Available online at www.sciencedirect.com

SciVerse ScienceDirect

Fuzzy Sets and Systems 221 (2013) 48–64

FUZZY sets and systems

www.elsevier.com/locate/fss

Clustering documents with labeled and unlabeled documents using fuzzy semi-Kmeans

Chien-Liang Liu[a,*], Tao-Hsing Chang[b], Hsuan-Hsun Li[c]

[a] *Information and Communications Research Laboratories, Industrial Technology Research Institute, Rm. 709, Bldg. 51, 195, Sec. 4, Chung Hsing Rd., Chutung, Hsinchu 310, Taiwan, ROC*

[b] *Department of Computer Science and Information Engineering, National Kaohsiung University of Applied Sciences, Chien Kung Campus 415, Chien Kung Road, Kaohsiung 807, Taiwan, ROC*

[c] *Department of Computer Science, National Chiao Tung University, 1001 University Road, Hsinchu 300, Taiwan, ROC*

Received 18 February 2012; received in revised form 6 January 2013; accepted 8 January 2013
Available online 16 January 2013

Abstract

While focusing on document clustering, this work presents a fuzzy semi-supervised clustering algorithm called fuzzy semi-Kmeans. The fuzzy semi-Kmeans is an extension of *K*-means clustering model, and it is inspired by an EM algorithm and a Gaussian mixture model. Additionally, the fuzzy semi-Kmeans provides the flexibility to employ different fuzzy membership functions to measure the distance between data. This work employs Gaussian weighting function to conduct experiments, but cosine similarity function can be used as well. This work conducts experiments on three data sets and compares fuzzy semi-Kmeans with several methods. The experimental results indicate that fuzzy semi-Kmeans can generally outperform the other methods.

Keywords: Fuzzy clustering; Semi-supervised learning; Text mining; Fuzzy semi-Kmeans

1. Introduction

Text clustering has attracted an increasing amount of interest recently. Clustering can be used to automatically group retrieved documents into a list of meaningful categories and it is also one of the most widely used techniques for exploratory data analysis, since it can capture the natural structure of the data. Compared with supervised learning, clustering is an unsupervised learning approach, so it does not need labeled data during the course of clustering and its goal is to assign objects into groups so that objects from the same cluster are more similar to each other than objects from different clusters.

Clustering algorithms can be roughly divided into discriminative and generative types. Discriminative algorithms employ pairwise similarities between every document to determine an objective function and optimize this function to obtain clustering result. The *K*-means is a typical discriminative algorithm, which aims at the minimization of the average squared distance between the objects and the cluster centers. The *K*-means is a hard assignment clustering algorithm, where each object belongs to exactly one cluster. However, hard assignment may lead to some problems for those objects that are located in the boundaries among cluster centers. Fuzzy C-Means (FCM) clustering [1], which is a soft version of *K*-means, allows one piece of the object to belong to two or more clusters. Each object has

* Corresponding author. Tel.: +886 3 5913799; fax.: +886 3 5820098.
E-mail address: jackyliu@itri.org.tw (C.-L. Liu).

0165-0114/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.fss.2013.01.004

a membership degree to indicate the degree belonging to each cluster. On the other hand, generative algorithms assume that the data is modeled by underlying parametric distributions, and the objective is to estimate the parameters from observed data. Then, cluster centers can be further obtained from models and their parameters. The Gaussian mixture model is a typical generative algorithm, where a mixture of multiple Gaussian distributions is employed to model the data. A variety of approaches to the problem of mixture decomposition have been proposed, many of which focus on maximum likelihood methods such as expectation maximization (EM) [2] or maximum a posteriori (MAP) estimation. The fuzzy semi-Kmeans algorithm proposed in this work is an extension of *K*-means, but employs EM technique to incorporate a fuzzy membership function to allow the objects to belong to more than one cluster.

Although unsupervised learning approaches do not need labeled data to cluster the documents, proper seeding biases clustering toward a good region of the search space [3]. Meanwhile, it is very common that the experimenter possesses some background knowledge that could be useful in clustering the data. Basically, the background knowledge can be encoded as constraints of the clustering, and they should be satisfied when the clustering process is completed. Restated, the semi-supervised clustering problem can be encoded as an optimization problem with constraints.

In general, the objects should be transformed into a collection of feature vectors in advanced of machine learning process. For instance, a spam mail detection application has to transform each email into a term vector to represent email features, and then a classifier can classify each email into spam or non-spam according to the feature vector. This work employs vector space model and bag of words model to represent a document. A document is represented as an unordered collection of words, disregarding grammar and even word order. Clearly, each document is located in a high-dimensional space. One approach to simplification is to employ dimensionality reduction technique. This work proposes to employ probabilistic latent semantic analysis (PLSA) clustering model [4,5] to reduce dimensionality. Essentially, PLSA, which is inspired and influenced by latent semantic analysis (LSA), aims to analyze the co-occurrences of terms in a corpus of documents to find hidden/latent topics within the corpus. PLSA is a generative model and it is based on a mixture decomposition derived from a latent class model. The reduction process transforms each document from a term vector into a topic vector. Then, fuzzy semi-Kmeans performs semi-supervised clustering on the topic space. The fuzzy semi-Kmeans uses initial labeled examples for seeding. These seeds are used to initialize centers and keep the grouping of labeled data unchanged throughout the clustering process. Essentially, fuzzy semi-Kmeans can employ different fuzzy membership functions to measure the distances between each document and cluster centers. This work employs Gaussian weighting function to measure each document's class membership, but cosine similarity can function properly as well.

The experimental results indicate that fuzzy semi-Kmeans is stable even though only a small amount of labeled data is available. Meanwhile, fuzzy semi-Kmeans generally outperforms the other semi-supervised learning methods. In many real applications, background knowledge is ready, making it appropriate to employ background knowledge to make the learning more fast and effective. Although unsupervised learning does not need labeled data, the experimental results present that a small amount of labeled data can effectively improve the performance.

The main contribution of this work is that this work proposes a novel fuzzy semi-supervised learning algorithm called fuzzy semi-Kmeans. Moreover, this work proposes to use PLSA clustering model to reduce dimensionality. This work uses three data sets in the experiments and compares proposed fuzzy semi-Kmeans with several state-of-the-art algorithms. The experimental results indicate that fuzzy semi-Kmeans is robust when only a small amount of labeled data is available and it generally outperforms several semi-supervised learning methods.

The rest of this work is organized as follows. Section 2 presents related surveys. Section 3 then introduces the fuzzy semi-Kmeans algorithm. Next, Section 4 summarizes the results of several experiments. Conclusions are finally drawn in Section 5.

2. Related surveys

Semi-supervised learning, learning with both labeled and unlabeled data, has recently been studied by many researchers. A variety of semi-supervised algorithms have been proposed, including co-training [6,7], semi-supervised Naive Bayes [8], Transductive support vector machines (TSVM) [9], fuzzy clustering model [10–13], graph-based approaches [14,15], and clustering-based approaches [16,3,17]. Semi-supervised learning methods can be further classified into semi-supervised classification and semi-supervised clustering methods. Semi-supervised classification employs the labeled data along with unlabeled data to construct a more accurate classifier, whereas semi-supervised clustering employs small amount of labeled data to bias the clustering of unlabeled data.

Basically, unsupervised clustering does not need labeled data during the course of clustering and its goal is to assign objects into groups so that objects from the same cluster are more similar to each other than objects from different clusters. Thus, many clustering algorithms aim at the minimization of the cost function, which involves distortion measure between the objects and the cluster representatives. Intuitively, semi-supervised clustering can become an optimization problem with constraints, since labeled examples can be encoded as constraints of the clustering. This technique has been widely used by many researchers [10,12,18]. For instance, Bouchachia and Pedrycz [12] developed a semi-supervised clustering algorithm based on a modified Fuzzy C-Means objective function. In addition to the original FCM objective function, the labeled examples are encoded as an additional regularization term in the complete objective function. Miyamoto et al. [18] employed the same technique in fuzzy semi-supervised clustering. They introduced two variants of FCM that regard labeled data as a regularization term of the objective function. Many classification or clustering algorithms rely on a distance measure between patterns to determine the pattern similarities, so defining an appropriate distance measure between patterns is crucial to many machine learning algorithms. Bouchachia and Pedrycz [19] investigated and quantified the effect of various distance measures on the FCM performance. Essentially, metric learning and dimensionality reduction are highly related, since learning a Mahalanobis metric is identical to learning a linear subspace of the data. Huang and Zhang [20] devised locality sensitive clustering algorithms to preserve locality information in dimensionality reduction.

Moreover, the background knowledge can also be encoded as pairwise constraints of the clustering, and they should be satisfied when the clustering process is completed. Wagstaff et al. [16] devised a semi-supervised variant of *K*-means called COP-KMeans to employ constraints to represent background knowledge. There are two types of constraints, must-link (two instances have to be together in the same cluster) and cannot-link (two instances have to be in different clusters) and they are used in the clustering process to generate a partition that satisfies all the given constraints. Basu et al. [3] introduced two semi-supervised variants of *K*-means clustering that use initial labeled data for seeding. These two algorithms are Seeded-KMeans and Constrained-KMeans. In Seeded-KMeans, the seeds are only used to initialize the *K*-means algorithm, and they are not used in the clustering algorithm. In Constrained-KMeans, the seeds are used to initialize centers and keep the grouping of labeled data unchanged throughout the clustering process. Their experimental results showed that Constrained-KMeans outperforms Seeded-KMeans. Zhong's experimental results [21] supported the same conclusion. The fuzzy semi-Kmeans proposed in this work is inspired by Constrained-KMeans to use the seeds to initialize centers and keep the grouping of labeled data unchanged throughout the clustering process. However, fuzzy semi-Kmeans further employs EM to perform soft cluster assignment, which can incorporate different fuzzy membership functions into the algorithm; while Constrained-KMeans only allows each object to belong to exactly one cluster.

Besides the above approaches, there are many semi-supervised clustering approaches that are extended from the other algorithms. For instance, Finley and Joachims [22] presented an SVM algorithm that trains a clustering algorithm by adapting the item-pair similarity measure. Since all the constraints may not be satisfied, Wang et al. [23] developed an efficient soft-constraint algorithm to obtain a satisfactory clustering result so that the constraints are respected as much as possible. In spectral clustering, Ji et al. [24] proposed to incorporate prior knowledge of cluster membership for document cluster analysis. The prior knowledge indicates pairs of documents that have to be together in the same cluster. Then, the prior knowledge is transformed into a set of constraints. The document clustering task is accomplished by finding the best cuts of the graph under the constraints. Wang and Davidson [25] proposed a framework for constrained spectral clustering algorithm, which preserves the original graph Laplacian and explicitly encodes the constraints.

3. Fuzzy semi-Kmeans

3.1. Notation

The notations that will be used in the following sections are described in this section. Given a set of training documents $\mathcal{D} = \{d_1, \ldots, d_N\}$, the goal is to assign each document into one of the predefined class labels $\mathcal{C} = \{1, \ldots, C\}$. Meanwhile, this work assumes that only small subset of the documents $d_i \in D^l$ are given class labels $y_i \in \mathcal{C}$, and the rest of documents, in subset $\mathcal{D}^u$, do not contain class label information. Restated, the whole documents can be divided into two sets, that is, $\mathcal{D} = \mathcal{D}^l \cup \mathcal{D}^u$. Each document d_i is considered to be an ordered list of word events, $\langle w_{i,1}, \ldots, w_{i,M}\rangle$. This work uses $w_{i,j}$ to denote the word w_j in the document d_i, where w_j is a word in the vocabulary $\mathcal{W} = \langle w_1, \ldots, w_M\rangle$. The entry value for $w_{i,j}$ is represented as $n(d_i, w_j)$, meaning the number of times w_j occurring

in d_i. $P(d_i)$ is used to denote the probability that a word occurrence will be observed in a particular document d_i. $P(w_j|z_k)$ represents the class conditional probability of a specific word conditioned on the unobserved class variable z_k, and finally $P(z_k|d_i)$ denotes a document specific probability distribution over the latent variable space.

3.2. Dimensionality reduction

High-dimensional data sets present many mathematical challenges to machine learning tasks. One of the problems with high-dimensional data sets is that not all the measured variables are important for understanding the underlying phenomena of interest. One approach to simplification is to assume that the data of interest lies on an embedded linear subspace or non-linear manifold within the higher-dimensional space. Dimensionality reduction, which tries to find a lower dimensional representation of the data according to some criterion, is an active research field in machine learning. Many dimensionality reduction algorithms have been developed to accomplish these tasks. Principal component analysis (PCA) and multidimensional scaling (MDS) are classical methods that provide a sequence of best linear approximations to a given high-dimensional observation. In order to resolve the problem of dimensionality reduction in nonlinear cases, many recent techniques, including Isomap [26], locally linear embedding (LLE) [27], and Laplacian eigenmaps [28] have been proposed.

Hofmann et al. [29] proposed an unsupervised learning framework from dyadic data. The dyadic data refers to a domain with two sets of objects, $\mathcal{X} = \{x_1, \ldots, x_N\}$ and $\mathcal{Y} = \{y_1, \ldots, y_M\}$ in which observations are made for (x_i, y_j) with their co-occurrence information. The dyadic data representation is commonly used in many application domains, such as text analysis, computer vision, and computational linguistics. In text analysis, $\mathcal{X}$ represents a document collection and $\mathcal{Y}$ represents the vocabulary set appeared in $\mathcal{X}$. The co-occurrence information (x_i, y_j) represents the number of times term y_j occurring in document x_i. Given the above observations, latent semantic analysis (LSA) is a theory and method for analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSA applies singular value decomposition (SVD) to the term-document matrix and a low-rank approximation of the matrix could be used to determine patterns in the relationships between the terms and concepts contained in the text. LSA has been successfully applied to various applications [30,31]. Inspired by LSA, Hofmann [5] proposed PLSA for factor analysis of binary and count data. As an unsupervised learning method, PLSA does not require labeled data. Additionally, PLSA is a generative model based on a mixture decomposition derived from a latent class model. The latent variable introduced by PLSA can be viewed as topics or concepts embedded in the document collections.

The input of PLSA is a term-document matrix, and PLSA can decompose each document into a set of latent topic variables. Notably, two PLSA models are the aspect model and statistical clustering model [5,29]. In a clustering model for documents, PLSA clustering model assumes that each document belongs to exactly one cluster. Conversely, the aspect model assumes that every occurrence of a word in a document is associated with a unique state z_k of the latent class variable [5]. This work uses PLSA clustering model to reduce dimensionality.

The PLSA clustering model is based on two assumptions: (1) the data is generated by a mixture model and (2) there is a correspondence between mixture components and classes. Under these assumptions, each document is generated using a mixture model, which is parameterized by Φ. The generating process can be described using two steps: select a mixture component based on the mixture weights and, then, generate a document based on this selected mixture component and its parameters. Thus, the likelihood of document d_i is the sum of total probability over all mixture components as shown in the following equation:

$$P(d_i|\Phi) = \sum_{k=1}^{K} P(z_k)P(d_i|z_k;\Phi) \tag{1}$$

The number of topics is K and z_k represents the kth component. The likelihood of $\mathcal{D}$ is simply the product over all the documents, since each document is independent of the others, given the model. Then, log likelihood of $\mathcal{D}$ can be obtained using Eq. (2). The standard procedure for maximum likelihood estimation in latent variable models is the EM algorithm, which includes E-step and M-step. Similarly, the parameter estimation of PLSA clustering model can be achieved by using the expectation maximization (EM) algorithm.

$$\ln P(\mathcal{D}|\Phi) = \sum_{d_i \in \mathcal{D}} \ln \sum_{k=1}^{K} P(z_k)P(d_i|z_k;\Phi) \tag{2}$$

Algorithm 1. PLSA clustering algorithm.

Input: A $N \times M$ term-document matrix H, and the number of topics K.
Output: A $N \times K$ document-topic distribution matrix Q, where each entry Q_{ik} represents the probability of document d_i assigned to topic k.

begin
  Choose K vectors from H randomly and they are the initial values of $\Theta_1, \ldots, \Theta_K$
  Topic proportion components $\pi_1 = \cdots = \pi_K = \frac{1}{K}$
  repeat
    E-step: Compute latent variable posterior probability Q
    $$Q = P(z_k|d_i) = \pi_k \exp\left(\sum_{j=1}^{M} n(d_i, w_j) \ln \Theta_{kj}\right).$$
    M-step: Update proportion parameter π_k and Θ_k for $k = 1, \ldots, K$
    $$\pi_k = P(z_k) = \frac{\sum_{i=1}^{N} Q_{ik}}{\sum_{k'=1}^{K}\sum_{i=1}^{N} Q_{ik'}}.$$
    $$\Theta_{kj} = \frac{\sum_{i=1}^{N} P(z_k|d_i)n(d_i, w_j)}{\sum_{j=1}^{M}\sum_{i=1}^{N} P(z_k|d_i)n(d_i, w_j)} = \frac{\sum_{i=1}^{N} Q_{ik}n(d_i, w_j)}{\sum_{j'=1}^{M}\sum_{i=1}^{N} Q_{ik}n(d_i, w_{j'})}, \text{ where } j = 1, \ldots, M$$
  until *convergence*
  return Q
end

Algorithm 1 shows the PLSA clustering algorithm. The inputs of the algorithm include term-document matrix H and the number of topics K. The initial value of Θ_k is determined randomly. The E-step and M-step are estimated according to the equations listed in Algorithm 1. The output is a document-topic distribution matrix Q. The dimensionality reduction can be achieved by using the document-topic distribution matrix Q, since each document can be represented by a topic distribution vector.

3.3. Fuzzy semi-Kmeans

Many clustering algorithms aim at the minimization of the cost function, which involves distortion measure between the objects and the cluster representatives. The K-means locally minimizes the average squared distance between the objects and the cluster centers. The Fuzzy C-Means has similar objective function, but it extends K-means to include the degree of membership information, which indicates the confidence in the assignment of the object to the cluster. The above two clustering algorithms can be derived from the optimization of cost functions.

The EM algorithm is another popular technique for analyzing clustering algorithms, since it is a statistically formalized method and it provides more detailed information about the clustering result. As mentioned above, the EM algorithm is a general method of finding maximum likelihood solutions for models having latent variables. If the set of all observed data is $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the set of model parameters is denoted by Θ and the set of all latent variables is $\mathbf{Z}$, the E-step employs the current parameter values Θ^{old} to estimate the posterior distribution of the latent variables given by $P(\mathbf{Z}|\mathbf{X}, \Theta^{old})$. Then, the posterior distribution is used to compute the expectation of complete data log likelihood to estimate new parameter value Θ^{new}. The expectation of complete data log likelihood over the posterior distribution of latent variables is denoted by $Q(\Theta, \Theta^{old})$ as shown in Eq. (3). The M-step estimates new parameter Θ^{new} from the maximization of this function as shown in Eq. (4)

$$Q(\Theta, \Theta^{old}) = \mathbf{E}_{\mathbf{Z}|\mathbf{X},\Theta^{old}}[\ln\ P(\mathbf{X}, \mathbf{Z}|\Theta)] \tag{3}$$

$$\Theta^{new} = \underset{\Theta}{\operatorname{argmax}}\ Q(\Theta, \Theta^{old}) \tag{4}$$

The K-means can be modeled using EM algorithm on a mixture of C Gaussians under certain assumptions, where C is the number of clusters of K-means. Whereas the K-means algorithm performs a hard assignment of data points to clusters, in which each data point is associated uniquely with one cluster, the EM algorithm makes a soft assignment based on the posterior probabilities. Consider a Gaussian mixture model with C components in which the means of these components are $\mu_1, \ldots, \mu_C$ and the common covariance matrices of the mixture components are given by $\Sigma = \varepsilon\mathbf{I}$, where $\mathbf{I}$ represents identity matrix and ε is a shared parameter by all of the components. As $\varepsilon \longrightarrow 0$, the expected

complete data log likelihood can be written as [32]

$$\mathbf{E}_{\mathbf{Z}|\mathbf{X},\Theta^{old}}[\ln\ P(\mathbf{X},\mathbf{Z}|\Theta)] = -\frac{1}{2}\sum_{i=1}^{N}\sum_{c=1}^{C} P(z_c|\mathbf{x}_i;\Theta)\|\mathbf{x}_i-\mu_c\|^2 + \text{const} \tag{5}$$

Thus, the maximization of the above expected complete data log likelihood is equivalent to the minimization of the K-means objective function with the hard assignment restriction, that is, each data is assigned to one cluster as shown in the following equation:

$$P(z_c|\mathbf{x}_i;\Theta) = \begin{cases} 1 & \text{if } c = \underset{l}{\operatorname{argmin}}\ \|\mathbf{x}_i-\mu_l\|^2 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Essentially, the posterior probability function is not limited to the hard assignment as shown in Eq. (6), the other soft assignments can be used as well. The connection between K-means and EM described above inspires us to propose a fuzzy semi-supervised K-means algorithm. The posterior probability function is replaced by a fuzzy membership function with constraints. Fuzzy membership can then be incorporated into the E-step of EM algorithm. This study uses a matrix U to keep track of fuzzy membership information, where U_{ic} denotes the degree of membership of document i in the cluster c. In other words, U_{ic} is used to represent the posterior probability $P(z_c|\mathbf{x}_i)$. In M-step, U_{ic} is used to estimate new model parameters, which are cluster centers $\mu_1, \ldots, \mu_C$, from the maximization of log likelihood function.

Moreover, a small amount of labeled examples is available. The fuzzy semi-Kmeans employs these labeled examples for two main purposes. First, these labeled examples can determine the initial guess of the centers. Practically, clustering algorithms such as K-means and Fuzzy C-Means are sensitive to the initial guess. Thus, the seeds can provide a better guess for the algorithm to cluster the documents. Second, these labeled examples can bias clustering toward a better searching space during the course of clustering.

Algorithm 2. Fuzzy semi-Kmeans algorithm.

Input: A $N \times M$ term-document matrix H, the number of topics K, the number of clusters C and the seeds $S_1, \ldots, S_C$. Without loss of generality, S_c represents the document seeds for cluster c ($c = 1, \ldots, C$).
Output: A $N \times C$ document-cluster membership matrix U

```
1  begin
     H_i ← H_i / ||H_i||, where H_i is the ith row of H and i = 1, ..., N
2    H̃ ← PLSA_Clustering(H, K)
3    μ_c ← (1/|S_c|) Σ_{d_i ∈ S_c} H̃_i, where μ_c is the center of cth cluster and c = 1, ..., C
4    repeat
5      for i = 1 To N do
6        for c = 1 To C do
7          if d_i is a document of S_c then
8            U_ic ← 1
9            U_ic' ← 0, for c' = 1, ..., C and c' ≠ c
10           break
11         else
12           |U_ic ← e^{-||H̃_i - μ_c||^2 / 2σ^2}
13         end
14       end
15       Normalize U_ic so that the sum of each row of U is 1.
16     end
17     for c = 1 To C do
18     |Z_c ← Σ_{i=1}^{N} U_ic
19     end
20     for c = 1 To C do
21     |μ_c ← (1/Z_c) Σ_{i=1}^{N} U_ic × H̃_i
22     end
23   until convergence
24   return U
25 end
```

Algorithm 2 presents the fuzzy semi-Kmeans algorithm. The fuzzy membership function used in Algorithm 2 is a Gaussian weighting function as shown in Eq. (7), where μ_c is the location of the center and σ is used to control the degree of membership of $\mathbf{x}_i$ in the cluster c and $\| \mathbf{x}_i - \mu_c \|$ represents the distance between $\mathbf{x}_i$ and μ_c. The points close to the center will be important and points far away will be relatively insignificant. The distance is represented using a fuzzy degree number, since the value of U_{ic} is a number ranging from 0 to 1. Additionally, the cluster centers $\mu_1, \ldots, \mu_C$ are the parameter Θ of the model, and they are updated iteratively

$$U_{ic} = e^{-\|\mathbf{x}_i - \mu_c\|^2 / 2\sigma^2} \tag{7}$$

The inputs of fuzzy semi-Kmeans algorithm include a $N \times M$ term-document matrix H, the number of clusters C, the number of topics K and document seeds $S_1, \ldots, S_C$. Each row of H represents a document, and each column represents a term feature of the document. Initially, each row of H has to be normalized, then the normalized matrix and the number of topics K are fed into PLSA clustering algorithm to obtain a topic-document matrix $\tilde{H}$. The main difference between H and $\tilde{H}$ is that each document in H is denoted by a term vector; while each row in $\tilde{H}$ is represented by a topic vector. Initial seeds are labeled documents, making it feasible to use the seeds $S_1, \ldots, S_C$ to calculate initial cluster centers, namely $\mu_1, \ldots, \mu_C$. The above processes are listed in Lines 2–4 of fuzzy semi-Kmeans algorithm.

When the parameters $\mu_1, \ldots, \mu_C$ are obtained, the membership degree U_{ic} can be calculated by using Gaussian weighting function with distance measurement. Additionally, each seed's cluster information is known, explaining why this work assigns seed's corresponding cluster membership as 1. The membership degree matrix U is then normalized. The above processes are listed in Lines 6–17 of fuzzy semi-Kmeans algorithm.

When the membership degree matrix U is changed, the algorithm must use the new posterior probability of latent variable to calculate new parameters, which are cluster centers, with labeled and unlabeled documents. This work uses a normalization vector Z to represent a normalization factor, where each Z_c represents the sum of membership degrees of documents in cluster c. Then, the algorithm can calculate new cluster centers. The above processes are listed in Line 18-23 of fuzzy semi-Kmeans algorithm. When the algorithm converges, the algorithm outputs a membership degree matrix U.

Besides Gaussian weighting function, cosine similarity can be used as well. In information retrieval (IR) or natural language processing (NLP), vector space model is often used to represent documents, where each document is represented as a vector and each dimension corresponds to a distinct term. Cosine similarity, which is a measure of similarity between two vectors by measuring the cosine of the angle between them, is often used for distance measurement in IR or NLP. The result of cosine similarity is a number ranging from 0 to 1. Moreover, the membership function employed in the algorithm should be normalized, since it is derived from posterior probability distribution.

4. Experiments

4.1. Data corpora

Three corpora are used in the experiments. The 20 Newsgroups and Classic3 are popular corpora, which are conventionally used in text analysis experiment. In addition to the above two corpora, system performances are evaluated using academic paper information collected from the web site CiteULike. [1]

- The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned evenly across 20 different newsgroups. Some of the newsgroups are similar to each other, while others are highly unrelated. The 20 Newsgroups data set has become a popular data set for experiments in text analysis applications of machine learning methods, such as text classification and text clustering.
- As a conventional benchmark data set used in text mining, Classic3 data set [2] comprises three document collections: CISI (1460 information retrieval abstracts), CRAN (1400 aeronautical systems abstracts), and MED (1033 medical abstracts).
- CiteULike, a social bookmarking web site, promotes and fosters the sharing of scientific references among researchers. Scientists can annotate relevant academic papers with tags and share the information with other in-

[1] CiteULike: http://www.citeulike.org/
[2] Classic3: ftp://ftp.cs.cornell.edu/pub/smart/

Table 1
The CiteULike corpus.

Attribute	Graphics	Databases	Programming languages
Number of papers	741	1289	1364
Number of terms in abstractions	65,372	115,346	110,184
Number of tags	2013	3126	4983

dividuals. CiteUlike integrates two categories of software: the new Web 2.0 breed of social bookmarking services and conventional bibliographic management software. While web bookmarks are simple URLs, citations are slightly more complex and include meta-data such as journal names, authors, and page numbers. However, meta-data information lacks paper category information, which is necessary to evaluate the performance of classification or clustering. In this work, papers are assigned to communities according to their venues by using the classification system developed by Microsoft's academic search service [3]; this service provides the ranking of publications in different fields. For instance, the "Graphics" field includes ACM Transactions on Graphics (TOG) and IEEE Computer Graphics and Applications (CGA). A paper published in TOG is classified as "Graphics" field. The above paper classification mechanism is also used by Shi et al. [33]. Obviously, some publications may belong to more than one field, explaining why this work focuses on fields that are highly unrelated. The computer science domain is selected, in which 3394 articles are selected from three fields. The paper's full text is unavailable in CiteULike, explaining why the paper's abstract and its tags annotated by users are used here as the paper content. This corpus can be downloaded from http://islab.cis.nctu.edu.tw/download/. Table 1 summarizes the information of the data set.

During preprocessing, the stop words are removed from these data sets since stop words fail to provide sufficient information for the clustering task. Meanwhile, punctuation marks are removed and all English letters are converted into lower case. Finally, the stemming process is applied to the words.

4.2. Evaluation metric

As mentioned earlier, semi-supervised learning methods can be further classified into semi-supervised classification and semi-supervised clustering methods. Both semi-supervised classification methods and semi-supervised clustering methods are compared, respectively, with the proposed fuzzy semi-Kmeans algorithm. Many semi-supervised clustering methods use must-link and cannot-link constraints to bias the clustering of unlabeled data without using labeled information. Thus, these methods cannot be evaluated using a classification evaluation metric. Notably, the experiments are made more objective by using two evaluation metrics in the experiments. The fuzzy semi-Kmeans is a semi-supervised clustering method, but the labeled examples can provide category information. Thus, fuzzy semi-Kmeans can be evaluated using a clustering evaluation metric as well as a classification evaluation metric. When the fuzzy semi-Kmeans is compared with semi-supervised classification methods, system performances are evaluated using a classification evaluation metric. When the fuzzy semi-Kmeans is compared with semi-supervised clustering methods, a clustering evaluation metric is used. These two evaluation metrics are similar, as described in the following sections.

4.2.1. Clustering evaluation metric

This work compares the generated clusters by using the F_1 cluster evaluation measure [34]. The F_1 cluster evaluation measure considers both precision and recall, where precision and recall here are computed over pairs of documents of which the two label assignments either agree or disagree

- *True positives* (*TP*): The clustering algorithm placed the two articles in a pair into the same cluster, and data corpus has them in the same class.
- *False positives* (*FP*): The clustering algorithm placed the two articles in a pair into the same cluster, but data corpus has them in differing classes.

[3] Microsoft Academic Search: http://academic.research.microsoft.com

- *True negatives* (*TN*): The clustering algorithm placed the two articles in a pair into differing clusters, and data corpus has them in differing classes.
- *False negatives* (*FN*): The clustering algorithm placed the two articles in a pair into differing clusters, but data corpus has them in the same class.

Similar to traditional information retrieval definition, Eq. (8) shows the formulas of precision, recall and F_1 evaluation

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP} \\ \text{Recall} &= \frac{TP}{TP + FN} \\ F_1 &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned} \tag{8}$$

4.2.2. Classification evaluation metric

For each class, the correctness of a classification can be evaluated by computing the number of correctly recognized class examples (true positives), the number of correctly recognized examples that do not belong to the class (true negatives), and examples that either were incorrectly assigned to the class (false positives) or that were not recognized as class examples (false negatives) [35]. Eq. (9) shows the definition of precision, recall and F_1 score, where *TP* represents the number of true positives, *TN* represents the number of true negatives, *FP* represents the number of false positives, and *FN* represents the number of false negatives. Meanwhile, many classification tasks employed in the experiments belong to multi-class problem, so the evaluation should take into account the prediction result of every class. Macro-average F_1, which is the average on F_1 scores of all the classes, is used in system performance evaluation. Eq. (10) shows the definition of macro-average F_1 score, where K is the number of classes and F_{1i} is the F_1 score of ith class

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP} \\ \text{Recall} &= \frac{TP}{TP + FN} \\ F_1 &= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned} \tag{9}$$

$$\text{Macro-average } F_1 = \frac{\sum_{i=1}^{K} F_{1i}}{K} \tag{10}$$

4.3. Comparison methods

Several semi-supervised learning methods are used for comparison. Among these methods, graph-based and TSVM are semi-supervised classification methods. Meanwhile Constrained-KMeans, constrained spectral clustering and constrained normalized cut are semi-supervised clustering methods:

- *Graph-based semi-supervised learning*: Graph-based approach has been extensively adopted in semi-supervised learning methods. This work adopts the approach developed by Goldberg and Zhu [36]. Similar to the other graph-based semi-supervised learning approaches, their approach uses a graph to represent labeled and unlabeled data. Each document is a node in the graph, and each node is connected to an observed node called a dongle. The edge weight between a labeled document and its dongle is a large number M, while the weight between an unlabeled document and its dongle is 1. Each unlabeled document x_i is connected to the k nearest labeled documents and the k' nearest unlabeled documents. Different weight coefficients are given in the above two cases. The original problem can then be transformed into a optimization problem with constraints. In this case, a closed form solution can be obtained. Goldberg and Zhu [36] used support vector regression (SVR) in their proposed graph-based semi-supervised learning approach to perform an initial prediction, but this work focuses on document classification or cluster problems. The experimental results show that graph-based semi-supervised with support vector machines

(SVM) outperforms graph-based semi-supervised with SVR, explaining why this work employs graph-based semi-supervised with SVM. This work conducts graph-based semi-supervised learning experiments using LIBSVM [37] package with radial basis function (RBF) kernel function. Moreover, the value of k' is 5, and k is the number of seeds divided by 10 in the experiments.

- *TSVM*: TSVM is an extension of the standard SVM with unlabeled data. The experiment is performed using SVM light [9] with a linear kernel function. For multi-class classification, the one-against-all approach is used in the experiment.
- *Constrained-KMeans*: Basu et al. [3] devised two semi-supervised variants of *K*-means clustering that use initial labeled data for seeding. The two algorithms are Seeded-KMeans and Constrained-KMeans. According to their experimental results, Constrained-KMeans outperforms Seeded-KMeans. Meanwhile, Constrained-KMeans also outperforms COP-KMeans [16].
- *Constrained spectral clustering* (*abbreviated as CSC*): Wang and Davidson [25] devised a flexible and generalized framework for constrained spectral clustering. Constrained spectral clustering can be formulated as a constrained optimization problem by adding a constraint to the original objective function of spectral clustering. Constrained spectral clustering encodes the degree of belief (weight) in must-link and cannot-link constraints. In this experiment, discrete values are used to represent the constraints rather than degree of belief. Restated, user supervision is encoded with a constraint matrix Q, which only uses binary constraints:

$$Q_{ij} = Q_{ji} = \begin{cases} +1 & \text{if } ML(i, j) \\ -1 & \text{if } CL(i, j) \\ 0 & \text{no supervision available} \end{cases}$$

  Besides affinity matrix and constraint matrix, an additional variable β is necessary in constrained spectral clustering. The β can be used to convert the optimization problem into a generalized eigenvalue problem. Wang and Davidson discussed how to set β in [25]. A collection of N data instances is modeled by an undirected, weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{A})$, where each data instance corresponds to a vertex in $\mathcal{V}$; $\mathcal{E}$ is the edge and $\mathcal{A}$ is the associated affinity matrix. Meanwhile, $\bar{Q}$ is the normalized constraint matrix. The β in this experiment is

$$\beta = 0.5 \times \lambda_{max}\, \mathrm{vol}(\mathcal{G})$$

  where λ_{max} is the largest eigenvalue of $\bar{Q}$.
- *Constrained normalized cut* (*abbreviated as CNC*): Ji et al. [24] devised a semi-supervised clustering model that incorporates prior knowledge about documents' membership for document cluster analysis. The prior knowledge indicates several pairs of documents which the user wishes to group into the same cluster. That work used the normalized cut [38] as cost function and also introduced a penalty term with a penalty coefficient matrix to incorporate the user's prior knowledge on the data set. The document clustering task is then performed by identifying the cluster set that globally optimizes the constrained cost function. Meanwhile, users can control the degree of enforcement of the prior knowledge by using a set of parameters. In this experiment, the control parameter β is 20, i.e. the same as the value used in [24]. Meanwhile, this work adopts Ji's experimental procedure, which uses a normalized term frequency-inverse document frequency (TF-IDF) vector to represent each document.

4.4. Semi-supervised learning experiments

Owing to the focus on semi-supervised learning performances, the experiments use a small amount of labeled examples. Examples are randomly selected as the labeled ones, and the remaining examples are unlabeled. Exactly how the number of labeled examples impacts system performances is further evaluated in the experiments by using different percentages of labeled examples. Each evaluation is performed 10 times, and the average of the results becomes its performance outcome. Although a time-consuming process, this evaluation can be more objective, since the labeled examples are selected randomly. In the fuzzy semi-Kmeans algorithm, the value of σ is 5 in all of the experiments. The additional parameters and values for the other comparison methods are described above. For instance, the control parameter β in CNC method is 20 in the experiments.

Semi-supervised classification methods and semi-supervised clustering methods are used in the experiments for comparison. Meanwhile, different metrics are applied to these two kinds of methods. The first data set is 20 Newsgroups, which is a popular data set for text classification evaluation. As mentioned above, some of the newsgroups

Table 2
Semi-supervised learning results on talk subject newsgroups (4 Newsgroups).

Percentage of labeled examples	Semi-supervised classification			Semi-supervised clustering			
	FSK (dimension=10)	Graph-based	TSVM	FSK	Constrained-KMeans	CSC	CNC
1%	0.6020	0.3051	0.4794	0.5315	0.3575	0.3919	0.4065
2%	0.6672	0.5118	0.5546	0.5700	0.3624	0.3998	0.4135
3%	0.6569	0.5317	0.6197	0.5711	0.3536	0.3998	0.4122
4%	0.6715	0.6061	0.6249	0.5776	0.3638	0.3999	0.4037
5%	0.6763	0.6080	0.6745	0.5754	0.3660	0.4069	0.4132

Table 3
Semi-supervised learning results on "comp.graphics", "rec.autos", "sci.crypt" and "talk.politics.guns" Newsgroups.

Percentage of labeled examples	Semi-supervised classification			Semi-supervised clustering			
	FSK (dimension=5)	Graph-based	TSVM	FSK	Constrained-KMeans	CSC	CNC
1%	0.9339	0.5119	0.5559	0.8738	0.3550	0.4057	0.7231
2%	0.9339	0.6399	0.7084	0.8737	0.3599	0.4056	0.7148
3%	0.9343	0.7280	0.7431	0.8745	0.3636	0.3998	0.7202
4%	0.9344	0.7364	0.7803	0.8746	0.3647	0.3998	0.7211
5%	0.9343	0.7767	0.8141	0.8745	0.3643	0.3998	0.7327

Table 4
Semi-supervised learning results on Classic3 data set.

Percentage of labeled examples	Semi-supervised classification			Semi-supervised clustering			
	FSK (dimension=10)	Graph-based	TSVM	FSK	Constrained-KMeans	CSC	CNC
1%	0.9833	0.7804	0.9771	0.9687	0.9233	0.5077	0.8506
2%	0.9859	0.8545	0.9804	0.9729	0.9256	0.5076	0.8285
3%	0.9865	0.9115	0.9811	0.9740	0.9266	0.5077	0.8320
4%	0.9867	0.9128	0.9803	0.9743	0.9250	0.5075	0.8320
5%	0.9872	0.9394	0.9820	0.9750	0.9265	0.5075	0.8855

are very closely related to each other, while others are highly unrelated. Different combinations of the newsgroups are employed to evaluate system performances. The first combination is about the newsgroups which are very close. The newsgroups are all talk subjects, including "talk.politics.misc", "talk.politics.guns", "talk.politics.mideast", and "talk.religion.misc". Table 2 shows the experimental results, where fuzzy semi-Kmeans is abbreviated as "FSK" and the dimension represents the reduced dimensionality using PLSA clustering model. The second combination is about the newsgroups which are highly unrelated to each other. Table 3 presents the experimental results.

The second data set is Classic3 data set, including three categories. Table 4 presents the experimental results. The third data set is CiteULike data set. Table 5 shows the experimental results. Besides, additional experiments are conducted to compare the effect of σ parameter of Gaussian weighting function and dimensionality reduction. This work uses different values of σ^2 to evaluate system performances on Classic3 and 20 Newsgroups data sets. In dimensionality reduction, fuzzy semi-Kmeans is applied to two data sets, including 20 Newsgroups and Classic3 data sets. The experimental results are presented in Tables 6 and 7, respectively.

4.5. Discussion

The first experiment is the evaluation on 20 Newsgroups data set. Two combinations of newsgroups are used in the experiments. The purposes of these experiments focus on two issues. The first one focuses on whether these

Table 5
Semi-supervised learning result on CiteULike data set.

Percentage of labeled examples	Semi-supervised classification			Semi-supervised clustering			
	FSK (dimension=15)	Graph-based	TSVM	FSK	Constrained-KMeans	CSC	CNC
1%	0.8379	0.4944	0.6678	0.7239	0.4798	0.5220	0.4789
2%	0.8507	0.6220	0.7087	0.7465	0.4799	0.5222	0.4795
3%	0.8617	0.6953	0.7768	0.7596	0.4851	0.5222	0.4769
4%	0.8614	0.7146	0.8084	0.7590	0.4872	0.5219	0.4791
5%	0.8653	0.7449	0.8200	0.7619	0.4911	0.5220	0.4771

Table 6
Dimensionality reduction effect on talk newsgroups (dimension=10).

Percentage of labeled examples	Semi-supervised classification		Semi-supervised clustering	
	Dimensionality reduction	Without dimensionality reduction	Dimensionality reduction	Without dimensionality reduction
1%	0.6020	0.5138	0.5315	0.3687
2%	0.6672	0.5715	0.5700	0.4178
3%	0.6569	0.6025	0.5711	0.4409
4%	0.6715	0.6201	0.5776	0.4571
5%	0.6763	0.6498	0.5754	0.4863

Table 7
Dimensionality reduction effect on Classic3 data set (dimension=10).

Percentage of labeled examples	Semi-supervised classification		Semi-supervised clustering	
	Dimensionality reduction	Without dimensionality reduction	Dimensionality reduction	Without dimensionality reduction
1%	0.9833	0.9143	0.9687	0.8482
2%	0.9859	0.9434	0.9729	0.8987
3%	0.9865	0.9495	0.9740	0.9093
4%	0.9867	0.9515	0.9743	0.9129
5%	0.9872	0.9534	0.9750	0.9163

methods can function well on multi-class problems. Some algorithms are designed for binary class classification or clustering, so these experiments can be used to evaluate whether these methods can be extended to multi-class problems. For instance, TSVM is a binary classifier, and this work employs one-against-all approach for multi-class problems. The constrained spectral clustering proposed by Wang and Davidson [25] is also a binary cluster method. In general, spectral clustering can be extended to multi-class method by using spectral embedding technique, which embeds data points in the subspace of the K eigenvectors of graph Laplacian matrix. Then, K-means can be applied to cluster embedded points. The second one is to evaluate whether these methods can function properly when the boundaries among clusters are not clear.

Fig. 1a and b presents the experimental results on 20 Newsgroups talk subject data set. There are 4 Newsgroups in this data set and they are all related to talk subject. Clearly, the boundaries among these newsgroups are not clear. In both semi-supervised clustering and semi-supervised classification experiments, fuzzy semi-Kmeans outperforms the other methods. The fuzzy semi-Kmeans works stably even though only a small amount of labeled documents is available.

Fig. 2a and b shows the experimental results on "comp.graphics", "rec.autos", "sci.crypt" and "talk.politics.guns" newsgroups. It is apparent that the boundaries among these newsgroups are clear and this is also a multi-class data set. Compared with the experiments on talk subject, most semi-supervised learning methods' performances can be improved a lot. In semi-supervised learning experiments, fuzzy semi-Kmeans outperforms the other methods. These two experimental results also show that it seems like CSC fails to function properly in multi-class problems.

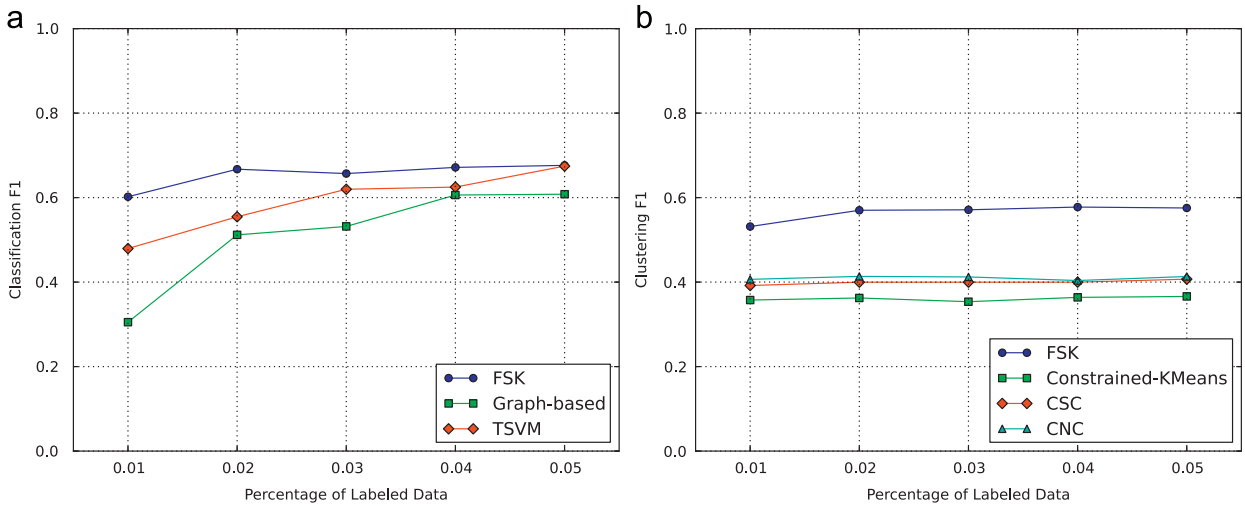

Fig. 1. Experimental results on 20 Newsgroups talk subject: (a) semi-supervised classification result and (b) semi-supervised clustering result.

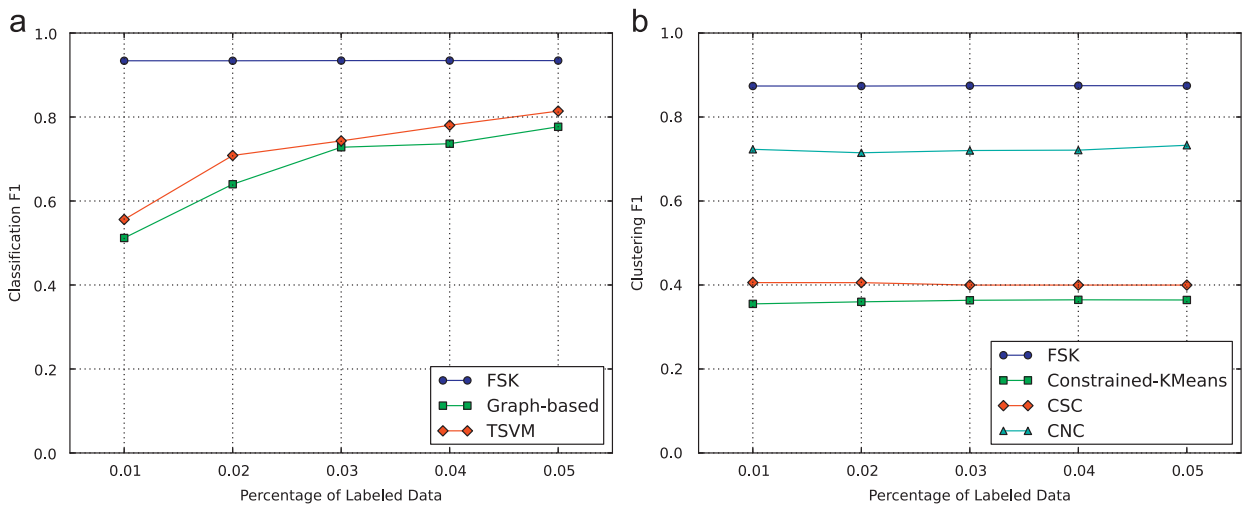

Fig. 2. Experimental results on comp.graphics, rec.autos, sci.crypt and talk.politics.guns: (a) semi-supervised classification result and (b) semi-supervised clustering result.

The second experiment is the evaluation on Classic3 data set. It is a well-known benchmark data set used in text mining. This data set comprises three different document collections and it has different characteristics with newsgroup data set. The boundaries among the clusters are very clear in this data set. Similarly, all of the methods are applied to this data set. Fig. 3a and b shows the experimental results. In semi-supervised classification experiments, fuzzy semi-Kmeans and TSVM can work very well and their performances are almost identical. In semi-supervised clustering experiments, fuzzy semi-Kmeans outperforms the other methods.

The final experiment is the evaluation on CiteULike data set. There are three fields in this data set. The abstracts and tags annotated by users are used to represent papers. In general, the number of words of an abstract is about 200–250. The abstract is similar to the summary of a paper, while tags are similar to the keywords of a paper. Thus, abstract and tags can be viewed as condensed information of a paper. The purpose of this experiment focuses on whether these methods can be applied to the data set, where only condensed information is available.

Fig. 4a and b shows the experimental results on CiteULike data set. In semi-supervised classification and semi-supervised clustering experiments, fuzzy semi-Kmeans outperforms the other methods. The performances of

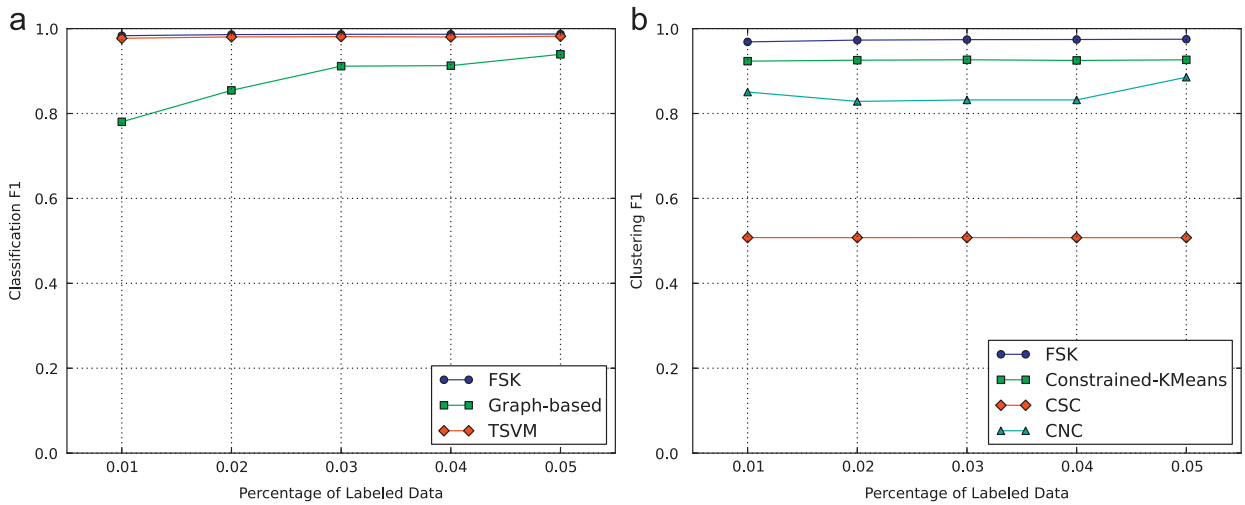

Fig. 3. Experimental results on Classic3: (a) semi-supervised classification result and (b) semi-supervised clustering result.

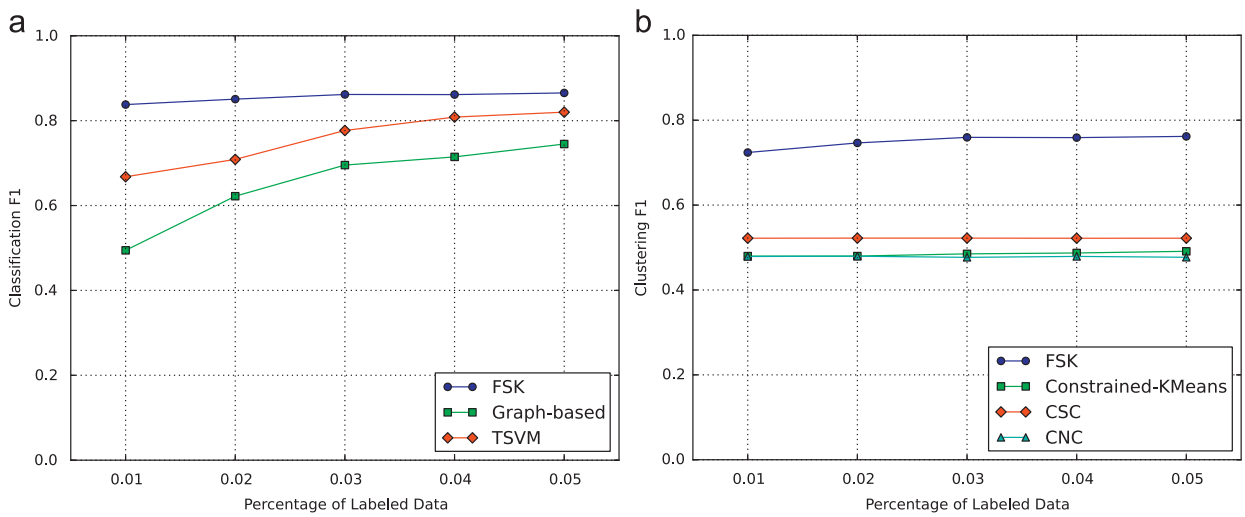

Fig. 4. Experimental results on CiteULike: (a) semi-supervised classification result and (b) semi-supervised clustering result.

Constrained-KMeans, CSC and CNC are almost identical, but they all fail to function properly on this data set. On the other hand, fuzzy semi-Kmeans can function properly and stably.

This work further conducts several experiments to evaluate the effects of σ parameter of Gaussian weighting function and dimensionality reduction. Fig. 5a and b shows the experimental results on Classic3 and 20 Newsgroups data sets using different values of σ^2, ranging from 0.001 to 100. The two experimental results show similar results, system performances become stable when σ^2 exceed 0.09. Practically, cross-validation technique can be used to determine the best σ, but cross-validation technique may fail to function properly in semi-supervised learning applications. One of the reasons is that the number of labeled examples may be few in semi-supervised learning applications, making it infeasible to use available labeled examples to determine the parameter. Compared to Gaussian weighting function, cosine similarity measures the similarity of documents without additional parameters, making it feasible to use cosine similarity function in text analysis applications.

Additionally, this work analyzes whether the proposed algorithm can benefit from dimensionality reduction. The fuzzy semi-Kmeans is applied to two data sets with and without dimensionality reduction. Tables 6 and 7 present the experimental results on 20 Newsgroups talk subject and Classic3 data sets, respectively. The experimental results

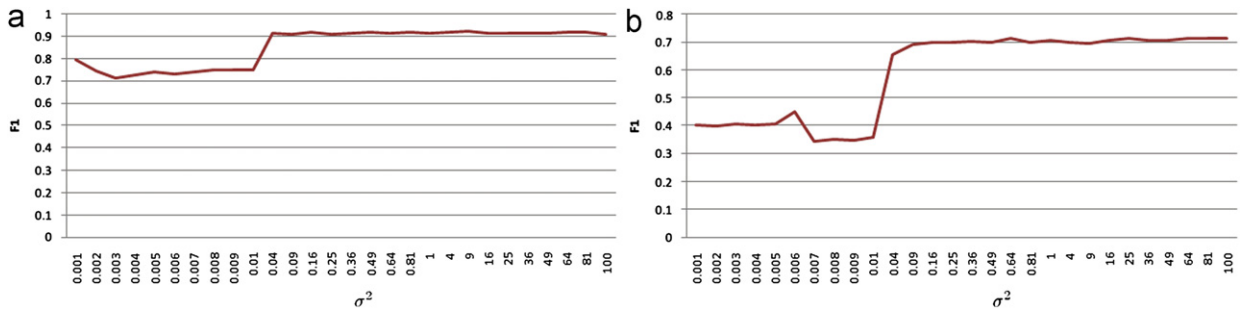

Fig. 5. Experimental results on the effect of σ: (a) Classic3 clustering result and (b) 20 Newsgroups (comp.graphics, rec.autos, sci.crypt and talk.politics.guns) clustering result.

Table 8
Experimental results on the effect of fuzzy semi-Kmeans fixing mechanism.

Percentage of labeled examples	20 Newsgroups talk data set (without dimensionality reduction)	
	With fixing mechanism	Without fixing mechanism
1%	0.3794	0.3268
2%	0.4264	0.3539
3%	0.4552	0.3607
4%	0.4662	0.3741
5%	0.4869	0.3795

indicate that fuzzy semi-Kmeans can benefit from dimensionality reduction. The main reason is that each document is a sparse and high-dimensional vector when using term vector representation. The PLSA can discover the hidden concepts embedded in a document collection, and concept vector representation can provide more distinctive information for text analysis tasks.

As mentioned above, fuzzy semi-Kmeans fixes the clusters of labeled examples during the course of clustering. To further evaluate the impact of fixing mechanism on system performances, we conduct experiments on 20 Newsgroups talk subject data set. Similarly, different percentages of labeled examples are used in the experiments and clustering F_1 is used as evaluation metric. Table 8 presents the experimental results. The experimental results indicate that fuzzy semi-Kmeans can improve performances when fixing mechanism is used in the proposed algorithm. In semi-supervised learning, the number of labeled examples is insufficient for the system to train an accurate and robust model, explaining why semi-supervised learning methods must use unlabeled examples to enhance their models. The proposed method uses fixing mechanism to propagate the label information to unlabeled examples, and it can bias clustering toward a good region of the search space. The proposed method without fixing mechanism will behave like an unsupervised learning method, since the algorithm does not consider labeled examples in the clustering process. Moreover, the clusters of labeled examples can be viewed as prior knowledge, giving base for the system to use the prior knowledge iteratively.

Furthermore, we further analyze the words for different clusters when the proposed method completes the clustering task. Table 9 presents the experimental results in which the top 10 frequent words for four talk newsgroups are presented. To further identify their corresponding newsgroups, we compare these words with the words appeared in the talk newsgroups. Cluster 2 and cluster 3 are “talk.politics.mideast” and “talk.politics.guns”, respectively. However, it is difficult to identify cluster 1 and cluster 4 only from the top 10 frequent words. The main reason is that these two newsgroups, “talk.politics.misc” and “talk.religion.misc”, are both about miscellaneous news articles, explaining why it is difficult to distinguish these two newsgroups by using the words. The clustering experimental results also conform to the results, since the F_1 clustering value on this data set is about 0.5.

This work employs three data sets and various combinations to evaluate system performances. The experimental results indicate that fuzzy semi-Kmeans works stably and it can benefit from a small amount of labeled examples. Even though the boundaries among clusters are not clear, fuzzy semi-Kmeans can function properly. This work compares

Table 9
Top 10 frequent terms for talk data set.

Ranking	Cluster 1	Cluster 2	Cluster 3	Cluster 4
1	tho	amid	calgari	bondag
2	survivalist	busload	vancouver	kej
3	rcander	jen	sloan	bidder
4	guy	unintention	kellerman	now
5	atf	cue	blackman	ccrtkba
6	ncoast	bimac	criminologist	saf
7	assault	gideon	canada	sumpin
8	romp	hasan	nejm	nit
9	oleari	polari	regimen	racket
10	cbnewsh	async	stricter	employers

fuzzy semi-Kmeans with semi-supervised classification methods and semi-supervised clustering methods, the proposed method can generally outperform the other methods.

5. Conclusion

This work focuses on semi-supervised clustering and proposes a novel algorithm called fuzzy semi-Kmeans to perform document clustering with a small amount of labeled documents. This algorithm extends *K*-means clustering model and uses the seeds to bias clustering toward a good region of the search space. Moreover, fuzzy semi-Kmeans provides the flexibility to employ different fuzzy membership function to measure the distance between data. This work employs Gaussian weighting function to conduct experiments, but cosine similarity function can be used as well. This work conducts experiments on three data sets and compares fuzzy semi-Kmeans with several methods. The experimental results indicate that fuzzy semi-Kmeans can generally outperform the other methods. Even though the boundaries among clusters are not clear, fuzzy semi-Kmeans can function properly. In many real applications, background knowledge is ready, so it is appropriate to employ background knowledge to make the learning more fast and effective.

Acknowledgment

This work was supported in part by the National Science Council under the Grants NSC-101-2221-E-009-163.

References

[1] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Kluwer Academic Publishers, Norwell, MA, USA, 1981.
[2] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. Ser. B 39 (1) (1977) 1–38.
[3] S. Basu, A. Banerjee, R.J. Mooney, Semi-supervised clustering by seeding, in: Proceedings of the 19th International Conference on Machine Learning, ICML'02, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002, pp. 27–34.
[4] T. Hofmann, Probabilistic latent semantic analysis, in: Proceedings of Uncertainty in Artificial Intelligence, UAI '99, 1999.
[5] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, Mach. Learn. 42 (1–2) (2001) 177–196.
[6] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the 11th Annual Conference on Computational Learning Theory, COLT '98, ACM, New York, NY, USA, 1998, pp. 92–100.
[7] W. Wang, Z.-H. Zhou, A new analysis of co-training, in: J. Fürnkranz, T. Joachims (Eds.), Proceedings of the 27th International Conference on Machine Learning (ICML-10), Omnipress, Haifa, Israel, 2010, 1135–1142.
[8] K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Mach. Learn. 39 (2000) 103–134.
[9] T. Joachims, Making large-scale support vector machine learning practical, Advances in Kernel Methods, MIT Press, Cambridge, MA, USA, 1999, pp. 169–184.
[10] W. Pedrycz, J. Waletzky, Fuzzy clustering with partial supervision, IEEE Trans. Syst. Man Cybern., Part B Cybern. 27 (5) (1997) 787–795.
[11] A.M. Bensaid, L.O. Hall, J.C. Bezdek, L.P. Clarke, Partially supervised clustering for image segmentation, Pattern Recognition 29 (1996) 859–871.
[12] A. Bouchachia, W. Pedrycz, A semi-supervised clustering algorithm for data exploration, in: Proceedings of the 10th International Fuzzy Systems Association World Congress Conference on Fuzzy Sets and Systems, IFSA '03, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 328–337.

[13] Y. Hamasuna, Y. Endo, S. Miyamoto, Semi-supervised fuzzy C-means clustering using clusterwise tolerance based pairwise constraints, in: Proceedings of the 2010 IEEE International Conference on Granular Computing, GRC '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 188–193.
[14] A. Blum, S. Chawla, Learning from labeled and unlabeled data using graph mincuts, in: Proceedings of the 18th International Conference on Machine Learning, ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 19–26.
[15] A.B. Goldberg, X. Zhu, Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization, in: Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, TextGraphs-1, Association for Computational Linguistics, Stroudsburg, PA, USA, 2006, pp. 45–52.
[16] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, Constrained K-means clustering with background knowledge, in: Proceedings of the 18th International Conference on Machine Learning, ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 577–584.
[17] S. Basu, M. Bilenko, R.J. Mooney, A probabilistic framework for semi-supervised clustering, in: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04, ACM, New York, NY, USA, 2004, pp. 59–68.
[18] S. Miyamoto, M. Yamazaki, W. Hashimoto, Fuzzy semi-supervised clustering with target clusters using different additional terms, in: IEEE International Conference on Granular Computing, 2009, GRC '09, 2009, pp. 444–449.
[19] A. Bouchachia, W. Pedrycz, Enhancement of fuzzy clustering by mechanisms of partial supervision, Fuzzy Sets Syst. 157 (13) (2006) 1733–1759.
[20] P. Huang, D. Zhang, Locality sensitive C-means clustering algorithms, Neurocomputing 73 (16–18) (2010) 2935–2943.
[21] S. Zhong, Semi-supervised model-based document clustering: a comparative study, Mach. Learn. 65 (2006) 3–29.
[22] T. Finley, T. Joachims, Supervised clustering with support vector machines, in: Proceedings of the 22nd International Conference on Machine Learning, ICML '05, ACM, New York, NY, USA, 2005, pp. 217–224.
[23] J. Wang, S. Wu, H.Q. Vu, G. Li, Text document clustering with metric learning, in: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, ACM, New York, NY, USA, 2010, pp. 783–784.
[24] X. Ji, W. Xu, Document clustering with prior knowledge, in: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '06, ACM, New York, NY, USA, 2006, pp. 405–412.
[25] X. Wang, I. Davidson, Flexible constrained spectral clustering, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10, ACM, New York, NY, USA, 2010, pp. 563–572.
[26] J.B. Tenenbaum, V. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[27] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[28] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (2003) 1373–1396.
[29] T. Hofmann, J. Puzicha, M.I. Jordan, Learning from dyadic data, in: Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II, MIT Press, Cambridge, MA, USA, 1999, pp. 466–472.
[30] S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, R.A. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf. Sci. 41 (1990) 391–407.
[31] C.-L. Liu, W.-H. Hsaio, C.-H. Lee, G.-C. Lu, E. Jou, Movie rating and review summarization in mobile environment, IEEE Trans. Syst. Man Cybern., Part C 42 (3) (2012) 397–407.
[32] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York Inc., Secaucus, NJ, USA, 2006.
[33] X. Shi, B.L. Tseng, L.A. Adamic, Information diffusion in computer science citation networks, in: E. Adar, M. Hurst, T. Finin, N.S. Glance, N. Nicolov, B.L. Tseng (Eds.), International Conference on Weblogs and Social Media, The AAAI Press, 2009.
[34] C.D. Manning, P. Raghavan, H. Schtze, Introduction to Information Retrieval, Cambridge University Press, New York, NY, USA, 2008.
[35] M. Sokolova, L. Guy, A systematic analysis of performance measures for classification tasks, Inf. Process. Manage. 45 (2009) 427–437.
[36] A.B. Goldberg, X. Zhu, Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization, in: TextGraphs '06: Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing on the First Workshop on Graph Based Methods for Natural Language Processing, Association for Computational Linguistics, Morristown, NJ, USA, 2006, pp. 45–52.
[37] C.-C. Chang, C.-J. Lin, LIBSVM: A Library for Support Vector Machines, Software Available at ⟨http://www.csie.ntu.edu.tw/cjlin/libsvm⟩, 2001.
[38] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.