**
3	70.63 ± 15.05	79.07 ± 17.64	85.20 ± 14.77	86.47 ± 16.31	93.51 ± 10.24	95.56 ± 9.37	94.75 ± 15.43	**94.70 ± 11.16**
4	58.88 ± 10.63	59.43 ± 17.03	70.23 ± 15.27	77.50 ± 14.33	83.63 ± 11.28	92.43 ± 9.05	90.88 ± 13.60	**89.78 ± 10.18**
5	55.82 ± 7.63	59.92 ± 7.02	70.72 ± 9.31	75.48 ± 9.62	78.20 ± 11.14	88.90 ± 9.48	85.26 ± 13.07	**87.66 ± 9.99**
6	47.95 ± 9.47	54.95 ± 11.16	61.17 ± 8.60	68.33 ± 8.03	74.67 ± 14.12	85.25 ± 10.24	74.41 ± 9.33	**81.25 ± 8.68**
7	53.57 ± 7.83	52.76 ± 4.56	58.67 ± 9.29	66.89 ± 12.31	76.87 ± 15.20	79.75 ± 11.72	70.18 ± 11.03	**72.89 ± 11.72**
8	46.55 ± 12.20	54.76 ± 9.59	58.78 ± 9.09	65.41 ± 5.19	71.98 ± 7.86	75.78 ± 10.13	76.54 ± 5.42	**80.63 ± 7.12**
9	44.20 ± 5.63	47.17 ± 8.14	54.89 ± 6.86	53.75 ± 6.66	66.58 ± 8.42	72.80 ± 10.22	71.24 ± 2.79	**74.06 ± 10.34**
10	42.40 ± 4.04	45.15 ± 4.63	49.50 ± 5.89	52.89 ± 12.07	64.73 ± 9.76	71.76 ± 9.84	73.55 ± 5.07	**72.27 ± 5.80**
Avg.	55.98	60.63	67.27	71.53	78.75	84.68	81.70	**96.78**

Table 13
Clustering NMI performance on 20Newsgroup when t = 20%.

k	KMeans	CF	LCCF	CCF	SemiLCCF	CNPCF	GDCF	RCF
2	46.16 ± 24.98	73.12 ± 23.79	78.28 ± 10.91	80.45 ± 15.22	85.75 ± 19.37	98.45 ± 4.77	98.75 ± 5.63	**100.00 ± 4.30**
3	42.96 ± 15.96	65.32 ± 19.15	68.43 ± 14.73	68.25 ± 16.41	84.05 ± 14.64	93.85 ± 14.87	86.27 ± 10.75	**97.07 ± 11.25**
4	32.64 ± 13.43	46.70 ± 19.08	56.73 ± 12.36	65.76 ± 13.44	78.89 ± 12.43	90.21 ± 13.37	80.69 ± 9.08	**96.67 ± 10.50**
5	34.13 ± 9.06	49.65 ± 10.41	61.01 ± 8.65	63.68 ± 9.76	79.25 ± 10.66	88.97 ± 11.20	81.74 ± 11.08	**97.95 ± 9.89**
6	29.33 ± 8.92	49.63 ± 11.07	53.27 ± 6.54	60.78 ± 10.07	77.24 ± 9.82	85.88 ± 9.14	78.95 ± 10.78	**97.50 ± 7.07**
7	37.46 ± 7.98	48.97 ± 5.82	53.50 ± 8.90	61.39 ± 8.87	72.79 ± 9.73	81.50 ± 5.64	73.38 ± 7.09	**93.21 ± 10.72**
8	33.35 ± 12.98	50.49 ± 10.47	53.54 ± 8.38	65.68 ± 11.27	65.41 ± 12.37	79.48 ± 11.36	66.77 ± 10.08	**97.76 ± 6.41**
9	34.32 ± 5.45	47.18 ± 7.98	51.47 ± 6.33	58.28 ± 9.74	63.85 ± 8.29	75.67 ± 8.81	66.25 ± 7.46	**94.07 ± 9.48**
10	32.07 ± 4.81	45.41 ± 4.07	48.69 ± 4.05	55.37 ± 10.21	62.64 ± 7.47	74.06 ± 5.45	65.40 ± 9.28	**92.80 ± 3.44**
Avg.	35.82	52.94	58.32	64.40	74.43	85.34	77.58	**96.34**

always achieves a higher result. This indicates that RCF can efficiently leverage the power of dual connected constraints and graph Laplacian regularization to obtain a greater performance.

4.4. Constraint effect

The effect of the constraints is evaluated in this subsection. 20Newsgroup data set is chosen to conduct a thorough study on the effect of the must-connected and cannot-connected constraints.

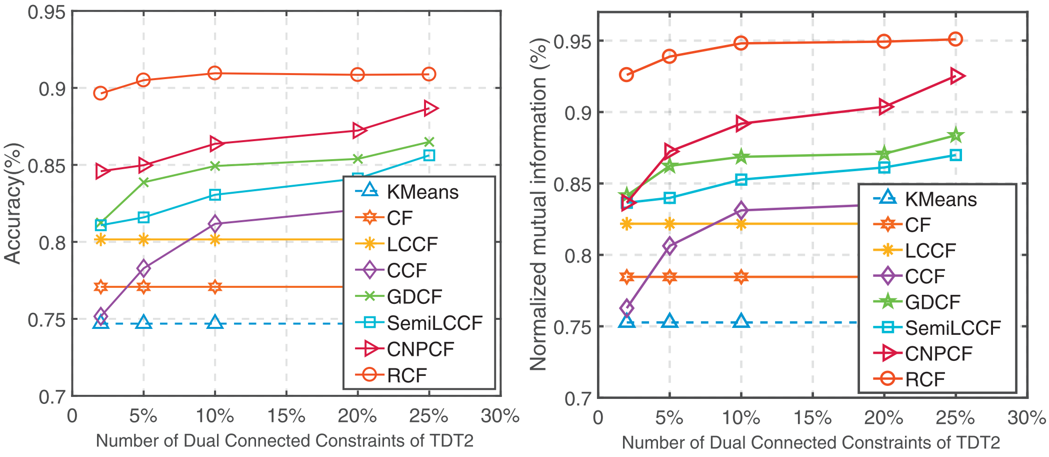
Fig. 3 shows the average clustering accuracy verses increasing number of dual connected constraints on 20Newsgroup data set. It can be seen that RCF achieves the best performance when both must-connected and cannot-connected constraints are utilized, where the must-connected constraints have larger importance than cannot-connected constraints. Despite this, cannot-connected can also improve the performance of the proposed method as the accuracy with cannot-connected constraints is higher than the case when it does not use any cannot-connected constraint.

4.5. Parameters analysis

We further investigated the influences of different parameter settings on the clustering performance of RCF, leading to the best choice of the learning model. There are three parameters in RCF, i.e. the number of nearest neighbors p, the constraint propagation parameter α, and the regulation parameter λ. We randomly selected nine classes and five classes from TDT2 and Reuters-21578, respectively, and generated $t = 2\%$ and $t = 20\%$ dual connected constraints from these classes, while independently performing each trail 20 times. The average clustering performance is recorded as the final result. Figs. 4–6 show how AC and NMI vary with these parameters. For TDT2, the parameters are set to $\alpha = 0.5$, $p = 5$ and $\lambda = 100$. For Reuters-21578, α is set to 0.3, p is set to 4, and λ is set to 100.

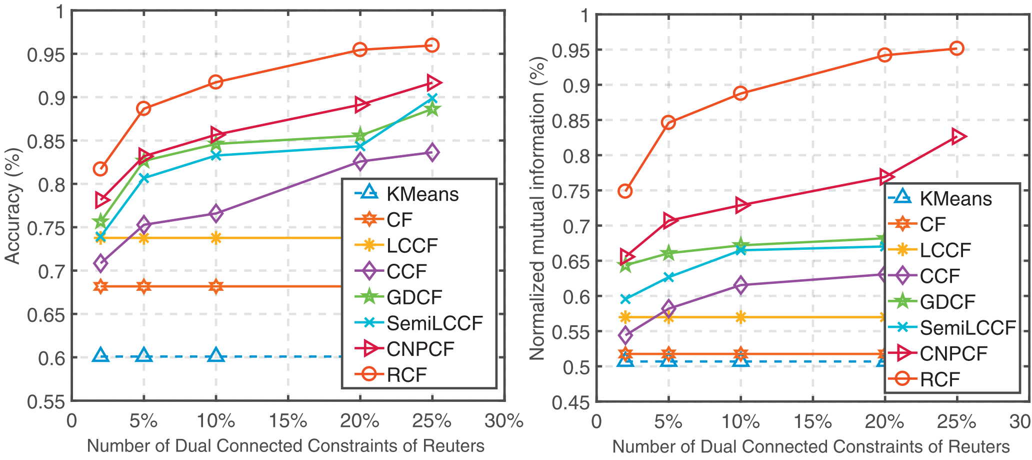
5. Conclusions and future work

In this paper, we propose a novel regularized concept factorization (RCF) method, which propagates scarce constraint information to obtain more supervisory information and uses this supervisory information to preserve the geometrical structure of the data space. The proposed method imposes restrictions on samples both belonging to the same class and a different class such that it can achieve a more discriminating ability. In addition, the updating rules of our proposed algorithm are given and its convergence is proved theoretically. The experimental results on three data sets (TDT2, Reuters-21578, and 20Newsgroup) demonstrate that the proposed method can achieve a more superior performance than other state-of-the-art clustering algorithms.



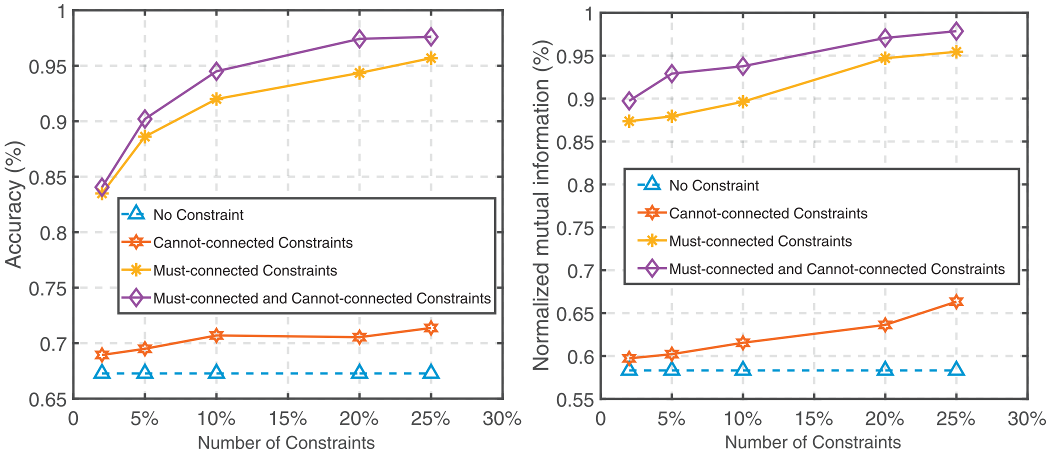
(a) (b)

Fig. 1. Clustering performance on TDT2 with different numbers of dual connected constraints (a) Accuracy (b) Normalized mutual information.



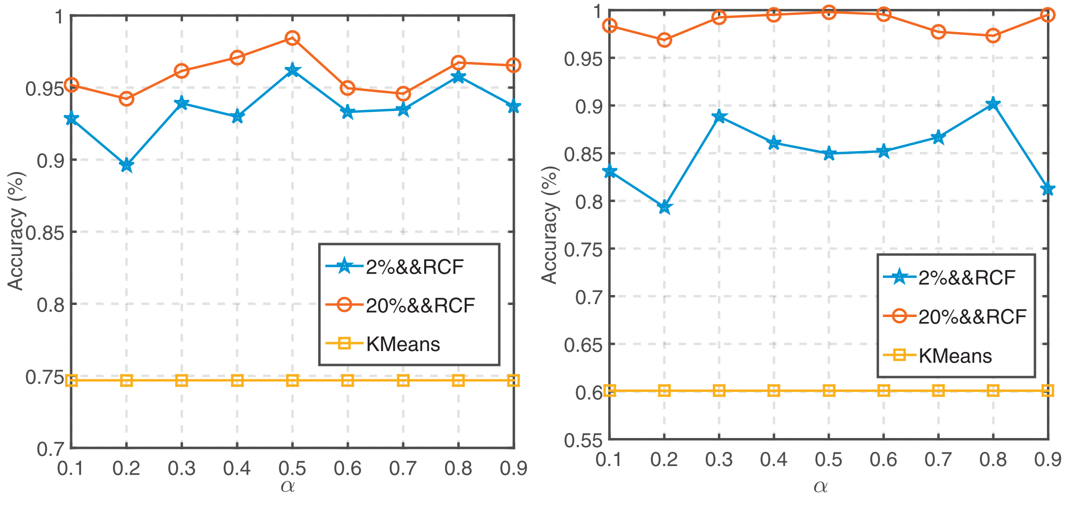
(a) (b)

Fig. 2. Clustering performance on Reuters with different numbers of dual connected constraints (a) Accuracy (b) Normalized mutual information.



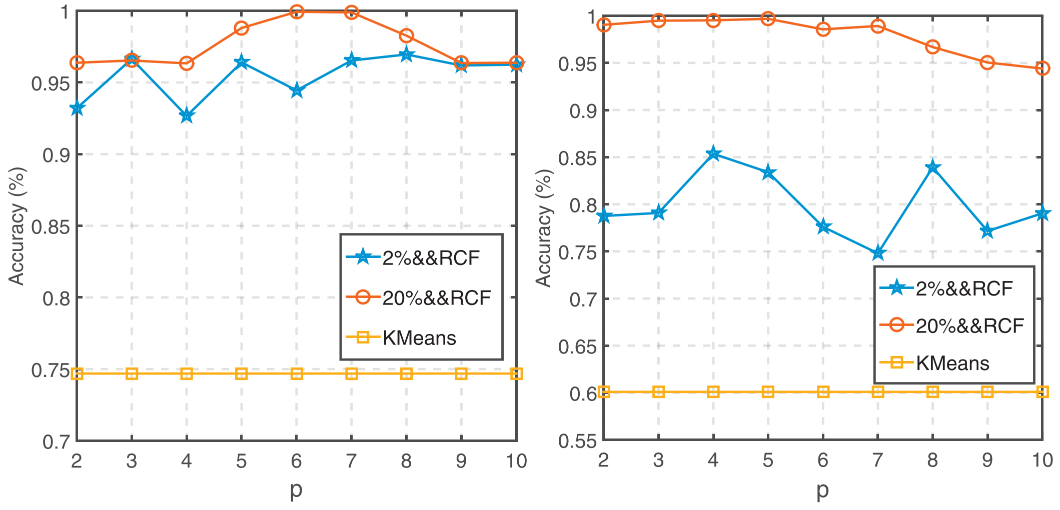
(a) (b)

Fig. 3. Clustering performance with varied number of constraints (a) Accuracy (b) Normalized mutual information.



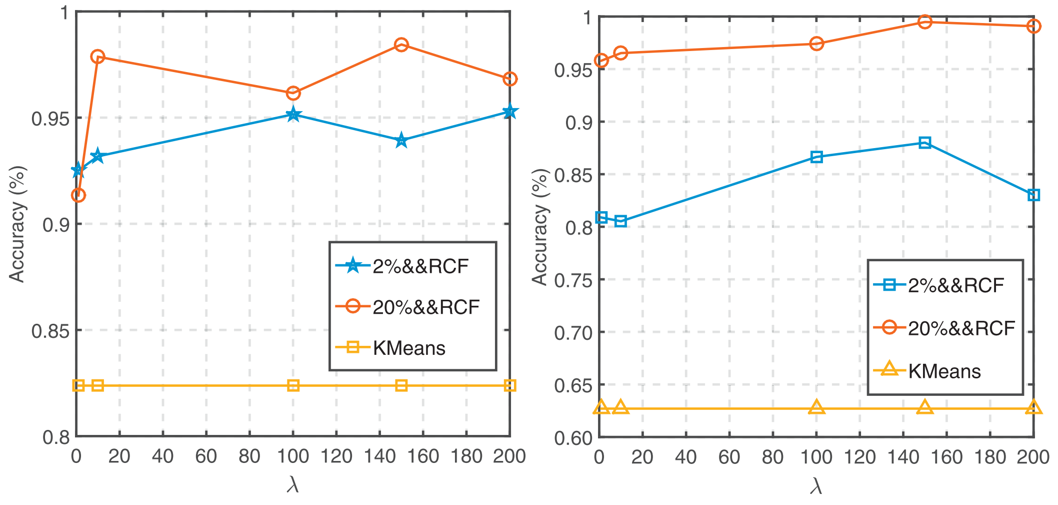
(a) (b)

Fig. 4. Clustering performance Accuracy with varied parameter α on different data sets (a) TDT2 (b) Reuters.



(a) (b)

Fig. 5. Clustering performance Accuracy with varied parameter p on different data sets (a) TDT2 (b) Reuters.



(a) (b)

Fig. 6. Clustering performance Accuracy with varied parameter λ on different data sets (a) TDT2 (b) Reuters.

In the future, we will bring other kinds of supervisory information such as multi-label and tag into RCF and we plan to extend RCF to a version that can be applied to multi-view document clustering.

Acknowledgments

This work is supported by the Science and Technology Development Fund (FDCT) of Macao SAR (128/2013/A and 124/2014/A3), the National Natural Science Foundation of China (61602540, 61722304), the Guangdong Natural Science Funds for Distinguished Young Scholar (2014A030306037), the Pearl River S&T Nova Program of Guangzhou (201610010196), and the Guangdong Program for Support of Top-notch Young Professionals (2014TQ01X144).

References

[1] B.S. Kumar, V. Ravi, A survey of the applications of text mining in financial domain, Knowl. Based Syst. 114 (2016) 128–147.
[2] X. Pei, T. Wu, C. Chen, Automated graph regularized projective nonnegative matrix factorization for document clustering, IEEE Trans. Cybern. 44 (10) (2014) 1821–1831.
[3] W. Xu, X. Liu, Y. Gong, Document clustering based on non-negative matrix factorization, in: Proceedings of the ACM SIGIR'03, 2003, pp. 267–273.
[4] Z. Yang, Y. Zhang, W. Yan, Y. Xiang, S. Xie, A fast non-smooth nonnegative matrix factorization for learning sparse representation, IEEE Access 4 (2016) 5161–5168.
[5] H. Li, J. Zhang, J. Hu, C. Zhang, J. Liu, Graph-based discriminative concept factorization for data representation, Knowl. Based Syst. 118 (2017) 70–79.
[6] W. Xu, Y. Gong, Document clustering by concept factorization, in: Proceedings of the ACM SIGIR'04, 2004, pp. 202–209.
[7] D. Cai, X. He, J. Han, Locally consistent concept factorization for document clustering, IEEE Trans. Knowl. Data Eng. 23 (6) (2011) 902–913.
[8] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.
[9] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1548–1560.
[10] S.E. Palmer, Hierarchical structure in perceptual representation, Cogn.Psychol. 9 (4) (1977) 441–474.
[11] H. Liu, Z. Yang, J. Yang, Z. Wu, X. Li, Local coordinate concept factorization for image representation, IEEE Trans. Neural Netw. Learn. Syst. 25 (6) (2014) 1071–1082.
[12] O. Chapelle, B. Schlkopf, A. Zien, Semi-Supervised Learning, The MIT Press, 2010.
[13] W. Zhang, X. Tang, T. Yoshida, Tesc: an approach to text classification using semi-supervised clustering, Knowl. Based Syst. 75 (2015) 152–160.
[14] N. Piroonsup, S. Sinthupinyo, Semi-supervised cluster-and-label with feature based re-clustering to reduce noise in thai document images, Knowl. Based Syst. 90 (2015) 58–69.
[15] H. Liu, G. Yang, Z. Wu, D. Cai, Constrained concept factorization for image representation, IEEE Trans. Cybern. 44 (7) (2014) 1214–1224.
[16] Z. Shu, C. Zhao, P. Huang, Local regularization concept factorization and its semi-supervised extension for image representation, Neurocomputing 158 (2015) 1–12.
[17] M. Lu, L. Zhang, X.-J. Zhao, F.-Z. Li, Constrained neighborhood preserving concept factorization for data representation, Knowl. Based Syst. 102 (2016) 127–139.
[18] Z. Lu, Y. Peng, Exhaustive and efficient constraint propagation: a graph-based learning approach and its applications, Int. J. Comput. Vis. 103 (3) (2013) 306–325.
[19] Z. Fu, Z. Lu, H.H.S. Ip, H. Lu, Y. Wang, Local similarity learning for pairwise constraint propagation, Multimed. Tools Appl. 74 (11) (2015) 3739–3758.
[20] X. Li, X. Shen, Z. Shu, Q. Ye, C. Zhao, Graph regularized multilayer concept factorization for data representation, Neurocomputing 238 (2017) 139–151.
[21] J. Ye, Z. Jin, Dual-graph regularized concept factorization for clustering, Neurocomputing 138 (2014) 120–130.
[22] W. Hua, X. He, Discriminative concept factorization for data representation, Neurocomputing 74 (18) (2011) 3800–3807.
[23] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency., in: Adv. Neural Inf. Process, Syst., 16, 2003, pp. 321–328.
[24] T. Jin, J. Yu, J. You, K. Zeng, C. Li, Z. Yu, Low-rank matrix factorization with multiple hypergraph regularizer, Pattern Recognit. 48 (3) (2015) 1011–1022.
[25] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Adv. Neural Inf. Process, Syst., 2001, pp. 556–562.
[26] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the em algorithm, J. R. Stat. Soc. Ser. B (Methodological) (1977) 1–38.
[27] L. Lovász, M.D. Plummer, Matching theory, 367, American Mathematical Society, 2009.
[28] R.G. Rossi, R.M. Marcacini, S.O. Rezende, Benchmarking Text Collections for Classification and Clustering Tasks, Institute of Mathematics and Computer Sciences, University of Sao Paulo, 2013.