Towards Subjectifying Text Clustering

Sajib Dasgupta and Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
{sajib,vince}@hlt.utdallas.edu

ABSTRACT

Although it is common practice to produce only a single clustering of a dataset, in many cases text documents can be clustered along different dimensions. Unfortunately, not only do traditional text clustering algorithms fail to produce multiple clusterings of a dataset, the only clustering they produce may not be the one that the user desires. In this paper, we propose a simple active clustering algorithm that is capable of producing multiple clusterings of the same data according to user interest. In comparison to previous work on feedback-oriented clustering, the amount of user feedback required by our algorithm is minimal. In fact, the feedback turns out to be as simple as a cursory look at a list of words. Experimental results are very promising: our system is able to generate clusterings along the user-specified dimensions with reasonable accuracies on several challenging text classification tasks, thus providing suggestive evidence that our approach is viable.

Categories and Subject Descriptors

I.5.3 [**Clustering**]: Algorithms

General Terms

Algorithms, Experimentation

Keywords

interactive clustering, active clustering, spectral clustering, multiple clusterings, disparate clusterings

1. INTRODUCTION

Text clustering has been widely researched. The inherent ambiguity of natural language and the fact that many text clustering problems involve a large, complex feature space (usually represented as a bag of words) and a large number of data points make it an exciting clustering task. However, there is an important aspect of text clustering that is often overlooked by text mining researchers: many text datasets can be naturally clustered along multiple *dimensions*. For instance, *news articles* can be clustered by topic, the source of the news (e.g., AP, CNN), or the date the article was written; *political blog postings* can be clustered not only by topic, but also by the author's stance on an issue (e.g., support, oppose) or her political affiliation; and *movie reviews* can be clustered by genre (e.g., action, documentary), sentiment (e.g, positive, negative), or even the main actors/actresses. In some text mining applications, it is desirable and sometimes important to recover as many clusterings of a dataset along its important clustering dimensions as possible.

Unfortunately, not only do traditional text clustering algorithms fail to produce multiple clusterings of a dataset, the only clustering they produce may not be the one the user desires. The traditional "optimal objective" approach to clustering is overly constrained in the sense that it forces an algorithm to produce a clustering along a single dimension, specifically the dimension along which the objective function is optimized. Although many different objective functions have been used, the basic qualitative criteria employed to evaluate the *structure* of a clustering (e.g., intra-cluster similarity, inter-cluster dissimilarity, and the size of the clusters) have remained more or less the same over the years. One important notion that is commonly left out of a qualitative measure is the human factor. As mentioned before, different subsets of features might lead to different kinds of clusterings of a dataset. Although a clustering algorithm identifies a particular clustering as the most structured one (i.e., the one that optimizes the objective), it might not be deemed fit by an end user, as she may be interested in a clustering that is different than the *optimal* clustering.

The question, then, is: can a clustering algorithm produce a clustering along the user-specified dimension (which may be suboptimal with respect to the objective function)? This also leads us to our second question: is it possible for a clustering algorithm to produce multiple clusterings of the same data simultaneously according to user interest?

One may argue that it is possible to design the feature space in a way that helps induce the user-desired clustering. This typically involves having the user identify features that are useful for inducing the desired clusters [14]. However, manually identifying the "right" set of features is both time-consuming and knowledge-intensive, and may require a lot of domain expertise. To overcome this weakness, researchers have attempted constrained clustering [2, 18] and learning a similarity metric from *side* information [19] such as constraints on which pairs of documents must or must

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.
Copyright 2010 ACM 978-1-60558-896-4/10/07 ...$10.00.

not appear in the same cluster. However, enough *side* information might not be readily available for many scenarios.

By contrast, recent work has focused on *active clustering*, where a clustering algorithm can incorporate user feedback *during* the clustering process to help ensure that the documents are grouped along the user-desired dimension. One way to do this is to have the user incrementally construct a set of relevant features in an interactive fashion [3, 16]. Another way is to have the user correct the mistakes made by the clustering algorithm in each clustering iteration by identifying the set of clusters that need to be *merged* or *split* [1]. A major drawback associated with these active clustering algorithms is that they involve a considerable amount of human feedback, which needs to be provided in *each* iteration of the clustering process.

In this paper, we attack the problem of *subjectifying* text clustering, or clustering documents according to user interest, from a different angle. We aim to develop a *knowledge-lean* approach to this problem — an approach that can produce multiple user-desired clusterings without relying on human knowledge for fine-tuning the similarity function or selecting the relevant features, unlike existing approaches. To this end, we propose a novel active clustering algorithm, which assumes as input a simple feature representation (composed of unigrams only) and a simple similarity function (i.e., the dot product), and operates by (1) inducing the important clustering dimensions of a given set of documents, where each clustering dimension is represented by a (small) number of automatically generated words that are representative of the dimension; and (2) have the user select the dimension(s) along which she wants to cluster the documents by examining these automatically generated words. In comparison to previous work on feedback-oriented clustering, the amount of user feedback required by our algorithm is minimal. In fact, the feedback turns out to be as simple as a cursory look at these automatically generated words and is required only once. Experimental results are very promising: our system is able to generate the user-specified clustering with reasonable accuracies on several challenging text classification tasks, thus providing suggestive evidence that our approach is viable.

The rest of the paper is organized as follows. In Section 2, we enumerate the properties that are desirable of an algorithm for producing multiple clusterings and discuss related work on multiple clusterings. In Section 3, we present our active clustering algorithm. We describe our evaluation results in Section 4 and present our conclusions in Section 5.

2. PRODUCING MULTIPLE CLUSTERINGS

In this section, we enumerate the desiderata for an algorithm for producing multiple meaningful clusterings of a dataset. By meaningful, we mean that each of these clusterings should be qualitatively strong. As mentioned in the introduction, there are different ways to evaluate the quality of a clustering (e.g., intra-cluster similarity, inter-cluster dissimilarity), which is typically captured by the objective function employed by the clustering algorithm. A clustering is meaningful if it is not overly suboptimal with respect to the objective function.

Before we formalize the concept, let us introduce some notation. Let $X = \{x_1, \ldots, x_n\}$ be a set of n documents to be clustered, where each document $x_i, i = 1\!:\!n$, is represented by a bag of d unigrams $w_1, w_2, \ldots, w_d$. We want to learn m different partitioning functions $f^i, i = 1\!:\!m$, that correspond to m different clusterings $C^i, i = 1\!:\!m$. Specifically, each partitioning function f^i takes data X as input, and outputs a 2-way partition $C^i = \{C^i_1, C^i_2\}, i = 1\!:\!m$, such that $C^i_1 \cup C^i_2 = X$ and $C^i_1 \cap C^i_2 = \phi$.[1] Finally, a clustering algorithm employs an objective function, $o : C^i \rightarrow \Re$, which assigns a qualitative score to each clustering.

To produce multiple meaningful clusterings (henceforth *multiple clusterings* for brevity), we require a clustering algorithm to satisfy three *multi-clustering criteria*:

Multiplicity: Given data X, a clustering algorithm should be able to produce m (where $m > 1$) different clusterings of the data, $C^i, i = 1 : m$, without having to change the feature space and the similarity function.

Distinctivity: Each clustering $C^i, i = 1 : m$, should be *distinctively different*. By distinctively different, we mean that two clusterings are highly dissimilar. That is, the similarity of two clusterings should be close to zero.

Quality: Each clustering $C^i, i = 1 : m$, should be qualitatively strong (i.e., close to optimal) with respect to the objective function o. This condition ensures that none of the clusterings that the algorithm produces is overly suboptimal and thus completely structure-less.

Now, the question is: do existing clustering algorithms produce multiple clusterings of a dataset and satisfy multiplicity, distinctivity and quality? It turns out that a few of them do [5, 7, 9, 11]. Below we first describe each of these algorithms, and then explain how our algorithm differs from them. These existing clustering algorithms can be broadly divided into two categories:

Semi-supervised methods. In these methods, one of the clusterings is provided (by the human) as input, and the goal is to produce the other clustering, assuming that there are only two distinct ways to cluster the data. For instance, Gondek and Hofmann's approach [9] learns a non-redundant clustering that maximizes the conditional mutual information $I(C;Y|Z)$, where C, Y and Z denote the clustering to be learned, the relevant features and the known clustering, respectively. Their approach turns out to be difficult to implement, since it requires modeling the joint distribution of the cluster labels and the relevant features. On the other hand, Davidson and Qi's approach [7] first learns a distance metric D_C from the original clustering C, and then reverses the transformation of D_C using the Moore-Penrose pseudo-inverse to get the new distance metric D'_C, which is used to produce a distinctively different clustering.

Unsupervised methods. Here, each possible clustering of a dataset is produced in an unsupervised manner (i.e., without using any labeled data). Caruana et al.'s approach [5], also known as meta clustering, produces multiple clusterings of the same data by running k-means multiple times, each time with a random selection of seeds and a random weighting of features. The goal is to present each local minimum of k-means as a possible clustering. It suffers from two weaknesses. First, it does not ensure that the aforementioned distinctivity and quality criteria are satisfied. Second, k-means tends to produce similar clusterings regardless of the number of times it is run (see our meta clustering results in Section 4). Jain et al.'s approach [11] is more sophisticated, as it tries to learn two clusterings in a "decorrelated" k-means

[1] While we present our algorithm for 2-way clustering tasks, it can be extended to produce n-way ($n > 2$) clusterings.

framework. Its joint optimization model achieves typical k-means objectives and at the same time ensures that each of the two induced clusterings are distinctively different. Note that Jain et al. use this framework to produce only two clusterings of the data, as the optimization objective becomes too convoluted to generate more clusterings.

Before providing the details of our active clustering algorithm, we describe the primary differences between our algorithm and the aforementioned approaches. First, our algorithm operates in an unsupervised setting, i.e., it neither uses any labeled data nor assumes the existence of a prior clustering, unlike the semi-supervised methods. Second, it satisfies all three multi-clustering criteria (i.e., multiplicity, distinctivity, and quality). Finally, it is *not* restricted to producing only two clusterings.

3. OUR APPROACH

In this section, we describe our active clustering algorithm, which can produce multiple clusterings of the same data according to user interest. At the core of our algorithm resides spectral clustering. Even though spectral clustering is widely researched, it has traditionally been used to produce a single clustering of a dataset. To our knowledge, we are the first to exploit spectral clustering to produce multiple clusterings of the same data. Interestingly, a spectral clustering algorithm *naturally* satisfies all three multi-clustering criteria that we desire: multiplicity, distinctivity, and quality. In the rest of this section, we first show that a small extension of a spectral clustering algorithm (namely, Shi and Malik's spectral clustering algorithm [17]) satisfies the three multi-clustering criteria and hence can be used to produce multiple clusterings. We then show how to incorporate human feedback into the spectral clustering algorithm to produce the clusterings that the user desires.

3.1 Spectral Learning and Multi-clustering

Many variants of spectral clustering have been proposed (e.g., [15, 17]). Here, we use Shi and Malik's spectral clustering algorithm [17], as it is widely used. We first show how to extend their algorithm to produce multiple clusterings of a dataset. More specifically, we propose an extension of their algorithm that takes a set of n documents, $X \in \Re^{n \times d}$, as input, and outputs m different document clusterings, where d is the number of unigrams, as defined in Section 2.[2] In addition, as spectral algorithms work in a matrix space, we introduce another notation. Let $s : X \times X \to \Re$ be a symmetric similarity function over X (i.e., $s(x_i, x_j) = s(x_j, x_i)$), and S be the similarity matrix that captures pairwise similarities (i.e., $S_{i,j} = s(x_i, x_j)$).

Before describing our algorithm, we define two concepts: *optimal* and *suboptimal* clusterings. Given a clustering algorithm with a predefined objective function o, the *optimal clustering* is the clustering that optimizes o. All other clusterings are *suboptimal clusterings*. Below we show how to learn the optimal clustering and several suboptimal clusterings using Shi and Malik's spectral algorithm.

Learning the optimal clustering. Spectral clustering employs a graph-theoretic notion of grouping. A set of n data points in an arbitrary feature space is represented as an undirected graph, where each node corresponds to a data point, and the edge weight between two nodes is their similarity as defined by S. The goal is to induce a clustering, or equivalently, a *partitioning function* f, which is typically represented as a vector of length n such that $f(i) \in \{1, -1\}$ indicates which of the two clusters data point i should be assigned to. Note that the cluster labels are interchangeable and can even be renamed without any loss of generality.

Normalized cut [17] is a widely used objective function in spectral clustering. Shi and Malik show that if we embed the optimization problem in the real domain (i.e, we allow f to be a real-valued vector rather than a binary-valued vector), then the normalized cut partition of X can be derived from the solution to the following constrained optimization problem:

$$\underset{f \in \Re^n}{\operatorname{argmin}} \sum_{i,j} S_{i,j} \left(\frac{f(i)}{\sqrt{d_i}} - \frac{f(j)}{\sqrt{d_j}}\right)^2 \qquad (1)$$

$$\text{subject to} \quad ||f||^2 = \sum_i D_{i,i} \text{ and } D^{-1/2} f \perp \mathbf{1}$$

where D is a diagonal matrix with $D_{i,i} = \sum_j S_{i,j}$ and $d_i = D_{i,i}$. It can be proved that the closed form solution to this optimization problem is the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix $L = D^{-1/2}(D - S)D^{-1/2}$ [17].[3] Clustering using the second eigenvector[4], $\mathbf{e}_2$, is trivial: we can just apply 2-means to the n data points represented by the second eigenvector [15].

Learning suboptimal clusterings. As mentioned before, suboptimal clusterings are useful if they can capture the user-desired clustering. Our algorithm for producing multiple suboptimal clusterings exploits a useful but rarely utilized fact: if we consider only those vectors that are orthogonal to $\mathbf{e}_2$ as candidate solutions to the constrained optimization problem above, then $\mathbf{e}_3$ is the solution. More generally, if our candidate solutions are restricted to those vectors that are orthogonal to the first n eigenvectors of L, then $\mathbf{e}_{n+1}$ is the solution. In other words, all $\mathbf{e}_n$ where $n > 2$ are *suboptimal* solutions to the minimization problem, with $\mathbf{e}_n$ being more suboptimal as n increases. Hence, we can apply 2-means to each of the eigenvectors, $\mathbf{e}_n$, separately to produce a suboptimal clustering. To our knowledge, *employing suboptimal partitioning functions to produce multiple clusterings is an unexplored idea*: existing work has focused on using only $\mathbf{e}_2$ (or a combination of $\mathbf{e}_2$ and other eigenvectors) to derive a *single* partition of the data; in contrast, we use each of the $\mathbf{e}_i$s (with $i \geq 2$) separately to produce *multiple* clusterings of the data.

So, our algorithm for producing multiple clusterings is simple: given data X and a symmetric similarity function s, we form the Laplacian L, compute its second through $(m+1)$-th eigenvectors, and apply 2-means to each of these m eigenvectors separately to produce m different clusterings.

Next, we show that our system satisfies each of the three multi-clustering criteria:

Multiplicity: Our algorithm produces m different clusterings, where $m > 1$, without changing the feature space and the similarity function. Hence, it satisfies multiplicity.

[2] As we only apply spectral clustering to produce 2-way clusterings, we will center our discussion on 2-way clusterings.

[3] Note that the solution to Equation (1) is only an approximation to the (discrete) normalized cut solution. Our definition of optimal and suboptimal clusterings refers to the continuous normalized cut objective.

[4] We refer to the nth smallest eigenvector of the Laplacian simply as the nth eigenvector, and denote it by $\mathbf{e}_n$.

Distinctivity: With some algebra, one can show that (1) L is symmetric when the similarity matrix S is symmetric, and (2) the eigenvectors of L are orthogonal to each other when L is symmetric. Since we always use a symmetric similarity function, S and L are symmetric. As a result, the eigenvectors of L are orthogonal to each other, and their similarity (obtained via the dot product) is zero. Since we employ the principal eigenvectors of L (starting from $\mathbf{e}_2$) as real-valued partitioning functions, the clusterings produced by these functions are distinctively different.[5]

Quality: As noted before, if we disallow the first n eigenvectors of L to be the solution to our optimization problem, then $\mathbf{e}_{n+1}$ is the solution. This implies that clustering using $\mathbf{e}_3$ achieves the next optimal objective value after $\mathbf{e}_2$, and more generally, clustering using $\mathbf{e}_{m+1}$ achieves the next optimal objective value after $\mathbf{e}_2, \ldots, \mathbf{e}_m$. Hence, $\mathbf{e}_3, \ldots, \mathbf{e}_{m+1}$ constitute the $(m-1)$-best suboptimal solutions that can be achieved by a spectral system. This gives us direct control over suboptimality: if we do not desire overly suboptimal solutions, we can simply put restrictions on m. In our experiments, we set m to 4, thus producing one optimal and three suboptimal clusterings for each dataset.[6]

3.2 Incorporating Human Feedback

So far, we have shown how to produce multiple clusterings ($C^i, i = 1\!:\!m$) of a dataset. To determine which of these m clusterings is the user-desired clustering, one can possibly have the user inspect the clusterings and decide which one corresponds most closely to the desired clustering. The main drawback associated with this kind of user feedback is that the user may have to inspect a large number of documents in order to make a decision. To reduce human effort, we employ an alternative procedure: we (1) identify the most informative unigrams characterizing each cluster, and (2) have the user inspect just these "features" rather than the documents.

To select these informative features, we rank them by their weighted log-likelihood ratio (WLLR):

$$P(w_i \mid C_j) \cdot \log \frac{P(w_i \mid C_j)}{P(w_i \mid \neg C_j)},$$

where w_i and C_j denote the ith feature and the jth cluster respectively, and each probability is add-one smoothed. Informally, feature w_i will have a high rank with respect to cluster C_j if it appears frequently in C_j and infrequently in $\neg C_j$. This correlates reasonably well with what we think an informative feature should be. Now, for each of the m partitions, we (1) derive the top 100 features for each cluster according to the WLLR, and then (2) present the ranked lists to the user. The user then selects the feature lists that are most relevant to her interest by inspecting as many features in the ranked lists as needed.

There is a caveat, however. The presence of a large number of *ambiguous* documents can adversely affect the identification of informative features, owing to the fact that many text documents are ambiguous with respect to the dimension along which they are clustered. For instance, many product reviews are sentimentally ambiguous (i.e., they contain both positive and negative sentiment-bearing words), as a reviewer often likes certain aspects of the product and dislikes the others. Hence, we remove the ambiguous documents before deriving informative features from a partition.

We employ a simple method for identifying unambiguous documents. In the computation of eigenvalues, each data point factors out the orthogonal projections of each of the other data points with which they have an affinity. Ambiguous points receive the orthogonal projections from both the positive and negative points, and hence they have near zero values in the pivot eigenvectors. In other words, the points with near zero values in the eigenvectors are more ambiguous than those with large absolute values. We therefore sort the data points according to their corresponding values in the eigenvector, and keep only the top $n/8$ and the bottom $n/8$ data points. We induce the informative features only from the resulting 25% of the data points, and present them to the user so that she can select the desired partition.[7]

In the event that the user identifies more than one eigenvector as relevant to the desired clustering dimension, we apply 2-means to re-cluster the n documents in the space defined by all of the human-selected eigenvectors. Below is the final algorithm.

```
Algorithm: Active-Spectral-Clustering
Input: Data $X$, Similarity Matrix $S$
Output: Clustering $C = \{C_1, C_2\}$
  1. Construct the Laplacian matrix $L = D^{-1/2}(D-S)D^{-1/2}$,
     where $D$ is a diagonal matrix with $D_{i,i} = \sum_j S_{i,j}$.
  2. Compute $E = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{m+1}\}$, the eigenvectors of
     $L$ that correspond to its $(m+1)$ smallest eigenvalues.
  3. For each $e_i \in E \setminus \{\mathbf{e}_1\}$, induce the top feature list.
  4. Ask the user to identify $E'$, the subset of $E$ that is relevant
     to the desired clustering dimension, by inspecting
     the top feature lists.
  5. Create the clustering $C = \{C_1, C_2\}$ by using 2-means
     to cluster the data points in the space defined by $E'$.
```

Note that this active clustering algorithm produces a single clustering of a dataset along the dimension that the user selects. If we want to use our algorithm to produce multiple clusterings of the same data along different dimensions, we just need to repeat steps 4 and 5 with a different set of eigenvectors chosen by the user for each intended dimension.

4. EVALUATION

4.1 Experimental Setup

Datasets. We employ five evaluation datasets.

Blitzer et al.'s book (BOO) and DVD datasets [4] each contains 1000 positive and 1000 negative customer reviews of books or movies, and can therefore be used to evaluate our algorithm's ability to cluster by *sentiment*. Since we desire that each evaluation dataset possesses at least two clustering dimensions, we also manually annotate each review with a *subjectivity* label that indicates whether it is "mostly subjective" (where the reviewer mainly expresses her sentiment) or

[5] Note that we only compare two partitioning functions (or, more precisely, two clusterings) in the continuous space. Their similarity might be different in the discrete space.

[6] We assume that the clusterings we are interested in can be captured using four eigenvectors. Note that using only up to $\mathbf{e}_5$ is by no means a self-imposed limitation of our algorithm, since we can employ as many eigenvectors as we desire.

[7] 25% is a somewhat arbitrary choice. Additional experiments revealed that the list of top-ranked features is not particularly sensitive to slight changes to the number of unambiguous documents used in the feature induction process.

"mostly objective" (where the reviewer focuses on describing the content of the book or the movie). Details of the annotation process are described later in this subsection.

The MIX dataset is a 4000-document dataset consisting of the 2000 BOO reviews and the 2000 DVD reviews, as described above. We can therefore cluster these reviews by *topic* (i.e., book or DVD), *sentiment* or *subjectivity*.

Schler et al.'s MAN dataset [13] contains 19,320 blog posts. We randomly selected 1000 blog postings, half of which were written by males and half by females. We can therefore cluster these blog posts by the author's *gender*. Since the author's *age* information is also available in each blog post, we can also cluster them by age. To do so, we automatically generate a 2-way partitioning of the documents by imposing an age threshold of 25. Specifically, the 932 documents written by bloggers aged below 25 are marked as *young*, and the remaining 1068 are marked as *old*.

Our own POA dataset consists of 2000 political articles written by columnists, 1000 of whom identified themselves as *Republicans* and the remaining 1000 identified themselves as *Democrats*.[8] Hence, we can cluster these articles by the author's *political affiliation*. We also create a second clustering dimension by annotating each article as either *foreign* or *domestic*, depending on the policy that the article discusses. For example, the policy on the Iraq war is foreign, whereas the policy on regulating the job market is domestic.

Table 1 summarizes the dimensions along which the documents are annotated. Note that each of the seven distinct dimensions yields a 2-way partitioning of the documents: (1) Sentiment (positive/negative); (2) Subjectivity (subjective/objective); (3) Topic (book/DVD); (4) Gender (man/woman); (5) Age (young/old); (6) Political affiliation (Democrat/Republican); and (7) Policy (domestic/foreign).

Human annotation. As mentioned above, we need to annotate the BOO, DVD, and MIX datasets with respect to *Subjectivity* and POA with respect to *Policy*.[9] We had two computer science graduate students independently annotate the documents. For POA, we asked them to use commonsense knowledge to annotate each document with respect to the policy that the article discusses. If both foreign and domestic policies are discussed in the same article, we asked them to assign the label based on the one that is discussed more frequently. On the other hand, given a BOO or DVD review, we asked them to first label each of its sentences as subjective or objective; if a sentence contains both subjective and objective materials, its label should reflect the type of material that appears more frequently. The review is then labeled as subjective (objective) if more than half of its sentences are labeled as subjective (objective).

The inter-annotator agreement rate in terms of Cohen's κ is 0.774 (BOO), 0.796 (DVD), and 0.820 (POA), indicating substantial agreement. The annotators examined each case on which they disagreed and decided on the final label together. They reported that essentially all disagreements arose from labeling the ambiguous data points (e.g., articles that discuss both foreign and domestic policies, and sentences that contain both subjective and objective materials). In the end, we obtained 1205 subjective and 795 objective documents for BOO, 1124 subjective and 876 objective documents for DVD, and 875 foreign and 1125 domestic documents for POA. To stimulate research, we make these annotations publicly available.[10]

	Dimension 1	Dimension 2	Dimension 3
BOO	Sentiment	Subjectivity	–
DVD	Sentiment	Subjectivity	–
MIX	Topic	Sentiment	Subjectivity
MAN	Gender	Age	–
POA	Political Affiliation	Policy	–

Table 1: Clustering dimensions for the five datasets.

Document preprocessing. To preprocess a document, we first tokenize and downcase it, and then represent it as a vector of unstemmed unigrams, each of which assumes a value of 1 or 0 that indicates its presence or absence in the document. In addition, we remove from the vector punctuations, numbers, words of length one, and words that occur in only a single document. Following the common practice in the information retrieval (IR) community, we also exclude words with a high document frequency, many of which are stopwords or domain-specific general-purpose words. Details of this preprocessing step can be found in Dasgupta and Ng [6]. We compute the similarity between two documents by taking the dot product of their feature vectors.

Evaluation metrics. We employ two evaluation metrics. First, we report results for each dataset in terms of accuracy, which is the fraction of documents for which the label assigned by our system is the same as the gold-standard label.[11] Second, following Kamvar et al. [12], we evaluate the clusters produced by our approach against the gold-standard clusters using the Adjusted Rand Index (ARI) [10]. ARI is the adjusted-for-chance form of the Rand Index, which computes the pairwise accuracy given two partitions. ARI ranges from -1 to 1; better clusterings have higher values.

4.2 Baseline Systems

Clustering using the second eigenvector only. As our first baseline, we adopt the commonly-used approach introduced by Shi and Malik [17] and cluster the reviews using only the second eigenvector, $\mathbf{e}_2$, which induces the partitioning that is optimal with respect to spectral clustering's objective function, as described previously. Results of this baseline, reported in terms of accuracy and ARI, are shown in row 1 of Table 2 and Table 3, respectively.[12] Since this method can propose only one clustering per dataset but each dataset contains at least two gold-standard clusterings (one for each dimension), the results are obtained by comparing this proposal clustering against each of the gold-standard clusterings. As we can see, accuracy ranges from 52.9 to 77.1, and ARI ranges from 0.003 to 0.291. Note that these and other results involving 2-means are averaged over ten independent runs owing to the randomness in seed selection.

Non-negative Matrix Factorization. As our second base-

[8] These articles were chosen randomly among those written in 2006 from http://www.commondreams.org/archives.

[9] Note that the subjectivity labels for MIX can be derived from BOO and DVD.

[10] http://www.utdallas.edu/∼sajib/multi-clusterings.html

[11] Since a human also labels each cluster by inspecting the induced features, we are able to compute accuracy.

[12] For each dataset shown in Tables 2 and 3, Dim$n$ refers to the nth dimension listed for the dataset in Table 1. For instance, Dim1 and Dim2 of BOO correspond to Sentiment and Subjectivity, respectively.

		BOO		DVD		MIX			MAN		POA	
	System Variation	Dim1	Dim2	Dim1	Dim2	Dim1	Dim2	Dim3	Dim1	Dim2	Dim1	Dim2
1	2nd eigenvector	58.9	58.6	55.3	61.0	**77.1**	52.9	**66.6**	64.2	62.4	54.2	63.1
2	NMF	52.1	57.8	50.3	60.5	69.2	51.7	58.6	55.6	58.0	53.0	62.8
3	Meta clustering	50.8	51.2	53.9	**71.0**	50.2	50.2	58.6	51.2	53.6	59.4	58.8
4	Feature removal	58.9	63.2	51.2	60.5	**77.1**	50.0	51.0	64.2	57.8	57.8	63.1
5	Our approach	**69.5**	**63.8**	**70.7**	60.5	**77.1**	**68.9**	59.7	**66.4**	**62.9**	**69.7**	**76.3**

Table 2: Results in terms of accuracy for the five datasets. The best result for each dimension is boldfaced.

		BOO		DVD		MIX			MAN		POA	
	System Variation	Dim1	Dim2	Dim1	Dim2	Dim1	Dim2	Dim3	Dim1	Dim2	Dim1	Dim2
1	2nd eigenvector	0.031	0.027	0.011	0.044	**0.291**	0.003	**0.109**	0.080	0.061	0.007	0.066
2	NMF	0.002	−0.03	0.000	0.005	0.147	0.000	0.026	0.012	0.022	0.003	0.063
3	Meta clustering	0.000	−0.04	0.006	**0.155**	0.000	0.000	0.021	0.001	0.003	0.035	0.023
4	Feature removal	0.031	0.07	0.000	0.044	**0.291**	−0.03	0.010	0.080	0.024	0.024	0.066
5	Our approach	**0.152**	**0.074**	**0.171**	0.044	**0.291**	**0.142**	0.036	**0.107**	**0.065**	**0.155**	**0.276**

Table 3: Results in terms of ARI for the five datasets. The best result for each dimension is boldfaced.

line, we use Non-negative Matrix Factorization (NMF), which has recently been demonstrated to be more effective than Latent Semantic Analysis (LSA) [8] for document clustering [20]. As in the first baseline, we compare the clustering proposed by NMF against each of the gold-standard clusterings for each dataset. Since the algorithm involves choosing seeds at random, the NMF results shown in row 2 of Tables 2 and 3 are averaged over ten independent runs. Despite its algorithmic sophistication, NMF performs consistently worse than the first baseline in terms of both accuracy and ARI.

Meta clustering. Since our algorithm produces multiple clusterings, it is desirable to have a baseline that also produces multiple clusterings. However, many algorithms that produce multiple clusterings operate in a semi-supervised setting [7, 9]. The notable exceptions are Caruana et al.'s meta clustering algorithm [5] and Jain et al.'s approach [11]. Since some of our datasets can be clustered in three different ways but Jain et al.'s approach produces only two clusterings for a given dataset, we evaluate meta clustering only. As mentioned before, meta clustering produces multiple clusterings of the same data by running k-means multiple times, each time with a random selection of seeds and a random weighting of features. We produce multiple clusterings for each dataset by running this algorithm 100 times and report in row 3 of Tables 2 and 3 the *best* result obtained for each dimension of each dataset.[13] Even though the best results are reported, meta clustering underperforms the first two baselines for all but two dimensions (DVD/Dim2 and POA/Dim1). The poor performance can be attributed to the fact that k-means is generally a weaker clustering algorithm than its more recently developed counterparts.

Iterative feature removal. We designed another simple baseline for producing multiple clusterings. Given a dataset, we (1) apply spectral clustering to produce a 2-way clustering using the second eigenvector, and then (2) remove from each cluster the informative features that are identified using WLLR. To produce another clustering, we repeat these two steps, but without the features removed in step 2. Hence, we can generate as many clusterings as we want by repeating these two steps, each time with a smaller number of features. The motivation behind this algorithm is that by repeatedly removing the informative features from a partition and re-clustering, we can potentially yield different clusterings. To obtain the results of this algorithm for each dataset in row 4 of Tables 2 and 3, we (1) run it for m iterations to produce m clusterings, where m is the number of dimensions the dataset possesses; and (2) find the bipartite matching between the proposal clusterings and the gold-standard clusterings that has the highest average accuracy/ARI. Since we need to specify the number of features to remove from each cluster in each iteration, we tested values from 100 to 5000 in steps of 100, reporting the *best* result.

As we can see, except for BOO/Dim2 and POA/Dim1, this algorithm never surpasses the performance of the first baseline. One reason for its poorer performance can be attributed to the fact that the informative features for the different dimensions of a dataset are *not* disjoint. For example, *terrific* is likely to be an informative feature when clustering by sentiment and by subjectivity, but since this algorithm removes informative features in each iteration, *terrific* will only be accessible to one but not both clustering dimensions.

4.3 Our Active Clustering Algorithm

Human experiments. An important step in our algorithm involves having a user identify the dimensions along which she wants to cluster the documents by inspecting the features. To evaluate the feasibility of this step, we performed the experiment independently with ten humans (all of whom are computer science graduate students not involved in data annotation) and computed the agreement rate.

Specifically, for each dataset, we generated four clustering dimensions using the second through fifth eigenvectors of the Laplacian. After that, we showed the human judges the top 100 features for each cluster of each dimension according to WLLR (see Tables 4 and 5 for a snippet, where the dimension and cluster labels are manually assigned by the majority of the judges). To mimic the realistic situation where a user of our algorithm knows *a priori* the dimension(s) along which she wants to cluster the documents, we informed each judge of the dimensions she was expected to identify: for example, for BOO, she was told to identify the Sentiment and Subjectivity dimensions. To ensure that

[13] These results are obtained using a publicly-available implementation of meta clustering, which is available at http://www.cs.cornell.edu/∼nhnguyen/metaclustering.htm.

DVD				MIX			
e_2	e_3	e_4	e_5	e_2	e_3	e_4	e_5
Subjective	**Positive**	**C_1**	**C_1**	**Book**	**Subjective**	**Positive**	**C_1**
fan	wonderful	video	saw	reader	bought	music	loved
bought	music	music	watched	information	disappointed	wonderful	children
money	collection	workout	loved	example	information	excellent	enjoyed
video	excellent	found	series	subject	reviews	highly	wonderful
waste	quality	videos	season	important	waste	collection	novel
quality	cast	bought	fan	nature	recipes	special	mother
reviews	extras	moves	comedy	text	workout	video	bought
series	song	doing	family	provides	video	classic	child
buying	special	kids	enjoy	science	boring	disc	parents
worth	highly	right	whole	human	easy	version	novels
Objective	**Negative**	**C_2**	**C_2**	**DVD**	**Objective**	**Negative**	**C_2**
young	money	series	quality	music	men	boring	version
between	waste	fan	money	actors	young	waste	quality
director	thought	cast	video	watched	dirctor	novel	waste
played	worst	comedy	sound	script	cast	pages	original
cast	boring	stars	picture	films	scene	worst	sound
role	actually	actors	version	loved	role	felt	edition
place	saw	episodes	waste	saw	actors	person	disc
performance	maybe	original	found	video	films	disappointed	features
actors	nothing	season	disk	horrible	plays	finish	review
men	felt	set	transfer	tv	war	reviews	video
Subjectivity	**Sentiment**			**Topic**	**Subjectivity**	**Sentiment**	

Table 4: Top ten features induced for each dimension for the DVD and MIX datasets. The shaded columns correspond to the dimensions selected by the judges. e_2, ..., e_5 are the top eigenvectors; C_1 and C_2 are the clusters.

POA				MAN			
e_2	e_3	e_4	e_5	e_2	e_3	e_4	e_5
Foreign	**Foreign**	**Foreign**	**Republican**	**Woman**	**Woman**	**C_1**	**Young**
muslim	iran	oil	voters	omg	blonde	faith	ur
israel	iraqi	growth	conservative	lol	danced	deeply	wmd
islamic	forces	rate	gop	haha	wore	happiness	ashcroft
islam	israeli	food	win	sooo	worn	emotions	qaeda
muslims	israel	poverty	polls	bye	romantic	prayed	haha
jews	weapons	living	candidates	hahaha	dancing	strength	omg
israeli	east	average	hilary	hehe	dresses	existence	iraqi
peace	nuclear	rates	kerry	soo	porch	actions	voters
religion	region	economy	clinton	ppl	dall	soul	ppl
saddam	regime	god	conservatives	wanna	lips	courage	ghraib
Domestic	**Domestic**	**Domestic**	**Democrat**	**Man**	**Man**	**C_2**	**Old**
tax	god	court	agency	policies	issues	ebay	kitchen
economy	love	constitutional	information	voters	linux	dvd	bedroom
spending	kids	supreme	companies	democracy	orkut	cable	laundry
income	church	judiciary	department	opposition	xml	upgrade	meal
corporate	im	intelligence	justice	coverage	developers	browser	dishes
jobs	myself	committee	warrant	networks	software	xp	delicious
taxes	person	presidential	criminal	policy	debian	users	grocery
budget	parents	constitution	investigation	credibility	rss	software	cozy
rates	folks	nsa	legal	governments	interface	feature	wrapped
prices	family	judicial	documents	federal	mozilla	camera	salad
Policy			**Political Aff.**	**Gender**			**Age**

Table 5: Top ten features induced for each dimension for the POA and MAN datasets. The shaded columns correspond to the dimensions selected by the judges. e_2, ..., e_5 are the top eigenvectors; C_1 and C_2 are the clusters.

the judges have a consistent understanding of the clustering dimensions, we explained to them the seven clustering dimensions shown in Table 1 with definitions and examples prior to the experiment. (We omitted them here due to space limitations.) Also, we told them that a clustering dimension could be captured by multiple eigenvectors. If they determined that more than one eigenvector was relevant to a dimension, they were instructed to rank the eigenvectors in terms of their degree of relevance, where the most relevant one would appear first in the list. Finally, they were given the option of not choosing any eigenvector for a given dimension if it was not possible for them to do so. Note that while only the top ten features are shown in Tables 4 and 5, they based their judgment on the top 100 features.

After the judges completed the experiment, we (1) selected for each dimension the largest set of eigenvectors that

	Dimension 1	Dimension 2	Dimension 3
BOO	4 (100%)	3 (70%)	–
DVD	3 (100%)	2 (90%)	–
MIX	2 (100%)	4 (80%)	3 (100%)
MAN	2,3 (70%)	5 (60%)	–
POA	5 (40%)	2,3,4 (80%)	–

Table 6: Human-selected eigenvectors and the agreement rate for the five datasets.

was ranked first by the majority of the judges, and (2) computed the agreement rate as the percentage of judges who assigned the highest rank to the eigenvector set obtained in step 1. Results are shown in Table 6, where the agreement rate is shown in parentheses. As we can see, reasonably high agreement is achieved for all dimensions in all datasets except Political Affiliation (Dimension 1 of POA) and Age (Dimension 2 of MAN). An inspection of the feature lists induced for POA and MAN (see Table 5) reveals that they are fairly noisy, which makes it difficult to identify these two dimensions. In fact, many judges could not find any relevant eigenvector for these dimensions.

Overall, these results, together with the fact that a judge took 8–9 minutes on average to identify each dimension, indicate that asking a human to identify the intended dimension(s) based on the induced features is a viable task.

Clustering results. Next, we cluster the documents in each dataset along each dimension using the eigenvectors selected by the majority of the judges. If more than one eigenvector is selected for a dimension, the documents will be clustered using 2-means in the space defined by all of the selected eigenvectors. The clustering results are shown in row 5 of Tables 2 and 3. As we can see, our clustering algorithm frequently outperforms the four baselines by a large margin. These results substantiate our claim that our algorithm can cluster documents along multiple dimensions according to user interest. In addition, clustering accuracies are generally higher for the topic-related dimensions (e.g., Book vs. DVD and Domestic vs. Foreign) than the other dimensions (e.g., Gender, Age, Sentiment). This should not be surprising: non-topic-based classification tasks can be difficult even for supervised systems that are trained on a large amount of labeled data (e.g., [13]).

5. CONCLUSIONS

Overall, we believe that our work on subjectifying text clustering makes three contributions:

Generation of multiple clusterings. With a few exceptions (e.g., [5, 11]), existing clustering algorithms can only produce a *single* clustering of a dataset along its most prominent dimension. In contrast, our algorithm can produce *multiple* clusterings of the same data according to user interest without using any labeled data or considerable feedback.

Interactivity in IR. The active clustering algorithm that we proposed allows for more user interactivity in an easy, low effort manner. Perhaps the implications of our work for interactivity in IR are even more important: we believe this and other interactive algorithms that allow the user to make small, guiding tweaks, and thereby get results better than would otherwise be possible is the future of IR.

Improved understanding of spectral clustering. While spectral clustering has been employed for many years to produce a single clustering of a dataset, to our knowledge we are the first to empirically demonstrate that the top eigenvectors of the Laplacian can be used in isolation or in combination to produce semantic clusterings.

6. ACKNOWLEDGMENTS

We thank the three anonymous reviewers for their invaluable comments on an earlier draft of the paper. This work was supported in part by NSF Grant IIS-0812261.

7. REFERENCES

[1] M.-F. Balcan and A. Blum. Clustering with interactive feedback. In *Proc. of ALT*, pages 316–328, 2008.

[2] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proc. of SDM*, pages 333–344, 2004.

[3] R. Bekkerman, H. Raghavan, J. Allan, and K. Eguchi. Interactive clustering of text collections according to a user-specified criterion. In *Proc. of IJCAI*, pages 684–689, 2007.

[4] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. of the ACL*, pages 440–447, 2007.

[5] R. Caruana, M. F. Elhawary, N. Nguyen, and C. Smith. Meta clustering. In *Proc. of ICDM*, pages 107–118, 2006.

[6] S. Dasgupta and V. Ng. Topic-wise, sentiment-wise, or otherwise? Identifying the hidden dimension for unsupervised text classification In *Proc. of EMNLP*, pages 580–589, 2009.

[7] I. Davidson and Z. Qi. Finding alternative clusterings using constraints. In *Proc. of ICDM*, pages 240–249, 2007.

[8] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of American Society of Information Science*, 41(6):391–407, 1990.

[9] D. Gondek and T. Hofmann. Non-redundant data clustering. In *Proc. of ICDM*, pages 75–82, 2004.

[10] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[11] P. Jain, R. Meka, and I. S. Dhillon. Simultaneous unsupervised learning of disparate clusterings. In *Proc. of SDM*, pages 858–869, 2008.

[12] S. Kamvar, D. Klein, and C. Manning. Spectral learning. In *Proc. of IJCAI*, pages 561–566, 2003.

[13] J. Schler, M. Koppel, S. Argamon, and J. Pennebaker. Effects of age and gender on blogging. In *AAAI Symposium on Computational Approaches for Analyzing Weblogs*, 2006.

[14] B. Liu, X. Li, W. S. Lee, and P. S. Yu. Text classification by labeling words. In *Proc. of AAAI*, pages 425–430, 2004.

[15] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in NIPS*, pages 849–856, 2001.

[16] H. Raghavan and J. Allan. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proc. of SIGIR*, pages 79–86, 2007.

[17] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[18] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *Proc. of ICML*, pages 577–584, 2001.

[19] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. Distance metric learning with application to clustering with side-information. In *Advances in NIPS*, pages 505–512, 2002.

[20] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proc. of SIGIR*, pages 267–273, 2003.