# Enhancing Semi-Supervised Document Clustering with Feature Supervision

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

Traditional semi-supervised clustering uses only limited user supervision in the form of labeled instances and pairwise instance constraints to aid unsupervised clustering. However, user supervision can also be provided in alternative forms for document clustering, such as labeling a feature by indicating whether it discriminates among clusters. This paper thus fills this void by enhancing traditional semi-supervised clustering with feature supervision which asks the user to label discriminating features during labeling the instance or pairwise instance constraints. Various types of semi-supervised clustering algorithms were explored with feature supervision. Our experimental results on several real-world datasets demonstrate that augmenting the instance-level supervision with feature-level supervision can significantly improve document clustering performance.

## 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

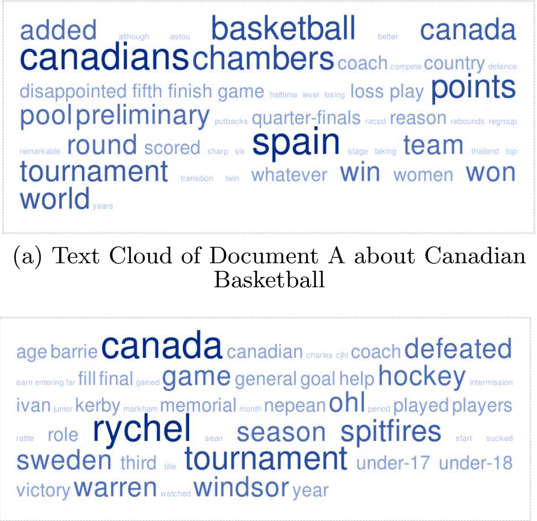
Algorithm, Document Clustering, Features

## Keywords

User Supervision, Feature Supervision, Feature Reweighting, Text Cloud

## 1. INTRODUCTION

Traditional document clustering is an unsupervised categorization of a given document collection into clusters so that documents within the same cluster are more topically similar than those in different clusters. However, given a document collection, different users may want it organized in their own point of view instead of a universal one. Consider a collection of news articles about international sports. One user may like to organize the collection by country while another may want it organized by sport, of which unsupervised clustering is incapable. This is addressed by incorporating user supervision in the clustering process. In this paper, we use two types of user supervision, i.e., document supervision and feature supervision for document clustering. *Document Supervision* involves labeling documents, i.e., assigning a document to a cluster or specifying a "must-link" or "cannot-link" [15] for a pairwise constraint between two documents. *Feature Supervision* involves labeling features, i.e., indicating whether a feature discriminates clusters. Note that a labeled features is not assigned to a cluster but known for its usefulness for clustering.



(a) Text Cloud of Document A about Canadian Basketball

(b) Text Cloud of Document B about Canadian Hockey

**Figure 1: Text Clouds of Two Documents**

Traditional semi-supervised clustering, which uses both labeled and unlabeled instances, has shown its usefulness in generating clusters matching user expectations. User supervision usually takes the form of document supervision. However, the user can also provide alternative forms of user supervision such as feature supervision involving labeling features for document clustering. Since this paper focuses on document clustering, we may use *instance* and *document*,

*SAC'12* March 25-29, 2012, Riva del Garda, Italy.
Copyright 2011 ACM 978-1-4503-0857-1/12/03 ...$10.00.

*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 or specifies a pairwise constraint based on the document's text cloud, the words appearing in the text cloud can also be labeled by being clicked or highlighted.

**Example 1.** Documents A and B in Fig. 1 can be specified as a "must-link" when clustered by country but a "cannot-link" when clustered by sport. Correspondingly, the user would label the words "Canada" "Canadian" "Spain" in the first case but "basketball" "points" "hockey" "rychel" (last name of a hockey player) in the latter case. □

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. It has been argued that document supervision and feature supervision are complementary rather than completely redundant and their joint use has been called *dual supervision* [1].

In this paper, we assume that the user labels a document or establish a pairwise constraint by reading a fraction of the documents' contents. At the same time, the user can label a word by indicating (e.g. highlighting) whether it discriminates among clusters. The text cloud could be used to visualize the fraction of the content and augment the labeling. Since the labeled features are not associated with specific clusters, we incorporate them into the semi-supervised clustering through feature reweighting. Despite of its simplicity, our proposed method is proved to be quite robust and effective under different experimental settings. We enhance semi-supervised clustering algorithms in different categories mentioned in Section 2. We also compare those algorithms using only labeled documents to our proposed method using only labeled features. Finally, we did experiments by allowing that the user makes error in labeling features, that the user only reads a fraction of a document content, and that various numbers of documents are labeled per cluster.

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 present the methodology for incorporating the feature supervision. 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 [14]. In this paper, we enhance the first three methods with user labeled features obtained from feature supervision.

Liu et al. [12] propose to ask the user to assign features with class labels and use the set of features labeled for each class to find a set of documents for training classifiers. Druck et al. [7] use labeled features with class labels to constrain the probabilistic model estimation on unlabeled instances instead of creating pseudo-instances as done in other approaches. Raghavan et al. [13] make use of feature feedback in the active learning with support vector machine by upweighting the accepted features. All those methods ask the user to assign class labels to features and require labeled features for each class. In our paper, we do not ask the user to label features for each cluster. In fact, the user does not even give the cluster label for a feature but just indicates whether it is useful for clustering. We also assume that the user has the document content as context to label words instead of from a standalone ranked list of features. In addition, the labeled features are used to modify document representations when cluster labels of the features are not given. 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 while we only ask the user to indicate whether a feature is good or not for clustering. 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, their work asks the user to label features from a standalone ranked list of features. More importantly, they did not explore the usefulness of integrating labeling documents and features together or compare feature supervision and document supervision for clustering.

## 3. METHODOLOGY

In this section, we first briefly describe basic $K$Means and $K$Means-based semi-supervised clustering algorithms, into which we incorporate feature supervision. Then, we describe feature supervision, feature reweighting, and the framework for semi-supervised clustering with feature supervision.

### 3.1 Background

$K$Means 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 $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}, x_i \in \mathbb{R}^d$, $K$Means algorithm generates $K$ clusters $\{\mathcal{X}_l\}_{l=1}^K$ of $\mathcal{X}$ such that the objective function

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

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

COP-$K$Means [15] is a constraint-based method, where user supervision is used in the form of must-link and cannot-link constraints. During the clustering process, all the constraints in must-link set $\mathcal{M}$ and cannot-link set $\mathcal{C}$ must be satisfied. Otherwise, COP-KMeans fails.

Seeded-$K$Means [2] uses the seed set $\mathcal{S} \subseteq \mathcal{X}$ to initialize the KMeans algorithm, where $\mathcal{S} = \{\mathcal{S}_l\}_{l=1}^K$. During the clustering iterations, the seed documents can change their cluster memberships. In Constrained-$K$Means [2], the seed set is also used to initialize the $K$Means algorithm. Unlike Seeded-$K$Means, the memberships of the seeds are not recomputed, but kept unchanged in the subsequent clustering steps.

Xing et al. [16] tries to learn a better Euclidean distance metric[1] based on the labeled pairwise constraints. Since the learned Euclidean distance metric works with $K$Means, we call it Xing-$K$Means. Xing-$K$Means tries to learn a positive semi-definite matrix $\mathcal{A}$ based on must-link set $\mathcal{M}$ and cannot-link $\mathcal{C}$. Since the dimensionality of document vectors $\{x_i\}_{i=1}^{N}$ is very high, we compute the diagonal matrix $\mathcal{A} = diag(\mathcal{A}_{11}, \mathcal{A}_{22}, \ldots, \mathcal{A}_{nn})$ instead of the full matrix $\mathcal{A}$.

## 3.2 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 are included in the labeled feature set $\mathcal{W}$. 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. Note that the user does not need to associate a feature with a specific cluster.

**Definition 1.** Labeled Feature Set $\mathcal{W}^{\mathcal{L}} = \{w|M(w) = \mathcal{L}\}$, where $M$ is the function to produce the label of a feature:

$$M(w) = \begin{cases} \mathcal{L} & \text{if } w \text{ is labeled for clustering} \\ \mathcal{U} & \text{if } w \text{ is unlabeled} \end{cases} \tag{2}$$

□

**Feature Reweighting** We make use of feature reweighting [8] to incorporate the labeled features for our proposed algorithms. Feature reweighting for $K$Means is performed as follows: the *TFIDF* values of labeled features in $\mathcal{W}^{\mathcal{L}}$ are multiplied by a given weight $g$ $(> 1)$:

$$R_w^{d_i}(tfidf) = \begin{cases} O_w^{d_i}(tfidf) \times g & \text{if } w \in \mathcal{W}^{\mathcal{L}} \\ O_w^{d_i}(tfidf) & \text{otherwise} \end{cases} \tag{3}$$

where $O_w^{d_i}(tfidf)$ and $R_w^{d_i}(tfidf)$ are the original and reweighted $tfidf$ values of feature $w$ in document $d_i$ separately. And then the vector of *TFIDF* values is normalized; Since Xing-$K$Means learns the feature weights based on the pairwise constraints, we use another heuristic to incorporate the labeled features. We first perform Euclidean distance metric learning and obtain the feature weights. Next, we set the the weight of all labeled features $\in \mathcal{W}^{\mathcal{L}}$ to the highest weight learned based on the labeled constraints.

## 3.3 Semi-supervised Clustering with Feature Supervision

The procedure of semi-supervised clustering with feature supervision is presented in Algorithm 1. Since traditional semi-supervised clustering methods employ user supervision in the form of pairwise constraints or cluster seeds, adding feature supervision to semi-supervised clustering therefore amounts to dual supervision for clustering, i.e., both document supervision and feature supervision [1]. Dual supervision takes place together and before the clustering algorithms begin. The clustering algorithms will use both labeled documents and features to guide the clustering process and produce clusters better matching user expectations.

**Algorithm 1** Semi-supervised Clustering with Feature Supervision

```
Input: Set of data points $\mathcal{X}$
Output: $K$ clusters $\{\mathcal{X}_l\}_{l=1}^{K}$
Method:
Perform dual supervision, i.e., document supervision and feature supervision
Obtain the labeled feature set $\mathcal{W}^{\mathcal{L}}$ and the document seed set $\mathcal{S}$ or must-link set $\mathcal{M}$ and cannot-link set $\mathcal{C}$
if Xing-$K$Means then
    Learn diagonal matrix $\mathcal{A}$ and set weights of labeled features to the maximum value in $\mathcal{A}$
    Perform basic $K$Means Clustering using the learned weights
else
    Perform feature reweighting based on labeled feature set $\mathcal{W}^{\mathcal{L}}$.
    Cluster the documents using semi-supervised clustering algorithm.
end if
```

**Algorithm 2** Construction of a Feature Oracle

```
Data Input: Set of unordered features $\mathcal{F}$, Training set $\mathcal{CL}$ – documents and their class labels in the dataset
Parameter Input: Noise level $p_n$, the percentage of noise features the feature oracle will mislabel as "accept", Feature Oracle Capacity $f$ – the number of features the oracle labels as "accept", $f \ll |\mathcal{F}|$
Output: List of ordered features $\mathcal{L}$ – the list of features the feature oracle labels as "accept"
Method:
Compute $\chi^2$ values of all features in $\mathcal{F}$ based on $\mathcal{CL}$
Sort all features in $\mathcal{F}$ according to the computed $\chi^2$ values and obtain ordered list $\mathcal{T}$ of the same size as $\mathcal{F}$
for $i = 1$ to $f$ do
    Flip a coin with the probability $p_n$ getting the tail and obtain the outcome $O$
    if $O$ is tail then
        Randomly pick a feature from the bottom half of $\mathcal{T}$, which is considered to be a noisy feature
        Swap $i^{th}$ feature with the picked noisy feature in $\mathcal{T}$
    end if
end for
Generate $\mathcal{L}$ by taking the top $f$ features of $\mathcal{T}$
```

## 3.4 Oracles

Most research involving labeling documents simulates human input by a document oracle that uses the underlying class labels of the documents in the dataset [1, 2, 3, 4, 10, 14]. 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 methods described by [1, 7]. The feature

[1] Note that our document vectors are normalized and Euclidean distance operating on unit-length instances is equivalent to cosine similarity [14].

**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 | 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 | | 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 | | 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 | webpages from different universities | 4:student, faculty, course, project | 250 | 1000 |
| sector-multi-10-100 | 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 | 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 |

oracle is constructed as described in Algorithm 2. Note that our feature oracle is different from those previous feature oracles in two aspects: (1) Our feature oracle only indicates whether a feature is useful for clustering instead of giving the feature a class/cluster label; (2) Our feature oracle can be noisy by introducing $p_n f$ noisy (mislabelled) features when $p_n > 0$.

# 4. EXPERIMENTAL RESULTS

In this section, we demonstrate the effectiveness of our proposed methods on several real-word datasets. Specifically, we study the performance of different weight values for feature reweighting, the size of the feature oracle vocabulary, the fraction of a document's content the user reads, and the noise level of the user (feature oracle). We enhance several semi-supervised clustering algorithms with feature supervision and compare algorithms with and without feature supervision.

## 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[2] and three more datasets from webkb[3], industry sector[4], and reuters21578[5] 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[6]. 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 documents [5]. Finally, a feature vector for each document is constructed with TFIDF weighting and then normalized.

## 4.2 Evaluation Measures

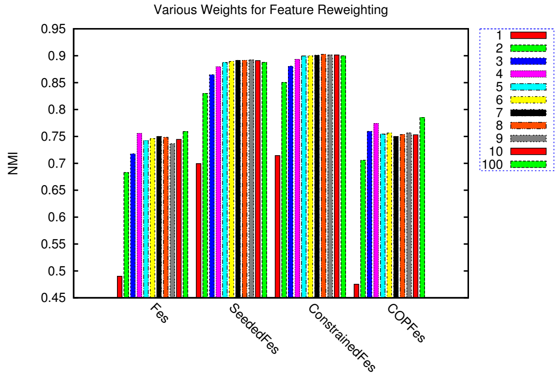


**Figure 2: Feature Reweighting with Different Weights, sector-multi-10-100**

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:

$$NMI(S, L) = \frac{I(S, L)}{(H(S) + H(L))/2} \tag{4}$$

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

Other than explicitly stated, we assume the whole content is read to label a document and a noise-free feature oracle is employed to label the words in the documents. In addition, we set the number of seeds for each cluster to 10, feature capacity per cluster $f$ to 30 if not explicitly described.

Due to the limit of space, we are not able to present all the experimental results for all datasets. Therefore, we mainly use dataset sector-multi-10-100 to illustrate our points. The results for all other datasets have similar pattern as presented here. However, we includes the results of all datasets

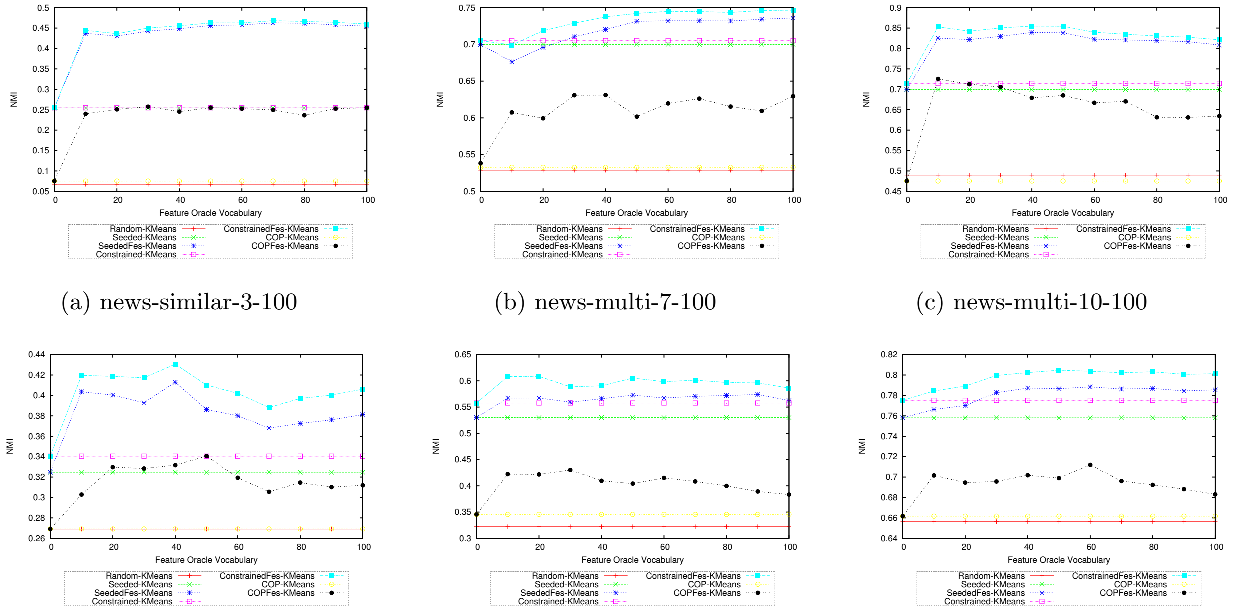
[2] http://people.csail.mit.edu/jrennie/20Newsgroups/

[3] http://www.cs.cmu.edu/~webkb

[4] http://www.cs.umass.edu/~mccallum/data.html

[5] http://kdd.ics.uci.edu

[6] A word is defined as a sequence of alphabetic characters delimited by non-alphabetic characters.



(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 3: Performance as a Function of the Size of Feature Vocabulary, i.e., Feature Oracle Capacity**

for discussion of the feature oracle capacity and number of the seeds for completeness.

**Feature Reweighting $g$:** Different weight values, $g$ (Details in § 3.2), might lead to different clustering results. We conducted experiments with different values of $g$ to show the robustness of our algorithms. Results show that different datasets and algorithms achieved their best performance with different values of $g$ (Fig. 2). However, all weights used improve over their corresponding baselines ($g = 1$), namely, Random-$K$Means, Seeded-$K$Means, Constrained-$K$Means, and COP-$K$Means. Due to the limit of space, we select $g = 2$ to report the results on the following experiments. Weight 2 is selected since it is seldom the weight to achieve the best performance for various algorithms on all datasets. Namely, we give the benefit to the baseline algorithms.

**Feature Oracle Capacity $f$:** The number of features the feature oracle can recognize and label as "accept", namely, the size of the feature oracle vocabulary. If we assume the size of feature oracle vocabulary for each cluster is $s$, then $f = s \times K$, where is $K$ is the number of clusters. Since we do not know the best value of $s$ for clustering, we conducted experiments with different values of $s$. We say the $f$ features that the user labels as "accepted" belong to the feature oracle vocabulary. The general hypothesis is that neither too large nor small values of $s$ can produce good clusters.

Assuming the whole content of the labeled documents read and a noise-free feature oracle, semi-supervised clustering with feature supervision shows significantly[7] improved performance over the method without feature supervision (Fig. 3). With feature supervision, Constrained-$K$Means and Seeded-$K$Means still works much better than COP-$K$Means. It is noticeable that the performance of the clustering algorithms stays relatively stable after the feature oracle vocabulary per cluster reaches a small size 10 to 30. In practice, it means that the user does not have to know all the discriminative features but only a few of the most discriminative ones. As $f$ grows, clustering performances may decrease, e.g., Fig. 3(d). Since the algorithm used to construct the feature oracle is not perfect, it is unavoidable to include some features which are not discriminating for clustering in the feature oracle vocabulary as $f$ grows. We conjecture that clustering performance declines due to the presence of such features introduced by the construction algorithm. The behavior of a noisy feature oracle with explicitly injected poor features is explored later.

**Content Fraction $p_c$:** Since the user does not have to read the whole content of a document to label it, we assume that the user reads a fraction $p_c$ of its content. The general hypothesis is that the more content the user reads, the more features the user will label and the better the performance is if the the user can label the features correctly. However, if the user is not confident with feature labeling, reading more content might not help or even harm the clustering performance.

Assuming a noise-free feature oracle, the clustering performance with feature supervision is improved with more content of labeled documents being read (Fig. 4). At the same time, regardless of the fraction of the content read (at least 10% in our experiments), the performance of semi-supervised clustering with feature supervision is much better over the method with only labeled constraints. In fact, the clustering performance only increases moderately with more than 10% of the content of a document being read. Therefore, the user does not need to read the whole content of a document for effective feature supervision, just as the user does not have to read the whole content to label a document.

**Noisy Feature Fraction $p_n$:** Since the user can make

[7] Two-tailed paired t-test with $p = 0.05$. Also applies to other significance statements.

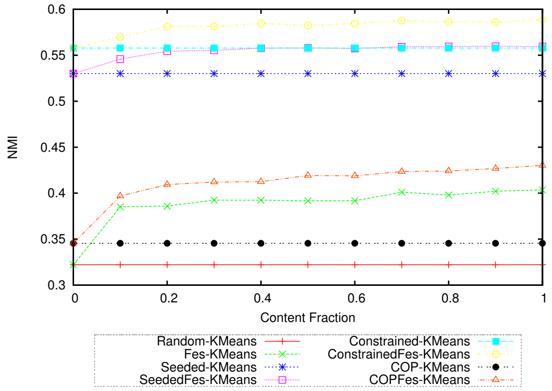


**Figure 4: Enhanced with Feature Supervision with Varying Content being Read, sector-multi-10-100**

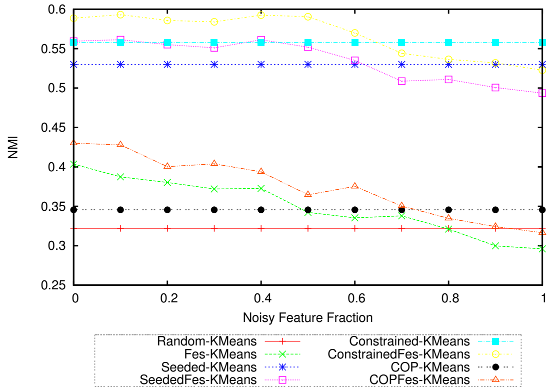


**Figure 5: Enhanced with Feature Supervision with Varying Noise Feature Fraction, sector-multi-10-100**

mistakes by accepting poor features for clustering, we constructed feature oracles with various fractions of noisy features (See Algorithm 2).

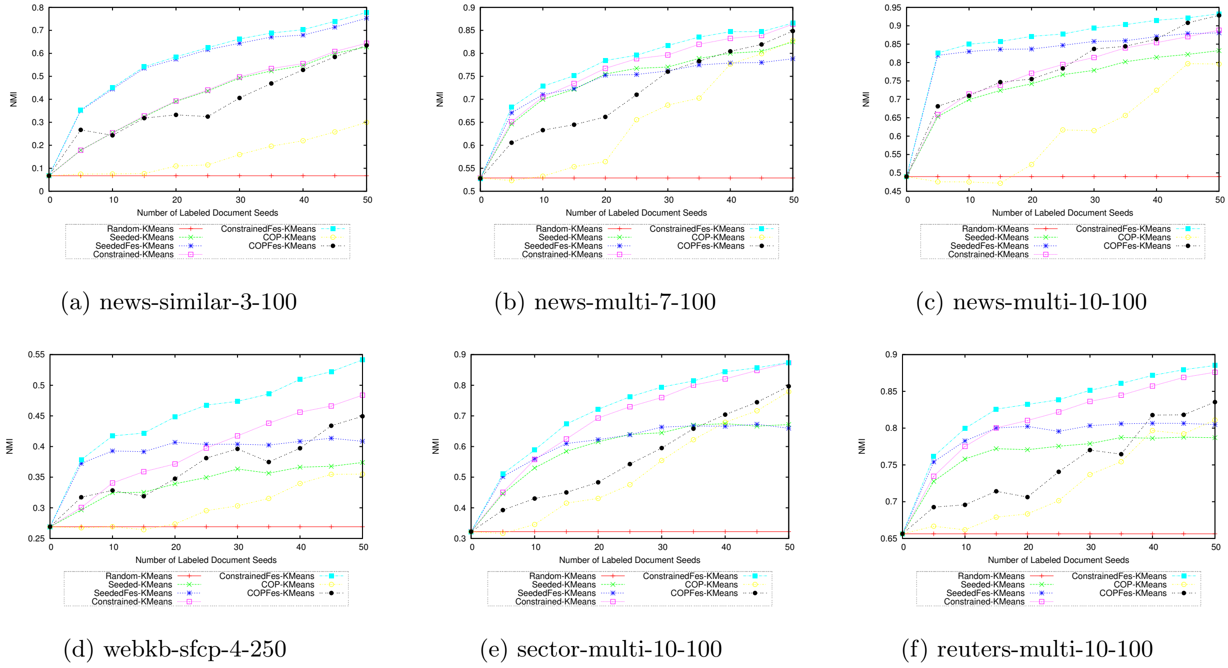
Assuming the whole content of labeled documents being read, we study the behavior of the noisy feature oracle, which can make mistakes in labeling features. Through the experiments, we find that the clustering performance decreases as more noisy features are introduced by the feature oracle, namely, the more mistakes the feature oracle makes, the worse the performance is (Fig. 5). However, even with some incorrect features being labeled as "accepted", the performance of semi-supervised clustering with feature supervision can still improve over the pure document supervision. In fact, Fig. 5 demonstrates that our algorithms have high tolerance of mistakes in labeling features. It may be due to the fact the very few labeled features that are highly discriminative dominate the clustering despite the presence of many non-discriminative features.

**Number of Seeds or Constraints:** We used different numbers of cluster seeds and constraints for the semi-supervised clustering algorithms. The cluster seeds for Seeded-$K$Means and Constrained-$K$Means are randomly sampled and labeled. Since we compare the COP-$K$Means, Seeded-$K$Means and Constrained-$K$Means, *the constraints used for COP-$K$Means* are constructed from the cluster seeds by "must-link"ing the seeds with the same cluster labels and "cannot-link"ing the seeds with different cluster labels. *The constraints for Xing-$K$Means* are randomly sampled.

Feature supervision with only a few documents labeled can improve the clustering performance significantly compared with the pure document supervision method (Fig. 6). To achieve the same performance without feature supervision, a lot more documents have to be labeled. For example, 20 documents per cluster have to be labeled in order to achieve the same performance as 15 documents per cluster labeled with feature supervision (Fig. 6(e)). With more documents labeled, feature supervision becomes less important than when there are only few labeled documents. This implies that feature supervision can help us save user effort from labeling unnecessary documents. Since the user labels features while labeling documents, feature supervision in our proposed methods does not have to involve much extra effort.

**Feature Supervision vs. Document Supervision:** Besides the semi-supervised clustering with/without feature supervision, we also ran the random$K$Means only with the labeled features, i.e., Fes-$K$Means, the algorithm used in Hu et al. [8] to incorporate labeled features. Random-$K$Means with feature supervision works better than COP-$K$Means and comparatively with COPFes-$K$Means (Fig. 5). Although Fes-$K$Means works worse than Seeded-$K$Means and Constrained-$K$Means, semi-supervised clustering with feature supervision always works better than without feature supervision on all datasets. The distance metric learning method based on labeled document constraints works much worse than Random-$K$Means even when quite a large number of constraints is given (Fig. 7). Our explanation is that the high-dimensional and sparse document vectors require too many document constraints to learn a correct distance metric. With only a few document constraints, some unimportant features are unavoidably over-weighted. However, Random-$K$Means with feature supervision only requires a few constraints and features to be labeled to improve the clustering performance. Note that Xing-Fes-$K$Means can still improve the clustering performance further compared to Fes-$K$Means. However, the Euclidean distance metric learning algorithm is quite computationally expensive (hours for metric learning versus seconds for feature reweighting for labeled features) even when a diagonal matrix is assumed because of the high-dimensional vector representation of documents.

**Noisy Feature Oracle and Content Fraction:** Instead of assuming a noise-free feature oracle and that the user reads the whole content of a document to label it, we explore the behavior of the noisy feature oracle while only part of content is read to label a document. In Fig. 8, each curve represents the clustering performance of a noisy feature oracle with different noise level when different fractions of content is read. Fig. 8 verifies that the clustering performance improves as the user reads more content of a labeled document and when the feature oracle is less noisy. More importantly, those figures demonstrate that a noisy feature oracle still works very well even when only a small amount of content of a document is read for labeling. This observation allows human users to make mistakes in feature supervision while reading only part of content for labeling



(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 6: Different Number of Document Seeds (Constraints for COP-$K$Means and COPFes-$K$Means are Generated from Document Seeds, See § 4.3 for Details)**

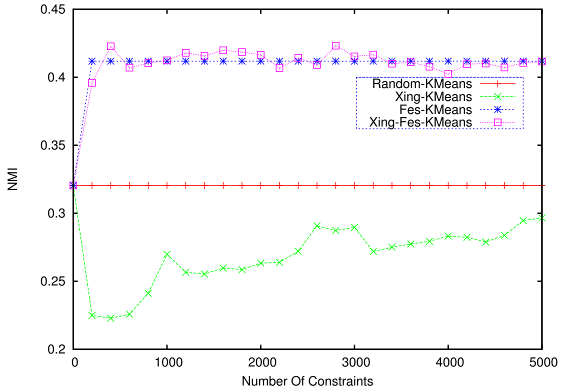


**Figure 7: Metric Learning Method and Feature Supervision Method, sector-multi-10-100**

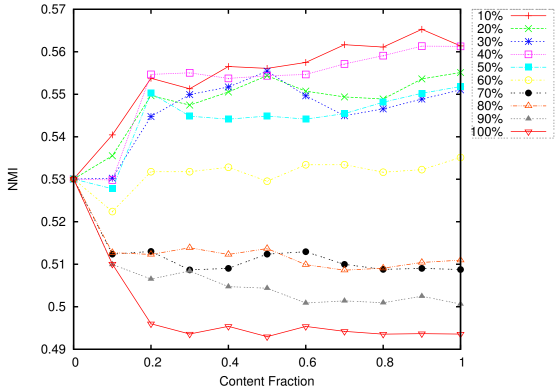


**Figure 8: SeededFes-KMeans with Varying Content being Read for Feature Oracle with Different Noisy Feature Level. Each curve represents a Feature Oracle with the Corresponding Level of Noisy Features. sector-multi-10-100**

a document and validates the practicality of our feature supervision model that feature supervision during document supervision can improve clustering performance. However, for a very noisy feature oracle such as one with 80% noisy features (Fig. 8), the clustering performance decreases when more content of a document is read, since the more content is read, the more noisy features are introduced. Due to the limit of space, only the results for Seeded-$K$Means are presented. The results for Fes-$K$Means, COPFes-$K$Means and ConstrainedFes-$KMeans$ have similar patterns.

## 5. CONCLUSIONS AND FUTURE WORK

In this paper, we enhance the traditional semi-supervised document clustering with feature supervision, which asks the user to label features by indicating whether they discriminate among clusters. We make the assumption that the user can label features while he is labeling a document so that the discriminating features are obtained without too much extra work. The labeled features are incorporated into semi-supervised clustering by feature reweighting, which explicitly gives more weight to the features that, according to the user, discriminate among clusters. We explore this enhancement by employing different types of semi-supervised clustering algorithms. Experimental results demonstrate that all types of semi-supervised clustering algorithms enhanced with feature supervision improved clustering performance significantly. Specifically, the distance metric learned using feature supervision on top of document constraints works significantly better than the one learned only based on document constraints. We also find that feature supervision improves clustering performance even when only a small amount of content of the labeled documents is read and some mistakes are made in labeling features.

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] 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.

[14] 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.

[15] 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.

[16] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. *Advances in Neural Information Processing Systems*, pages 521–528, 2003.