Semi-supervised Document Clustering with Dual Supervision through Seeding

Yeming Hu
Dalhousie University
Faculty of Computer Science
6050 University Avenue
Halifax, Canada
yeming@cs.dal.ca

Evangelos E. Milios
Dalhousie University
Faculty of Computer Science
6050 University Avenue
Halifax, Canada
eem@cs.dal.ca

James Blustein
Dalhousie University
Faculty of Computer Science
and School of Information
Management
jamie@cs.dal.ca

ABSTRACT

Semi-supervised clustering algorithms for general problems use a small amount of labeled instances or pairwise instance constraints to aid the unsupervised clustering. However, user supervision can also be provided in alternative forms for document clustering, such as labeling a feature by associating it with a document or a cluster. Besides labeled documents, this paper also explores labeled features to generate cluster seeds to seed the unsupervised clustering. In this paper, we present a unified framework in which one can use both labeled documents and features in terms of seeding clusters and refine this information using intermediate clusters. We introduce two methods of using labeled features to generate cluster seeds. Experimental results on several real-world data sets demonstrate that constraining the clustering by both documents and features seeding can significantly improve document clustering performance over random seeding and document only seeding.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*; I.5.4 [**Pattern Recognition**]: Application—*Text Processing*

General Terms

Algorithm, Document Clustering, Features

Keywords

User Supervision, Feature Supervision, Seeding, Text Cloud

1. INTRODUCTION

Traditional document clustering is an unsupervised categorization that partitions a given document collection into clusters so that topically similar documents are placed into the same clusters. However, given the same document collection, different users may want to organize it in their own point of view instead of a universal one, which is addressed to some extent by incorporating document supervision [2]. In this paper, we have two types of user supervision, namely, *document supervision* and *feature supervision* for document clustering. *Document Supervision* involves labeling documents, i.e., assigning a document to a cluster. *Feature Supervision* involves labeling features, i.e., associating a feature with a document if that feature describes the topic of that document.

Most prior semi-supervised clustering algorithms use user supervision in the form of document supervision such as labeled instances [2] or instance pairwise constraints [16] for general clustering problems. However, user supervision can also be provided in alternative forms such as labeling features (words) for document clustering in addition to labeling instances (documents). Since this paper focuses on document clustering, we may use *instance* and *document*, *feature* and *word* interchangeably. Labeling documents and words can be performed at the same time, with *little additional effort* for labeling words, if an appropriate document visualization is used, such as text clouds [11]. While the user assigns a document to a cluster based on the document's text cloud, the words appearing in the text cloud can also be labeled by being clicked or highlighted.

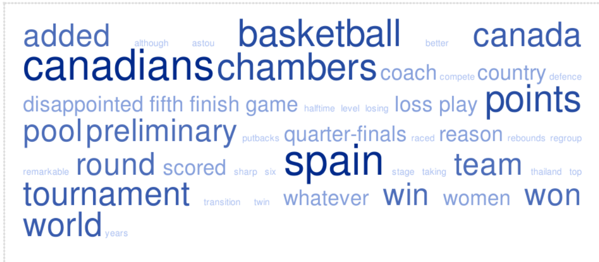


Figure 1: Text Cloud of a Document about Canadian Basketball

Example 1. Consider a collection of news articles about international sports. While the user labels the document displayed as text cloud (Fig. 1) to a cluster, the words associating the document with the specific cluster can also be labeled by being clicked or highlighted. In one scenario, the document (Fig. 1) can be labeled to cluster "Canada", in which the words "Canada", "Canadians" should labeled (associated) with the document.In another scenario, the document would be labeled to cluster "Basketball", in which the words "basketball", "points" should be associated with the document. □

Different labeled words reflect different organizations and the user forms his point of view based on the perception of the words in the text clouds. By using the text cloud for labeling documents, the user can not only label documents to seed the clustering but also label the words discriminating among clusters. It has been argued that document supervision and feature supervision are complementary rather than completely redundant and this joint use has been called *dual supervision* [1].

In this paper, we assume that the user labels a document by reading its content. At the same time, the user can label a word by indicating (e.g. highlighting) whether it is associated with the document or the specific cluster. The text cloud could be used to visualize the document content and enhance the labeling. We extend two methods incorporating the labeled features from document classification to document clustering, namely, feature-vote-model [7] and feature-generative-model [13]. In (semi-supervised) document classification, labeled documents and features are required for each category. However, knowledge of the relevant categories is incomplete in many domains. Semi-supervised document clustering can group documents into partial clusters with labeled documents and features, as well as extend and modify the existing set of clusters to reflect other topical groupings in document collection [2]. The model built from both the labeled documents the labeled features can be used to guide the clustering process. The knowledge from the labeled documents and features will be refined by intermediate clusters in an iterative manner. To this end, we present a unified framework which combines knowledge from labeled documents, labeled features, and unlabeled documents by an iterative clustering process. Finally, we demonstrate the effectiveness of the framework on several real-word data sets.

The rest of this paper is organized as follows. Related work on semi-supervised clustering and feature supervision is discussed in Section 2. In Section 3, we introduce the models to incorporate the labeled features and present the unified framework to combine knowledge from labeled documents, labeled features and intermediate clusters. The details of the experimental results on several real-world text datasets are presented and discussed in Section 4. We conclude this paper and discuss the future work in Section 5.

2. RELATED WORK

Existing semi-supervised clustering techniques, employing user supervision in the form of instance-level constraints, are generally grouped into four categories. First, constraints are used to modify the loss function [3, 10]. Second, cluster seeds derived from the constraints initialize the cluster centers [2]. Third, constraints are employed to learn adaptive distance metrics using metric learning techniques [4]. Finally, the original high-dimensional feature space can be projected into low-dimensional feature subspaces guided by constraints [15]. However, alternative forms of user supervision exist when we apply semi-supervised clustering algorithms to group documents. In this paper, we explore words labeled by being associated with a document when the document is assigned to a cluster.

Liu et al. [12] propose to ask the user to label features for each class and use the set of features labeled for each class to label a set of documents for training classifiers. Druck et al. [7] use labeled features for each class to constrain the probabilistic model estimation on unlabeled instances instead of creating pseudo-instances as done in other approaches. Raghavan et al. [14] make use of feature feedback in the active learning with support vector machine by up-weighting the accepted features. Unlike the above classification methods which require labeled documents and/or features for each class, our framework can deal with partial clusters with labeled documents and/or features. In addition, it explores the unlabeled documents to refine the prior knowledge provided by the user. Huang and Mitchell [9] propose a generative probabilistic framework to incorporate various types of user feedback including feedback on features. In their work, the user needs to assign a feature to an intermediate cluster, which requires the user browse the intermediate clusters and understand them. In our framework, the user associates the features with documents through text clouds, which is much easier and more convenient than understanding intermediate clusters. Hu et al. [8] propose an interactive framework for feature selection for document clustering, in which the user only indicates whether a feature is suitable for clustering. However, they ask the user to label features from a standalone ranked list of features, which requires extra effort for labeling. In addition, they did not explore the usefulness of integrating labeling documents and features together or compare feature supervision with document supervision for clustering.

3. METHODOLOGY

In this section, we first briefly describe basic KMeans algorithm and then present a unified framework to combine the document supervision, feature supervision, and unlabeled documents.

3.1 Background

KMeans is a clustering algorithm based on iterative assignments of data points to clusters and partitions a dataset into K clusters so that the average squared distance between the data points and the closest cluster centers are locally minimized. For a dataset with data points $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}, x_i \in \mathbb{R}^d$, KMeans algorithm generates K clusters $\{\mathcal{X}_l\}_{l=1}^{K}$ of $\mathcal{X}$ so that the objective function

$$J = \sum_{l=1}^{K} \sum_{x_i \in \mathcal{X}_l} ||x_i - \mu_l||^2 \quad (1)$$

is locally minimized and $\{\mu_1, \mu_2, \ldots, \mu_K\}$ represents the centers of the K clusters.

3.2 Algorithms

In this section, we first introduce document supervision and feature supervision in the form of document seeding and feature seeding separately. Then, we present two methods to model feature seeding. At the end, we describe a unified framework to incorporate both document seeding and feature seeding into the KMeans algorithm, namely, *DualSeeded$K$Means*.

3.2.1 Document Seeding

Given a dataset $\mathcal{X}$, as previously described, KMeans can partition it into K clusters $\{\mathcal{X}_l\}_{l=1}^K$. Then, we can define the document seed set $\mathcal{D}^L \subseteq \mathcal{X}$ as the following subset of data points: for each $x_i \in \mathcal{D}^L$, the user provides the cluster $\mathcal{X}_l$ to which it belongs. We assume that there is at least one data point x_i for each cluster $\mathcal{X}_l$. Note that there is a K-disjoint partitioning $\{\mathcal{D}_l^L\}_{l=1}^K$ of the seed set $\mathcal{D}^L$ such that all $x_i \in \mathcal{D}_l^L$ belong to $\mathcal{X}_l$ according to the supervision. We define the centers of the document seed set $\{\mathcal{D}_l^L\}_{l=1}^K$ as $\{\mu_l^d\}_{l=1}^K$:

$$\mu_l^d = \frac{\sum_{x_i \in \mathcal{X}_l^d} x_i}{|\mathcal{X}_l^d|} \tag{2}$$

3.2.2 Feature Seeding

Similar to document seed set $\mathcal{D}^L$, we can define the feature seed set $\mathcal{W}^L$ as the following subset of features: for each $w_i \in \mathcal{W}^L$, the user indirectly associates it with the cluster $\mathcal{X}_l$ through document $x_j \in \mathcal{X}_l$ in which w_i occurs and is labeled from. We assume that each cluster has a topic and at least one feature is associated with it. Note that there does not exist a K-disjoint partitioning $\{\mathcal{W}_l^L\}_{l=1}^K$ of the feature seed set because one feature can be associated with multiple clusters. We define the centers of the feature seed set $\{\mathcal{W}_l^L\}_{l=1}^K$ as $\{\mu_l^w\}_{l=1}^K$, which can be derived from either feature-vote-model (§3.2.4) or feature-generative-model (§3.2.5).

3.2.3 Feature supervision

A document d can be considered as a list of words in the order in which the words occur in the document, i.e., $< w_1, w_2, \ldots, w_{|d|} >$, where $|d|$ is the length of the document. To label a document, we assume that the user needs to read a fraction of its content, i.e., $< w_1, w_2, \ldots, w_m >$, where $m \leq |d|$. While reading a document, the user is assumed to be able to label words he encounters. The labeled words should describe the topic of the document from which they are labeled. The fraction of document content could be displayed as a text cloud and the user could label words by highlighting them on the text clouds. The user labels a feature if it is a good description of the topic of a cluster and discriminates the cluster from others. The features can be associated with a cluster indirectly through the labeled documents.

3.2.4 Feature-Vote-Model

In this method, we use the labeled features in the feature seed set to vote on cluster labels for the unlabeled documents. A similar approach was introduced for document classification [7, 17]. For each labeled feature w in a document d, it contributes one vote for each of its cluster labels (could be associated with multiple clusters). Then, we normalize the vote totals to get a probabilistic distribution over the cluster labels for each document, i.e., $\{P_{li}\}$ for document d_i and cluster $\mathcal{X}_l$. With this soft labeled documents, we can derive the center of μ_l^w from the feature seed set as:

$$\mu_l^w = \sum_{x_i \in \mathcal{X}} P_{li} x_i \tag{3}$$

3.2.5 Feature-Generative-Model

This model was introduced for binary sentiment analysis [13] and we extend it for document clustering with multiple clusters. In this method, we generate each cluster center from the feature seed set directly. We choose to represent the cluster center as a multinomial distribution which generates documents for the corresponding cluster. Without losing generality, we derive the cluster center for cluster $\mathcal{X}_l$ and *words* and *features* are used interchangeably. We define the following notations to aid our derivations:

$\mathcal{V}$ – set of words used for clustering
$\mathcal{P}_{\mathcal{X}_l}$ – set of words labeled for cluster $\mathcal{X}_l$
$\mathcal{N}_{\mathcal{X}_l}$ – set of words labeled for the other clusters
$\mathcal{U}$ – set of unlabeled words used for clustering
m – size of vocabulary, i.e. $|\mathcal{V}|$
$p_{\mathcal{X}_l}$ – number of words labeled for cluster $\mathcal{X}_l$, i.e. $|\mathcal{P}_{\mathcal{X}_l}|$
$n_{\mathcal{X}_l}$ – number of words labeled for the other clusters, i.e. $|\mathcal{N}_{\mathcal{X}_l}|$

In order to derive the multinomial distribution for cluster center of $\mathcal{X}_l$, we assume the following properties about the relationships between words and clusters.

Property 1: All words in $\mathcal{P}_{\mathcal{X}_l}$ are equally likely to occur in a document from cluster $\mathcal{X}_l$.

$$P(w_i|\mathcal{X}_l) = P(w_j|\mathcal{X}_l), \forall w_i, w_j \in \mathcal{P}_{\mathcal{X}_l} \tag{4}$$

We refer to the probability of any word in $\mathcal{P}_{\mathcal{X}_l}$ appearing in a document from cluster $\mathcal{X}_l$ simply as $P(w_p|\mathcal{X}_l)$.

Property 2: All words in $\mathcal{N}_{\mathcal{X}_l}$ are equally likely to occur in a document from cluster $\mathcal{X}_l$.

$$P(w_i|\mathcal{X}_l) = P(w_j|\mathcal{X}_l), \forall w_i, w_j \in \mathcal{N}_{\mathcal{X}_l} \tag{5}$$

We refer to the probability of any word in $\mathcal{N}_{\mathcal{X}_l}$ appearing in a document from cluster $\mathcal{X}_l$ simply as $P(w_n|\mathcal{X}_l)$.

Property 3: The unlabeled words are treated equally in each cluster.

$$P(w_i|\mathcal{X}_l) = P(w_j|\mathcal{X}_l), \forall w_i, w_j \in \mathcal{U} \tag{6}$$

We refer to the probability of any word in $\mathcal{U}$ appearing in a document from cluster $\mathcal{X}_l$ simply as $P(w_u|\mathcal{X}_l)$.

Property 4: A document from cluster $\mathcal{X}_l$ is more likely to contain a word from $\mathcal{P}_{\mathcal{X}_l}$ than a word from $\mathcal{N}_{\mathcal{X}_l}$

$$P(w_p|\mathcal{X}_l) = r \times P(w_n|\mathcal{X}_l) \tag{7}$$

where r is referred to as polarity level, which measures how much more likely a word in $\mathcal{P}_{\mathcal{X}_l}$ occurs in a document from cluster $\mathcal{X}_l$ compared with a word in $\mathcal{N}_{\mathcal{X}_l}$. Since a word in $\mathcal{P}_{\mathcal{X}_l}$ is more likely occurs in a document from cluster $\mathcal{X}_l$, we have $0 < 1/r \leq 1$.

Property 5: The multinomial probability distribution learned from labeled features for each cluster is constrained by summing to one.

$$\sum_{i}^{m} P(w_i|\mathcal{X}_l) = 1 \tag{8}$$

We use property 5 as constraints to derive the appropriate probability distribution based on labeled features. By Eq. 8 it follows that

$$pP(w_p|\mathcal{P}_{\mathcal{X}_l}) + nP(w_n|\mathcal{P}_{\mathcal{X}_l}) + (m-p-n)P(w_u|\mathcal{P}_{\mathcal{X}_l}) = 1 \tag{9}$$

which gives us the following inequality using Eq. 7,

$$pP(w_p|\mathcal{X}_l) + nP(w_n|\mathcal{X}_l) \leq 1$$

$$\Rightarrow pP(w_p|\mathcal{X}_l) + n\frac{P(w_p|\mathcal{X}_l)}{r} \leq 1$$

Since $0 < 1/r \leq 1$, it follows that,

$$P(w_p|\mathcal{X}_l) \leq \frac{1}{p+n}$$

By assigning the maximum probability mass to the known words, $P(w_p|\mathcal{X}_l)$ is set to the maximum value possible, i.e.

$$P(w_p|\mathcal{X}_l) = \frac{1}{p+n} \tag{10}$$

Now, it follows from Eq. 7,

$$P(w_n|\mathcal{X}_l) = \frac{1}{p+n} \times \frac{1}{r} \tag{11}$$

Now, solving Eq. 9, we can have the probabilities for the unlabeled words:

$$P(w_u|\mathcal{X}_l) = \frac{n(1-1/r)}{(p+n)(m-p-n)} \tag{12}$$

Finally, we use Eqs. 10, 11 and 12 to derive the center μ_l^w of cluster $\mathcal{X}_l$. The cluster center μ_l^w is defined as a vector, whose elements are the probabilities of words in $\mathcal{V}$ given the cluster $\mathcal{X}_l$, namely,

$$\mu_l^w = (P(w_1|\mathcal{X}_l), P(w_2|\mathcal{X}_l), \ldots, P(w_m|\mathcal{X}_l)) \tag{13}$$

where $w_i \in \mathcal{V}$ and $m = |\mathcal{V}|$ as previously defined.

In our experiments, we set $r = 100$ based on previous experimental results [13].

3.2.6 Combining Multiple Centers

Opinion pool is a general approach to combine information from multiple sources, such as the centers derived from document seed set and feature seed set in our document clustering problem. Particularly, we use *linear opinion pool* approach to aggregate multiple centers. which was used to combine probability distributions for text classification [13]. In this approach, the aggregated (pooling) center is defined as

$$\mu_l = \sum_{s=1}^{S} \alpha_s \mu_l^s \tag{14}$$

where S is the number of sources we have.

In addition, we compute the weights α's of individual sources based on their error in labeling the document seed set. In particular, we use the same weighting scheme as [13]:

$$\alpha_s = \log \frac{1 - err_s}{err_s} \tag{15}$$

where err_s is error of source s when the derived centers is used to label document seed set. All α_s's are normalized to one.

3.2.7 Dual Semi-supervised KMeans Algorithm

In *DualSeededKMeans*, both the document and feature seeds are used to initialize the KMeans algorithm. Therefore, the center of the l^{th} cluster is initialized with the pooling center derived from μ_l^d and μ_l^w. During the clustering, the cluster centers are refined using the information contained in the intermediate clusters. This information is expressed in the form of intermediate cluster center μ_l^c

$$\mu_l^c = \frac{\sum_{x_i \in \mathcal{X}_l^c} x_i}{|\mathcal{X}_l^c|} \tag{16}$$

where $\mathcal{X}_l^c$ is the l^{th} intermediate cluster. Then, we can incorporate μ_l^c to the algorithm using the *linear opinion pool* technique (Eq. 14). The algorithm is described in detail in Alg. 1. Note that *DualSeededKMeans* can be specialized to *DocumentSeededKMeans* when feature seed set is empty and *FeatureSeededKMeans* when document seed set is empty.

Algorithm 1 *DualSeededKMeans*

Input: Set of data points $\mathcal{X}$, the document seed set $\mathcal{D}^L = \cup_{l=1}^K \mathcal{D}_l^L$, the feature seed set $\mathcal{W}^L = \cup_{l=1}^K \mathcal{W}_l^L$
Output: K clusters $\{\mathcal{X}_l\}_{l=1}^K$
Method:

Compute $\{\mu_l^d\}$ from $\{\mathcal{D}_l^L\}$ using Eq. 2
Compute $\{\mu_l^w\}$ from $\{\mathcal{W}_l^L\}$ using Eq. 3 or Eq. 13
initialize: $\mu_l^{(0)} = \alpha_d \mu_l^d + \alpha_w \mu_l^w$, for $l = 1, \ldots, K; t \leftarrow 0$
repeat
    for all $x_i \in \mathcal{X}$ **do**
        Assign x_i to the closest cluster $\mathcal{X}_l^{(t+1)}$ based on $\{\mu_l^t\}$ and get $\{\mathcal{X}_l^{(t+1)}\}_{l=1}^K$
    end for
    Update intermediate cluster centers: $\mu_l^c \leftarrow \frac{1}{|\mathcal{X}_l^{(t+1)}|} \sum_{x \in \mathcal{X}_l^{(t+1)}} x$
    Update cluster centers: $u_l^{(t+1)} \leftarrow \alpha_d \mu_l^d + \alpha_w \mu_l^w + \alpha_c \mu_l^c$
    $t \leftarrow t + 1$
until *convergence*

3.3 Oracles

Most research involving labeling documents simulates human input by a document oracle that uses the underlying class labels of documents in the dataset [1, 2, 3, 4, 10, 15]. However, in the case of features, we do not have a gold-standard set of feature labels. Ideally, we should have a human expert in the loop labeling the selected features. However, such a manual process is not feasible for repetitive large-scale experiments. Therefore, we construct a feature oracle similar to the method described by [7]. Using the document labels, the oracle computes the χ^2 value of each feature with cluster/class label, and accept a feature if the χ^2 value is above a threshold β. In this paper, the β value is the mean of the top f most predictive features, where $f = 100K$, namely, 100 times the number of clusters. If accepted, the feature oracle labels a feature with the cluster in which it occurs the most and any other clusters in which the feature occurs at least half of the most occurrences.

4. EXPERIMENTAL RESULTS

4.1 Datasets

We conducted our experiments on several real-word datasets of different sizes and also consisting of different types of text documents. We derive three datasets of different sizes from the 20-Newsgroup corpus[1] and three more datasets from webkb[2], industry sector[3], and reuters21578[4] separately. The descriptions and details of the datasets are summarized in Table 1.

We pre-processed each document by tokenizing the text into bags-of-words[5]. Then, we removed the stop words and stemmed all the remaining words. Next, we selected the top 2000 words using mutual information between words and

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/
[2] http://www.cs.cmu.edu/~webkb
[3] http://www.cs.umass.edu/~mccallum/data.html
[4] http://kdd.ics.uci.edu
[5] A word is defined as a sequence of alphabetic characters delimited by non-alphabetic characters.

Table 1: Six Datasets from the 20-newsgroups, Webkb, Industry Sectors and Reuters21578

Dataset	Description	Categories included	Category Doc.	Total Doc.
news-similar-3-100 (D1)	The 20-Newsgroup data set consists of 20 different Usenet newsgroups, each of which has approximately 1000 newsgroup messages.	3:comp.graphics,comp.os.ms-windows.misc,comp.windows.x	100	300
news-multi-7-100 (D2)		7:alt.atheism,comp.sys.mac.hardware, misc.forsale,rec.sport.hockey,sci.crypt, talk.politics.guns,soc.religion.christian	100	700
news-multi-10-100 (D3)		10:alt.atheism,comp.sys.mac.hardware,misc.forsale, rec.autos,rec.sport.hockey,sci.crypt,sci.med, sci.electronics, sci.space, talk.politics.guns	100	1000
webkb-sfcp-4-250 (D4)	webpages from different universities	4:student, faculty, course, project	250	1000
sector-multi-10-100 (D5)	webpages from different industrial sectors	10:basic.materials,capital.goods,consumer.cyclical, oil.and.gas.integrated, investment.services, biotechnology.and.drugs, hotels.and.motels, communications.equipment, railroad, water.utilities	100 (railroad-95)	995
reuters-multi-10-100 (D6)	news articles from Reuters21578. We use the top 10 most frequent categories, documents of which does not have multiple labels.	10:acq, coffee, crude, earn, gold, interest, money-fx, ship, sugar, trade	100 (gold-90)	990

documents [5]. Finally, a feature vector for each document is constructed with TFIDF weighting and then normalized.

4.2 Evaluation Measures

In this paper, we used normalized mutual information (NMI) [6] as the clustering evaluation measure. NMI measures the share information between the cluster assignments S and class labels L of documents. It is defined as:

$$NMI(S, L) = \frac{I(S, L)}{(H(S) + H(L))/2} \tag{17}$$

where $I(S, L)$, $H(S)$, and $H(L)$ denote the mutual information between S and L, the entropy of S, and the entropy of L respectively. The range of NMI values is 0 to 1.

4.3 Analysis of Results

First, we have two sets of comparisons in our experiments. The first set of comparisons is designed to see whether the user provided information can be refined by the intermediate clusters:

- Supervised$K$Means, which performs clustering by assigning documents to nearest cluster centers inferred from either document seed set or feature seed set or both. It can be achieved by running the *DualSeededKMeans* or its specialized cases, i.e., *DocumentSeededKMeans* and *FeatureSeededKMeans*, with only one iteration. Correspondingly, we have *DualSupervisedKMeans*, *DocumentSupervisedKMeans*, and *FeatureSupervisedKMeans*.

- *DualSeededKMeans*, or its specialized algorithms when one of the seed set is empty, i.e., *DocumentSeededKMeans* and *FeatureSeededKMeans*. Note that FeatureSeeded$K$Means has two variants, namely, Feature-Vote-Model and Feature-Generative-Model to derive cluster centers.

We did thorough pair comparisons (Table 2) to demonstrate that incorporating unlabeled documents can refine the information provided by the user and produce better clusters. Concretely, we compared the following pairs of algorithms:

- DocumentSeedsed$K$Means vs. DocumentSupervised$K$Means

- FeatureSeeded$K$Means vs. FeatureSupervised$K$Means using Feature-Vote-Model and Feature-Generative-Model.

- DualSeeded$K$Means vs. DualSupervised$K$Means using Feature-Vote-Model and Feature-Generative-Model.

From Table 2, we can tell that all algorithms with refinement by intermediate clusters improve its clustering performance over the peer algorithms of Supervised$K$Means except when Feature-Vote-Model with only feature supervision works on dataset D3 (news-multi-10-100) and DualSeeded$K$Means and DualSupervised$K$Means using Feature-Vote-Model on D1 (news-similar-3-100) (indicated by * in Table 2). Therefore, intermediate clusters are helpful in improving clustering performance in addition to labeled information.

The second set of comparisons is designed to see whether dual supervision performs better than any single supervision. Thus, we compare *DualSeededKMeans* with *DocumentSeededKMeans*, and *FeatureSeededKMeans*. Again, we have two variants when feature seed set is involved. From Table 3, we can tell that dual supervision with both document and feature generally improve the clustering performance over any single supervision except with feature-generative-model on D1 (news-similar-3-100) and D4 (webkb-sfcp-4-250) indicated by * in Table 3. Note that algorithms with dual supervision works better than document only supervision on all datasets. Therefore, it is worth labeling features.

Second, we ran experiments with incomplete seeding, namely, only a fraction of categories are seeded (Fig. 2 and Fig. 3). It can be seen that the performances decreases with increase number of unseeded clusters. However, the performances do not decrease substantially, showing that *DualSeededKMeans* can extend the seeded clusters and generate more clusters to fit the regularities in the dataset.

Finally, we study the behaviors of the *DualSeededKMeans* with different numbers of document seeds. Note that the more document seeds, the more feature seeds because the feature seeds are labeled while a document is being labeled. We have the following observations from Fig. 4.

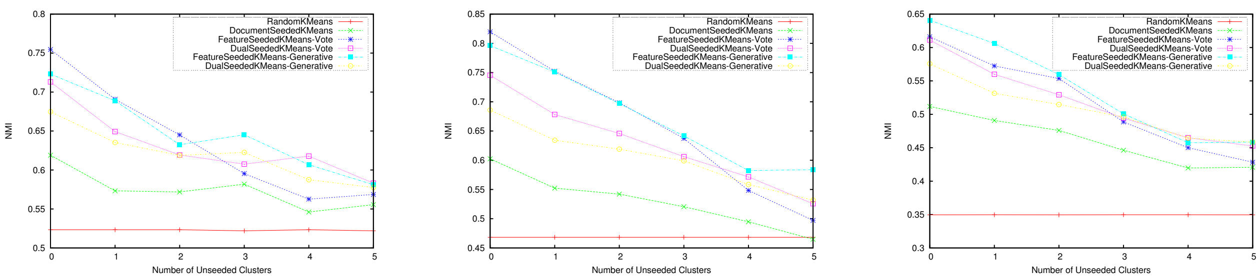
- *DualSeededKMeans* always works better than *DocumentSeededKMeans*. However, the performances of the two algorithms are getting close when more documents are provided. It suggests that the feature la-

Table 2: Supervised$K$Means compared to peer algorithms refined by intermediate clusters. 10 documents are labeled for each cluster and features are labeled by feature oracle from the labeled documents. We did two-tailed paired t-test with p = 0.05 for comparing pairs of algorithms. In this table, we compare algorithms by pairs, i.e., DocumentSeedsed$K$Means vs. DocumentSupervised$K$Means, FeatureSeeded$K$Means vs. FeatureSupervised$K$Means using Feature-Vote-Model and Feature-Generative-Model. DualSeeded$K$Means vs. DualSupervised$K$Means using Feature-Vote-Model and Feature-Generative-Model. All algorithms refined by intermediate clusters works significantly better than peer Supervised$K$Means algorithm except FeatureSeeded$K$Means and FeatureSupervised$K$Means using Feature-Vote-Model on D3 (news-multi-10-100) and DualSeeded$K$Means and DualSupervised$K$Means using Feature-Vote-Model on D1 (news-similar-3-100) indicated by *.

Supervision	Algorithm		D1	D2	D3	D4	D5	D6
No Supervision	Basic KMeans		0.069	0.523	0.468	0.341	0.710	0.350
Document Only	DocumentSeeded$K$Means		0.276	0.692	0.686	0.397	0.815	0.637
	DocumentSupervised$K$Means		0.266	0.625	0.624	0.319	0.786	0.581
Feature Only	Feature-Vote-Model	FeatureSeeded$K$Means	0.551	0.770	0.820*	0.464	0.795	0.649
		FeaureSupervised$K$Means	0.548	0.766	0.820*	0.428	0.791	0.637
	Feature-Generative-Model	FeatureSeeded$K$Means	0.515	0.724	0.791	0.470	0.805	0.692
		FeatureSupervised$K$Means	0.512	0.681	0.747	0.413	0.734	0.660
Dual Supervision	Feature-Vote-Model	DualSeeded$K$Means	0.482*	0.757	0.783	0.421	0.822	0.687
		DualSupervised$K$Means	0.482*	0.745	0.765	0.372	0.815	0.660
	Feature-Generative-Model	DualSeeded$K$Means	0.423	0.732	0.738	0.443	0.824	0.684
		DualSupervised$K$Means	0.421	0.703	0.700	0.391	0.812	0.642

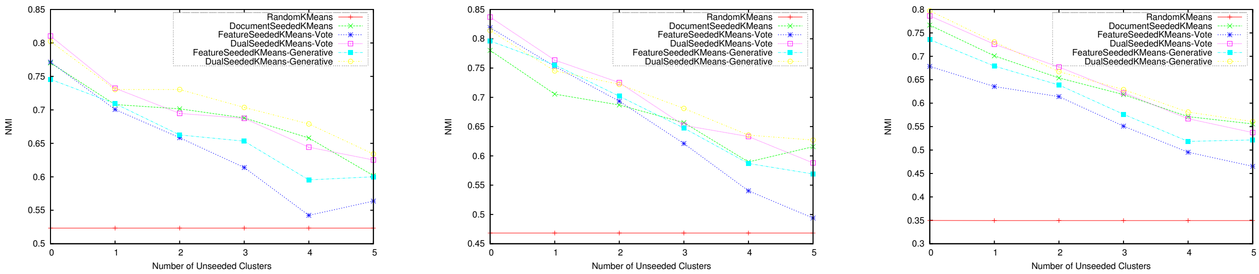
Table 3: Comparison of algorithms with dual supervision to algorithms with any single supervision. 20 documents are labeled for each cluster and features are labeled by feature oracle from those labeled documents. We did two-tailed paired t-test with p = 0.05 for comparing pairs of algorithms. In this table, we compared DualSeeded$K$Means with DocumentSeeded$K$Means, DualSeeded$K$Means with FeatureSeeded$K$Means using Feature-Vote-Model or Feature-Generative-Model. DualSeeded$K$Means works better than DocumentSeeded$K$Means on all datasets. DualSeeded$K$Means works better than FeatureSeeded$K$Means on all datasets except D1 (news-similar-3-100) and D4 (webkb-sfcp-4-250) with Feature-Generative-Model indicated by *.

Feature Model	Algorithm		D1	D2	D3	D4	D5	D6
No Supervision	Basic KMeans		0.069	0.523	0.468	0.341	0.710	0.350
Document Only	DocumentSeeded$K$Means		0.416	0.770	0.780	0.466	0.847	0.767
Feature-Vote-Model	Feature Only	FeatureSeeded$K$Means	0.560	0.771	0.819	0.468	0.796	0.679
	Dual Supervision	DualSeeded$K$Means	0.561	0.810	0.837	0.484	0.845	0.786
Feature-Generative-Model	Feature Only	FeatureSeeded$K$Means	0.515*	0.746	0.796	0.504*	0.808	0.736
	Dual Supervision	DualSeeded$K$Means	0.507*	0.802	0.814	0.502*	0.852	0.797



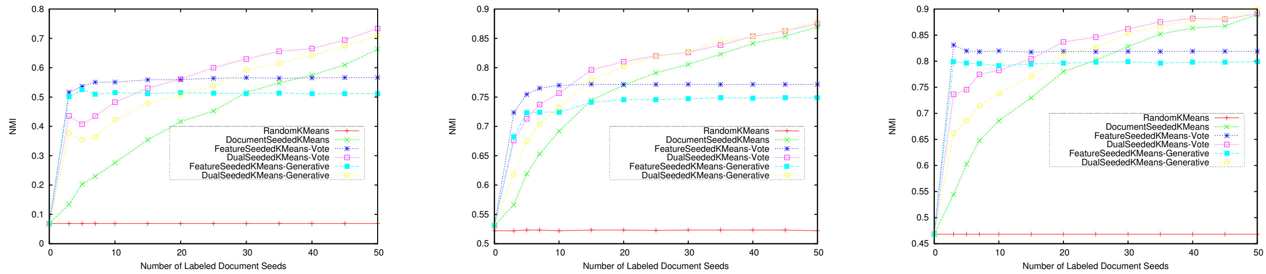
(a) news-multi-7-100 (b) news-multi-10-100 (c) reuters-multi-10-100

Figure 2: Performance as a Function of the Number of Unseeded Clusters. 5 Documents Are Labeled for Each Seeded Cluster where FeatureSeeded$K$Means works better than DocumentSeeded$K$Means and DualSeeded$K$Means

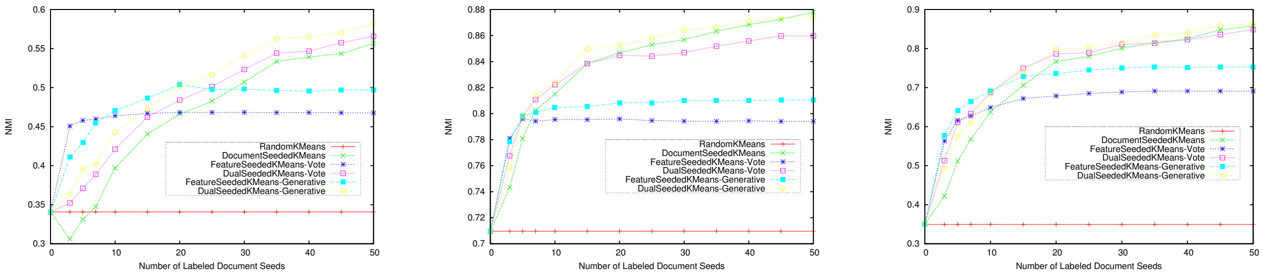


(a) news-multi-7-100 (b) news-multi-10-100 (c) reuters-multi-10-100

Figure 3: Performance as a Function of the Number of Unseeded Clusters. 20 Documents Are Labeled for Each Seeded Cluster where DualSeeded$K$Means works better than DocumentSeeded$K$Means and FeatureSeeded$K$Means



(a) news-similar-3-100 (b) news-multi-7-100 (c) news-multi-10-100



(d) webkb-sfcp-4-250 (e) sector-multi-10-100 (f) reuters-multi-10-100

Figure 4: Performance as a Function of the Number of Labeled Documents. The more documents labeled, the more features labeled and the better performance. The usefulness of labeled features are more obvious when there are only a few documents labeled, e.g., < 10. In fact, the feature supervision even works better than dual supervision at the beginning of the curves, indicating that feature supervision is more reliable when only few documents are labeled.

beling is more useful when there are few documents labeled, i.e., little effort.

- When there are only few documents labeled, *FeatureSeededKMeans* (fewer feature seeds) performs better than *DualSeededKMeans* and *DocumentSeededKMeans*. It suggests that feature supervision is more reliable than document supervision when only little supervision can be provided. However, *DualSeededKMeans* and *DocumentSeededKMeans* improve their performances quickly than *FeatureSeededKMeans* when more document seeds labeled. When there are enough document seeds labeled, both *DualSeededKMeans* and *DocumentSeededKMeans* performs better than *FeatureSeededKMeans*.

- Learning curves of *FeatureSeededKMeans* are steep at the beginning but become flat quickly. Our explanation is that enough feature seeds are labeled after a few document seeds labeled at first. The number of feature seeds labeled does not change much when more document seeds are labeled later.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we incorporate feature supervision in the form of feature seeding. *DualSeededKMeans* is a unified framework to combine document supervision, feature supervision and unlabeled documents in the form of seeding. *DocumentSeededKMeans* and *FeatureSeededKMeans* are two specialized cases of *DualSeededKMeans*. Experimental results demonstrate that unlabeled documents can help to refine the information provided by the user and feature supervision is much more helpful when only few documents can be labeled due to manually cost.

The research presented in this paper is in the context of a document management system that support user-driven organization of document collections. Evaluation of the effectiveness of the system through user studies is in progress.

6. ACKNOWLEDGMENTS

We would like to thank anonymous reviewers for their insightful comments. This research was supported by the funding of NSERC and MITACS NCE.

References

[1] Josh Attenberg, Prem Melville, and Foster Provost. A Unified Approach to Active Dual Supervision for Labeling Features and Examples. In *ECML PKDD 2010 Part I, LNAI 6321*, pages 40–55. Springer, 2010.

[2] S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *International Conference on Machine Learning*, pages 19–26, 2002.

[3] S. Basu, M. Bilenko, and R.J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 59–68. ACM, 2004.

[4] H. Cheng, K.A. Hua, and K. Vu. Constrained locally weighted clustering. *Proceedings of the PVLDB'08*, 1 (1):90–101, 2008.

[5] I.S. Dhillon, S. Mallela, and D.S. Modha. Information-theoretic co-clustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 89–98. ACM, 2003. ISBN 1581137370.

[6] B.E. Dom. An information-theoretic external cluster-validity measure. Technical Report RJ 10219, IBM Research Division, 2001.

[7] G. Druck, G. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 595–602. ACM, 2008.

[8] Y. Hu, E. Milios, and J. Blustein. Interactive feature selection for document clustering. In *the 26th Symposium On Applied Computing*, pages 1148–1155. ACM Special Interest Group on Applied Computing, 2011.

[9] Y. Huang and T.M. Mitchell. Text clustering with extended user feedback. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 420. ACM, 2006.

[10] X. Ji and W. Xu. Document clustering with prior knowledge. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 412. ACM, 2006.

[11] Joe Lamantia. Text Clouds: A New Form of Tag Cloud? http://www.joelamantia.com/tag-clouds/text-clouds-a-new-form-of-tag-cloud, 2007.

[12] B. Liu, X. Li, W.S. Lee, and P.S. Yu. Text classification by labeling words. In *Proceedings of the National Conference on Artificial Intelligence*, pages 425–430, 2004.

[13] P. Melville, W. Gryc, and R.D. Lawrence. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1275–1284. ACM, 2009.

[14] H. Raghavan, O. Madani, and R. Jones. Interactive feature selection. In *Proceedings of IJCAI 05: The 19th International Joint Conference on Artificial Intelligence*, pages 841–846, 2005.

[15] W. Tang, H. Xiong, S. Zhong, and J. Wu. Enhancing semi-supervised clustering: a feature projection perspective. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 707–716. ACM, 2007.

[16] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, 2001.

[17] X. Wu and R. Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 326–333. ACM, 2004. ISBN 1581138881.