An Effective Document Clustering Method using User-Adaptable Distance Metrics

Han-joon Kim and Sang-goo Lee
School of Computer Science and Engineering
Seoul National University
San 56-1, Shillim-dong, Gwanak-gu, Seoul, Korea
{hjkim,sglee}@europa.snu.ac.kr

ABSTRACT
Document clustering is inherently an unsupervised learning process that organizes document (or text) data into distinct groups without depending on pre-specified knowledge. However, real-world applications, such as building a topical hierarchy for a large document collection, need to perform clustering under various kinds of constraints. This paper presents a new type of supervised clustering to organize information in a way that reflects knowledge provided by a user. As a means by which external human knowledge can be incorporated into the clustering process, a quadratic form distance metric is employed that contains a weight matrix. Also, we propose a way of representing knowledge to guide the clustering process and a variant of the gradient descent search technique to find a user-specific weight matrix under the hierarchical clustering strategy.

Keywords
document clustering, information organization, quadratic form distance, user knowledge, hierarchical clustering

1. INTRODUCTION
In order to effectively address the modern information overload problem, it is extremely important to organize documents according to topic. Commonly, this can be achieved by using clustering techniques [1, 8, 11]. Since clusters are distinct groups of similar documents, they can be thought of as representing topicly coherent subtopics in the collection. In particular, when the document collection is large and no classification scheme is available, such as in the web environment, clustering is crucial for constructing a well-organized information structure. Such a clustering that is used for organizing the repositories of an information system is called ‘persistent clustering’.

Persistent clustering should have the following two characteristics. First, it must be adaptable to specific users. That is, it should be able to more closely reflect the preferences of individual users or the specific requirements of an application than the fixed viewpoints provided by conventional clustering methods. Secondly, for reliable information organization, persistent clustering needs to allow an underlying structure to be identified more precisely in terms of users’ viewpoints, even if this means higher costs for both time and resources. Unfortunately, most methods that have been proposed for document clustering do not support persistent clustering. The reason is that conventional clustering algorithms are strongly dependent on simple distance measures (such as the standard Euclidean distance), which do not consider human knowledge. A feasible way to overcome these limitations is to make use of prior or external knowledge in cluster analysis [4, 7, 12]. However, this type of clustering method has not been extensively studied for persistent document clustering.

In this paper, a new approach to persistent clustering is proposed that can effectively accommodate user-specified external knowledge. The basic idea behind our approach is to vary the distance metrics by weighting different dimensions. The proposed method can automatically build a user-adaptable distance function by ‘learning’ from a set of previously defined relevant (or irrelevant) documents.

2. BACKGROUND
Before describing our proposed method for document clustering, the basic notions and related terminology are presented.

2.1 The Representation of Documents
As in standard information retrieval systems, a vector space model is used to represent documents as points in a high-dimensional topic space, where each dimension corresponds to a unique term from the document collection. Therefore, each document can be represented as a vector of the form \( \vec{d}_i = (d_{i1}, d_{i2}, \cdots, d_{in}) \), where \( n \) is the total number of index terms in the system and \( d_{ij} \) (\( 1 \leq j \leq n \)) denotes the weighted frequency that term \( t_j \) occurs in document \( d_i \). Here, we use a normalized tf weighing scheme, where the size of a document vector is scaled to one. In an un-normalized tf weighting scheme [5], a longer document may contain components that cause it to become too far from a given document when computing the distance between vectors.

2.2 The Distance Measure

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC 2002, Madrid, Spain
© 2002 ACM 1-58113-445-2/02/03 ...$5.00.
In order for the distance measure to reflect a user's subjectivity, we employ the quadratic form distance that can model dependencies between different components of feature vectors. The distance between two documents $d_x$ and $d_y$ regarding weights $W$ is given by:

$$\text{dist}_W(d_x, d_y) = \sqrt{(d_x - d_y) \cdot W \cdot (d_x - d_y)} \quad (1)$$

where $\top$ means the transpose of vectors, and when there are $n$ terms, $W$ is an $n \times n$ symmetrical weight matrix whose entry $w_{ij}$ denotes the interrelationship between the components (terms) $i$ and $j$ of the vectors. In order for this function to obtain a non-negative distance value, $W$ is only required to be 'semi-positive definite'; that is, $Z^\top W Z$ is more than or equal to zero for all $Z \in \mathbb{R}^n$, $Z \neq 0$. Hence, the semi-positive definiteness of $W$ needs to be checked during the clustering process. This measure is similar to the Mahalanobis distance where $W$ represents the co-variance matrix for training document vectors. In other special cases, if $W$ is an identity matrix or a diagonal, this distance function represents the un-weighted standard or the weighted Euclidean distance, respectively.

2.3 The Representation of User Knowledge

In order to represent user-specified constraint or knowledge, the proposed method introduces one or more groups of relevant (or irrelevant) document examples to the clustering system. We call each of these document groups a 'document bundle' or simply a 'bundle'. Here, we specify two types of document bundles: positive ones and negative ones. A positive (or negative) bundle is a group of example documents that are judged to be relevant (or irrelevant) by the user. Documents within positive bundles must be placed in the same cluster, while documents within negative bundles must be located in different clusters. Given a document set $D$, suppose that documents $d_x, d_y, \ldots, d_z$ in $D$ are judged to be (ir-) relevant by a user. Then, this subset is denoted by $(d_x, d_y, \ldots, d_z)$ as a document bundle.

**Definition 1.** Given a document set $D$, a positive (negative, respectively) bundle set of $D$ is a collection of positive (or negative) bundles created from $D$, which is denoted by $B^+(D^-)$. That is, $B^+(D^-) \subset 2^D$.

Intuitively, 'all' the pairs of documents within a positive (or negative) bundle are supposed to be judged to be relevant (or irrelevant) by the user. Thus, each bundle can be explained in the context of a binary relation on $D$.

**Definition 2.** Given document positive bundle sets $B^+$ and negative bundle set $B^-$, a positive (negative, respectively) bundle relation of $B^+$ is defined by:

$$B^+ = \{(d_x, d_y) | d_x \in b, d_y \in b \text{ for any bundle } b \in B^+\}$$

By simplicity of knowledge representation, a bundle relation does not explicitly describe the document pair $(d_x, d_y)$ whose element $d_x$ is the identical to its partner $d_y$, or whose symmetric pair $(d_y, d_x)$ also exists in it. Throughout the paper, we will call these bundle relations the 'bundle constraints'.

**Example 1.** Consider the bundles $B^+ = \{(d_1, d_2), (d_3, d_6), (d_3, d_{12})\}$, $B^- = \{(d_1, d_2), (d_3, d_6), (d_3, d_{12})\}$. Each bundle in the two bundle sets is transformed into a bundle relation, then we get $R_{B^+} = \{(d_1, d_2), (d_3, d_6), (d_3, d_{12})\}$ and $R_{B^-} = \{(d_1, d_2), (d_3, d_6), (d_3, d_{12})\}$.

Furthermore, given a set of bundle constraints, we can derive additional constraints that hold. For instance, from $R_{B^+}$ of EXMPELE 1, document pairs $(d_6, d_{12})$ and $(d_3, d_{12})$ can be derived by the transitivity among relevant documents. Like this, the technique for deriving all constraints logically implied by a given bundle constraints is based on the following two axioms or rules.

- **Positive transitivity rule:** If $(d_x, d_y) \in R_{B^+}$ and $(d_y, d_z) \in R_{B^+}$, then $(d_x, d_z) \in R_{B^+}$.
- **Negative transitivity rule:** If $(d_x, d_y) \in R_{B^+}$ and $(d_y, d_z) \in R_{B^+}$, then $(d_x, d_z) \in R_{B^-}$ holds for all $d_x \in [d_x]_{R_{B^+}}$, where $[d_x]_{R_{B^+}}$ is an equivalence class$^1$ of $d_x$ on $R_{B^+}$.

As a result of the positive transitive rule, $R_{B^+}$ becomes an equivalence relation on documents occurring in $R_{B^+}$. The rationale of the negative transitivity rule is that if $d_x$ is irrelevant to $d_y$ and $d_z$ is relevant to $d_x$, then $d_y$ is irrelevant to all documents relevant to $d_z$.

**Example 2.** Let us consider the bundle constraints of EXAMPLE 1. By the above transitive rules, we can augment the constraints as follows.

$$R_{B^+} = \{(d_1, d_2), (d_3, d_6), (d_3, d_12), (d_3, d_5), (d_3, d_{12})\}$$

$$R_{B^-} = \{(d_1, d_2), (d_3, d_6), (d_3, d_12), (d_3, d_5), (d_3, d_{12})\}$$

3. THE USER-ADAPTABLE DISTANCE BASED CLUSTERING METHOD

The overall algorithm proceeds in three phases: 1) the supervising phase, 2) the learning phase, and 3) the clustering phase.

3.1 The Supervising Phase: generating document bundles

The clustering process begins by introducing one or more positive (or negative) document bundles, depending on the user’s judgment of the selected documents. After obtaining a given document collection, then, each of the two types of document bundle constraints is reformed into a corresponding binary relation $R_{B^+}$ (or $R_{B^-}$). Now, the fact that users do not always give consistent document bundles must be considered. There can be redundant document pairs within the same document bundle, or a conflict between two types of bundle constraint, since any pair of documents in a positive bundle must not exist in any negative bundle, and vice versa. Therefore, redundant document pairs are removed from each bundle, and all of the conflicting document pairs in $R_{B^+}$ and $R_{B^-}$ are also eliminated.

Furthermore, based upon the two transitive rules given in Section 2, each of the initial $R_{B^+}$ and $R_{B^-}$ are augmented.

$^1$An equivalence class is a subset whose elements are related to each other by an equivalence relation. A relation on a given set is an equivalence relation if it is reflexive, symmetric, and transitive.
In particular, augmenting a relation $R_B$ according to negative transitive rule can significantly enhance the quality of the resulting clusters, since negative bundle constraints play a role in separating documents within incoherent clusters. In the next phase, each of the document pairs within augmented document bundle relations is used as a training example.

3.2 The Learning Phase: adjusting the distance metric parameters according to user-specified bundle constraints

With augmented bundle constraints generated in the previous phase, the clustering system learns the distance metric parameters to satisfy the given constraints. The problem is how to find the weights that best fit the training examples. The distance metric must be adjusted by minimizing the distance between documents within positive bundles that belong to the same cluster, while maximizing the distance between documents within negative bundles. This dual optimization problem can be solved using the following proposed objective function $Q_B$, which is defined on document bundle sets $B$ (i.e., $B^+ \cup B^-$).

$$Q_B(W) = \sum_{(d_i', d_j') \in R_{B^+} \cup R_{B^-}} I(d_i', d_j') \cdot \theta(diag_W(d_i', d_j'))$$

\[ \text{where} \quad I(d_i, d_j) = \begin{cases} +1 & \text{if } (d_i', d_j') \in R_{B^+} \\ -1 & \text{if } (d_i', d_j') \in R_{B^-} \end{cases} \]

$$\theta(z) = \frac{\alpha}{1 + e^{-z}}$$

We characterize $Q_B$ as a function of weight matrix $W$ since the distance value depends on the weight matrix. A squashing function $\theta$ is used that converges to $\frac{1}{2}$ and whose steepness is determined by $\sigma$. The reason why the real distance value is squashed by $\theta$ is that the sum of the distance values of documents within $R_{B^-}$ during the learning process can dominate the total sum of the distance values of documents within $R_{B^+} \cup R_{B^-}$ when the distance function $diag_W(d_i', d_j')$ satisfies the constraints corresponding to the given document bundles, $Q_B$ decreases.

The goal of the learning phase is to find a weight matrix $W$ that minimizes $Q_B$. This learning problem is analogous to the problem of learning weights for perceptron in artificial neural networks [9]. Thus, a form of gradient descent search algorithm is adopted as a way of finding an optimal weight matrix. However, note that it should be solved in terms of producing a coherent cluster, not by classifying training examples correctly. The algorithm determines a weight matrix that minimizes an objective function by starting with the following initial weight matrix.

$$w_{ij} = \frac{df(t_i, t_j)}{df(t_i) + df(t_j) - df(t_i, t_j)}$$

where $df(t_i)$ (or $df(t_j)$) is the number of documents that contain the term $t_i$ (or $t_j$), and $df(t_i, t_j)$ is the number of documents that contain both terms. This formula is used as a normalized correlation factor with conventional clustering algorithms [10]. The idea behind such an initialization is that it can supplement the inter-correlations among the terms that the search process cannot learn from the previously-defined bundle constraints.

Then, the weight matrix is then repeatedly modified in small steps. At this time, each entry $w_{ij}$ is iteratively updated according to the following training rule:

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij} = w_{ij} + (-\eta \cdot \frac{\partial Q_B}{\partial w_{ij}})$$

$$\frac{\partial Q_B}{\partial w_{ij}} = \sum_{(d_i', d_j') \in R_{B^+} \cup R_{B^-}} I(d_i', d_j') \cdot \frac{\eta e^{-z} (d_{ij} - \bar{d}_{ij}) (d_{ij} - \bar{d}_{ij})}{2\sigma^2 (1 + e^{-z})^2}$$

where $z = \text{dist}_W(d_i', d_j')$, $\eta$ is a positive constant called the ‘learning rate’, which control the amount of weight adjustment at each step of training. Initially, it must be sufficiently small that we do not overstep the minimum in the surface of $Q_B$. According to this weight change rule, each weight changes by a different proportion of the partial derivative of the objective function with respect to that weight. As stated in Equation (5), the matrix of $\frac{\partial Q_B}{\partial w_{ij}}$ derivatives that form the gradient can be obtained by differentiating $Q_B$ from Equation (2). Consequently, by learning the weight matrix using user-specific document bundle constraints, the document space is formed into the document space desired by the user.

In this learning phase, to be noted is that the resulting weight matrix should be checked to make sure it is semi-positive definite (see Eq.(1)). For this, sophisticated techniques such as maximum likelihood estimation have been proposed in [3]. However, such techniques probably are not suitable to be applied to our learning process due to their computational complexity. Our solution is to transform the matrix into its ‘closest’ positive definite matrix by carefully perturbing the matrix so that it does not mingle with the original matrix. To do this, the off-diagonal entries in the weight matrix are multiplied by a number very close to 1 (say 0.995) until the semi-positive definite constraint is approximately satisfied.

3.3 The Clustering Phase: hierarchically clustering previously defined bundles and residual documents

The final phase of the clustering process involves the formation of genuine clusters in the entire document collection. In this phase, the system clusters a given document collection in the same way as the hierarchical agglomerative clustering (HAC) except that it uses the previously defined bundle constraints. First, the HAC algorithm starts with each ‘positive bundle’ and each residual document, not with each of all documents, as a separate cluster. Then it partitions them into a number of clusters based on the learned clustering criterion. Basically, each step of the algorithm involves merging the two clusters whose merger produces the smallest increase in diameter (or radius). At this time, the two clusters to be merged are examined whether they do not include both elements (documents) of a certain pair in $R_{B^-}$. If pairwise documents occurring each pair of $R_{B^-}$ are placed in the same clusters, then the clusters are ignored and the next possible clusters are checked to merge.

In addition, the number of clusters that are generated can be approximately determined from the bundle constraints. In general, how to get the right number of resulting clusters is an open problem. In our work, a minimum number $k_c$ of clusters can be given by using a negative bundle relation (which is described in Theorem 1). After determining the value of $k_c$, the clustering process is carried out until the number of clusters becomes $k_c$. Finally, the user cuts the
resulting tree of clusters at an appropriate level and obtains a set of clusters.

**Theorem 1.** Given negative document bundle sets $B^-$ from a document set $D$ and let $G = (V, E)$ be an undirected graph such that the set of nodes $V$ consists of all the document occurring in $B^-$, and $(d_i, d_j) \in R_{B^-}$ is represented by an edge from node $d_i$ to node $d_j$ in $E$. The number of the resulting clusters generated from $D$ is at least the number of the nodes in the largest complete subgraph of $G$.

**Proof.** Suppose all documents in a subset $D_i$ of $D$ are located in different clusters. Since two documents occurring in each pair in $R_{B^-}$ are located in a separate cluster, every pair of $D_i$ should exist in $R_{B^-}$. By the definition of a complete graph, which is a graph with an edge between every pair of distinct nodes, all nodes occurring in $D_i$ form a complete graph, which is simultaneously a subgraph of $G$. Thus, if the number of the resulting clusters created from $D$ is smaller than the number of the nodes in the largest complete subgraph in the clusters violate negative bundle constraints of $R_{B^-}$.

4. EXPERIMENTAL RESULTS

4.1 Experimental Setup

To evaluate our method, we used a controlled subset of the Reuters-21578 document collection [13] that has been accepted as a clean test collection. This data set consists of 21,578 articles, each one pre-labeled with one or more of 135 topics (categories). For performance evaluation, five different sets of experiments are presented. Each of these sets contains only documents with a single topic, to avoid the ambiguity of documents with multiple topics, and it cannot overlap with that of documents of other data sets. This generated the 5 test sets are shown in Table 1.

In addition, we synthetically generated the positive and negative document bundles. In the case of positive bundles, we picked the size of each bundle as a random variable from the number of documents belonging to each category. Once the number of bundles was determined, the size of each bundle was randomly determined; the size ranged from 2 to 5 documents. Documents belonging to each bundle may overlap with each other, considering a user's bundling behavior. For negative bundles, since each document of each bundle should be selected from a separate topic, the size of a bundle is, at most, the number of topics in a given test set. With that constraint, that size was determined in a pseudo-random way. Each of the documents for negative bundles was randomly selected from each topic. The probability of selecting a particular topic is proportional to the size of topic. The number of such negative bundles was set to less than the number of positive bundles, since we expected that the negative ones would be harder to generate than the positive ones.

4.2 Performance Metrics

A subset of documents corresponding to a selected topic should be as a cluster in the organization. Therefore, we compare how closely each cluster generated by the clustering algorithm matches the set of categories previously assigned to the documents by human judges. In order to measure this, we use Sharon's entropy measure [2] with respect to the topics of the contained documents. The entropy of a cluster $i$ is defined as:

$$e_i = -\sum_{t} \frac{c_{i,t}}{|t|} \log \frac{c_{i,t}}{|t|}$$

$$e_{total} = \sum_i e_i \cdot |t_i|$$

where $c_{i,t}$ is the number of times that the topic label $t$ occurs in cluster $i$ and $|t_i|$ is the number of documents belonging to cluster $i$. The entropy for a cluster is 0 if all the topic labels of all the documents are the same, otherwise it is positive. The total entropy $e_{total}$ can be computed as the weighted average of the individual cluster entropies, as shown in Equation (7). Thus, lower $e_{total}$ means better clustering on the whole.

4.3 Evaluation of Clustering Results

As the baseline for performance measurement, we used two methods: one is the complete-linkage HAC (unsupervised) clustering method that does not consider users' knowledge, and the other is a semi-supervised clustering method proposed in [6]. Note that our approach is not to propose a new unsupervised clustering strategy, but to allow user supervision to guide the clustering. Thus, we focus on eval-

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Dataset size</th>
<th>Topics</th>
<th>Set of topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1 (11)</td>
<td>1294 × 3,000</td>
<td>Sugar, (14), Coffee (17), Corn (10)</td>
<td>Wheat (152), Grain (75), Dollar (78), Natural-gas (108), Money-fx (121)</td>
</tr>
<tr>
<td>T2 (8)</td>
<td>347 × 3,000</td>
<td>Livestock (6), Carasse (38), Hog (7)</td>
<td>Veg-oil (48), Oiletseed (32), Palmoil (23), Corn (85), Soybean (60)</td>
</tr>
<tr>
<td>T3 (10)</td>
<td>477 × 3,000</td>
<td>Orange (24), Tobacco (29), Rice (12)</td>
<td>Cocoa (52), Silver (23), Copper (61), Tin (30), Iron-steel (52), Gold (92), Money-supply (75)</td>
</tr>
<tr>
<td>T4 (10)</td>
<td>290 × 3,000</td>
<td>Tp (39), Pet-chm (29), Retail (19)</td>
<td>Wpi (24), Zinc (25), Meat-feed (19), Cotton (57), Lumber (13), Yen (40), Rubber (39)</td>
</tr>
<tr>
<td>T5 (7)</td>
<td>1,733 × 3,000</td>
<td>Crude (66), Acry (15), Loan (304)</td>
<td>Interest (249), Ship (222), Money-fx (130), Trade (150)</td>
</tr>
</tbody>
</table>

**Figure 1:** The effects of supervision on clustering quality: $T\#$ and $T\#(\text{prev})$ denote the graphs obtained from the proposed method and the previous clustering method, respectively, for each dataset.
5. CONCLUSIONS

This paper has presented a new technique for persistent off-line clustering that can effectively adapt to different users. We developed a framework for representing user knowledge about clusters and a learning method by which user-specified constraints are effectively incorporated into the distance metric for clustering. In addition, we reformulated the classic HAC algorithm so as to control its flexibility in terms of the number of clusters created and the kind of document assigned to each cluster. Our learning approach is applicable to any clustering algorithm that is based on similarity scoring and even similarity-based classification algorithms such as k-Nearest Neighbor. Our experiments indicate that the proposed clustering method holds significant promise for isolating topicically coherent document groups, given a little human effort. Furthermore, we are continuing to investigate techniques for refining the process of knowledge acquisition, so that users can easily compile the document bundles.

6. REFERENCES


