Interactive Feature Selection for Document Clustering

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 document clustering techniques group similar documents without any user interaction. Although such methods minimize user effort, the clusters they generate are often not in accord with their users' conception of the document collection. In this paper we describe a new framework and experiments with it exploring how clustering might be improved by including user supervision at the level of selecting features that are used to distinguish between documents. Our features are based on the words that appear in documents (see §4.1 for details.) We conjecture that clusters better matching user expectations can be generated with user input at the feature level. In order to verify our conjecture, we propose a novel iterative framework which involves users interactively selecting the features used to cluster documents. Unlike existing semi-supervised clustering, which asks users to label constraints between documents, this framework interactively asks users to label features. The proposed method ranks all features based on the recent clusters using cluster-based feature selection and presents a list of highly ranked features to users for labeling. The feature set for next clustering iteration includes both features accepted by users and other highly ranked features. The experimental results on several real datasets demonstrate that the feature set obtained using the new interactive framework can produce clusters that better match the user's expectations. Moreover, we quantify and evaluate the effect of reweighting previously accepted features and of user effort.

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

Document clustering, Feature selection

SAC'11 March 21-25, 2011, TaiChung, Taiwan.

Keywords

Interactive clustering, Interactive feature selection, User supervision

1. INTRODUCTION

Traditional document clustering is an unsupervised classification of a given document collection into clusters so that the documents within the same cluster are more topically similar than those in different clusters. Such methods work by either (a) optimizing some loss function, such as K-Means [4], over all document assignments or (b) fitting a probabilistic model, such as the multinomial naïve Bayes model [14] onto the document collection. Unsupervised processes minimize user effort during clustering and output potential clusters. Users are often dissatisfied with the generated clusters because they are are either not intuitive or do not reflect the users' point of view.

We sought to determine if clusters better matching user expectation may be generated with some supervision by users. User supervision can be used in both components of clustering: in the algorithm itself and in the representation of the documents to be clustered. Semi-supervised clustering applies user-provided constraints such as "must-link" and "cannot-link" between documents to modify the clustering algorithm by changing either the loss function or the probabilistic model. Through optimizing the constrained loss function and forming the probabilistic model with constraints, user expectation is reflected in the clustering algorithm and finally in the generated clusters. Besides constraining the clustering algorithm, user supervision can also be used to achieve a document representation that is more in accord with the user's view. Users can influence the document representation by selecting the feature set to represent the documents. Document category information, which is not available in document clustering setting, is required for an effective feature selection. However, users can also give feedback at the feature level. Therefore, instead of asking users to label enough documents for an effective feature selection, we ask users to directly label features for clustering.

In this paper, we explore how user supervision can work when it is used for feature selection. The work is different from previous semi-supervised clustering as it asks users to label features instead of documents, and the supervision takes the form of selecting features from a list rather than labeling constraints. Because semi-supervised clustering and our framework work at different levels, their performance is not directly comparable, because it is difficult to establish a common quantification of user effort, when the user labels features versus documents. A key benefit of labeling features is that it may take less time than labeling documents as reported in the active learning setting [15].

An overview of the framework we use in our study is the following. We first obtain document clusters using the current feature set. Then, cluster-based feature selection is performed based on the obtained clusters serving as the classes, generating a ranked list of features. We present the top f features in the ranked list to users for labeling. Users must label every feature as "accept" or "don't know" according to their understanding of the document collection. The features users label as "accept" and other highly ranked features are used for the new document representation. The clustering algorithm iterates using the new document representation. In this framework, users are always presented with a number of features based on the recent clusters. The ranking of the features changes at each iteration. In our framework we try to present users the features which are the most promising to be accepted so that users are asked to label as few features as possible.

Our framework is related to the paradigm of active learning (AL) in the document classification setting. It differs from the interactive feature selection framework proposed in the following ways. First, AL is normally used with document classification algorithms but our framework works in the document clustering context. Compared to document clustering algorithm, classification algorithms requires labeled documents to train a classifier. Second, users label documents in AL but label features in our framework. Third, uncertain sampling [12] is used in AL to find the most uncertain document for labeling at each iteration. However, cluster-based feature selection is used to locate a list of the most promising features for labeling.

To explore whether user supervision at the feature level can generate clusters better matching user expectation, we propose an interactive framework for feature selection, in which the feature set obtained from the interactive feature selection is used for clustering. This framework includes several components: an underlying clustering algorithm, unsupervised feature selection, cluster-based feature selection, and user supervision. We use this framework to select the features for producing clusters and evaluate whether the generated clusters conform better to user expectation. We also use this framework to evaluate and quantify the effect of feature reweighting and user effort in terms of labeling features.

In our study, we use simulated users instead of human users for practicality. Simulated users label features based on the documents based on document labels (see §3.4 for details). In addition, both may make mistakes. More importantly, simulated users can be employed repeatedly. Future work will focus on evaluation by human users.

The rest of this paper is organized as follows. Section 2 introduces the related work. In Section 3, we present the interactive framework for feature selection and clustering. Specially, cluster-based feature selection based on clusters is described in detail. Details of the experimental Evaluation is given in Section 4. We conclude with a discussion of the implications of this work and the opportunities for further investigations in Section 5. Section 6 discusses possible future work.

2. RELATED WORK

Existing semi-supervised clustering makes use of user supervision in the form of document-level constraints. Those methods are generally grouped into four categories. First, constraints are used to modify the optimization of the loss function [11] or estimation of parameters [2]. Second, cluster seeds are derived from the constraints to initialize the cluster centroids [1]. Third, constraints are employed to learn adaptive distance using metric learning techniques [5, 18]. Finally, the original high-dimensional feature space can be projected into low-dimensional feature subspaces guided by constraints [19]. In this paper, users are asked to give feedback at the feature level instead of the document level. Except active learning of document constraints such as [10] and [9], most semi-supervised clustering algorithms involve the user supervision outside the clustering process. In this way, all the document constraints are defined before the clustering starts. In our interactive feature selection framework, users interact with the clustering process and label the presented features.

Interactive feature selection in the context of active learning is studied in [15], which used linear support vector machine as the base classifier. At each iteration of the active learning, users are asked to label both the most uncertain document and a list of features. Active learning works in the document classification setting and operates at the document level. It normally uses uncertainty sampling [12] which requires users to label the document about which the classifier(s) is (are) not certain about. In our framework, we explore the interactive feature selection for document clustering and no document labeling is required.

A set of representative features for each class is labeled in [13]. These features are then used to extract a set of documents for each class, which are used to form the training set. Then, the Expectation-Maximization (EM) algorithm [7] is applied iteratively to build new classifiers. The features are only labeled once for constructing cluster seeds in [13] but the feature set is iteratively updated in our approach for document clustering.

Cluster-based feature selection is also performed iteratively in the algorithm proposed in [16]. The main idea of this work is to label a few documents for cluster seeds and for supervised feature selection. It does not involve any user supervision inside the clustering process.

3. METHODOLOGY

In this section, we introduce the interactive feature selection and clustering framework, including an approach investigating the effect of user effort and cluster evaluation measures. We also give details about our simulation of users.

In Table 1, we define the variables we use in this paper.

3.1 Interactive Feature Selection Framework

The high dimensionality of the document text reduces the clustering algorithm performance. Feature selection can alleviate this problem and generate a feature set which is easily interpreted by users. This is one of the motivations for inviting users to label features during clustering. At each iteration, the features presented to users for confirmation are the top f features ranked by cluster-based feature selection, e.g. the χ^2, treating the most recent clusters as classes. Users give one of two answers when a feature is presented. If the feature is believed to be useful for discriminating among

Algorithm 1 Feature Selection with User Supervision

```
• Input: $m$, $f$, $FS^{t-1}_{accepted}$, $FS_{basic}$, $y^c$.
• Output: $FS^{t}_{accepted}$, $FS^m$.

$FS^{t}_{accepted} \leftarrow FS^{t-1}_{accepted}$
$FL^{all} \leftarrow$ Rank all features in $FS_{basic}$ by cluster-based feature selection, e.g. the $\chi^2$, based on $y^c$
{features accepted or rejected are only presented to users once}
$FL = FL^{all} - FS^{t-1}_{accepted}$
for all $i = 1$ to $f$ do
    Present $i^{th}$ feature in $FL$ to the user, get $reply
    if $reply == "accept" then
        Add $i^{th}$ feature into $FS^{t}_{accepted}$
    end if
end for
$FS^m \leftarrow FS^{t}_{accepted}$
$size \leftarrow size\ of\ FS^m$
for $i = 1$ to $m - size$ do
    Add $(f+i)^{th}$ feature in $FL$ to $FS^m$
end for
```

clusters, the user will give answer "accept"; otherwise, an answer "don't know" is given. The algorithm that incorporates feature selection by users, is presented in Algorithm 1. All features accepted by users will be included in the feature set for next clustering iteration. The remaining features, up to the total number m of features for clustering, are selected according to the ranking obtained by the cluster-based feature selection based on the most recent clusters.

3.2 Interactive Document Clustering Framework

After a new feature set with user supervision is obtained 1, the documents are re-clustered using this new feature set. During the re-clustering, the accepted features may be given higher weights. The algorithm for interactive document clustering based on interactive feature selection is given in Algorithm 2. At the beginning, clusters obtained from traditional K-Means with the feature set selected by mean-TFIDF [17]. There are not user accepted features at the beginning. It is worth noting that the feature set can be constructed automatically without user supervision by setting f to 0. In addition, the clustering process can terminate at any time when the user chooses to stop or when the generated clusters do not change. The user may choose to stop when generated clusters or the feature set is satisfactory.

3.3 Cluster-based Feature Selection

When document class labels are available, class-based feature selection can be performed. Examples are χ^2, information gain, and gain ratio. In our work, we apply those techniques without human attached labels, by treating clusters as classes. The cluster a document belongs to is treated as the label of the document. We make use of the class-based feature selection and the cluster labels in our paper to perform feature selection. To be unambiguous, we call it cluster-based feature selection as there is no user supervision in the artificial labels.

The cluster-based (class-based) feature selection ranks the features according to the corresponding measures [6]. Take χ^2 as an example and suppose there are K clusters. There is one χ^2 value between each feature t and each cluster c. Therefore, there are K χ^2 values for a feature t which we call 'local values'. In order to sort the features, we need one global value for each feature. The 'global value' can be defined either as the sum of the local values or the maximum of the local values. The larger the global value is the better the feature is in discriminating among clusters. In this paper, we compute global values as the sums of the local values.

Table 1: Definition of Variables

Variable	Definition
g	the weight for accepted features in $FS^{t}_{accepted}$
s	seed for the randomization of K-Means cluster centroids $\{u_j\}$
m	size of feature set for document clustering
f	number of features presented to users at each iteration
y^c	recent clusters
$\{r_{ij}\}$	assignment of document i to cluster j
FS^m	feature set selected for next clustering iteration
FS_{basic}	all features extracted
$FS^{t-1}_{accepted}$	set of features accepted until iteration $t-1$
$FS^{t}_{accepted}$	set of features accepted until iteration t
$\{d_1, d_2, \ldots, d_N\}$	document vectors

3.4 Simulating Users

In this paper, user supervision is used for feature selection. Users are asked to select good features in the interactive feature selection framework. Our goal in this paper is to compare our interactive framework with unsupervised feature selection. More importantly, we'd like to show that our interactive framework is significantly better in feature selection. In order to test for the statistical significance, many runs of the algorithms have to be performed, which is very costly in terms of human effort required. Unlike human users, the simulating user based on a data set with class labels is fast, cost nothing and is sufficient for an initial proof-of-concept demonstration.

Based on the document class label, a ranking of all features is obtained using class-based feature selection and the top m features can be taken to form a reference feature set. Then the simulated user works as follows: It gives the answer "accept" if the presented feature is included in the reference feature set. Otherwise, the answer is "don't know".

With simulated users, we can quantify performance of the clustering algorithm by comparing computed clusters against the known classes, which we consider as the clusters users expected.

In the simulated user scenario,the interactive framework terminates when the generated clusters do not change or the maximum number of iterations is reached.

3.5 Feature Sets

We compare interactive feature selection framework with

Algorithm 2 Interactive Document Clustering Framework with Feature Selection

- Input: $\{d_1, d_2, \ldots, d_N\}$, s, f, g, m, FS_{basic}.
- Output: $\{r_{ij}\}$

```
Obtain an initial set of clusters $y^c_{initial}$ using $K$-Means with given seed $s$ and feature set selected by unsupervised feature selection, e.g., mean-TFIDF
$y^c \leftarrow y^c_{initial}$
$t \leftarrow 0$
$FS^0_{accepted} \leftarrow \{\}$
repeat
    $t \leftarrow t + 1$
    Feature Selection with User Supervision, Algorithm 1
    Initialize the underlying clustering algorithm with previous iteration's parameters
    Cluster documents using the new feature set and the initialized underlying clustering algorithm and obtain new clustering $y^c_{new}$ and data point assignments $\{r_{ij}\}$
    $y^c \leftarrow y^c_{new}$
until No data point assignment changes or maximum number of iterations is reached or the user chooses to terminate
```

Table 2: Definition of Feature Sets

Feature Set	Definition
FS_{basic}	feature set including all features extracted, i.e., without doing any feature selection
$FS_{mean\text{-}TFIDF}$	feature set selected by mean-TFIDF feature selection method
$FS_{iterative}$	feature set selected by the interactive feature selection framework without user supervision, i.e., f is 0
$FS_{interactive}$	feature set selected by the interactive feature selection with user supervision
$FS_{reference}$	reference feature set, selected by the simulated user, namely, document class labels and class-based feature selection

unsupervised feature selection technique with the underlying algorithms. Since our framework aims to select better feature set for clustering, the underlying algorithms with feature sets selected by different methods are compared. The various feature sets are listed in Table 2.

3.6 Effect of User Effort

In this section, we investigate effect of user effort on the document clustering. To the best of our knowledge, our work is the first one to do that. A few variables are defined for the analysis. As we know, the size of the feature set f is given as an input parameter of the interactive feature selection and clustering framework. The f value can be thought of as the unit of effort as f features are confirmed by users at each iteration. Therefore, total amount of user input spent in the document clustering depends on the value of f. Suppose r is the number of iterations, then the total number of features inspected is defined as $f_{total} = f * r$. Out of the f_{total} features inspected, we define $f_{accepted}$ as the number of features accepted by users. Finally, user effort efficiency $eff\text{-}eff$ can be defined as:

$$eff\text{-}eff = \frac{f_{accepted}}{f_{total}} \tag{1}$$

The larger $eff\text{-}eff$ is, the larger portion of confirmed features is accepted, which are good for clustering.

Feature Reweighting.

Since feature reweighting can boost classification performance in active learning [15], feature reweighting is also incorporated in the interactive clustering framework. Different underlying clustering algorithms have their own method of integrating the feature re-weighting. We use K-Means and Multinomial Naïve Bayes model or EM-NB. For K-Means, the *TFIDF* values of accepted features is multiplied by the given weight g and then the vector of TFIDF values is normalized. In EM-NB, the posterior probability of a class is

$$P(c_j|d_i) = \frac{P(c_j)P(d_i|c_j)}{\sum_{r=1}^{|C|} P(c_r)P(d_i|c_j)} = \frac{P(c_j)\prod_{k=1}^{|d_i|} P(w_{d_i,k}|c_j)}{\sum_{r=1}^{|C|} P(c_r)\prod_{k=1}^{|d_i|} P(w_{d_i,k}|c_j)} \tag{2}$$

for a given document [13]. g affects Eq. 2 through the feature term frequency:

$$p(w_t|c_j) = \frac{1 + \sum_{i=1}^{|D|} \boldsymbol{g}_t \cdot N(w_t, d_i) \cdot P(c_j|d_i)}{|V| + \sum_{s=1}^{|V|}\sum_{i=1}^{|D|} \boldsymbol{g}_s \cdot N(w_s, d_i) \cdot P(c_j|d_i)} \tag{3}$$

where g_s is the weight given to feature w_s in the selected feature set. The weight g_s of a given feature is defined as :

$$|g_s| = \begin{cases} g & \text{if } w_s \text{ is accepted} \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

In our experiments, g is an integer between 1 and 10. Using the above definitions, the effect of user effort on clustering performance is divided into four questions:

1. How does clustering performance change with f?
2. How does clustering performance change with f_{total}?
3. How does feature reweighting affect clustering performance?
4. How does feature reweighting affect user effort?

3.7 Cluster Evaluation Measures

We use two measures to evaluate the cluster quality: clustering accuracy [3], normalized mutual information (NMI) [8]. Clustering accuracy and NMI are both external clustering validation metrics that estimate the clustering quality with respect to a given collection of labeled documents. They measure how close the reconstructed clusters are to the underlying classes of the documents.

Assume we have a clustering T and the underlying classes C. To estimate the clustering accuracy, we map each cluster $t \in T$ to one underlying class $c \in C$ when the documents from c dominate t. Then we define $n(t)$ as the number of dominating documents in t from c. The clustering accuracy CA of T with respect to C is defined as:

$$CA(T,C) = \frac{\sum_t n(t)}{\sum_t |t|} = \frac{\sum_t n(t)}{N} \quad (5)$$

where N is the size of the document collection.

Normalized mutual information (NMI) measures the share information between the cluster assignments S and class labels L of documents. NMI is defined as:

$$NMI(S,L) = \frac{I(S,L)}{(H(S)+H(L))/2} \quad (6)$$

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.

4. EXPERIMENTAL EVALUATION

In this section, we present the datasets used and the experimental results. In our experiments, we set the number of clusters, K, to the number of true classes in the datasets. Two underlying algorithms, K-Means algorithm and Multinomial Naïve Bayes Model (EM-NB) are tested. However, we expect that other clustering algorithm also work because our interactive framework does not depend on any specific algorithm. We use unsupervised mean-TFIDF feature selection and the χ^2 method for the cluster-based feature selection.

4.1 Datasets

We use three datasets to test our proposed framework and explore how clustering performance depends on user effort.

The first dataset is the widely used 20-Newsgroups collection [1] for text classification and clustering. Three reduced datasets, *News-Different-3*, *News-Related-3*, and *News-Similar-3*, are derived according to [2]. Since *News-Similar-3* has significant overlap between groups, it is the most difficult one to cluster.

The second dataset is a collection of papers in full text, which were manually collected by the authors from Association for Computing Machinery (ACM) Digital Library [2]. We use the 1998 ACM Computing Classification System to label the categories [3]. In this paper, we use categories listed in Table 3. H and I are related as they have overlap areas such as "Data Mining" and "Text Clustering" areas. Two datasets are derived from ACM paper collection. The first, *D2-D2&D3-D3*, contains papers which are only from category $D2$, from both categories $D2$ and $D3$, and only from the $D3$ category respectively. Each category has 87 papers in this dataset and is related to each other as they are all from D category. The second, *D-H-I*, consists of 100 papers from each of D,H,I categories.

The third dataset 3-*classic* is made by combining the CISI, CRAN, and MED from the SMART document collection [4]. MED is a collection of 1033 medical abstracts from the Medlars collection. CISI is a collection of 1460 information science abstracts. CRAN is a collection of 1398 aerodynamics abstracts from the Cranfield collection. One hundred documents from each category are sampled to form the reduced 3-*classic* dataset. The topics are quite different across categories, like *News-Different-3*.

Table 3: Legend of ACM Categories

ACM category code	ACM category name
D	Software
$D.2$	Software Engineering
$D.3$	Programming Languages
H	Information Systems
I	Computing Methodologies

We pre-process each document by tokenizing the text into bags-of-words[5]. Then, we remove the stop words and stem all other words. The top m features ranked either by mean-TFIDF or the χ^2 method are employed for clustering. For the K-Means-based algorithms, a feature vector for each document is constructed with TFIDF weighting and then normalized. For EM-NB-based algorithms, the term frequency of the selected features is directly used in the related algorithms.

4.2 Results

We first present the results of the underlying algorithms with feature sets selected by different feature selection techniques. Second, we explore the effect of feature set size on document clustering. Third, we explore how clustering performance depends on user effort.

4.2.1 *Performance of Different Feature Sets*

In this section, we compare and discuss the performance of the same underlying algorithm with different feature sets.

Each pair of algorithm and feature set was run 36[6] times with different initializations over all the datasets. In our experiments, we set the size of feature set m to 600. The average results are listed in Table 4 for K-Means and Table 5 for EM-NB. For the performance of interactive feature set, we take the average performance when the performance stabilizes with the number of feature f displayed to users, e.g. f is between 100 and 300.

As shown in Table 4 and Table 5, interactive feature selection framework can produce better clusters than other unsupervised feature selection methods. In these tables, the performance of the feature set improves significantly when it moves from column FS_{basic} to column $FS_{reference}$ except those in bold. In Table 5, the exception is between $FS_{mean\text{-}TFIDF}$ and $FS_{iterative}$ including both NMI and Accuracy measures of news-diff dataset and news-similar dataset. Although the automatically constructed feature set does not always perform better than the unsupervised feature set, the feature set selected with user supervision does. Especially, when the automated feature set works much worse than the unsupervised feature set for news-similar dataset, user supervision can bring the clustering back to the right track and obtain better performance. Also noted that interactive feature selection and clustering framework achieves comparable performance to the underlying algorithm with the reference

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

[2] http://portal.acm.org/dl.cfm

[3] http://www.acm.org/about/class/1998/

[4] ftp://ftp.cs.cornell.edu/pub/smart

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

[6] The number of times we ran the analysis was chosen to be large enough for calculating statistical significance.

Table 4: Comparison of Performances Of K-Means with Different Feature Sets, namely, FS_{basic}, $FS_{mean\text{-}TFIDF}$, $FS_{iterative}$, $FS_{interactive}$, $FS_{reference}$

Dataset	Measure	Performance by Feature Sets				
		basic	**mean-TFIDF**	**Iterative**	**Interactive**	**Reference**
news-diff	NMI	0.4051	0.5957	0.6651	0.7084	0.6804
	Accuracy	0.6941	0.7931	0.8335	0.8522	0.8330
news-related	NMI	0.1755	0.3341	0.4116	0.4702	0.4501
	Accuracy	0.5285	0.5931	0.6334	0.6722	0.6768
news-similar	NMI	0.0380	0.0765	0.1004	0.1938	0.1818
	Accuracy	0.4243	0.4669	0.4988	0.5479	0.5411
D2-D2&D3-D3	NMI	0.1609	0.2315	0.2727	0.2912	0.2736
	Accuracy	0.5404	0.5971	0.6293	0.6438	0.6235
D-H-I	NMI	0.1051	0.1786	0.2193	0.2594	0.2082
	Accuracy	0.4699	0.5335	0.5794	0.6115	0.5496
3-Classic	NMI	0.5779	0.7220	0.7626	0.8079	0.7854
	Accuracy	0.7544	0.8481	0.8755	0.9017	0.8744

Table 5: Comparison of Performances Of EM-NB Different Feature Sets, namely, FS_{basic}, $FS_{mean\text{-}TFIDF}$, $FS_{iterative}$, $FS_{interactive}$, $FS_{reference}$

Dataset	Measure	Performance by Feature Sets				
		basic	**mean-TFIDF**	**Iterative**	**Interactive**	**Reference**
news-diff	NMI	0.5267	**0.6742**	**0.6737**	0.7845	0.7879
	Accuracy	0.7622	**0.8474**	**0.8450**	0.9050	0.9034
news-related	NMI	0.1966	0.3756	0.3933	0.5227	0.5741
	Accuracy	0.5469	0.6093	0.6150	0.7051	0.7273
news-similar	NMI	0.0819	**0.1491**	**0.0259**	0.1925	0.2114
	Accuracy	0.4742	**0.4464**	**0.3481**	0.4793	0.5379
D2-D2&D3-D3	NMI	0.1834	0.2435	0.2486	0.3178	0.3281
	Accuracy	0.5582	0.5596	0.5653	0.6082	0.6493
D-H-I	NMI	0.1051	0.1786	0.2193	0.2920	0.2082
	Accuracy	0.4881	0.3678	0.4796	0.5967	0.5840
3-Classic	NMI	0.6829	0.8182	0.8412	0.8841	0.8960
	Accuracy	0.7946	0.9069	0.9179	0.9439	0.9503

feature set $FS_{reference}$.

4.2.2 Effect of User Effort

In this section, we study the effect of user effort on clustering performance with feature re-weighting.

Effect of user effort on news-related dataset is shown in Fig. 3 for K-Means and Fig. 5 for EM-NB while effect of user effort on news-similar dataset is demonstrated in Fig. 4 for K-Means and Fig. 6 for EM-NB. The four questions brought up in Section 3 will be answered one by one as follows.

For all datasets, the user effort spent in terms of f_{total} until algorithm converges increases with f, the number of features presented to users in each iteration, e.g., Fig. 1. We also note that the effort efficiency declines when more features displayed in each iteration, e.g., Fig. 2. This may be because the more features are displayed each time, and the higher proportion of features displayed are not in the reference feature set.

Generally speaking, clustering performance increases with more effort provided from users such as Fig. 4 and Fig. 5. However, when the interactive clustering framework with K-Means works with news-related dataset and ACM (D-H-I) dataset, the clustering performance declines after a certain amount of effort is provided. One possible reason is that the extra effort later is used to introduce noisy feature in the reference feature set $FS_{reference}$.

One important finding is that the algorithm converges very quickly when f is very small so that the total number of features accepted is only a small portion of the reference feature set. When weight g is greater than 1 and total accepted features f_{total} is very small, the accepted fea-

Figure 1: f vs. f_{total} with EM-NB on news-diff datasets.

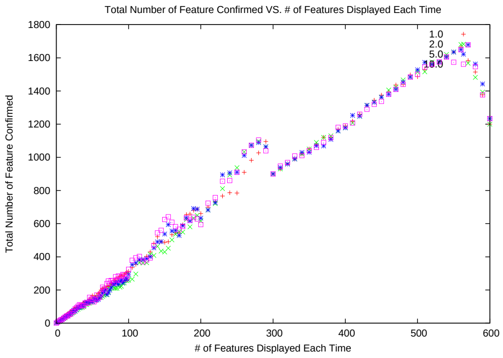


Figure 2: User effort efficiency with EM-NB on news-similar datasets.

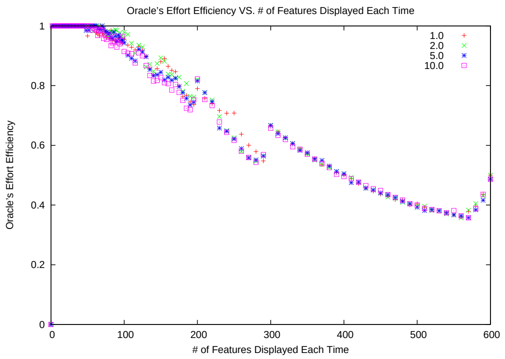
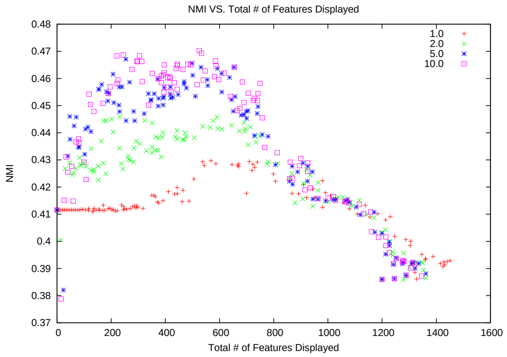


Figure 3: Effect of user effort with KMeans on news-related datasets.



tures could be over-emphasized and have negative effect on interactive clustering framework with EM-NB. For the interactive framework with EM-NB, probabilities of features in the feature set for clustering are affected through Eq. 3 and the performance in terms of NMI declines first and climbs back when more features are accepted by users.

In our experiments, we tried different g values from 1 to 10 for accepted features. Comparing the effect of different g values on various datasets, it can be found that feature reweighting helps the document clustering performance. It can either improve clustering accuracy (Fig. 3) or help reach maximum clustering performance earlier (Fig. 4), which saves user effort. When the interactive framework with EM-NB works with $g > 1$, it improves performance when applied to news-similar dataset(which represents the dataset that is the hardest to cluster) although it achieves comparable performance when applied to other datasets. We suggest $g = 5$ to avoid over-emphasis on accepted features.

5. CONCLUSION

Users can interact with the document clustering process at either the document- or feature-level. Existing semi-supervised clustering algorithms improve performance by

Figure 4: Effect of user effort with KMeans on news-similar datasets.

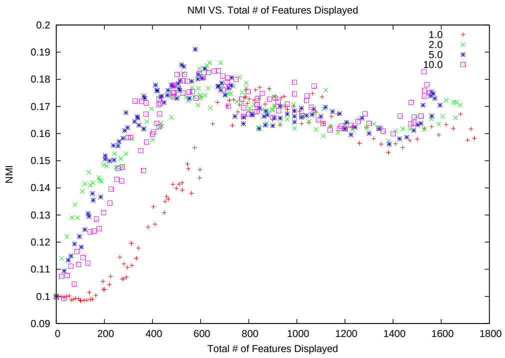


Figure 5: Effect of user effort with EM-NB on news-related datasets.

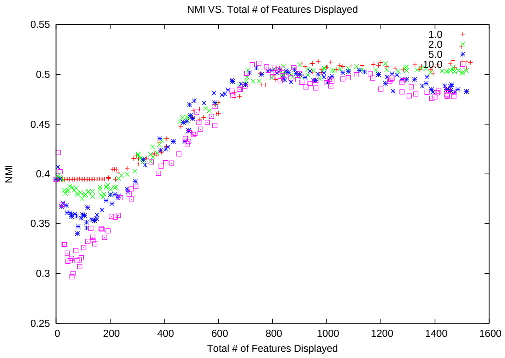
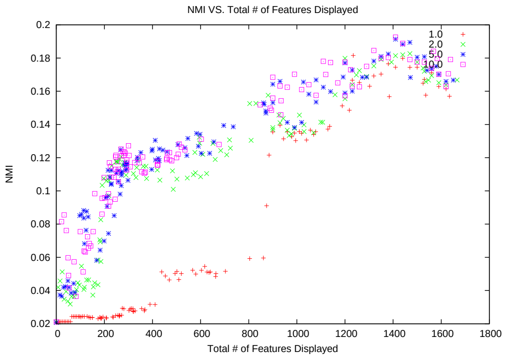


Figure 6: Effect of user effort with EM-NB on news-similar datasets.



exploiting the constraints between documents defined by users. In this paper, we have focused on user-guided clustering at the level of features.

We designed and created a new framework that enables users to guide the clustering process by selecting features which are meaningful to them. The framework interleaves interactive feature selection and clustering iteratively until users choose to stop or the underlying algorithm reaches its terminating conditions. At each iteration, users are presented a list of the top f features ranked by the cluster-based feature selection of the most recent clusters. Since the ranking is based on the recent clusters, meaningful features are likely to have higher ranking. Users rate each of those features by selecting one of the two options: "accept" and "don't know". A revised feature set including the features users "accept"-ed and highly ranked features is used to re-cluster the documents.

This novel method was evaluated by comparison with unsupervised clustering using three different unsupervised feature selection techniques over six varied document datasets. The novel method was significantly better in all cases.

We also studied the effect of user effort on clustering performance in the new framework. Our experiments indicate that a certain number of features must be labeled by users for clustering performance to improve and to avoid early convergence of the algorithm at a local optimum. After a certain amount of user input, the performance can either stay the same or decline a little. Our results show that reweighting of previously "accept"-ed features can also improve clustering performance. However, large weights should be avoided to prevent over-emphasizing the accepted features for some datasets.

6. FUTURE WORK

For the future work, we plan to explore whether the results extend to the case when humans are employed to label features. Since our experiments demonstrated the potential of labeling features to generate clusters better matching users' expectations, future work will focus on the experimental design involving user interaction, e.g. presenting one list of all features to users or presenting K lists of features, one list per cluster. We also plan to present cluster summaries to users to help them in feature selection.

7. REFERENCES

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

[2] S. Basu, M. Bilenko, and R. 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.

[3] R. Bekkerman, M. Scholz, and K. Viswanathan. Improving clustering stability with combinatorial MRFs. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 99–108. ACM, 2009.

[4] C. Bishop. *Pattern Recognition and Machine Learning.* Springer New York, 2006.

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

[6] F. Debole and F. Sebastiani. Supervised term weighting for automated text categorization. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, pages 784–788. ACM, 2003.

[7] A. Dempster, N. Laird, D. Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

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

[9] A. Huang, D. Milne, E. Frank, and I. Witten. Clustering documents with active learning using Wikipedia. In *Eighth IEEE International Conference on Data Mining, 2008. ICDM'08*, pages 839–844, 2008.

[10] R. Huang and W. Lam. An active learning framework for semi-supervised document clustering with language modeling. *Data & Knowledge Engineering*, 68(1):49–67, 2009.

[11] 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.

[12] D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 148–156, 1994.

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

[14] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to classify text from labeled and unlabeled documents. In *Proceedings of the National Conference on Artificial Intelligence*, pages 792–799, 1998.

[15] 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.

[16] L. Rigutini and M. Maggini. A semi-supervised document clustering algorithm based on EM. In *Proceedings of the 2005 IEEE/WIC/ACM International conference on Web Intelligence (WI'05)*, 2005.

[17] B. Tang, M. Shepherd, E. Milios, and M. Heywood. Comparing and Combining Dimension Reduction Techniques for Efficient Text Clustering. In *International Workshop on Feature Selection for Data Mining*, in conjunction with 2005 SIAM International Conference on Data Mining, Newport Beach, California, April 23 2005.

[18] N. Tang and V. Vemuri. User-interest-based document filtering via semi-supervised clustering. *Foundations of Intelligent Systems*, pages 573–582, 2005.

[19] 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.