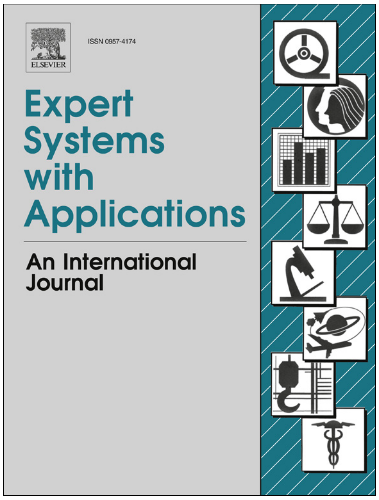


Proximity-based k-partitions clustering with ranking for document categorization and analysis

Jian-Ping Mei, Lihui Chen

PII: S0957-4174(14)00357-1
DOI: http://dx.doi.org/10.1016/j.eswa.2014.06.016
Reference: ESWA 9388

To appear in: *Expert Systems with Applications*

Received Date: 15 April 2014
Revised Date: 3 June 2014
Accepted Date: 7 June 2014

Please cite this article as: Mei, J-P., Chen, L., Proximity-based k-partitions clustering with ranking for document categorization and analysis, *Expert Systems with Applications* (2014), doi: http://dx.doi.org/10.1016/j.eswa.2014.06.016

Proximity-Based k-Partitions Clustering with Ranking for Document Categorization and Analysis

Jian-Ping Mei[a,*], Lihui Chen[b]

[a]College of Computer Science and Technology, Zhejiang University of Technology,
288 Liuhe Road, Hangzhou 310023, China
[b]Division of Information Engineering, Nanyang Technological University,
50 Nanyang Avenue, Singapore 639798, Republic of Singapore

Abstract

As one of the most fundamental yet important methods of data clustering, center-based partitioning approach clusters the dataset into k subsets, each of which is represented by a centroid or medoid. In this paper, we propose a new medoid-based k-partitions approach called Clustering Around Weighted Prototypes (CAWP), which works with a similarity matrix. In CAWP, each cluster is characterized by multiple objects with different representative weights. With this new cluster representation scheme, CAWP aims to simultaneously produce clusters of improved quality and a set of ranked representative objects for each cluster. An efficient algorithm is derived to alternatingly update the clusters and the representative weights of objects with respect to each cluster. An annealing-like optimization procedure is incorporated to alleviate the local optimum problem for better clustering results and at the same time to make the algorithm less sensitive to parameter setting. Experimental results on benchmark document datasets show that, CAWP achieves favourable effectiveness and efficiency in clustering, and also provides useful information for cluster-specified analysis.

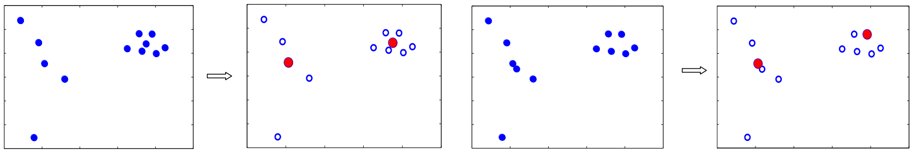
Keywords: Clustering, similarity-based, k-medoids, partitioning, document categorization

[*]Corresponding author. Tel.:+86 571 8529 0527
Email address: meijianping10@gmail.com (Jian-Ping Mei)

1. Introduction

Due to the tremendous growth of on-line documents, clustering becomes an important tool for automatic document categorization and analysis. Document clustering is a process that automatically groups a set of documents into subgroups, called clusters, such that documents in the same cluster are more similar to each other in content than those in different clusters. The technique of document clustering has been used in various applications across different areas, such as text mining, information retrieval, and Web mining. It can be used to detect topics in a document collection, or as a pre-step to facilitate further mining procedures, e.g. cluster-based information retrieval by focusing on a subset rather than the whole data collection (Salton, 1971; Cutting et al., 1992). Another application of document clustering is for effective representation, i.e., present the document collection in an organized structure. For example, clustering the results returned by a search engine in response to a query allows the user to browse the content of the relevant topic more efficiently (Zamir & Etzioni, 1998).

During the past years, various clustering approaches have been proposed and applied for document categorization. A wide range of document clustering approaches, including k-means and its extensions (Zhong, 2005), model-based clustering (Zhong & Ghosh, 2003; Blei et al., 2003), random projection based approach (Havaliwala et al., 2000), and approaches using matrix factorization (Xu et al., 2003), are based on the vector space model (VSM) of document representation, where each document is treated as "a bag of words". Although by only capturing statistical information of the original documents makes the simplicity of VSM, it is very difficult to perform clustering directly in the space it forms which is very sparse and high dimensional. Dimension reduction techniques may be used to reduce the dimensionality so that the transformed data can be handled by existing clustering approaches more easily (Jun et al., 2014). However, in many cases, the vector representation of documents may not be able to preserve enough information of the original documents that is useful for clustering purpose. Recently, researchers have been working on other models



(a) Data1 (b) Data2

Figure 1: Medoids selected with traditional k-medoids approach ($k = 2$) for the two datasets with different distributions. Each medoids is labeled as a red dot.

for better representations of documents (Chim & Deng, 2007; Shehata et al., 2007; Wan, 2007; Merlo et al., 2003; Hai-Tao Zheng, 2009; D'hondt et al., 2010; Lin et al., 2013). Experimental results reported in these studies show that document similarities obtained based on more sophisticated representation models lead to improvement in the clustering results. Since many complicated document representation models generate pairwise document similarities directly without producing explicit features and weights, vector-based clustering, i.e., clustering approaches using vector representation as the input form are thus not readily applicable in these cases. Proximity-based approaches, such as k-medoids clustering (Kaufman & Rousseeuw, 1990), hierarchical clustering (Zhao & karypis, 2005), spectral clustering (Shi & Malik, 2000), kernel-kmeans (Schölkopf et al., 1998) and fuzzy relational clustering (Davé & Sen, 2002; Skabar & Abdalgader, 2013), produce clusters only based on pairwise document proximity, i.e. similarity or dissimilarity. Since the (dis)similarity can be calculated by various proximity measures under different document representation models, proximity-based clustering is more generic than vector-based clustering in this sense. In this paper, we focus on proximity-based approaches, i.e. approaches that cluster objects only based on pairwise (dis)similarities.

Center-based partitioning clustering, such as k-means and k-medoids, is one type of the most popular clustering methods due to its good balance between simplicity and effectiveness. As mentioned earlier, unlike k-means, the k-medoids approach is a proximity-based approach and is more robust than k-

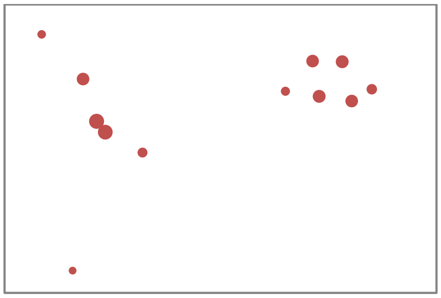


Figure 2: Weighted medoids generated by CAWP. Each cluster is represented by multiple objects with different weights. The size of the dot is proportional to the value of the weight of the corresponding object.

means (Kaufman & Rousseeuw, 1990). In k-means, each cluster center called centroid is calculated as the mean of all objects in that cluster; while in k-medoids, each cluster center is constrained to a real sample object, called medoid. Such cluster representation enables k-medoids to produce clusters of objects only based on the similarity matrix. However, representing a cluster by one object may not be sufficient enough. Figure 1 illustrates the distributions of two simulated datasets. The medoids identified by a k-medoids clustering approach are labeled as red dots. It is shown that for Data1, each medoid locates at the central place of the cluster that it represents. It seems to be reasonable to select these two objects as representatives of the two clusters, respectively. However, for Data2, there are two objects in the center of the left cluster which are close to each other and both of them may be selected as the representatives of this cluster, while for the other cluster, no single object is more significantly representative than others. Therefore, the one-medoid representation seems to be insufficient for Data2. In this paper, we propose a new similarity-based k-partitions clustering approach called Clustering Around Weighted Prototypes (CAWP) for document categorization and analysis. In this approach, each cluster is represented by more than one objects with various weights. The larger of the weight an object is assigned, the more representative this object is in the corresponding cluster. The weighted medoids for Data2 produced by CAWP is illustrated in Fig. 2, where the size of the dot is proportional to the value of the weight. It can be observed that multiple objects together represent a cluster but with different weights. Generally, the more central an object is located in a cluster, the larger weight is assigned to that object. This representation enhances the ability to capture more information of the cluster structures. We formulate the clustering problem into a constrained maximization problem. An efficient algorithm is derived to obtain the document clusters and the weights of documents in each cluster. We have reported some preliminary results of this approach in (Mei & Chen, 2011). As many objective-based clustering approaches, CAWP only converges locally. To improve the optimization result, we further developed an enhanced algorithm using an "annealing-like" procedure in the optimization process, which is inspired by the deterministic annealing technique proposed by Rose et al. (1990). This mechanism enables our algorithm to have an increased probability of escaping from local maximums during optimization and at the same time less sensitivity to parameter setting. We conducted extensive simulations on benchmark document datasets extracted from several corpora. Experimental results show that compared with existing approaches, the proposed approach achieves significant improvement in effectiveness and efficiency of document clustering. This demonstrates the helpfulness of the new cluster representation scheme for improving the clustering result as well as for providing more description to understand each document cluster generated.

In the following section, we review some related work on proximity-based clustering and formally present the details of the proposed approach in Section 3. In Section 4, experimental results are reported and discussed. Finally, conclusions are drawn in Section 5.

2. Related work

(Dis)similarity-based clustering has been studied for a long time. Hierarchical clustering generates a dendrogram consisting of a series of nested clusters in an agglomerative or divisive way (Sneath & Sokal, 1973; Jain & Dubes, 1988).

In hierarchical clustering, single linkage, complete linkage and group average linkage are the three classic ways to measure the closeness of two clusters based on the pairwise similarities between objects in the two clusters. In these conventional linkage-based hierarchical clustering approaches, each cluster is represented by all the objects in that cluster. Experimental study on document clustering by Zhao & karypis (2005) shows that partitioning approaches are superior than hierarchical ones for lower computational cost and better quality of clusters.

Another typical proximity-based partitioning clustering approach is the k-medoids approach. It generates k partitions or clusters of the dataset, where each cluster is represented by one of the objects belonging to that cluster. A well known k-medoids algorithm is the PAM algorithm proposed by Kaufman & Rousseeuw (1990). Due to the variety of data structure in real world, the conventional "one medoid for one cluster" strategy used is k-medoids approaches may not be sufficient enough. To be able to capture the cluster structure better, researchers developed clustering approaches based on multiple representative objects. Guha et al. (2001) proposed an hierarchical clustering approach CURE, where a specified number of representative objects are selected to be well separated in order to capture rich cluster shapes. Those representative objects are equally weighted and are not necessary to be real objects. An enhanced version of CURE is proposed in (Bellec & Kechadi, 2007) to select different numbers of representatives for different clusters based on cluster density. A density-based multi-representative approach is proposed in (Halkidi & Vazirgiannis, 2008). In the fuzzy clustering approach called PFC (Mei & Chen, 2010), each cluster is characterized by a number of weighted medoids. Different from other multi-representatives based approaches (Guha et al., 2001; Halkidi & Vazirgiannis, 2008), where representatives of each cluster are selected to be a pre-specified number with equal weights, in PFC, the weights as well as the number of representative objects in each cluster are decided based on the nature of the given dataset. Experimental results have shown the good performance of PFC in proximity-based clustering. However, the soft assignment and additional computation required in order to guarantee the non-negativeness of memberships and weights make the PFC algorithm not efficient enough for applications to large scaled document datasets.

Recently developed spectral clustering and kernel clustering may also be considered as proximity-based approaches. Spectral clustering is based on eigendecomposition of a well-defined affinity matrix. It has been applied for graph partitioning problem and achieves good result especially for non-linearly separable clusters (Shi & Malik, 2000). However, the eigendecomposition of an affinity matrix can become prohibitive when the dataset is very large. Another method known for clustering of non-linearly separable clusters is the kernel method. In kernel clustering approaches, such as kernel k-means (Schölkopf et al., 1998), a particular kernel is used to perform implicit mapping from the input space to a high dimensional feature space where clusters tend to be identified more easily. The equivalence between graph partitioning approach and kernel k-means was studied in (Dhillon et al., 2007).

To sum up, clustering data in a form of pairwise (dis)similarities is important in many real-world applications. Efficiency and scalability are two important factors to applications with large scaled data. Medoid-based partitioning clustering is more efficient and scalable compared to some other types of clustering such as hierarchical clustering and spectral clustering. However, the simple one-medoid cluster representation limits the effectiveness of classic k-medoids approach and its variants. This study aims to develop a new medoid-based clustering algorithm with enhanced cluster representation in a way to obtain improved effectiveness while maintain high efficiency.

3. Proposed Approach

In this section, we present the details of the proposed proximity-based clustering approach CAWP. We first formulate the clustering task into a constrained maximization problem and then develop an efficient algorithm to solve the problem in an iterative way.

3.1. Problem Formulation

Given a dataset with n objects $\mathcal{X} = \{x_1, x_2, \ldots x_n\}$ and the similarity matrix $\mathbf{S}$, where each entry $s_{ij} \in \mathbf{S}$ records the similarity between x_i and x_j, we want to cluster these objects into k non-overlapping clusters, i.e. $\mathcal{A}_1 \bigcup \mathcal{A}_2 \bigcup \ldots \bigcup \mathcal{A}_k = \mathcal{X}$, and $\forall\ c \neq f$, $\mathcal{A}_c \bigcap \mathcal{A}_f = \emptyset$. To derive the objective function, we first define the following concepts and notations.

Assumption 1. *For $c = \{1, 2, \ldots, k\}$, each cluster $\mathcal{A}_c$ is represented by a number of objects $\mathcal{R}_c = \{r_h\}_{h=1,\ldots,s}$ called representative objects, and each representative object $r_h \in \mathcal{R}_c$ is associated with a weight v_{ch}. The larger the value of v_{ch}, the more representative r_h is in $\mathcal{A}_c$.*

Here $\forall c, h$, we let $0 < v_{ch} < 1$ and $\forall c$, $\sum_h v_{ch} = 1$. It is worth noting that the same object may represent multiple clusters at the same time but with different weights regarding to different clusters. This is because overlapping may exist between clusters. For example, in document clustering, assume that one document cluster corresponds to the topic of “data mining” while another cluster corresponds to the topic of “machine learning”, then a paper on document clustering is representative in both clusters but with a slight larger weight in “data mining” than in “machine learning”. Based on Assumption 1, next we give two definitions as follows:

Definition 1. *The object-to-cluster closeness $sim(x_i, \mathcal{A}_c)$, or the closeness of an object x_i to a cluster $\mathcal{A}_c$, is calculated as the sum of weighted similarities from x_i to each of the representative objects of that cluster: $sim(x_i, \mathcal{A}_c) = \sum_{r_h \in \mathcal{R}_c} v_{ch} sim(x_i, r_h)$*

Definition 2. *The compactness of a cluster c is denoted as Q_c and measured by the sum of object-to-cluster closenesses with those objects belonging to that cluster: $Q_c = \sum_{x_i \in \mathcal{A}_c} sim(x_i, \mathcal{A}_c)$.*

Like many popular clustering approaches, the objective of CAWP is to maximize the total quality of all clusters in terms of compactness defined above, i.e.,

$$J = \sum_{c=1}^{k} Q_c = \sum_{c=1}^{k} \sum_{x_i \in \mathcal{A}_c} \sum_{r_h \in \mathcal{R}_c} v_{ch} sim(x_i, r_h) \tag{1}$$

Instead of performing representative object selection and weighting separately, these two tasks are accomplished simultaneously through assigning weights valued in the range of $[0, 1]$ to each of the objects in the dataset. More specifically, for each $x_i \in \mathcal{X}$, we assign it a weight $0 \leq w_{ci} \leq 1$ with respect to each cluster $\mathcal{A}_c$. Thus, objects have non-zero weights in a cluster are the representative objects of that cluster, i.e. $\mathcal{R}_c = \{x_q\}_{x_q \in \mathcal{X}, w_{cq} > 0}$. It is clear that in $\mathcal{R}_c$, if $r_h = x_q$, then $v_{ch} = w_{cq}$ and all objects x_j with $w_{cj} = 0$ are not included in $\mathcal{R}_c$. Thus, the objective function (1) is rewritten into the following form

$$\begin{aligned} J &= \sum_{c=1}^{k} \sum_{x_i \in \mathcal{A}_c} \Big(\sum_{x_q \in \mathcal{X}, w_{cq} > 0} w_{cq} s_{iq} + \sum_{x_j \in \mathcal{X}, w_{cj} = 0} w_{cj} s_{ij} \Big) \\ &= \sum_{c=1}^{k} \sum_{x_i \in \mathcal{A}_c} \sum_{j=1} w_{cj} s_{ij} \end{aligned} \tag{2}$$

where s_{ij} is short for $sim(x_i, x_j)$.

However, solving the above objective function directly gives an undesired result, as it is maximized when for each cluster, we simply select the object with the largest object-to-cluster closeness alone to represent that cluster. To produce more than one representative objects with different weights in each cluster, we add a penalty term into (2) to give the regularized form as below:

$$J_{\mathrm{CAWP}} = \sum_{c=1}^{k} \sum_{x_i \in \mathcal{A}_c} \sum_{j=1}^{n} w_{cj} s_{ij} - \frac{T}{2} \sum_{c=1}^{k} \sum_{j=1}^{n} w_{cj}^2 \tag{3}$$

subject to

$$\sum_{j=1}^{n} w_{cj} = 1 \quad \forall c; \tag{4}$$

$$w_{cj} \geq 0 \quad \forall c, j \tag{5}$$

Now the objective function J_{CAWP} in (3) is comprised of two terms. The first term is the total within-cluster compactness that needs to be maximized and the second term is used as a regularization or penalty of the first term. It is easy to observe that with constraint in (4), $\forall c$, the value of $\sum_{j=1}^{n} w_{cj}^2$ is minimized or the second term of (3) is maximized when $w_{cj} = \frac{1}{n}$. This means the second term is maximized when all objects are equally weighted to represent a cluster. Since (3) takes both terms into consideration, the solution to maximize (3) is a tradeoff between one object totally and all objects equally for representing a cluster. The parameter $T > 0$ controls the contribution of the regularization term and needs to be set properly in order to get a reasonable result.

3.2. Solution and Algorithm

As weight w_{cj} is a continuous variable, we can use the method of Lagrange multiplier to derive the solution of w_{cj} by locally maximizing J_{CAWP} in (3) under constraint in (4). Introducing the Lagrange Multipliers β_c, the Lagrangian is constructed as

$$L_{\text{CAWP}} = J_{\text{CAWP}} + \sum_{c=1}^{k} \beta_c (\sum_{j=1}^{n} w_{cj} - 1) \tag{6}$$

By calculating $\frac{\partial L_{\text{CAWP}}}{\partial w_{cj}} = 0$ and $\frac{\partial L_{\text{CAWP}}}{\partial \beta_c} = 0$, the update rule of w_{cj} is derived as

$$w_{cj} = \frac{1}{n} + \frac{1}{T}\left[\sum_{x_i \in \mathcal{A}_c} s_{ij} - \frac{1}{n}\sum_{q=1}^{n}\sum_{x_i \in \mathcal{A}_c} s_{iq}\right]. \tag{7}$$

With the above update formula, negative values of w_{cj} may appear during the iteration. To guarantee that all w_{cj} values are non-negative, we need to use the Karush-Kuhn-Tucker (KKT) conditions as in (Mei & Chen, 2010) to add the non-negative constraints (5) into the Lagrangian and derive a more complex update formula of w_{cj}. However, updating using that formula could be time consuming for large datasets. To speed up the process, here we use the following heuristic as an approximate solution. At each iteration, for each c, we check $\mathbf{w}_c = (w_{c1}, w_{c2}, \ldots, w_{cn})^T$, and set negative elements of $\mathbf{w}_c$ to 0. After that normalize $\mathbf{w}_c$ to unit l_1 norm so that the condition in (4) is still valid. Our experimental results show that this trick makes the algorithm run faster without sacrificing much the clustering accuracy and convergence speed.

After the update rule of weights has been derived, we then need to decide the way of cluster assignment. A typical way is to assign each of the objects to the cluster which is the closest. According to Definition 1, the object-to-cluster closeness can be expressed as

$$Sim(x_i, \mathcal{A}_c) = \sum_{r_h \in \mathcal{R}_c} v_{ch} sim(x_i, r_h) = \sum_{j=1}^{n} w_{cj} s_{ij} \tag{8}$$

Thus, x_i is assigned to cluster $\mathcal{A}_c$ where

$$c = \arg \max_{f=\{1,2,\ldots,k\}} \sum_{j=1}^{n} w_{fj} s_{ij} \tag{9}$$

Algorithm 1 AlternatingOptimization

```
Input: Similarity matrix $\mathbf{S}_{n \times n}$, the number of clusters $k$, parameter $T > 0$ .
Output: Cluster labels of objects, weight matrix $\mathbf{W}_{k \times n}$.
Method:
Generate an initial partition $\{\mathcal{A}_c^0\}_{c=1}^k$, and set iteration counter $l \leftarrow 0$;
repeat
    $\forall c, j$ update the weight $w_{cj}^{(l)}$ by (7) with the current partition; for each $c$
    check if for some $h$, $w_{ch}^{(l)} < 0$, set $w_{ch}^{(l)} = 0$ and $w_c^{(l)} = w_c^{(l)} / |w_c^{(l)}|_1$;
    $\forall c, j$ update the cluster assignment by (9) with $w_{cj}^{(l)}$;
    $l \leftarrow l + 1$.
until convergence
```

The alternating optimization is widely used to get local solutions of objective-based clustering. Following this procedure, we successively update the cluster assignment and representative weights based on each other. The representative objects are adjusted through the updating of weights based on the current partitions, and the updated representative objects of each cluster in turn are used to form new clusters. As been discussed earlier, when $T \to 0$, CAWP decreases to one object representation for each cluster; while $T \to \infty$ makes all objects equally represent all each cluster. Therefore, a proper value of T is important in order to generate reasonable results.

In this paper, we present an enhanced version of the basic alternating optimization with “annealing”. The deterministic annealing technique (Rose et al., 1990) has been proposed to alleviate the local optimum problem of clustering with non-convex objective functions. The annealing process starts with a high-temperature which generates highly fuzzy clusters, i.e., objects have very close memberships in all the clusters, and then the temperature is gradually decreased to produce less fuzzy clusters. The way of incorporating “randomness” into the objective function produce more opportunities for escaping from local solutions. Inspired by the deterministic annealing technique, we incorporate an “annealing like” process into the CAWP algorithm. With this procedure, CAWP is expected to avoid more local maximums and hence tends to achieve better clustering results. The detailed algorithm of CAWP is given as Algorithm 2, where the basic alternating optimization is given as Algorithm 1.

Algorithm 2 Clustering Around Weighted Prototypes (CAWP)

Input: Similarity matrix $\mathbf{S}_{n\times n}$, the number of clusters k, parameter T_0, T_f, maximum iteration number M.

Output: Cluster labels of objects, weight matrix $\mathbf{W}_{k\times n}$.

Method:

```
1: $T = T_0$, set $t \leftarrow 0$, generate an initial partition (0);
2: repeat
3:     $\{\mathcal{A}_c^{t+1}\}_{c=1}^{k} \leftarrow$ AlternatingOptimization ($T$, $\{\mathcal{A}_c^{t}\}_{c=1}^{k}$);
4:     $T = T_0 \times (T_f/T_0)^{t/M}$;
5:     $t \leftarrow t + 1$.
6: until $t > M$ or $T < T_f$
```

As shown in Algorithm 2, instead of fixing the parameter T to a specified value, the CAWP algorithm starts with a large T_0 and gradually decreases it when the process continues. This is analogy to the annealing process, where the parameter T can be treated as the temperature. When the temperature is high, i.e. T is large, the distribution of $\mathbf{w}_c$ is close to random and the representative objects are searched in a large space; when T becomes small, the distribution of $\mathbf{w}_c$ becomes stable and the new representative objects are searched only from the neighborhood of the current ones. Such a cooling process enables CAWP to avoid some of the local optimums and is more likely to converge to a better result.

It requires $O(n^2)$ for updating weights and cluster assignment, where n is the size of the dataset. The time complexity of CAWP for M temperature levels is $O(Mn^2)$. As $M \ll n$, and in many actual cases, less iterations are needed to converge when the temperature or the value of T decreases, the time complexity of CAWP is $O(n^2)$. It reduces to $O(E)$ for a sparse similarity matrix with E non-zero entries. Compared with existing (dis)similarity-based approaches, CAWP has the same complexity with kernel-kmeans, which is lower than that of Spectral clustering $O(n^3)$ (Shi & Malik, 2000) and group average based hierarchical clustering $O(n^2 logn)$ (Jain & Dubes, 1988).

3.3. Analysis

The weighted-prototype cluster representation, i.e., each cluster is represented by a set of weighted objects and each object is allowed to represent multiple clusters to certain degrees, plays a critical role in CAWP. In some approaches, such as graph-based partitioning clustering, all objects in the cluster together are used to represent that cluster. The only concern in graph-based clustering is to find the best cut of the graph to form several subgraphs without differentiating the importance or representativeness among nodes in each subgraph.

Allowing each cluster to be described by more than one objects which are weighted based on their importance in the cluster enhances the ability of CAWP to capture various data structures and hence produce better clustering result. Meanwhile, it provides more detailed information on the structure of each individual cluster. During the optimization process of CAWP, the representatives of each cluster are searched from the whole dataset rather than within the current members. It helps our approach to escape from some less optimal states where several other approaches are trapped, and makes CAWP more robust to outliers. Let us take a look at the updating of weights (7) in iteration l in the following form

$$w_{cj}^{l} = G_1 + \frac{1}{T}\left[G_2(x_j, \mathcal{A}_c^l) - G_3(\mathcal{A}_c^l)\right] \tag{10}$$

where $G_1 = \frac{1}{n}$ is a constant, $G_2 = \sum_{x_i \in \mathcal{A}_c} s_{ij}$ is the sum of similarities from x_j to objects in $\mathcal{A}_c$, and $G_3 = \frac{1}{n}\sum_{q=1}^{n}\sum_{x_i \in \mathcal{A}_c} s_{iq}$ is the average of sum of similarities from all the objects in the dataset to objects in $\mathcal{A}_c$. Given the current partitions, the weights of objects with respect to a specific cluster $\mathcal{A}_c$ are decided by G_2. The term G_2 reflects the overall similarity of each object x_j to cluster $\mathcal{A}_c$, no matter the object is currently assigned to $\mathcal{A}_c$ or not.

According to (10), we may have some interesting properties of w_{cj}^{l} as follows:

- Object x_j may be in the representative object sets of multiple clusters, i.e. $w_{cj}^{l} > 0$ for $c \in \mathcal{G}$ with $|\mathcal{G}| > 1$.
- Object x_j may have a very low level or no representativeness in all the clusters, i.e. $\forall c$, $w_{cj}^{l} < \delta$, where δ is a small positive value.
- For an object x_j with cluster label c at iteration l, i.e., $x_j \in \mathcal{A}_c^l$, the representative weight of x_j with respect to $\mathcal{A}_c^l$ may be smaller than that of some other clusters, i.e. $w_{cj}^{l} < w_{fj}^{l}$, $f \neq c$.

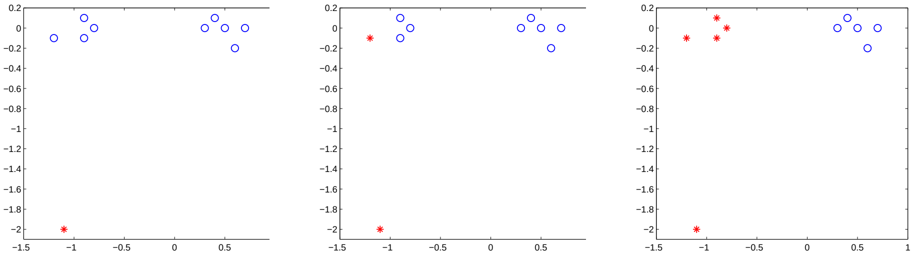
With these properties, CAWP becomes more flexible in capturing various cluster structures and to reduce as much as possible the negative impact of outliers during the clustering process.

3.4. A running example

Next, we give a running example of CAWP to illustrate the clustering process of CAWP. A simple 2-D data is used for the convenience of discussion and visualization. This dataset contains ten objects. Nine of them form two clusters, i.e., object 1 to 4 in one cluster $\mathcal{A}_1$ and object 6 to 10 in the other cluster $\mathcal{A}_2$. Object 5 can be treated as an outlier in the dataset. The coordinates of the ten objects and the running result of CAWP at each iteration are given in Table 1. We also plot the clusters at each iteration in Fig. 3.

The initial partitions as shown in Fig. 3a place the outlier in $\mathcal{A}_1$ and all other objects into the other cluster $\mathcal{A}_2$. Conventional centroid-based k-partitioning clustering such as kernel k-means and k-means gets stuck at this bad initialization. Since in these approaches the center of a cluster is updated based on the objects in the current cluster, the center of $\mathcal{A}_1$ is always the outlier x_5 and the center of $\mathcal{A}_2$ keeps locating at the central place of all the other objects. Thus, the two partitions are not to be updated. However, the proposed CAWP is able to overcome the difficulty in the given situation gradually at each iteration and finally converge to a reasonable result after three iterations as shown in Fig. 3b and Fig. 3c.

The success of CAWP to produce expected clusters by escaping from the bad initialization is attributed to the new cluster representation scheme. Since representative objects of a cluster are updated without being limited to the current members of that cluster, objects which are currently in $\mathcal{A}_2$ are able to represent $\mathcal{A}_1$ to a certain degree in the next iteration. It is observed from Table 1 that seven objects, i.e. $\{x_1, x_2, x_3, x_4, x_6, x_9, x_{10}\}$ which are initially in $\mathcal{A}_2$ are assigned different weights to represent $\mathcal{A}_1$ after being updated once. The updating of $\mathcal{R}_1$ implicitly drags the cluster center of $\mathcal{A}_1$ towards other objects, especially $\{x_1, x_2, x_3, x_4\}$, away from the outlier x_5. The new cluster representatives result in repartition of the dataset with x_1 being included in $\mathcal{A}_1$ together with x_5. Based on the updated partitions, the representative objects of two clusters are further adjusted and converges after two iterations. Since every object will be assigned a label, the outlier is finally grouped in $\mathcal{A}_1$. However, it is useful to observe that the weight of the outlier is 0 in both clusters. This means that the outlier does not engage in representing any of the two clusters even the cluster it is assigned to. This property is important from two aspects. First, it indicates that the presence of outliers may not affect much the labeling of other objects since those outliers have a low level of representativeness in all clusters; second, after convergence, objects with very small weights in all



(a) (b) (c)

Figure 3: Clustering results of CAWP with a bad initialization: (a) initial clusters, (b) clusters after one iteration, (c) clusters after two iterations (converge). Different colors and symbols label different clusters.

clusters may be identified as outliers or noise.

4. Experimental Results

In this section, we simulate the task of document categorization to evaluate the effectiveness and efficiency of the proposed algorithms. To make comparison, we run the proposed approach CAWP, together with several other popular (dis)similarity-based approaches, including spectral clustering, kernel k-means, k-medoids, fuzzy clustering of weighted medoids and hierarchical clustering. We also show how the document weights can be used for cluster-based analysis. Since the experiential study concentrates on evaluating the performance of the presented proximity-based clustering algorithm, we use the simple vector space model to represent the document set as $x_{ij} \in X_{n\times m}$, where n, m are the number of documents and distinct words, respectively. The similarity matrix is then generated as the input data.

4.1. Data sets and Preprocessing

We use the datasets extracted from several benchmark document collections that are frequently used in the literature for document clustering. A summary of ten datasets with various scales and topic balance is given in Table 2. A highlight of each is given as below:

Table 1: Label and Weight updating

Object	coordinate	initial	iteration 1			iteration 2			iteration 3		
ID		label	w_1	w_2	label	w_1	w_2	label	w_1	w_2	label
1	(-1.2, -0.1)	2	0.1299	0	1	0.2367	0	1	0.2362	0	1
2	(-0.9, 0.1)	2	0.0896	0.0874	2	0.1628	0.0397	1	0.2469	0	1
3	(-0.8, 0.0)	2	0.1066	0.1188	2	0.1674	0.0733	1	0.253	0	1
4	(-0.9, -0.1)	2	0.1284	0.0917	2	0.1966	0.0403	1	0.2639	0	1
5	(-1.1, -2.0)	1	0.5006	0	1	0.2367	0	1	**0**	**0**	1
6	(0.3, 0.0)	2	0.025	0.1841	2	0	0.2003	2	0	0.1964	2
7	(0.4, 0.1)	2	0	0.1676	2	0	0.1905	2	0	0.2032	2
8	(0.7, 0.0)	2	0	0.0871	2	0	0.1287	2	0	0.1946	2
9	(0.6, -0.2)	2	0.0182	0.1067	2	0	0.142	2	0	0.1932	2
10	(0.5, 0.0)	2	0.0016	0.1567	2	0	0.1851	2	0	0.2126	2

Multi5 and **Multi10** are two subsets extracted from the **20Newsgroups** (Lang, 1995) collection. The total collection consists of approximately 20,000 newsgroup articles collected from 20 different newsgroups. **Multi5** contains around 100 documents from each of the five categories: *comp.graphics, rec.motorcycles, rec.sports.baseball, sci.space*, and *talk.politi-cs.mideast*. **Multi10** contains around 50 documents from each of the following 10 topics: *alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space*, and *talk.politics.guns*.

tr12/31 [1] datasets are derived from the TREC collections (http://trec.nist.gov). **la1** and **la2** are derived from Los Angeles Times (TREC). **re0** and **re1** are derived from Reuters-21578 Text Categorization Test Collection Distribution 1.0.[2] The **Reuters**-21578 collection contains the documents appeared on the Reuters newswire in 1987. It is a standard text categorization benchmark that contains 135 categories in total. **k1a** and **wap** are derived from the **WebACE** project

[1] Available at http://www.cse.fau.edu/zhong/pubs.htm

[2] http://www.daviddlewis.com/resources/testcollections

Table 2: Summary of document datasets (n: # of documents, m: # of words, k: # of classes, $\overline{n}_c$: average number of documents per class, and $Balance$: the size ratio of the smallest class to the largest one)

Data	Source	n	m	k	$\overline{n}_c$	$Balance$
Multi5	20Newsgroups	494	1000	5	99	0.95
Multi10	20Newsgroups	497	1000	10	50	0.98
tr12	TREC	313	5804	8	39	0.0968
tr31	TREC	927	10128	7	132	0.0688
la1	LA Times (TREC)	3204	13195	6	534	0.290
la2	LA Times (TREC)	3075	12432	6	513	0.274
re0	Reuters-21578	1504	2886	13	115	0.0181
re1	Reuters-21578	1657	3758	25	66	0.0270
wap	WebACE	1560	8460	20	78	0.0254
k1a	WebACE	2340	21839	20	117	0.0182

(Boley et al., 1999). Each document corresponds to a web page listed in the subject hierarchy of Yahoo! (http://www.yahoo.com).

For datasets from 20Newgroups, the rainbow toolkit developed by McCallum (1996), which is available at: http://www.cs.cmu.edu/ mccallum/bow/, is used for indexing. Top 1000 words with the largest information gain are selected. We get the processed version of all other datasets from the CLUTO toolkit [3]. We further remove words that occur in less than three documents. After that, each word is weighted with the *tf-idf* weighting (Salton & Buckley, 1988), and *cosine* measure is used to calculate the similarity between each pair of documents.

4.2. Algorithms and Evaluations

For comparison, the following algorithms are run:

- PAM (Kaufman & Rousseeuw, 1990): the k-medoids approach.

[3] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download

- KernelKmeans (Schölkopf et al., 1998): the kernel k-means.
- NCut (Shi & Malik, 2000): the spectral clustering with normalized cut, where k-means is used to produce the final cluster labels.
- UPGMA (Zhao & karypis, 2005): the Unweighted Pair Group Method with Arithmetic Mean or hierarchical agglomerative clustering with group average linkage.
- PFC (Mei & Chen, 2010): the fuzzy clustering with weighted prototype
- CAWP: the proposed approach.

We use *F-measure* (Larsen & Aone, 1999) and *Normalized Mutual Information* (NMI) (Strehl & Ghosh, 2002) to evaluate the quality of the clusters produced by different approaches. Both of *F-measure* and *NMI* compare the clusters produced by a clustering algorithm with the ground-truth partitions and taking values in range of [0, 1]. The larger the *F-measure* and *NMI* , the better the clustering result is. If *cluster* refers to the clustering result created automatically with algorithms and *class* refers to the ground truth, then

$$F\text{-}measure(j,c)_\alpha = \frac{(1+\alpha)P(j,c)R(j,c)}{\alpha P(j,c)+R(j,c)} \tag{11}$$

$$F\text{-}measure = \sum_{j=1}^{f} \frac{n_j}{n} \text{argmax}_c\{F\text{-}measure(j,c)\} \tag{12}$$

where

$$P(j,c) = n_c^j/n_c, \quad R(j,c) = n_c^j/n_j \tag{13}$$

are *precision* and *recall*, respectively. In the above equations, n is the total number of documents, n_c, n_j are the numbers of documents in the cth *cluster* and jth *class*, respectively, and n_c^j is the number of common documents in class j and cluster c. The metric *F-measure* reflects the overall quality of the clusters produced with a weighted combination of *precision* and *recall* as in (11). Typically, equal weights are assigned to *precision* and *recall*, $i.e., \alpha = 1$.

Another metric *NMI* is rooted from the information theory which is defined as

$$NMI = \frac{\sum_{c=1}^{k} \sum_{j=1}^{f} n_c^j \log(\frac{n \cdot n_c^j}{n_c \cdot n_j})}{\sqrt{(\sum_{c=1}^{k} n_c \log \frac{n_c}{n})(\sum_{j=1}^{f} n_j \log \frac{n_j}{n})}}. \tag{14}$$

For each dataset, the result of each approach except UPGMA is the average of 30 trials with random initializations. To make a direct comparison, the same set of initializations are used for PAM, KernalKmeans, PFC and CAWP. For all datasets, we set $T_u = 0.0002, T_v = 1$ for PFC, and $T_0 = 1000, T_f = 1, M = 4$ for CAWP.

4.3. Significance of Results

We also perform statistical test to see whether the results produced by the proposed approach are significantly better than those of others. We use the *two-sample t-test* to determine if there is a significant difference between the mean values of each measure (F-measure or NMI) obtained by two approaches. The *null hypothesis* H_0 is “no significance of differences”. Within 95% confidence interval, the critical value of t, $t_{critical}$ is 2.002 at degree of freedom $df = 58$ ($df = h_1 + h_2 - 2$, and in our experiments $h_1 = h_2 = 30$). Thus, if the calculated $t < t_{critical}$, the *null hypothesis* H_0 is *Accepted*, otherwise it is *Rejected*, which means there is a significance of difference between two results. The *t-test* results are shown together with the F-measure and NMI scores in Table 3 to Table 7, where “$\gg$” indicates that the result of the algorithm on the left side is significantly better than the one on the right, and “>” indicates better but not significant.

4.4. Results and Discussions

4.4.1. CAWP vs. Existing Partitioning Approaches

We first compare the performance of CAWP with other three partitioning approaches namely kernel-based approach KernelKmeans, k-medoids clustering PAM and Spectral clustering NCut. In this group of experiments, the number of clusters is set to be the real number of classes. The means and standard deviations of *F-measure* and *NMI* scores of the four approaches on ten datasets from five different corpus are given in Table 3 to Table 7. The results of significance test are also included in these tables. It is seen that CAWP performs better or comparative with NCut, and significantly better than KernelKmeans and PAM on all the datasets. For three existing approaches, the spectral clustering NCut performs well on a majority of cases, but its results on two Reuters-21578 data, namely *re0* and *re1* are not as well as expected. For the two existing k-partitions approaches, KernelKmeans is overall slightly better than PAM. Overall, the new k-partitions approach CAWP makes significant improvement compared with two existing k-partitions approaches, and it also performs better or comparable with Spectral clustering NCut, which requires to perform eigendecompositions.

4.4.2. CAWP vs. PFC

Now let us compare the performances of CAWP with PFC, the fuzzy clustering of weighted medoids. Fig. 4 plots the F-measure and NMI scores of the two approaches, and Fig. 5 shows the average running time needed by each approach for generating those results. It is observed from Fig. 4 that the two approaches overall perform comparably. CAWP achieves better or comparable results with those of PFC on all the datasets except Multi10, on which the fuzzy approach PFC produces a better result. In CAWP, although each object is assigned to only one cluster, or the partitions are hard, objects are allowed to have some degree of representativeness in multiple clusters. In other words, some clusters may share representative objects to some degree. This property allows CAWP to be able to capture common features among clusters and produce good clusters even the partitions are hard. The good performance of CAWP is also attributed to the annealing process. Although, the performance of PFC is comparable to those of CAWP, it requires much more computational cost to update and store the fuzzy membership matrix. It is clearly seen from Fig. 5 that CAWP is much faster than PFC. Taking both the effectiveness and efficiency into consideration, CAWP is shown to be a more favorable choice than PFC for document categorization. In Mei & Chen (2010), it was shown that PFC outperform

4.4.3. CAWP vs. Hierarchical clustering

Finally we compare the performance of CAWP with hierarchical clustering with respective to different value of k. For the hieratical clustering, the group-average linkage is used. In Fig. 6, we plots the NMI values only of UPGMA and the means of CAWP over 30 trials on ten datasets. Similar conclusions can be made with respect to the F-measure scores. It is seen that the performance of CAWP is more stable than UPGMA when k varies, and CAWP gives higher NMI values than UPGMA on six datasets across different k values. CAWP performs better than UPGMA on *tr31* with $k = 5$, but slighter worse than that of UPGMA when k increases. The performance of two approaches on the other three datasets are comparable and do not change much when k changes.

4.4.4. Cluster-Based Rank

In many real-world applications of document categorization, the size of each document cluster generated through clustering may still be too large to carry out subsequent processing or to get a quick idea about the content of each cluster. Therefore, along with a number of document groups, additional information is needed in order to make cluster-specified analysis. To provide an efficient overview of a large document cluster, the information on the relative importance of documents within each cluster is helpful. Based on such kind of information, documents in a cluster can be ranked and users may only need to take a look at those top ranked documents to get a rough understanding of the content of the whole cluster. Other than that, removing those lowly ranked documents reduces the storage requirement without losing of any important information.

Table 8 lists the top five key words for each of the clusters of Multi5. For each cluster, we first select the top five ranked documents which have the largest representative weights, and then obtain the top ranked key words associated the largest *tf-idf* values in the five representative documents. The key words captures the main content of the representative documents. By scanning these key words or the content of the representative documents, we have a general idea that is consistent with the topic or the content of the corresponding cluster

Table 3: Results on 20Newsgroups data

Data sets	F-measure			
	PAM	KernelKmeans	NCut	CAWP
Multi5	0.7793 ± 0.0000	0.7734 ± 0.0969	0.8805 ± 0.0809	0.8994 ± 0.0627
Multi10	0.5213 ± 0.0197	0.4726 ± 0.0420	0.5747 ± 0.0222	0.6628 ±0.0423
	Significant test based on F-measure values			
Multi5	CAWP $>$ NCut $\gg$ PAM $>$ KernelKmeans			
Multi10	CAWP $\gg$ NCut $\gg$ PAM $\gg$ KernelKmeans			

Data sets	NMI			
	PAM	KernelKmeans	NCut	CAWP
Multi5	0.5113 ± 0.0000	0.6322 ± 0.1001	0.7748 ± 0.0551	0.7942 ± 0.0496
Multi10	0.3898 ± 0.0163	0.3468 ± 0.0417	0.5166 ± 0.0192	0.6046 ± 0.0364
	Significant test based on NMI values			
Multi5	CAWP $>$ NCut $\gg$ KernelKmeans $\gg$ PAM			
Multi10	CAWP $\gg$ NCut $\gg$ PAM $\gg$ KernelKmeans			

in an efficient way.

5. Conclusions

In this study, we have proposed a new proximity-based clustering approach CAWP for document categorization. The clustering problem has been formulated into an optimization problem and an efficient and effective algorithm has been developed to maximize the new objective function. Unlike conventional k-medoids clustering, each cluster is represented by multiple weighted objects while each object is only assigned to one cluster in CAWP. This formulation enables CAWP to be able to capture the cluster structure more accurately, more robust to outlier, and remains efficient for large document datasets. Along with the clusters produced, the proposed approach provides more detailed information on each of the individual clusters, which is very useful for analyzing

Table 4: Results on TREC data

Data sets	F-measure			
	PAM	KernelKmeans	NCut	CAWP
tr12	0.5169 ± 0.0270	0.5726 ± 0.0668	0.7375 ± 0.0408	0.7398 ± 0.0469
tr31	0.5886 ± 0.0209	0.6059 ± 0.0587	0.6922 ± 0.0217	0.7124 ± 0.0595
	Significant test based on F-measure values			
tr12	CAWP $\gg$ NCut $\gg$ KernelKmeans$\gg$ PAM			
tr31	CAWP $>$ NCut $\gg$ KernelKmeans$\gg$ PAM			

Data sets	NMI			
	PAM	KernelKmeans	NCut	CAWP
tr12	0.4535 ± 0.0263	0.4893 ± 0.0702	0.6482 ± 0.0407	0.6725 ± 0.0439
tr31	0.3883 ± 0.0082	0.4998 ± 0.0489	0.5502 ± 0.0224	0.5844 ± 0.0655
	Significant test based on NMI values			
tr12	CAWP $\gg$ NCut $\gg$ KernelKmeans$\gg$ PAM			
tr31	CAWP $\gg$ NCut $\gg$ KernelKmeans$>$ PAM			

Table 5: Results on LA Times (TREC) data

Data sets	F-measure			
	PAM	KernelKmeans	NCut	CAWP
la1	0.6033 ± 0.0000	0.5498 ± 0.0572	0.6106 ± 0.0090	0.6724 ± 0.0474
la2	0.5246 ± 0.0000	0.5554 ± 0.0348	0.6673 ± 0.0013	0.7117 ± 0.0235
	Significant test based on F-measure values			
la1	CAWP ≫ NCut ≫ PAM≫ KernelKmeans			
la2	CAWP ≫ NCut ≫ KernelKmeans≫ PAM			

Data sets	NMI			
	PAM	KernelKmeans	NCut	CAWP
la1	0.3265 ± 0.0000	0.3948 ± 0.0505	0.4516 ± 0.0073	0.5216 ± 0.0346
la2	0.2728 ± 0.0000	0.4059 ± 0.0447	0.4989 ± 0.0003	0.5539 ± 0.0263
	Significant test based on NMI values			
la1	CAWP ≫ NCut ≫ KernelKmeans≫ PAM			
la2	CAWP ≫ NCut ≫ KernelKmeans≫ PAM			

Table 6: Results on Reuters-21578 data

Data sets	F-measure			
	PAM	KernelKmeans	NCut	CAWP
re0	0.4164 ± 0.0184	0.4304 ± 0.0336	0.3904 ± 0.0177	0.5220 ± 0.0293
re1	0.4396 ± 0.0134	0.4356 ± 0.0287	0.4734 ± 0.0162	0.4802 ± 0.0225
	Significant test based on F-measure values			
re0	CAWP ≫ KernelKmeans > PAM ≫ NCut			
re1	CAWP > NCut ≫ PAM >KernelKmeans			

Data sets	NMI			
	PAM	KernelKmeans	NCut	CAWP
re0	0.3429 ± 0.0008	0.3874 ± 0.0323	0.3700 ± 0.0100	0.4123 ± 0.0165
re1	0.4999 ± 0.0070	0.5068 ± 0.0187	0.4970 ± 0.0100	0.5475 ± 0.0171
	Significant test based on NMI values			
re0	CAWP ≫ KernelKmeans≫ NCut ≫ PAM			
re1	CAWP ≫ KernelKmeans > PAM> NCut			

Table 7: Results on WebACE data

Data sets	F-measure			
	PAM	KernelKmeans	NCut	CAWP
wap	0.4341 ± 0.0046	0.4625 ± 0.0417	0.5294 ± 0.0215	0.5485 ± 0.0436
k1a	0.4231 ± 0.0027	0.4582 ± 0.0283	0.4934 ± 0.0122	0.5391 ± 0.0400
	significant test based on F-measure values			
wap	CAWP ≫ NCut ≫ KernelKmeans≫ PAM			
k1a	CAWP ≫ NCut ≫ KernelKmeans≫ PAM			
Data sets	NMI			
	PAM	KernelKmeans	NCut	CAWP
wap	0.4533 ± 0.0043	0.4913 ± 0.0256	0.5520 ± 0.0128	0.5585 ± 0.0171
k1a	0.4264 ± 0.0021	0.4994 ± 0.0241	0.5430 ± 0.0086	0.5498 ± 0.0148
	significant test based on NMI values			
wap	CAWP > NCut ≫ KernelKmeans≫ PAM			
k1a	CAWP ≫ NCut ≫ KernelKmeans≫ PAM			

Table 8: Key words of representative documents produced by CAWP in each topic of Multi 5

Topics	Key Words in Representative documents
graphics	{image, data, graphic, mac, code}
motorcycles	{bike,motorcycle, shaft, dod, forces}
baseball	{game, baseball, team, cubs, braves}
space	{space,launch, shuttle, mike elescope}
mideast	{jews, israel, arabs, armenian, palestine}

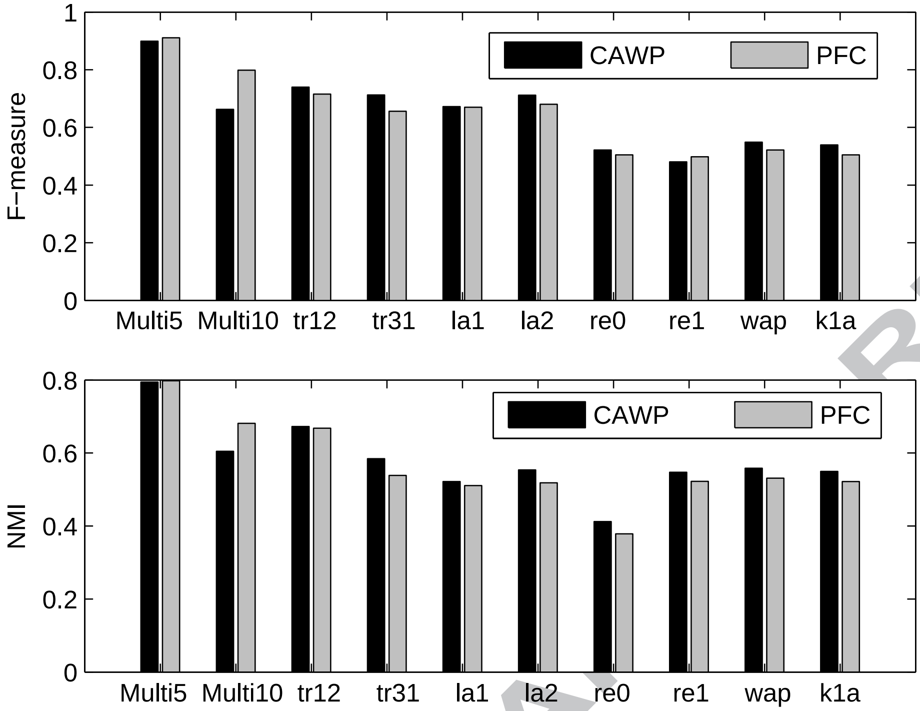


Figure 4: F-measure and NMI of CAWP and PFC

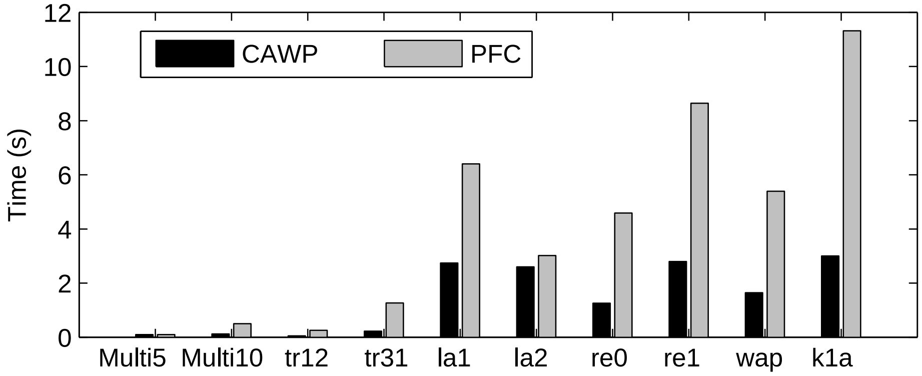
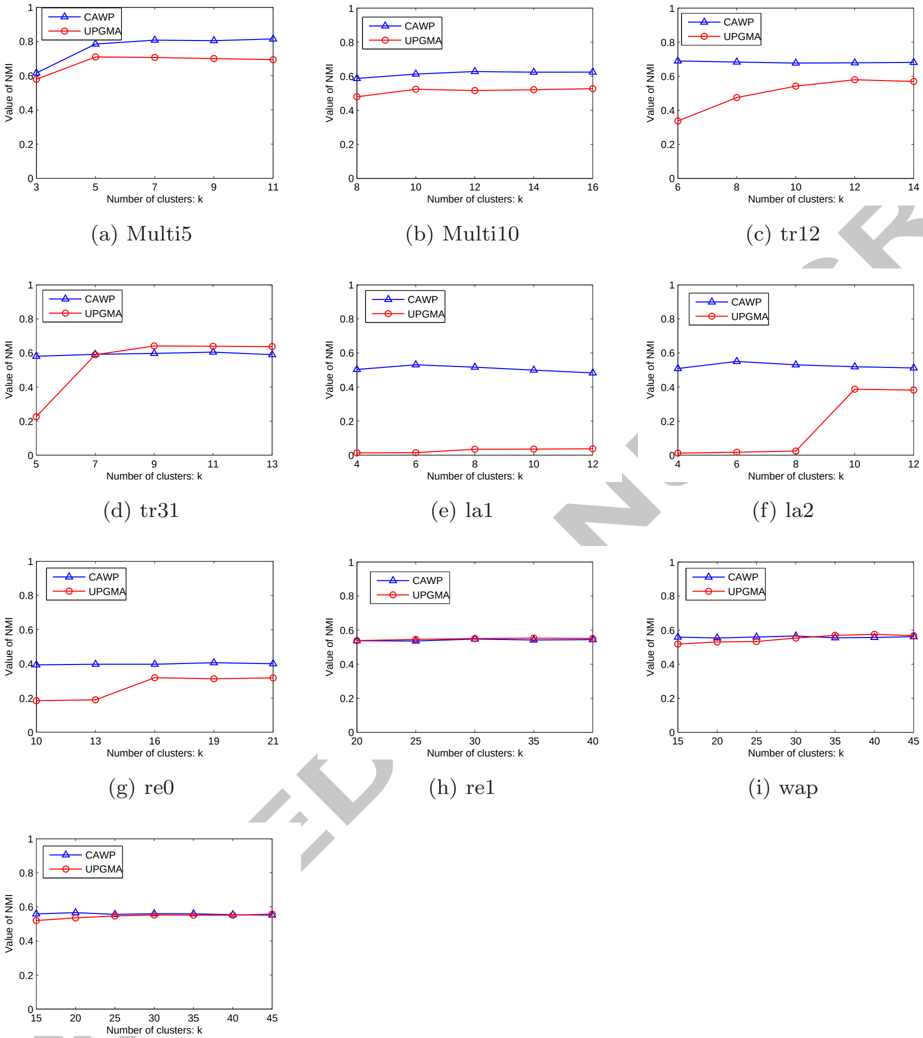


Figure 5: CPU execution time comparison of CAWP and PFC.



(a) Multi5

(b) Multi10

(c) tr12

(d) tr31

(e) la1

(f) la2

(g) re0

(h) re1

(i) wap

(j) k1a

Figure 6: Comparison of CAWP and UPGMA with respect to the number of clusters k on ten datasets.

the characteristics of each produced cluster. Systematic experimental study on benchmark document datasets shows that CAWP outperforms other center-based k-partitions clustering, including Kernel K-means and k-medoids. Its performance is also overall better and more stable than hierarchical clustering. Compared with spectral clustering, CAWP achieves the better or comparable accuracy with a scalable time complexity. All these results show the great potential of CAWP as similarity-based clustering.

Clustering is a very important data mining tool for automatic discovery of underlying data structure and efficient organization of data content. As documents take up the majority of data on the Web, document clustering for automatic document categorization comes out to be a critical component in expert systems involving document processing and analysis, such as topic detection in news, grouping of Web pages for advanced browsing, and patent document analysis Jun et al. (2014). Our proposed clustering algorithm may be used by any expert systems that provide cluster analysis and is especially useful when proper pairwise (dis)similarities are available.

There are several possible research directions may be worked on in the future to further extend and enhance the work made in this paper. First, evolutionary algorithms such as Genetic Algorithms (GAs), Particle Swarm Optimization (PSO), and the recently developed cohort intelligence Krishnasamy et al. (2014) may be integrated to find global solutions of the objective function. Moreover, as proximity-based clustering which provides a convenient interface to be integrated with any systems that output pairwise document similarities, it is beneficial to consider some advanced text representations and similarity measures with NLP techniques to further improve the performance of document clustering. Another important research direction that could be worked on is to further extend the current batch algorithm to an incremental one for large data or stream data.

References

Bellec, J.-H., & Kechadi, M.-T. (2007). CUFRES: Clustering using fuzzy representative events selection for the fault recognition problem in telecommunication networks. In *PIKM* (pp. 55–62).

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Boley, D., Gini, M., Gross, R., Han, E., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., & Moore, J. (1999). Document categorization and query generation on the World Wide Web using WebACE. *AI Review, volume*, *13*, 365–391.

Chim, H., & Deng, X. (2007). A new suffix tree similarity measure for document clustering. In *Proceedings of the 16th international conference on World Wide Web* (pp. 121–129).

Cutting, D. R., Karger, D. R., Pedersen, J. O., & W.Tukey, J. (1992). Scatter/gather: a cluster-based approach to browsing large document collections. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 318–329).

Davé, R. N., & Sen, S. (2002). Robust fuzzy clustering of relational data. *IEEE Trans. Fuzzy Syst.*, *10*, 713–727.

Dhillon, I. S., Guan, Y., & Kulis, B. (2007). Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, *29*, 1944–1957.

D'hondt, J., Vertommen, J., Verhaegen, P.-A., Cattrysse, D., & Duflou, J. R. (2010). Pairwise-adaptive dissimilarity measure for document clustering. *Information Sciences*, *180*, 2341–2358.

Guha, S., Rastogi, R., & Shim, K. (2001). CURE: An efficient clustering algorithm for large databases. *Information Systems*, *26*, 35–58.

Hai-Tao Zheng, H.-G. K., Bo-Yeong Kang (2009). Exploiting noun phrases and semantic relationships for text document clustering. *Information Sciences*, *179*, 2249–2262.

Halkidi, M., & Vazirgiannis, M. (2008). A density-based cluster validity approach using multi-representatives. *Pattern Recogn. Lett.*, *29*, 773–786.

Havaliwala, T. H., Gionis, A., & Indyk, P. (2000). Scalable techniques for clustering the web. In *Proc. WebDB Workshop*.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice Hall.

Jun, S., Park, S.-S., & Jang, D.-S. (2014). Document clustering method using dimension reduction and support vector clustering to overcome sparseness. *Expert Systems with Applications*, *41*, 3204–3212.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley.

Krishnasamy, G., Kulkarni, A. J., & Paramesran, R. (2014). A hybrid approach for data clustering based on modified cohort intelligence and k-means. *Expert Systems with Applications*, *41*, 60096016.

Lang, K. (1995). NewsWeeder: learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning* (pp. 331–339).

Larsen, B., & Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 16–22).

Lin, Y.-S., Jiang, J.-Y., & Lee, S.-J. (2013). A similarity measure for text classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, .

McCallum, A. K. (1996). Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. URL: `http://www.cs.cmu.edu/~mccallum/bow/`.

Mei, J.-P., & Chen, L. (2010). Fuzzy clustering with weighted medoids for relational data. *Pattern Recognition*, *43*, 1964–1974.

Mei, J.-P., & Chen, L. (2011). Document clustering around weighed medoids. In *Proceedins of the International Conference on Information, Communications and Signal Processing* (pp. 1 – 5).

Merlo, P., Henderson, J., Schneider, G., & Wehrli, E. (2003). Learning document similarity using natural language processing. *Linguistik online*, *17*, 99–115.

Rose, K., Gurewitz, E., & Fox, G. (1990). A deterministic annealing approach to clustering. *Pattern Recognition Letters*, *11*, 589–594.

Salton, G. (1971). Cluster search strategies and the optimization of retrieval effectiveness. In G. Salton (Ed.), *The SMART Retrieval System* (pp. 223–242). N. J.: Prentice Hall Englewood Cliffs.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, *24*, 513–523.

Schölkopf, B., Smola, A. J., & Müller, K. R. (1998). Non-linear conponent analysis as a kernel eigenvalue problem. *Neural Comput.*, *10*, 1299–1319.

Shehata, S., Karray, F., & Kamel, M. (2007). A concept-based model for enhancing text categorization. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 629–637).

Shi, J., & Malik, J. (2000). Normalized cuts and imgage segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, *22*, 888–905.

Skabar, A., & Abdalgader, K. (2013). Clustering sentence-level text using a novel fuzzy relational clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering*, *25*, 62 – 75.

Sneath, P. H. A., & Sokal, R. R. (1973). *Numerical Taxonomy — The Principles and Practice of Numerical Classification*. San Francisco: W. H. Freeman.

Strehl, A., & Ghosh, J. (2002). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research*, *3*, 583–617.

Wan, X. (2007). A novel document similarity measure based on earth mover??s distance. *Information Sciences*, *177*, 3718–3730.

Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval* (pp. 267–273).

Zamir, O., & Etzioni, O. (1998). Web document clustering: A feasibility demenstration. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 46–54).

Zhao, Y., & karypis, G. (2005). Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, *10*, 141–168.

Zhong, S. (2005). Efficient online spherical k-means clustering. In *Proceedings of the IEEE International Joint Conference on Neural Networks* (pp. 3180–3185).

Zhong, S., & Ghosh, J. (2003). A comparative study of generative models for document clustering. In *SDW Workshop Clustering High-Dimensional Data and Its Applications*.