Possibilistic fuzzy co-clustering of large document collections

William-Chandra Tjhi, Lihui Chen*

Nanyang Technological University, Republic of Singapore

Received 19 September 2006; received in revised form 11 April 2007; accepted 21 April 2007

Abstract

In this paper we propose a new co-clustering algorithm called possibilistic fuzzy co-clustering (PFCC) for automatic categorization of large document collections. PFCC integrates a possibilistic document clustering technique and a combined formulation of fuzzy word ranking and partitioning into a fast iterative co-clustering procedure. This novel framework brings about simultaneously some benefits including robustness in the presence of document and word outliers, rich representations of co-clusters, highly descriptive document clusters, a good performance in a high-dimensional space, and a reduced sensitivity to the initialization in the possibilistic clustering. We present the detailed formulation of PFCC together with the explanations of the motivations behind. The advantages over other existing works and the algorithm's proof of convergence are provided. Experiments on several large document data sets demonstrate the effectiveness of PFCC.

© 2007 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: Co-clustering; Possibilistic clustering; Fuzzy clustering; Document clustering; Text mining; Information retrieval

1. Introduction

A recent study estimates that there are at least 11.5 billion Web documents in the surface Web alone, i.e. a portion of the Web that is indexed by search engines [1]. This arguably makes them the primary source of information in our world today. As we know it, textual contents still make up the majority of all the published contents in the Web. Development of techniques that enable the analysis and organization of large collections of textual data should therefore remain a vital part in the advancement of the knowledge discovery technology. Categorization of data into a set of similar groups is a natural and basic mean for the data analysis purpose [2]. In the data mining field, data clustering is one of the two techniques (the other is classification) that could realize an automatic categorization of a set of objects [3]. There are many existing clustering algorithms suitable to categorize large document collections. They range from being the variants of the simple \( K \)-means or HAC [4], to those that adopt more complex approaches such as the ones found in Refs. [5,6].

* Corresponding author. Tel.: +65 67904484.
E-mail addresses: william_chandra@pmail.ntu.edu.sg (W.-C. Tjhi), elhchen@ntu.edu.sg (L. Chen).

In this paper, our aim is to introduce a new clustering algorithm called possibilistic fuzzy co-clustering (PFCC) for automatic categorization of large document collections. This algorithm simultaneously and efficiently addressed several issues in document clustering including: outlier-documents detection, rich representations of clusters through possibilistic and fuzzy memberships modeling, and a dual document–word clustering for the generation of more interpretable and accurate document clusters. To our knowledge, there has not been any similar effort made before.

In PFCC, we target to keep the benefits of co-clustering, possibilistic clustering, and fuzzy clustering simultaneously. As indicated by its name, the new algorithm is essentially a dual possibilistic document clustering and fuzzy word clustering technique. Since it is a co-clustering algorithm, PFCC generates not only document clusters but also their respectively associated word clusters. The latter naturally captures the contents of the document clusters, making them more easily understandable. In addition, as the words are being clustered, the document clustering process moves into a lower-dimension subspace. With this, PFCC can avoid the curse of dimensionality problem. The proposed algorithm goes even further beyond what a standard co-clustering algorithm can offer. The combination of the possibilistic and the fuzzy clustering
not only provides richer representations but also improves the quality of the resulting co-clusters. On the one hand, the possibilistic document clustering ensures documents are categorized according to their natural typicality to the clusters. At the same time it also makes the algorithm less vulnerable to the presence of outliers. On the other hand, the fuzzy word clustering of PFCC is designed in such a way it captures both the words’ importance ranking measures in the clusters (word ranking) and the overlapping categorization of words (word partitioning). As we shall discuss in Section 3.1, this arrangement enables an easy identification of outlier words in the resulting word clusters and also reduces the sensitivity to initialization that is known to affect many possibilistic clustering algorithms.

The possibilistic fuzzy co-clustering in PFCC is formulated as a problem of optimizing a newly defined objective function. The optimization problem is then solved by applying a fast iterative procedure which, under one reasonable assumption for categorization of a large document collection, can be mathematically proven to converge to a local optimum. In the implementation, to derive possibilistic updating rule, we devise an incremental updating approach based on an objective function approximation. This way, the otherwise potentially more complex and time-consuming method (which involves solving a third degree polynomial problem) to achieve the same goal can be avoided. Consequently, PFCC can be run with efficient time complexity, making it fast enough for real-world categorization tasks.

The remaining of this paper is organized as follows. In Section 2, some explanations of related concepts accompanied by literature reviews of some related algorithms are presented. We then discuss the new algorithm, its benefits, and the theoretical study in Section 3. Section 4 shows the experimental results on several large document data sets, including some performance comparisons and discussions. Finally, the paper is concluded and some future directions are suggested in Section 5.

2. Background

This section discusses some related clustering techniques along with their important concepts that can be useful in understanding PFCC in the next section. Along the way, literature reviews of some related algorithms are also provided. We divide this section into three main subsections namely, fuzzy clustering, possibilistic clustering, and co-clustering. Table 1 shows the list of all mathematical notations used in this paper.

### 2.1. Fuzzy clustering

#### 2.1.1. Overview of fuzzy clustering

Perhaps the most influential fuzzy clustering algorithm is fuzzy C-means (FCM) originally proposed by Bezdek in Ref. [7]. The aim of FCM is to minimize the objective function in Eq. (1), subject to the membership constraint in Eq. (2). Note that here we are using the version of FCM in Ref. [8], which may be considered as a variant of the general maximum-entropy clustering framework studied in Refs. [9,10]. This version of FCM is more closely related to the proposed PFCC.

\[
J_{FCM} = \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci} \text{Dist}(x_i, p_c) + T_u \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci} \ln u_{ci}, \quad (1)
\]

\[
\sum_{c=1}^{C} u_{ci} = 1 \quad \text{for } i = 1, 2, \ldots, N. \quad (2)
\]

The algorithm essentially tries to group similar objects together based on their distances to the clusters (or cluster prototypes), i.e. \( \text{Dist}(x_i, p_c) \). Minimization of the first term in Eq. (1) forces each object to have a higher membership to a cluster it has a closer distance to. In this way, similar objects will be clustered together. The second term of Eq. (1) is the entropy term [8], which controls the fuzziness of the resulting clusters. The larger the parameter \( T_u \) value, the fuzzier the outcomes will be, and vice versa. If \( \text{Dist}(x_i, p_c) \) is a Euclidean distance, the minimization of Eq. (1) can be solved by alternatively updating Eqs. (3) and (4) until the convergence is achieved:

\[
u_{ci} = \frac{\exp\{-\text{Dist}(x_i, p_c)/T_u\}}{\sum_{f=1}^{C} \exp\{-\text{Dist}(x_i, p_f)/T_u\}}, \quad (3)
\]

\[
p_c = \frac{\sum_{i=1}^{N} u_{ci} x_i}{\sum_{i=1}^{N} u_{ci}}. \quad (4)
\]

#### 2.1.2. The partitioning membership

Based on Eq. (3), an object membership \( u_{ci} \) in FCM essentially measures the similarity between an object \( i \) and a cluster \( c \), relative to the similarities between the object \( i \) and all the other clusters. This notion of “relative to other clusters” is uniquely associated with the membership constraint in Eq. (2). This is because the constraint, from its definition, forces the determination of a membership to a cluster to be dependent on the states of the other clusters. This membership characteristic is important in understanding the difference between fuzzy clustering and possibilistic clustering. To avoid confusion in later discussions, from this point onward, we will refer to
The root of the problem lies in the constraint in Eq. (2), or the problem depicted in Fig. 1. In Fig. 1, set $P$. To illustrate this limitation, let us consider a clustering algorithm, possibilistic C-means (PCM) [14], was primarily motivated by the limitation of FCM in handling outliers in a data set. Despite both $P$ and $Q$ being right in the middle of the two clusters, intuitively it would be more appropriate if $Q$ has greater membership values than $P$ to $C_1$ and $C_2$. This is obvious because $P$ is physically farther away from the two clusters than $Q$ is. If we apply FCM to this problem however, we will end up with both $P$ and $Q$ having 0.5 membership to each $C_1$ and $C_2$ (i.e. $u_{1P} = u_{1Q} = 0.5$ and $u_{2P} = u_{2Q} = 0.5$). The root of the problem lies in the constraint in Eq. (2), or equivalently the partitioning membership of FCM. Let us revisit Eq. (3) and express it in the form of $u_{ei} = A_{ci}/\sum_{i=1}^{C} A_{fi}$, where $A_{ci} = \exp(-\text{Dist}(x_i, p_c)/\eta_c)$. The distance from $P$ to $C_1$ is greater than the distance from $Q$ to $C_1$, resulting in the similarities $A_{1P} < A_{1Q}$. But this similarity between $P$ and $C_1$, relative to the similarities between $P$ and all the other clusters (as captured by $A_{1P}/\sum_{i=1}^{C} A_{fi}$) is equal to the similarity between $Q$ and $C_1$, relative to the similarities between $Q$ and all the other clusters (as captured by $A_{1Q}/\sum_{i=1}^{C} A_{fi}$). For this reason, both $P$ and $Q$ have exactly the same memberships. It should be evident from this discussion that the cause of this misinterpretation is the notion of “relative to other clusters” in the partitioning membership.

Possibilistic clustering addresses this problem by removing the constraint in Eq. (2) from the formulation. In doing so, the partitioning membership is transformed into the possibilistic membership. Eq. (5) shows the objective function of the PCM [15]. We use the variant of PCM proposed in Ref. [15] because it is more closely related to PFCC.

$$J_{PCM} = \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ei}^{pos} \text{Dist}(x_i, p_c) + \sum_{c=1}^{C} \eta_c \sum_{i=1}^{N} (u_{ei}^{pos} \ln u_{ei}^{pos} - u_{ei}^{pos})$$

(5)

The goal of PCM is to minimize $J_{PCM}$, and unlike in FCM, this minimization is not subject to any membership constraint. The first term of Eq. (5) is exactly the same as the first term of Eq. (1). Without imposing any constraint on the $u_{ei}^{pos}$ membership, the minimization of this term forces the membership values to be as small as possible. The minimization of the second term of Eq. (5) on the other hand, balances this by pushing the membership values to be as large as possible. $\eta_c$ is a user-defined parameter and the detail on how to set its value is discussed in Ref. [14]. If $\text{Dist}(x_i, p_c)$ is a Euclidean distance, Eq. (5) can be minimized by alternatively updating Eq. (6) below and Eq. (4) shown in Section 2.1.1.

$$u_{ei}^{pos} = \exp \left\{ -\frac{\text{Dist}(x_i, p_c)}{\eta_c} \right\}$$

(6)

2.2.2. The possibilistic membership

Based on Eq. (6), it is obvious that the value of a possibilistic membership $u_{ei}^{pos}$ depends only on the distance between an object $i$ and a cluster $c$. In other words, unlike the partitioning membership, the possibilistic membership is independent from the states of other clusters. Consequently, an outlier like the object $P$ in Fig. 1 would be assigned low possibilistic membership values to $C_1$ and $C_2$ because its distances to both clusters are large. This explains how a possibilistic clustering, due to the characteristic of its possibilistic membership, is able to minimize the impact of outliers present in a data set [14].

2.2.3. The coincident cluster problem

Despite their known robustness, many possibilistic clustering algorithms including PCM suffer one notorious problem generally referred to as the coincident cluster problem [16]. To understand this problem, let us decompose $J_{PCM}$ in Eq. (5) into $J_{PCM_1}, J_{PCM_2}, \ldots, J_{PCM_C}$. Since the latter are independent from one another (due to the absence of the constraint in Eq. (2)), the global minimum of $J_{PCM}$ can be achieved if $J_{PCM_1} = J_{PCM_2} = \cdots = J_{PCM_C} = \min_{c=1,...,C} J_{PCM_c}$ [17]. This corresponds to a condition where all the clusters coinciding exactly in the same location. Because of this problem,
without any proper initialization, PCM, instead of discovering the intended clusters, often results in coincident clusters. This coincident cluster problem does not affect FCM due to the influence of the constraint in Eq. (2). In Section 3, we will discuss how we tackle the coincident cluster problem in the formulation of PFCC by integrating possibilistic and (fuzzy) partitioning memberships in the objective function.

2.2.4. Other possibilistic clustering algorithms

A number of extensions to PCM have been proposed in the literatures. Many of them partially share the same objective of finding a better formulation to avoid the coincident cluster problem. The primary approach here is by incorporating some element of fuzzy clustering (i.e. the partitioning membership) into the formulation [18–20]. Some other approaches are to tackle the coincident cluster problem by including the robust estimation [21] and the cluster repulsion [17] techniques.

2.3. Co-clustering

2.3.1. Overview of co-clustering

Co-clustering is a technique that simultaneously clusters objects and features (the object’s attributes) [22]. There are several reasons why the co-clustering is argued to be better than the standard clustering. In clustering a high-dimensional data set, usually not all features are important in any particular object cluster [23]. In co-clustering, it is possible to limit the scope of data analysis on relevant features only. This ability to dynamically filter out irrelevant features in the co-clustering approach provides a mechanism to perform a feature selection process on the fly. This in turn can increase the accuracy of the algorithm [24]. In a slightly different perspective, this dynamic feature selection process can also be seen as a dynamic dimensionality reduction technique. The latter makes the co-clustering suitable for high-dimensional data sets. Furthermore, by focusing on the most salient features only, as opposed to all the features, co-clustering can minimize the potential negative impact brought about by sparseness of data, which is common in a high-dimensional context. Another advantage of the co-clustering is that the simultaneously generated feature clusters can be used to interpret the contents of the object clusters [25]. For example, in the case of document–word co-clustering, the word clusters are commonly utilized to generate the corresponding document clusters’ descriptions. This makes the co-clustering generally more interpretative than its standard clustering counterpart.

2.3.2. Overview of fuzzy co-clustering

Fuzzy co-clustering allows fuzzy representations of co-clusters, and therefore gives a more natural depiction of the co-clustering structure of objects and features in a data set [26]. One existing prominent fuzzy co-clustering algorithm that is very related to the proposed PFCC is fuzzy clustering of categorical and multivariate data (FCCM) [27]. The goal of the FCCM algorithm is to maximize the objective function in Eq. (7) under the object membership constraint in Eq. (2) in Section 2.1.1 and the feature membership constraint in Eq. (8) below.

\[
J_{FCCM} = \sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{j=1}^{K} u_{ci} v_{cj} d_{ij} - T_u \sum_{c=1}^{C} \sum_{i=1}^{N} u_{ci} \ln u_{ci} - T_v \sum_{c=1}^{C} \sum_{j=1}^{K} v_{cj} \ln v_{cj},
\]

\[
\sum_{j=1}^{K} v_{cj} = 1 \quad \text{for } c = 1, \ldots, C.
\]

Notice that we have the separate object and feature memberships \(u_{ci}\) and \(v_{cj}\), respectively, a clear indication that the algorithm simultaneously clusters both objects and features. Another important point is that FCCM does not use any distance measure. Instead, it makes use of a measure of object and feature relatedness \(d_{ij}\). This makes FCCM suitable for co-clustering certain types of data such as categorical or high-dimensional data sets. The first term in Eq. (7) is called the degree of aggregation [27]. The maximization of this term makes highly related object and feature, as indicated by a high value of \(d_{ij}\), to be co-clustered together. This is because the maximization would force the value of \(v_{cj}\) to be high whenever the values of \(d_{ij}\) and \(u_{ci}\) are simultaneously high, or alternatively the value of \(u_{ci}\) to be high whenever the values of \(d_{ij}\) and \(v_{cj}\) are simultaneously high. The second and third terms are the entropy terms, which serve as fuzzifiers of object and feature memberships. These terms are used to control the levels of fuzziness of the memberships. Quite evidently from these entropy terms in its objective function, we can consider FCCM as a fuzzy co-clustering extension of the maximum entropy clustering framework in Refs. [9,10]. The constraint in Eq. (8) has a different orientation from the constraint in Eq. (2). This results in a different type of membership as will be discussed shortly. Eq. (7) can be maximized by alternatively updating the following membership equations until convergence is achieved:

\[
u_{ci} = \frac{\exp[\sum_{j=1}^{K} v_{cj} d_{ij} / T_u]}{\sum_{f=1}^{C} \exp[\sum_{j=1}^{K} v_{fj} d_{ij} / T_u]},
\]

\[
v_{cj} = \frac{\exp[\sum_{i=1}^{N} u_{ci} d_{ij} / T_v]}{\sum_{q=1}^{K} \exp[\sum_{i=1}^{N} u_{qi} d_{ij} / T_v]}.\]
2.3.3. The partitioning and ranking memberships in fuzzy co-clustering

By conforming to the membership constraint in Eq. (2), the object membership $u_{ci}$ of FCCM becomes partitioning in nature. This is also indicated by the resemblance between Eqs. (9) and (3). On the other hand, the FCCM’s feature membership in Eq. (10) is of different nature from any membership we have discussed so far. One can observe from the equation that $v_{cj}$ is a measure of similarity between a feature $j$ and a co-cluster $c$, relative to the similarities between all the other features and the co-cluster $c$. To comprehend this, we first express Eq. (10) in the form of $v_{cj} = A_{cj}/\sum_{q=1}^{K} A_{cq}$, where $A_{cj} = \exp \left\{ \frac{\sum_{i} d_{ij} u_{ci}}{v_{o}} \right\}$.

Since $A_{cj}$ and the co-cluster membership similarity between the feature $j$ and the co-cluster $c$, we have $v_{cj} = \frac{A_{cj}}{\sum_{q=1}^{K} A_{cq}}$ captures the similarity between a feature $j$ and a co-cluster $c$, relative to the similarities between all the other features and the co-cluster $c$.

Like in the case of the partitioning membership, the notion of “relative to other features” is uniquely associated with the constraint in Eq. (8), which forces the determination of a feature membership to one co-cluster to be dependent on the states of the other features in the same co-cluster. From this point onward, we will refer to a membership of this kind as the ranking membership, because such a membership reflects the rank (in this case, the feature’s rank) of importance in a given co-cluster. Unlike the partitioning membership but like the possibilistic membership, the ranking membership does not depend on the states of other clusters. Different from the possibilistic membership, however, the ranking membership does depend on the states of other features in the given cluster (or co-cluster).

2.3.4. Other co-clustering algorithms

Various techniques have been developed for co-clustering. In the literatures, we can find co-clustering formulated using information theory principles (such as information bottleneck and mutual information) [25,28], graph partitioning models [29,30], matrix manipulations [24,31], and block decomposition techniques [32]. Refs. [23] and [22] provide extensive surveys on many existing co-clustering algorithms. In fuzzy co-clustering, a variant of the FCCM called fuzzy CoDoK [33] was developed and shown to be more appropriate for co-clustering large data sets. In Ref. [26], we proposed the first dual-partitioning, namely, both object and feature are assigned the partitioning memberships, co-clustering algorithm called FCR. This type of algorithm can be useful for applications requiring fuzzy object and feature clusters (such as document and word clusters) simultaneously. Two other algorithms in Refs. [34] and [35] are examples of the distance-based fuzzy co-clustering approaches.

3. The PFCC algorithm

It is quite evident from our discussions in Section 2 that each of the three techniques has its respective shares of advantages and limitations. Fuzzy clustering is stable and not too sensitive to initialization, but its performance may suffer in the presence of outliers. Possibilistic clustering can handle outliers better, but without a good initialization it often ends up with coincident clusters. Co-clustering, specifically fuzzy co-clustering algorithms like FCCM and its variants can achieve accurate and interpretable clusters in high dimension, but with a limitation in handling outliers. Therefore in PFCC, our goal is to have a co-clustering algorithm that (1) is able to minimize the impact of outliers in both objects and features (possibilistic clustering); (2) has natural and rich representations of co-clusters (possibilistic and fuzzy memberships); (3) performs well in high dimension and produces highly interpretable clusters (co-clustering); (4) is not sensitive to initialization (fuzzy clustering); and (5) is fast enough to be practical. The last point is crucial in our decision to stick to the iterative optimization framework used by FCM, PCM, FCCM, and many of their respective variants. Being relatively fast, PFCC has been designed with a specific application to automatically categorize large document collections. However, it is possible to apply it for other applications under certain conditions (to be discussed in Section 3.2.2). Consistent with the target applications, throughout the remaining discussions, we use “objects” and “documents” interchangeably. The same rule applies for “features” and “words”. We start by explaining the PFCC objective function and the motivations behind it. It is followed by the derivations of the update rules and the algorithm’s pseudo-code. The proof of convergence of PFCC can be found in the appendix.

3.1. The objective function

The goal of PFCC is to maximize its objective function in Eq. (11), subject to the word ranking membership constraint in Eq. (8) shown in Section 2.3.2 and the word partitioning membership constraint in Eq. (12) shown below:

$$J_{PFCC} = \sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{j=1}^{K} \left( \frac{u_{ci}^{pos}}{\sum_{p=1}^{N} u_{cp}^{pos}} \right) (v_{cj} + w_{cj}) d_{ij}$$

$$- T_u \sum_{c=1}^{C} \sum_{i=1}^{N} (u_{ci}^{pos})^2 - T_v \sum_{c=1}^{C} \sum_{j=1}^{K} v_{cj} \ln v_{cj}$$

$$- T_u \sum_{c=1}^{C} \sum_{j=1}^{K} w_{cj} \ln w_{cj},$$

$$\sum_{c=1}^{C} \sum_{j=1}^{K} w_{cj} = 1 \quad for \quad j = 1, 2, \ldots, K.$$

As indicated by the constraints, the PFCC objective function has three types of membership: the document possibilistic membership $u_{ci}^{pos}$, the word ranking membership $v_{cj}$, and the word partitioning membership $w_{cj}$. The first term of Eq. (11) is the modified version of the FCCM’s degree of aggregation which is going to be elaborated more later. The second term of Eq. (11) serves as a balancing mechanism for the document possibilistic membership. The maximization of the first term would force the value of $u_{ci}^{pos}$ to be as large as possible, while the maximization of the second term counter-balance this by forcing the value of
$u_{ci}^{pos}$ to be as small as possible. The parameter $T_u$ can be used to adjust how these two terms balance each other. A small value of $T_u$ would increase the value of $u_{ci}^{pos}$, while a large value of $T_u$ would do just the opposite. The third and the fourth terms of Eq. (11) are the fuzzifiers of $v_{cj}$ and $w_{cj}$, respectively. They control the levels of fuzziness of the two respective memberships. To increase the fuzziness of the memberships, the values of the parameters $T_v$ and $T_w$ should be increased, and vice versa.

We now try to explain the motivations behind the modified degree of aggregation. We first address why there are three types of membership. As discussed in Section 2.2.2, the document possibilistic membership is there to make PFCC less prone to the presence of outlier documents compared to some existing fuzzy co-clustering algorithms like FCCM and its variants. The latter use the document partitioning membership instead.

The word ranking membership is also included for the same reason, albeit this time is against the presence of word outliers. From Section 2.3.3, we know that the ranking membership does not depend on the state of other clusters. For this reason, the ranking membership can be used to detect outliers just like its possibilistic counterpart. The main reason behind the word partitioning membership is because its interpretation ability to make PFCC less sensitive to initialization. We have explained earlier in Section 2.2.3 the coincident cluster problem suffered by the possibilistic clustering. It is not difficult to adapt this same explanation for our co-clustering context by replacing the notion of prototype into word membership. Furthermore, with the same reasoning as in the possibilistic case, it can also be shown that the ranking membership alone also suffers from the coincident cluster problem. Hence we need the partitioning membership to avoid sensitivity to the initialization in PFCC. In addition to all these, the presence of the three types of membership allows PFCC to generate richer co-clustering representations compared to the existing fuzzy co-clustering algorithms.

Next we explain the need to normalize $u_{ci}^{pos}$ (i.e. $\sum_{p=1}^{N} u_{cp}^{pos}$) in Eq. (11). The PFCC degree of aggregation can be represented as follows:

$$D_{OPFCC} = \sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{j=1}^{K} \left( \frac{u_{ci}^{pos}}{\sum_{p=1}^{N} u_{cp}^{pos}} \right) (v_{cj} + w_{cj}) d_{ij}$$

where $m_{ij} = \sum_{c=1}^{C} \left( \frac{u_{ci}^{pos}}{\sum_{p=1}^{N} u_{cp}^{pos}} \right) (v_{cj} + w_{cj})$ and $d_{ij}$ denotes the component-wise inner product of two matrices, i.e. $A : B = \sum_{i=1}^{N} a_{ij} b_{ij}$. For a given matrix $M$, we can calculate $\sum_{i=1}^{N} \sum_{j=1}^{K} m_{ij}$. From the above expression, it can be seen that in the process of maximizing $D_{OPFCC}$, a matrix $M$ with a larger value of $\sum_{i=1}^{N} \sum_{j=1}^{K} m_{ij}$ tends to be more favored because such a matrix has the advantage to produce a larger $D_{OPFCC}$ value. This bias may cause a distortion in the optimization because an optimal co-clustering actually has nothing to do with the value of $\sum_{i=1}^{N} \sum_{j=1}^{K} m_{ij}$. The normalization of $u_{ci}^{pos}$ is performed to remove such bias. One can easily verify, by taking into account the constraints involved, that the value of $\sum_{i=1}^{N} \sum_{j=1}^{K} m_{ij}$ is a constant (i.e. $K + C$) if $u_{ci}^{pos}$ is normalized.

The last question to address is about the advantage of using the possibilistic membership over using the ranking membership, given that both memberships can handle outliers equally effectively. Due to the space constraint, we are not going to provide the detailed answer to this question. This has been the topic of our earlier study in Ref. [35]. Hence below we highlight some of the conclusions made in Ref. [35] and recommend readers to refer to the paper for more details. One problem with the ranking membership is that its dependency on the memberships of other objects (or features) imposed by a constraint such as the constraint in Eq. (8) can sometimes result in misleading clusters’ representations, especially in the case of overlapping clusters. The possibilistic membership on the other hand is not subject to such constraint. This makes it more accurate than its ranking counterpart. Another important finding from Ref. [35] is that the problem affecting the ranking membership can be neutralized by the partitioning membership (i.e. the partitioning membership does not suffer from the problem). And this is precisely what we do in PFCC, that is, combining the word ranking membership with the word partitioning membership.

### 3.2. The update equations

The update equations are derived from the first necessary conditions of the Lagrangian function below, corresponding to the objective function in Eq. (11) and the constraints in Eqs. (8) and (12):

$$L_{PFCC} = \sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{j=1}^{K} \left( \frac{u_{ci}^{pos}}{\sum_{p=1}^{N} u_{cp}^{pos}} \right) (v_{cj} + w_{cj}) d_{ij}$$

$$- T_u \sum_{c=1}^{C} \sum_{i=1}^{N} (u_{ci}^{pos})^2$$

$$- T_v \sum_{c=1}^{C} \sum_{j=1}^{K} v_{cj} \ln v_{cj} - T_w \sum_{c=1}^{C} \sum_{j=1}^{K} w_{cj} \ln w_{cj}$$

$$+ \sum_{c=1}^{C} \gamma_c \left( \sum_{j=1}^{K} v_{cj} - 1 \right) + \sum_{j=1}^{K} \lambda_j \left( \sum_{c=1}^{C} w_{cj} - 1 \right).$$

(13)

We now organized this section into two subsections. In the first one, the relatively simpler update equations for $v_{cj}$ and $w_{cj}$ are derived. In the second one, a special technique to derive the update equation of $u_{ci}^{pos}$ is presented.
3.2.1. \( v_{cj} \) and \( w_{cj} \)

From the first necessary conditions of \( \partial L_{PFCC} / \partial v_{cj} = 0 \) and \( \partial L_{PFCC} / \partial w_{cj} = 0 \), and the constraints in Eqs. (8) and (12), we can derive the following update equations:

\[
v_{cj} = \frac{\exp((1/T_v)\sum_{i=1}^{N}(u_{ci}^{\text{pos}}/\sum_{p=1}^{N}u_{cp}^{\text{pos}})d_{ij})}{\sum_{q=1}^{K}\exp((1/T_v)\sum_{i=1}^{N}(u_{ci}^{\text{pos}}/\sum_{p=1}^{N}u_{cp}^{\text{pos}})d_{iq})},
\]

\[
w_{cj} = \frac{\exp((1/T_w)\sum_{i=1}^{N}(u_{ci}^{\text{pos}}/\sum_{p=1}^{N}u_{cp}^{\text{pos}})d_{ij})}{\sum_{f=1}^{C}\exp((1/T_w)\sum_{i=1}^{N}(u_{fi}^{\text{pos}}/\sum_{p=1}^{N}u_{fp}^{\text{pos}})d_{ij})}.
\]

By observing the equation structures, it should not be too difficult to appreciate that Eqs. (14) and (15) are ranking and partitioning memberships, respectively.

3.2.2. An incremental and approximation approach to derive \( u_{ci}^{\text{pos}} \)

If we try to derive \( u_{ci}^{\text{pos}} \) from the first necessary condition \( \partial L_{PFCC} / \partial u_{ci}^{\text{pos}} = 0 \), we will end up with a third degree polynomial problem. Instead of applying a direct mathematical approach to solve this problem, we devise a more practical alternative incremental approach that incorporates a simplified approximate version of the objective function. It is an incremental approach because the \( u_{ci}^{\text{pos}} \) is updated one membership at a time. In each incremental update, the objective function in Eq. (11) is approximated by its relatively simpler version. The \( u_{ci}^{\text{pos}} \) equation is then derived from the first necessary condition of this approximate objective function. There are two important points to be underlined: (1) the approximation is only valid under the Assumption 1 stated below, and (2) if Assumption 1 holds, we can prove that each incremental update of \( u_{ci}^{\text{pos}} \) never decreases the objective function. The latter is important to ensure the algorithm’s convergence (the proof of convergence of PFCC is detailed in the appendix).

**Assumption 1.** At any point in the PFCC optimization process, we have \( \sum_{p=1}^{N}u_{cp} \gg u_{ci} \) for all \( c = 1, \ldots, C \) and \( i = 1, \ldots, N \).

Assumption 1 is an equivalent way of saying that there should be a significantly large number of documents in every cluster. This assumption is arguably reasonable for our specific application of categorizing large document collections because in such a system, clusters are generally made up of significantly large sets of documents. With regard to the presence of small outlier clusters in a data set, since our algorithm utilizes the possibilistic clustering formulation, such outliers would be assigned low membership values, preventing the need to actually generate small-sized document clusters. In this way, the validity of Assumption 1 remains. In order to keep Assumption 1 maintained throughout the runtime of PFCC, it is important to ensure that the algorithm is not initialized with empty or small clusters. In practice, we find that a random initialization is usually effective for this purpose.

To proceed with the derivation, let us express the objective function \( J_{PFCC} \) in Eq. (11) as a function of a single variable \( u_{ab}^{\text{pos}} \), as such is the case during a single incremental update of the possibilistic membership of a document \( b \) to a co-cluster \( a \):

\[
J_{PFCC}(u_{ab}^{\text{pos}}) = \sum_{i=1}^{N}(u_{ai}^{\text{pos}}\sum_{j=1}^{K}(v_{aj} + w_{aj})d_{ij}) - T_u(u_{ab}^{\text{pos}})^2 + \text{constant},
\]

where

\[
\text{constant} = \sum_{c=1}^{C}\sum_{i=1}^{N}\sum_{j=1}^{K}\left\{ \left( \frac{u_{ci}^{\text{pos}}}{\sum_{p=1}^{N}u_{cp}^{\text{pos}}} \right) (v_{cj} + w_{cj})d_{ij} \right\} - T_u \sum_{c=1}^{C}\sum_{i=1}^{N}\sum_{j=1}^{K}v_{cj}\ln v_{cj} - T_u \sum_{c=1}^{C}\sum_{i=1}^{N}w_{cj}\ln w_{cj}.
\]

Note that the terms \( v_{aj} \) and \( w_{aj} \) in the \( J_{PFCC}(u_{ab}^{\text{pos}}) \) expression above are also considered as constants with values computed from the previous iteration. If Assumption 1 holds, \( J_{PFCC}(u_{ab}^{\text{pos}}) \) can be approximated by the function \( \tilde{J}_{PFCC}(u_{ab}^{\text{pos}}) \) as follows:

\[
\tilde{J}_{PFCC}(u_{ab}^{\text{pos}}) \approx \tilde{J}_{PFCC}(u_{ab}^{\text{pos}}) = \sum_{i=1}^{N}(u_{ai}^{\text{pos}}\sum_{j=1}^{K}(v_{aj} + w_{aj})d_{ij}) - T_u(u_{ab}^{\text{pos}})^2 + \text{constant}.
\]

From the first necessary condition of \( \partial \tilde{J}_{PFCC}(u_{ab}^{\text{pos}}) / \partial u_{ab}^{\text{pos}} = 0 \), we can derive the update equation for \( u_{ab}^{\text{pos}} \):

\[
u_{ab}^{\text{pos}} = \frac{\sum_{j=1}^{K}(v_{aj} + w_{aj})d_{bj}}{2T_u \sum_{i=1}^{N}(u_{ai}^{\text{pos}})}.
\]

Since \( J_{PFCC}(u_{ab}^{\text{pos}}) \approx \tilde{J}_{PFCC}(u_{ab}^{\text{pos}}) \), the stationery point of \( J_{PFCC}(u_{ab}^{\text{pos}}) \) in Eq. (16) above should be a good estimate of the stationery point of \( J_{PFCC}(u_{ab}^{\text{pos}}) \). In other words, if Assumption 1 holds, we can estimate the PFCC \( u_{ci}^{\text{pos}} \) update equation by Eq. (16).

The equation clearly reflects the characteristic of the possibilistic membership because it only depends on the similarity between the document \( b \) and the co-cluster \( a \). The normality on the memberships can be accomplished by simply normalizing them at the end of the optimization. This incremental and approximate approach is simpler and more practical to derive the \( u_{ci}^{\text{pos}} \) update equation. Furthermore as long as Assumption 1
holds, it can be proven that each incremental update from Eq. (16) leads to the algorithm’s convergence.

3.3. The pseudo-code

The following shows the pseudo-code of PFCC:

PFCC Algorithm
1. Set parameter $T_u$, $T_v$, $T_w$, $\epsilon$, and $\tau_{\text{max}}$;
2. Set $\tau = 0$;
3. Randomly initialize $u_{ci}^\text{pos}$, $0 \leq u_{ci}^\text{pos} \leq 1$;
4. REPEAT
5. Update $v_{ij}$ from Eq. (14);
6. Update $w_{ij}$ from Eq. (15);
7. Update $u_{ci}^\text{pos}$ from Eq. (16);
8. Update $\tau = \tau + 1$;
9. UNTIL $\max_{c,i}(u_{ci}^\text{pos} (\tau) - u_{ci}^\text{pos} (\tau - 1)) \leq \epsilon$ or $\tau = \tau_{\text{max}}$.

Like FCM, PCM, and FCCM/fuzzy CoDoK, the new algorithm has a linear time complexity of $O(CNK\tau)$, where $\tau$ denotes the number of iterations. This comes from the $O(CNK)$ time complexity needed to calculate each membership equation. For $v_{ij}$ and $w_{ij}$, this can be achieved by computing their respective numerator terms separately from the denominator terms. We can prove that if Assumption 1 holds, the PFCC algorithm converges to a local maximum of the optimization. The detailed proof of convergence can be found in the appendix. We are yet to perform any analytical study on the rate of convergence of PFCC, which may depend on a number of factors including the input $\epsilon$ parameter value, the data set, the initialization, and the setting of the fuzzifier parameters. We found that, in all our experiments, which are discussed in Section 4, PFCC always converged within 200 iterations. However, in case of a possible prolonged convergence, the algorithm is set to stop once $\tau$ has exceeded the manually defined parameter $\tau_{\text{max}}$.

4. Experimental results

Several experiments on a number of large document data sets have been conducted to demonstrate the advantages of PFCC. This section provides the results and some further discussions.

4.1. Data sets and implementation details

There are four sources of data sets: the large Web page data set (WPD) [36], the 20-Newsgroup data set, the Classic3 data set, and the Syskill & Webert (SW) data set from the UCI KDD repository. We generated several subsets from WPD and 20-Newsgroup. Table 2 provides the complete list of all data sets (subsets and original) used in our experiments.

The Matrix Creation (MC-toolkit) [37] was utilized to pre-process the document collections. In this phase, we removed the stopwords and words with frequencies beyond the range of $0.005N < tf < 0.995N$, where $N$ denotes the number of documents. For WPD and all its subsets, the MC-toolkit word parser was set to HTML. For the other data sets, the parser was set to NORMAL. The resulting feature vectors were represented by their normalized tf-idf [38]. Information on the sizes of the document–word matrix representations can be found in Table 2.

We adopted random initializations in all the experiments. Unless otherwise stated, every result shown in this section represents the average value of a 10-trial simulation. In our implementation, all user-defined parameters were set empirically. For every data set, we ran a series of simulations with different parameter settings. The sets of parameter values that were found to be consistently resulted in the best performance were the ones used to generate the results. In all our experiments, we use $\epsilon = 10^{-5}$.

For evaluation measure purpose, we applied a defuzzification process that assigned every document to the cluster it has the highest membership to. After this assignment, the precision, recall, and purity measures were calculated. For every resulting document cluster $c$, the precision and recall are defined as follows:

$$\text{precision} = \frac{|R_c|}{|R_c| + |S_c|},$$
$$\text{recall} = \frac{|R_c|}{|R_c| + |T_c|},$$

where $R_c$ denotes the set of all documents correctly assigned to $c$, $S_c$ denotes the set of all documents incorrectly assigned to $c$, and $T_c$ denotes the set of all documents incorrectly not assigned to $c$. For every data set, we recorded the average values of the precisions and recalls of all the clusters. To resolve any conflict in deciding which of the ground-truth categories a cluster represented, as much as possible we chose the most optimum assignment scheme (i.e. the one that resulted in the best average precisions and recalls). The purity is defined as follows:

$$\text{purity} = \sum_{c=1}^{C} \max_{x \in X} |G_{cx}| \cdot |R_c| + |S_c|,$$

where $X$ denotes the set of all ground-truth document categories, $G_{cx}$ denotes the set of all documents in the ground-truth category $x$ and assigned to cluster $c$, $R_c$ and $S_c$ are as defined above.

4.2. Results and discussions

First we demonstrate that, in the presence of document outliers, PFCC, the proposed possibilistic-fuzzy co-clustering, can perform document clustering more accurately than two existing algorithms: fuzzy CoDoK [33], a fuzzy co-clustering algorithm, and HFCM [13], a fuzzy clustering algorithm. We did not make any direct comparisons with any popular possibilistic clustering algorithm such as the ones mentioned in Section 2.2 because in the current state, these algorithms are generally formulated using the Euclidean distance, which is not suitable for high-dimensional data sets. Also we chose fuzzy CoDoK instead of FCCM because of the computational overflow problem.
Table 2
List of document data sets

<table>
<thead>
<tr>
<th>Data set</th>
<th>No. of docs.</th>
<th>No. of words</th>
<th>Clusters (no. of docs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WPD</td>
<td>11000</td>
<td>9026</td>
<td>Banking and finance (3000), programming languages (3000), science (2000), sport (3000)</td>
</tr>
<tr>
<td>WPD2</td>
<td>2000</td>
<td>7299</td>
<td>Commercial Bank (1000), Java (1000)</td>
</tr>
<tr>
<td>WPD6</td>
<td>6000</td>
<td>11571</td>
<td>Commercial Bank (1000), C/C++ (1000), astronomy (1000), biology (1000), soccer (1000), sport (1000)</td>
</tr>
<tr>
<td>WPD7</td>
<td>7000</td>
<td>10725</td>
<td>Commercial Bank (1000), building societies (1000), Java (1000), astronomy (1000), biology (1000), soccer (1000), sport (1000)</td>
</tr>
<tr>
<td>SM</td>
<td>2000</td>
<td>5450</td>
<td>Soccer (1000), motorsport (1000)</td>
</tr>
<tr>
<td>SS</td>
<td>2000</td>
<td>6337</td>
<td>Soccer (1000), sport (1000)</td>
</tr>
<tr>
<td>CJB</td>
<td>2250</td>
<td>6781</td>
<td>Commercial Bank (1000), Java (750), biology (500)</td>
</tr>
<tr>
<td>CAB</td>
<td>2250</td>
<td>6980</td>
<td>Commercial Bank (1000), astronomy (750), biology (500)</td>
</tr>
<tr>
<td>NG8</td>
<td>7581</td>
<td>3360</td>
<td>IBM-PC hardware (982), atheism (799), medical (981), hockey (990), guns (999), automobiles (910), electronics (940), middle-east (990)</td>
</tr>
<tr>
<td>Binary</td>
<td>500</td>
<td>3377</td>
<td>Politics (250), middle-east (250)</td>
</tr>
<tr>
<td>Multi5</td>
<td>500</td>
<td>2889</td>
<td>Computer graphics (100), motorcycle (100), baseball (100), space (100), middle-east (100)</td>
</tr>
<tr>
<td>Classic3</td>
<td>3891</td>
<td>2176</td>
<td>Medical (1033), aerospace (1398), inform. retrieval (1460)</td>
</tr>
<tr>
<td>SW</td>
<td>341</td>
<td>3437</td>
<td>Sheep (70), goats (74), band (61), biomedical (136)</td>
</tr>
</tbody>
</table>

![Outliers Effect on Precisions](image1.png)

![Outliers Effect on Recalls](image2.png)

Fig. 2. The impact of document outliers on algorithms’ performances.

affecting the latter when applied on large high-dimensional data sets [33]. Both fuzzy CoDoK and HFCM use the partitioning document memberships in their formulations. We performed experiments on six data sets. The first is WPD2, which has 2000 documents equally distributed in two categories Commercial Bank and Java. The second to the sixth data sets are WPD2 added, respectively, by 200, 400, . . . , 1000 outlier documents (i.e. 10% to 50% levels of outliers). The outlier documents were randomly taken from the 20-Newsgroup data set. The average precision and the average recall of each data set were then recorded, considering only the original 2000 non-outlier documents. Fig. 2 shows how they vary as the number of outliers was increased in the data sets. In this experiment, we set $T_u = 0.1$, $T_v = 0.01$, $T_w = 0.001$ for PFCC, $T_u = 0.05$ (0.0001 for the data set with 50% outliers), $T_v = 1.5$ for fuzzy CoDoK, and $m = 1.1$ for HFCM. As shown in the figure, PFCC shows almost no variation in its performance, while the other algorithms’ performances gradually decline as the number of outliers present increases. The explanation can come from the fact that the possibilistic clustering in principle always assigns low membership values to outliers and hence, minimizes their impacts on the overall process. This is as opposed to clustering based on the partitioning membership where outliers are assigned relatively higher membership values. This point is
Table 3
Document outliers’ memberships ($\mu_i^{\text{out}}$ and $u_i$)

<table>
<thead>
<tr>
<th>Outlier</th>
<th>PFCC</th>
<th>Fuzzy CoDoK</th>
<th>HFCM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C0</td>
<td>C1</td>
<td>C0</td>
</tr>
<tr>
<td>o1</td>
<td>0.13</td>
<td>0.2</td>
<td>0.43</td>
</tr>
<tr>
<td>o2</td>
<td>0.07</td>
<td>0.1</td>
<td>0.43</td>
</tr>
<tr>
<td>o3</td>
<td>0.13</td>
<td>0.14</td>
<td>0.45</td>
</tr>
<tr>
<td>o4</td>
<td>0.16</td>
<td>0.2</td>
<td>0.41</td>
</tr>
<tr>
<td>o5</td>
<td>0.11</td>
<td>0.16</td>
<td>0.46</td>
</tr>
<tr>
<td>o6</td>
<td>0.2</td>
<td>0.25</td>
<td>0.35</td>
</tr>
<tr>
<td>o7</td>
<td>0.21</td>
<td>0.22</td>
<td>0.43</td>
</tr>
<tr>
<td>o8</td>
<td>0.18</td>
<td>0.16</td>
<td>0.49</td>
</tr>
</tbody>
</table>

illustrated in Table 3, where the document memberships of eight randomly picked outlier documents from the data set with 200 outliers are displayed. It should be evident that PFCC consistently assigns significantly lower membership values to these outliers as compared to fuzzy CoDoK and HFCM (note that the possibilistic membership values in Table 3 have been normalized to allow fair comparisons). These results reaffirm that through possibilistic clustering, a better clustering accuracy and more informative memberships can be achieved.

We now discuss the performance comparisons of PFCC, fuzzy CoDoK, and HFCM in clustering a benchmark data set WPD6. The data set consists of 6000 documents equally distributed in 6 categories, some of which contain significant amount of overlap (e.g. soccer and sport). For each algorithm, we ran a 10-trial simulation on the data set and chose the one that resulted in the best performance. The corresponding document cluster distributions are shown in Fig. 3. The optimum parameter settings were found to be $T_u = 0.001$, $T_v = 0.01$, $T_w = 0.001$ for PFCC, $T_u = 0.00001$, $T_v = 1.5$ for fuzzy CoDoK, and $m = 1.02$ for HFCM. From the figure, PFCC is shown to uncover quite successfully the six categories in the data sets, i.e. with 83.2% precision, 82% recall, 82% purity. A few misclassifications can be observed in C4 between soccer and sport, and in C0 between soccer and Commercial Bank. The former is most likely caused by the similar theme shared by the two categories, while the latter is perhaps caused by the existing discussions on the financial aspect of soccer in some soccer documents. Fig. 3 also shows that fuzzy CoDoK is able to generate the clusters C0–C3 with relatively high precisions (albeit with relatively low recalls). However, the quality of the clusters C4 and C5, which correspond to soccer and sport, respectively, is poorer than its PFCC counterpart. The overall fuzzy CoDoK’s performance is calculated to be 78.56% precision, 73.53% recall, ad 73.53% purity. Lastly, Fig. 3 shows that HFCM, although can well generate C0 and C1, is unable to capture the remaining categories properly. From the figure it is evident that the astronomy and biology categories are merged together in C2. The same thing also happens to soccer and sport in C4. These misclassifications cause a low performance of 53.45% precision, 56.39% recall, and 58.7% purity. We therefore have PFCC outperforms both fuzzy CoDoK and HFCM in this experiment on WPD6.

To further enrich the discussions, we include Table 4, which shows the document clustering performance comparisons of PFCC, fuzzy CoDoK, and HFCM on the other remaining benchmark data sets. On almost all the data sets in the table, PFCC achieves either better or comparable performance to its counterparts. In line with our earlier result on WPD6, the results in Table 4 seem to strengthen our argument that the proposed PFCC can generally perform document clustering better than the two existing formulations, i.e. fuzzy co-clustering and fuzzy clustering. Another interesting thing to highlight from Table 4 is that except on the simplest data set Classic3, HFCM is shown to consistently result in the lowest performances of all the three algorithms. This outcome further confirms the benefit of the co-clustering over the standard clustering in the task of document categorization [33]. Due to the space limit, we omit the list of the optimum parameter settings from Table 4. From the experiments we observe that PFCC is not so sensitive to the value of the $T_u$ parameter. It is, however, quite sensitive to the setting of $T_v$ and $T_w$. This indicates that the eventual performance of the algorithm depends quite significantly on the balance between its word ranking and word partitioning. Further study will be conducted in the future to better understand the interaction between these two techniques and how they can affect the algorithm’s performance.

Besides document clusters, PFCC also simultaneously generates fuzzy word clusters. The absence of the ground-truth word categorization, however, makes it difficult to evaluate the accuracies of the resulting word clusters. We therefore decide to focus more on the interpretability aspect of the PFCC’s word clustering. Table 5 shows the top 10 words with the highest membership values in the six clusters (or co-clusters) of WPD6 generated by PFCC and fuzzy CoDoK. These results correspond to the document clusters in Fig. 3. Since PFCC has two types of word membership, i.e. $v_{cj}$ and $w_{cj}$, the values of $v_{cj} + w_{cj}$ were used to decide the top words. It can be observed from the table that these words can generally describe the contexts of their corresponding document clusters, e.g. from words such as “DNA” and “biotechnology”, one can deduce that biology is the main context of document cluster C3 in Fig. 3. Based on the results in Table 5, we can argue one benefit of PFCC over fuzzy CoDoK, that is, for every word enlisted, PFCC provides two types of meaningful information,
Fig. 3. WPD6 document clusters distribution.

Table 4
Document clustering performance

<table>
<thead>
<tr>
<th>Data set</th>
<th>PFCC</th>
<th></th>
<th></th>
<th></th>
<th>Fuzzy CoDoK</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>HFCM</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WPD</td>
<td>84.06</td>
<td>83.28</td>
<td>82.97</td>
<td>81.57</td>
<td>75.27</td>
<td>76.8</td>
<td>62.28</td>
<td>60.62</td>
<td>68.05</td>
<td>56.62</td>
<td>54.91</td>
<td>56.49</td>
<td></td>
</tr>
<tr>
<td>WPD7</td>
<td>65.98</td>
<td>68.07</td>
<td>68.96</td>
<td>60.41</td>
<td>59.87</td>
<td>60.17</td>
<td>56.62</td>
<td>54.91</td>
<td>56.49</td>
<td>56.62</td>
<td>66.5</td>
<td>66.5</td>
<td></td>
</tr>
<tr>
<td>SM</td>
<td>84.14</td>
<td>83.34</td>
<td>83.34</td>
<td>69.6</td>
<td>67.33</td>
<td>67.73</td>
<td>66.94</td>
<td>66.5</td>
<td>66.5</td>
<td>69.93</td>
<td>69.62</td>
<td>69.62</td>
<td></td>
</tr>
<tr>
<td>SS</td>
<td>87.08</td>
<td>87.25</td>
<td>87.25</td>
<td>77.46</td>
<td>77.02</td>
<td>77.02</td>
<td>75.39</td>
<td>76.74</td>
<td>82.45</td>
<td>64.81</td>
<td>64.53</td>
<td>73.16</td>
<td></td>
</tr>
<tr>
<td>CJB</td>
<td>87.84</td>
<td>89.27</td>
<td>88.98</td>
<td>87.43</td>
<td>87.11</td>
<td>87.56</td>
<td>75.39</td>
<td>76.74</td>
<td>82.45</td>
<td>64.81</td>
<td>64.53</td>
<td>73.16</td>
<td></td>
</tr>
<tr>
<td>CAB</td>
<td>82.34</td>
<td>82.88</td>
<td>84.9</td>
<td>72.84</td>
<td>67.7</td>
<td>73.66</td>
<td>64.81</td>
<td>64.53</td>
<td>73.16</td>
<td>76.78</td>
<td>76.92</td>
<td>80.31</td>
<td></td>
</tr>
<tr>
<td>NG8</td>
<td>80.39</td>
<td>80.35</td>
<td>81.35</td>
<td>80.28</td>
<td>82.35</td>
<td>84.32</td>
<td>76.78</td>
<td>76.92</td>
<td>80.31</td>
<td>76.4</td>
<td>67.5</td>
<td>67.05</td>
<td></td>
</tr>
<tr>
<td>Binary</td>
<td>90.7</td>
<td>90.6</td>
<td>90.6</td>
<td>70.73</td>
<td>70.22</td>
<td>70.22</td>
<td>76.4</td>
<td>67.5</td>
<td>67.05</td>
<td>82.5</td>
<td>81.6</td>
<td>84.74</td>
<td></td>
</tr>
<tr>
<td>Multi5</td>
<td>93.84</td>
<td>94.05</td>
<td>94.83</td>
<td>95.64</td>
<td>95.83</td>
<td>96.3</td>
<td>82.5</td>
<td>81.6</td>
<td>84.74</td>
<td>98.9</td>
<td>98.8</td>
<td>98.9</td>
<td></td>
</tr>
<tr>
<td>Classic3</td>
<td>98.63</td>
<td>98.49</td>
<td>98.56</td>
<td>98.6</td>
<td>98.3</td>
<td>98.4</td>
<td>98.9</td>
<td>98.8</td>
<td>98.9</td>
<td>78.49</td>
<td>85.76</td>
<td>87.42</td>
<td></td>
</tr>
</tbody>
</table>

i.e. how the word is ranked in the cluster \( v_{cj} \) and how the word is partitioned \( w_{cj} \). In fuzzy CoDoK, only the former is produced. While the word ranking membership measures the relative position of a word locally in one particular cluster, the word partitioning membership reveals the positioning of the word relative to all the data set’s clusters. By combining these two together, PFCC gives a more complete picture of how a word is set in a given data set. Therefore the new algorithm is arguably more informative than its existing counterpart. In Table 5 we can see some uncommon words such as
Table 5
WPD6’s top words

<table>
<thead>
<tr>
<th>Cluster</th>
<th>PFCC ($v_{ij}/10^{-3}w_{ij}$)</th>
<th>Fuzzy CoDoK ($v_{ij}/10^{-2}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commercial Bank</td>
<td>banking (1.6, 1), bank (0.88, 1), mortgage (0.88, 1), account (0.81, 1), loan (0.69, 1), menu (0.66, 1), interest (0.44, 1), business (0.45, 1), savings (0.35, 1), credit (0.36, 1)</td>
<td>banking (1.38), bank (1.19), mortgage (1.08), account (0.96), loan (0.92), nbsp (0.86), business (0.81), menu (0.8), interest (0.7), savings (0.63)</td>
</tr>
<tr>
<td>C/C++</td>
<td>int (0.42, 1), file (0.42, 1), code (0.45, 1), quot (1.1, 1), string (0.26, 1), programming (0.28, 1), visual (0.26, 0.99), char (0.19, 0.99), void (0.2, 0.99), class (0.28, 0.99)</td>
<td>quot (1.06), file (0.93), int (0.9), code (0.65), programming (0.59), string (0.56), watcom (0.52), char (0.5), void (0.48), program (0.48)</td>
</tr>
<tr>
<td>Astronomy</td>
<td>universe (1.3, 1), galaxies (1, 1), galaxy (0.46, 1), stars (0.35, 1), light (0.35, 1), infrared (0.24, 1), astronomy (0.34, 1), space (0.31, 1), telescope (0.12, 1), hubble (0.2, 0.99)</td>
<td>universe (1.23), galaxies (1.18), galaxy (0.81), light (0.62), stars (0.6), infrared (0.56), space (0.54), telescope (0.5), black (0.46)</td>
</tr>
<tr>
<td>Biology</td>
<td>dna (0.32, 1), human (0.35, 1), genome (0.26, 1), biotechnology (0.25, 1), gene (0.25, 0.99), research (0.33, 0.99), genetic (0.21, 0.99), genetics (0.19, 0.98), cell (0.19, 096), biology (0.19, 0.96)</td>
<td>dna (0.85), human (0.77), research (0.67), gene (0.64), genome (0.59), cell (0.58), biology (0.54), science (0.5), genetics (0.5), hiv (0.48)</td>
</tr>
<tr>
<td>Soccer</td>
<td>nbsp (1.2, 1), var (0.36, 1), document (0.45, 1), cup (0.18, 1), england (0.14, 1), football (0.14, 1), league (0.15, 1), united (0.09, 1), town (0.06, 1), soccer (0.07, 1)</td>
<td>src (1.15), document (1.11), nbsp (1.1), var (1.05), gif (0.94), width (0.87), height (0.87), img (0.8), font (0.73), border (0.66)</td>
</tr>
<tr>
<td>Sport</td>
<td>hockey (0.4, 1), basketball (0.25, 0.99), eteamz (0.19, 0.99), coach (0.24, 0.98), players (0.24, 0.96), player (0.22, 0.96), ball (0.19, 0.95), coaches (0.09, 0.93), game (0.25, 0.9), play (0.16, 0.8)</td>
<td>nbsp (4.24), hockey (0.69), team (0.47), cup (0.44), document (0.44), quot (0.43), england (0.4), var (0.38), basketball (0.35), players (0.35)</td>
</tr>
</tbody>
</table>

Table 6
WPD6’s word outliers (PFCC)

<table>
<thead>
<tr>
<th>Word outliers</th>
<th>$\sum_{c=1}^{6} v_{cj}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>nbsp</td>
<td>$3.944 \times 10^{-4}$</td>
</tr>
<tr>
<td>istringstream</td>
<td>$3.945 \times 10^{-4}$</td>
</tr>
<tr>
<td>launchquery</td>
<td>$3.946 \times 10^{-4}$</td>
</tr>
<tr>
<td>fetchurl</td>
<td>$3.946 \times 10^{-4}$</td>
</tr>
<tr>
<td>teamsite</td>
<td>$3.946 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

“nbsp” and “quot”. These are words that often appear in the tags used in WPD6 and they would have been removed had we performed a more proper document pre-processing. Their high frequencies may have misled the two algorithms to consider them as important words, which explains why they have high memberships.

Finally, we demonstrate the outlier-detection property of the word ranking membership in PFCC. Based on the definition of the word ranking membership, word outliers can be identified by their low ranking memberships in all the data set’s clusters. In mathematical terms, these words should have low $\sum_{c=1}^{C} v_{cj}$ values. We extracted five such words with the lowest $\sum_{c=1}^{C} v_{cj}$ scores in the WPD6 clusters produced by PFCC and tabulated them in Table 6. The latter shows the enlisted words and their respective aggregate memberships, $\sum_{c=1}^{C} v_{cj}$. As we can observe from the table, the words are untypical, given the contents of the WPD6 data set. For this reason they should indeed be considered as outliers.

5. Conclusions and future work

We have presented a novel framework to combine the possibilistic and fuzzy formulations of co-clustering for large and high-dimensional data analyses. Based on the proposed framework, we have introduced a new simple yet robust co-clustering algorithm called PFCC. Numerous benefits offered by PFCC are explained and verified through our extensive experiments on real large benchmark data sets. These benefits include robustness in the presence of document and word outliers; more informative representation of co-clusters; highly descriptive document clusters; and more accurate document clustering. In addition, the proposed algorithm maintains a comparable
Appendix A. The proof of convergence

We shall prove that PFCC always converges under Assumption 1.

**Theorem 1.** At every iteration \( \tau \), updating \( v = \{v_{11}, \ldots, v_{CK}\} \) from Eq. (14) never decreases the objective function \( J_{PFCC} \) in Eq. (11).

**Proof.** When updating \( v \), we can consider the objective function \( J_{PFCC} \) as a function of \( v \):

\[
J_{PFCC}(v) = \sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{j=1}^{K} \left( \frac{u_{ci}^{pos}}{\sum_{p=1}^{N} u_{ip}^{pos}} \right) v_{cj} d_{ij} - T_v \sum_{c=1}^{C} \sum_{j=1}^{K} v_{cj} \ln v_{cj} + \text{constant},
\]

where

\[
\text{constant} = \sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{j=1}^{K} \left( \frac{w_{ci}^{pos}}{\sum_{p=1}^{N} w_{ip}^{pos}} \right) w_{cj} d_{ij} - T_u \sum_{c=1}^{C} \sum_{i=1}^{N} (u_{ci}^{pos})^2 - T_w \sum_{c=1}^{C} \sum_{j=1}^{K} w_{cj} \ln w_{cj}.
\]

Note that the terms \( u_{ci}^{pos} \) in the \( J_{PFCC}(v) \) expression above are also considered as constants with values computed from the previous iteration. Theorem 1 can be proven by showing that the \( v^* \) (i.e. \( v \) updated from Eq. (14)) is a local maximum of \( J_{PFCC}(v) \) above, in the region where the constraints in Eqs. (8) and (12) are satisfied. For this to be true, we need to show that the Hessian matrix \( \nabla^2 J_{PFCC}(v^*) \) is negative definite. It can be shown that:

\[
\nabla^2 J_{PFCC}(v) = \begin{bmatrix}
\frac{\partial J_{PFCC}(v)^2}{\partial v_{11}} & \ldots & \frac{\partial J_{PFCC}(v)^2}{\partial v_{CK}} \\
\vdots & & \vdots \\
\frac{\partial J_{PFCC}(v)^2}{\partial v_{CK}^2} & \ldots & \frac{\partial J_{PFCC}(v)^2}{\partial v_{CK}} \\
\end{bmatrix}
\]

\[
= \begin{bmatrix}
-T_v & 0 & \cdots & 0 \\
0 & \ddots & \cdots & 0 \\
\vdots & \cdots & \ddots & -T_v \\
0 & \cdots & 0 & -T_v \\
\end{bmatrix}.
\]

At \( v^* \), we have \( v_{cj} > 0 \) for all \( c = 1, \ldots, C \) and \( j = 1, \ldots, K \) (from Eq. (14)). And we always assign a positive value to \( T_v \). Therefore, the Hessian matrix \( \nabla^2 J_{PFCC}(v^*) \) is negative definite. Since both the first necessary condition (\( \partial J_{PFCC}/\partial v_{cj} = 0 \)) and the second sufficient condition (\( \nabla^2 J_{PFCC}(v^*) \) are negative definite) are satisfied, \( v^* \) updated from Eq. (14) is indeed a local maximum of \( J_{PFCC}(v) \) in the region where the constraints in Eqs. (8) and (12) are satisfied. For this reason, at every iteration, updating \( v \) from Eq. (14) never decreases the objective function \( J_{PFCC} \).

**Theorem 2.** At every iteration \( \tau \), updating \( w = \{w_{11}, \ldots, w_{CK}\} \) from Eq. (15) never decreases the objective function \( J_{PFCC} \) in Eq. (11).

**Proof.** Theorem 2 can be proven in a similar fashion as Theorem 1.

**Theorem 3.** If Assumption 1 holds, each \( u_{ci} \) update from Eq. (16), for \( c = 1, \ldots, C \) and \( i = 1, \ldots, N \), never decreases the objective function \( J_{PFCC} \) in Eq. (11).

**Proof.** It can be shown that \( \partial J_{PFCC}(u_{ci})^2/\partial u_{ci} = -2T_u \) (\( J_{PFCC}(u_{ci}) \) is defined in Section 3.2.2). Since \( T_u \) is always set to be positive, we have both the first necessary condition (\( \partial J_{PFCC}(u_{ci})/\partial u_{ci} = 0 \)) and the second sufficient condition (\( \partial J_{PFCC}(u_{ci})^2/\partial^2 u_{ci} < 0 \)) satisfied. Thus, the \( u_{ci}^* \) (i.e. \( u_{ci} \) updated from Eq. (16)) is a local maximum of \( J_{PFCC}(u_{ci}) \). If Assumption 1 holds, we have \( J_{PFCC}(u_{ci}) \approx J_{PFCC}(u_{ci}) \) and thus, the local maximum of \( J_{PFCC}(u_{ci}) \) should be a good estimate of the local maximum of \( J_{PFCC}(u_{ci}) \). From this we can conclude that if Assumption 1 holds, each \( u_{ci} \) update from Eq. (16) never decreases the objective function \( J_{PFCC} \).

**Theorem 4.** The following is true, with \( J_{PFCC} \) is as defined in Eq. (11):

\[
J_{PFCC} \leq 2C \sum_{i=1}^{N} \sum_{j=1}^{K} d_{ij} - T_v C \ln \left( \frac{1}{K} \right) - T_u K \ln \left( \frac{1}{C} \right).
\]
Proof. The first term of $J_{PFC}$ can be expressed as follows:

$$
\sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{j=1}^{K} \left( \left( \frac{u_{ij}^{pos}}{\sum_{p=1}^{N} u_{ip}^{pos}} \right) (v_{cj} + w_{cj})d_{ij} \right)
$$

$$
= \begin{bmatrix}
  m_{11} & \cdots & m_{1K} \\
  \vdots & \ddots & \vdots \\
  m_{N1} & \cdots & m_{NK}
\end{bmatrix}
\begin{bmatrix}
  d_{11} & \cdots & d_{1K} \\
  \vdots & \ddots & \vdots \\
  d_{N1} & \cdots & d_{NK}
\end{bmatrix},
$$

where $m_{ij} = \sum_{c=1}^{C} \left( \left( \frac{u_{ij}^{pos}}{\sum_{p=1}^{N} u_{ip}^{pos}} \right) (v_{cj} + w_{cj}) \right)$ and $d_{ij}$ denotes the component-wise inner product of two matrices, i.e., $A : B = \sum_{i} \sum_{j} a_{ij}b_{ij}$. Since $\left( \frac{u_{ij}^{pos}}{\sum_{p=1}^{N} u_{ip}^{pos}} \right) \geq 0$, $v_{cj}, w_{cj} \leq 1$, we have $m_{ij} \leq 2C$, and consequently, the first term is loosely bounded:

$$
\sum_{c=1}^{C} \sum_{i=1}^{N} \sum_{j=1}^{K} \left( \frac{u_{ij}^{pos}}{\sum_{p=1}^{N} u_{ip}^{pos}} \right) (v_{cj} + w_{cj})d_{ij} \leq 2C \sum_{i=1}^{N} \sum_{j=1}^{K} d_{ij}.
$$

The second term of $J_{PFC}$ has an obvious maximum value of zero because $u_{ci}$ is always positive (from Eq. (16)). The third and the fourth terms of $J_{PFC}$ are fuzzy entropies. Given the constraints, they are maximized when $\forall v_{uj}, w_{uj} = 1/C$, respectively, which corresponds to the maximum entropy condition. This translates into the following:

$$
-T_{v} \sum_{c=1}^{C} \sum_{i=1}^{N} v_{cj} \ln v_{cj} - T_{w} \sum_{c=1}^{C} \sum_{i=1}^{N} w_{cj} \ln w_{cj}
$$

$$
\leq -T_{v} C \ln \left( \frac{1}{K} \right) - T_{w} K C \ln \left( \frac{1}{C} \right).
$$

Putting everything together gives us the proof of Theorem 4. □

We can now state the convergence of PFCC.

Corollary 1. If Assumption 1 holds, the PFCC algorithm converges to a local maximum of the optimization, with the updates given in Eqs. (14)–(16).

Proof. This corollary is a direct consequence of the above four theorems. If Assumption 1 holds, Theorems 1–3 indicate that the procedure on membership updates given in Section 3.3 never decreases the PFCC objective function. Theorem 4 states that there is a limit to how much this objective function can be increased. So eventually the procedure should stop somewhere before or when it reaches this limit. □

References


About the Author—WILLIAM-CHANDRA TJHI received his B. Eng. degree in Information Engineering, School of Electrical and Electronic Engineering at Nanyang Technological University, Singapore in 2003. He is currently working towards PhD degree at the same school. His research interests include the effectiveness and interpretability aspects of web document clustering and applications to Web Mining.

About the Author—LIHUI CHEN holds the B. Eng. degree in Computer Science and Engineering (Zhejiang University, China) and the PhD degree in Computational Science (University of St. Andrews, UK). She is currently an Associate Professor in the Division of Information Engineering at Nanyang Technological University (NTU) in Singapore. Her research interests include machine learning algorithms and applications, Data Mining and web intelligence.