Mining Clustering Dimensions

Sajib Dasgupta SAJIB@HLT.UTDALLAS.EDU
Vincent Ng VINCE@HLT.UTDALLAS.EDU
Human Language Technology Research Institute, University of Texas at Dallas, Richardson, TX 75083 USA

Abstract

Many real-world datasets can be clustered along multiple dimensions. For example, text documents can be clustered not only by topic, but also by the author's gender or sentiment. Unfortunately, traditional clustering algorithms produce only a single clustering of a dataset, effectively providing a user with just a single view of the data. In this paper, we propose a new clustering algorithm that can discover in an unsupervised manner each clustering dimension along which a dataset can be meaningfully clustered. Its ability to reveal the important clustering dimensions of a dataset in an unsupervised manner is particularly appealing for those users who have no idea of how a dataset can possibly be clustered. We demonstrate its viability on several challenging text classification tasks.

1. Introduction

Many real-world datasets can be naturally clustered along multiple *dimensions*. For instance, speech data can be clustered by the gender of the speaker or the language used by the speaker; 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, romantic, documentary), sentiment (e.g, positive, negative), or even the main actors/actresses. In some data mining applications, it is desirable to recover as many clusterings of a dataset along its important *clustering dimensions* as possible.

A natural question is: given data X, is it possible to *discover* in an unsupervised manner each dimension along which X can be *meaningfully* clustered? By a meaningful clustering, we mean a clustering that is both human interpretable and qualitatively strong in terms of basic qualitative criteria typically used to evaluate the *structure* of a clustering. The ability of a clustering algorithm to discover multiple clustering dimensions is particularly appealing for a user who may not know how a dataset can possibly be clustered. Even if the user knows how she wants to cluster the data, it would still be desirable if an algorithm can unveil clustering dimensions that she is not previously aware of, but may also be of interest to her.

Unfortunately, not only do traditional clustering algorithms fail to produce multiple clusterings[1] of a dataset, the only clustering they produce may not be the one that 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 employed by the algorithm is optimized. However, the optimal clustering might not be deemed fit by an end user, as she may be interested in a clustering that is different from the optimal clustering.

One may argue that it is possible to design the feature space differently to produce different clusterings of a dataset. This typically involves having a user identify a set of features that are relevant to a particular clustering task (Liu et al., 2004). However, manually identifying the "right" set of features is both time-consuming and knowledge-intensive, and may even require a lot of domain expertise. To overcome this weakness, researchers have attempted to cluster in a semi-supervised setting (e.g., constrained clustering (Wagstaff et al., 2001; Bilenko et al., 2004)) and learn a similarity metric from *side* information (Xing et al., 2002). Note that these approaches work under the assumption that the user knows each of the plausible clustering dimensions *a priori*, and "communicates" her intention to the clustering algorithm by designing

Appearing in *Proceedings of the 27^{th} International Conference on Machine Learning*, Haifa, Israel, 2010. Copyright 2010 by the author(s)/owner(s).

[1] For brevity, we henceforth use the term *multiple clusterings* to refer to *multiple meaningful clusterings*.

Dimension1	**Dimension2**	**Dimension3**
Book	Subjective	Positive
reader	bought	wonderful
information	workout	excellent
research	recipes	music
important	information	highly
text	disappointed	collection
DVD	Objective	Negative
music	young	boring
script	men	waste
actors	scene	novel
films	cast	worst
comdey	role	pages

Table 1. Three clustering dimensions for the BOOK-DVD dataset that are induced by our clustering algorithm.

the feature space and/or constraints for each clustering dimension accordingly. In contrast, we work in a setting where the user has little or no prior knowledge about the plausible clustering dimensions, and our goal is to help users identify and possibly visualize each of the clustering dimensions latent in the data.

In this paper, we propose a text clustering algorithm that can *induce* and *visualize* multiple clustering dimensions latent in a text collection without using any prior knowledge or supervision. Our main contribution lies in our demonstration of the feasibility to use *a single feature space* and *a single clustering algorithm* with *a single similarity metric* and *a single objective function* to produce multiple clusterings of a given dataset. The key idea behind our approach is to produce multiple *suboptimal* clusterings along the prominent clustering dimensions of a dataset on top of the *optimal* clustering obtained by optimizing the objective function. Specifically, our clustering algorithm 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; and (2) *representing* each clustering dimension by a (small) number of automatically chosen words, which help the user visualize and subsequently select the dimension(s) along which she wants to cluster the documents. Experimental results are very promising: our algorithm is able to produce multiple clusterings along induced dimensions with reasonable accuracies on several challenging text classification tasks.

Topic 1:	book read pages information chapter cover
Topic 2:	god christian bible jesus faith spiritual christ
Topic 13:	music live song great songs band show put
Topic 22:	bad worst acting dialogue terrible absolutely
Topic 33:	movie cast role performance actor plays
Topic 43:	work writing style fiction read writer works
Topic 69:	war battle german american men military
Topic 89:	love wonderful time loved enjoy heart list

Table 2. Selected topics induced by the Latent Dirichlet Allocation model for the BOOK-DVD dataset.

As a concrete example, we show in Table 1 three clustering dimensions that are induced by our algorithm for a dataset containing book and DVD reviews. Here, a clustering dimension corresponds to a 2-way clustering, and is represented by the top unigrams automatically extracted from each of the two clusters involved in the dimension. By inspecting the unigrams, it may be possible for a user to realize that this data can be clustered by topic (Book or DVD), sentiment (Positive or Negative), or subjectivity (Subjective or Objective). It is worth mentioning that our goal is fundamentally different from that of *topic modeling* (Blei et al., 2003): while a topic model attempts to discover latent topics from a set of documents, we attempt to discover latent clustering dimensions (compare Table 1 and 2). Nevertheless, the two models bear resemblance to each other: not only are both models unsupervised, they both display the learned information to the user using representative words. We believe that the impact of our work goes beyond text clustering: it can potentially enhance the capability of exploratory text analysis and summarization algorithms for the unsupervised discovery of information from a text collection.

The rest of the paper is organized as follows. Section 2 discusses related work on producing multiple clusterings. Section 3 describes our clustering algorithm. We present evaluation results in Section 4 and summarize our conclusions in Section 5.

2. Related Work

Previous work on inducing clustering dimensions has focused on producing multiple clusterings of a dataset, and can be broadly divided into two categories.

Semi-supervised methods. These methods are semi-supervised in the sense that one of the clusterings is provided (by the human) as input, and the goal is to produce another clustering that is distinctively different from the given one. For instance, Gondek & Hofmann's (2004) approach 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. It 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 & Qi (2007) first learn a distance metric D_C from the original clustering C, and then reverse 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 new clustering.

Unsupervised methods. Here, each of the possible clusterings is produced without using any labeled data. Meta clustering (Caruana et al., 2006) is an approach that produces multiple clusterings of a dataset by running k-means multiple times, each time with a random selection of seeds and a random weighting of features. Its goal is to present each local minimum found by k-means as a possible clustering. This approach has two weaknesses. First, many of these local minima are qualitatively poor. 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 (2008) approach is more sophisticated, as it learns two clusterings in a "decorrelated" k-means framework. Its joint optimization model aims to achieve typical k-means objectives and at the same time ensures that the two induced clusterings are distinctively different. Note that Jain et al. use this framework to produce only two clusterings of a dataset, as the objective function becomes too convoluted to allow more clusterings.

Before moving on to the details of our clustering algorithm, we describe the primary differences between our algorithm and the aforementioned approaches. First, our algorithm neither uses labeled data nor assumes the existence of a human-supplied clustering, unlike the semi-supervised models. Second, while we employ spectral clustering, none of the existing approaches do. To our knowledge, we are the first to exploit spectral clustering to produce multiple clusterings of a dataset. Finally, none of the aforementioned approaches are intended to provide visualization of the induced clustering dimensions, which is a key goal in our work.

3. Our Approach

As mentioned before, our ultimate goal is to induce and visualize the dimensions along which a dataset can be meaningfully clustered. To this end, we first describe an algorithm that produces multiple clusterings along the distinct dimensions of the data. Then, to visualize a clustering dimension, we show how to represent it using a small number of features that are automatically selected from each clustering produced in the first step.

3.1. Problem Formulation

Let us begin by introducing some notation. Let $X = \{x_1, \ldots, x_n\}$ be a set of n data points to be clustered, where each point $x_i, i = 1:n$ is represented by d features $w_1, w_2, \ldots, w_d$. Let $s : X \times X \rightarrow \Re$ be a similarity function over X, and S be a similarity matrix that captures pairwise similarities (i.e., $S_{i,j} = s(x_i, x_j)$). We desire a clustering algorithm G that can learn m ($m > 1$) different partitioning functions $f_i, i = 1:m$ that correspond to m different 2-way clusterings $C^i = \{C_1^i, C_2^i\}, i = 1:m$ such that $C_1^i \cup C_2^i = X$ and $C_1^i \cap C_2^i = \phi$.[2] Specifically, a partitioning function f assigns a cluster label to each of the n data points in X, and is typically represented as a vector of length n such that $f(i) \in \{1, -1\}$ indicates which of the two clusters contains data point i. G is associated with an objective function, $o : C \rightarrow \Re$, which assigns a qualitative score to each clustering.

To produce multiple clusterings, we require a clustering algorithm to satisfy two important properties:

(1) Each clustering $C^i, i = 1\!:\!m$ produced by the clustering algorithm should be *distinctively different*. By distinctively different, we mean that two clusterings are highly dissimilar w.r.t. some measure for comparing clusterings. More formally, if ψ is a non-negative function that measures the similarity between two partitioning functions, then $\forall_{i,j}\ \psi(f_i, f_j) \approx 0$. Note that distinctivity is a crucial property in the existing approaches that also aim to produce multiple clusterings (Davidson & Qi, 2007; Jain et al., 2008).

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

3.2. Achieving Multi-Clusterability

Next, we describe our algorithm for producing multiple clusterings of a dataset. At the core of our system resides spectral clustering. Although spectral clustering is widely researched, it has been traditionally 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 a dataset. As we will see, spectral clustering algorithm *naturally* satisfies the aforementioned distinctivity and quality criteria. Many variants of spectral clustering have been proposed. Here, we use Shi & Malik's (2000) spectral clustering algorithm, as it is widely used.

Our algorithm is unique in its use of spectral clustering to produce multiple *suboptimal* clusterings along distinct dimensions on top of the *optimal* clustering. Our key hypothesis is that suboptimal clusterings may reveal important clustering dimensions of a dataset. Below we show how to learn the optimal clustering and suboptimal clusterings using Shi & Malik's algorithm.

[2] Note that our algorithm can be extended fairly easily to produce k-way ($k > 2$) clusterings.

In spectral clustering, a set of n data points X 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 of spectral clustering is to induce a partitioning function obtained by optimizing an objective that typically involves maximizing within-cluster similarity and inter-cluster dissimilarity. The partitioning function that optimizes the objective is the optimal partitioning function. All other partitioning functions are suboptimal.

Learning the optimal partitioning function. Normalized cut (Shi & Malik, 2000) is a widely used objective function in spectral clustering. Note that finding the optimal normalized cut solution is NP-hard when f is constrained to be discrete (i.e., $f \in \{1, -1\}$). However, if we relax the optimization problem by allowing f to be continuous (i.e., $f \in \Re$), the normalized cut partition of X can be derived from the solution to the following constrained optimization problem:

$$\arg \min_{f \in \Re^n} \sum_{i,j} S_{i,j} (\frac{f(i)}{\sqrt{d_i}} - \frac{f(j)}{\sqrt{d_j}})^2 \qquad (1)$$

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

where D is a diagonal matrix with $D_{i,i} = \sum_j S_{i,j}$ and d is a n-dimensional vector with $d_i = D_{i,i}$. It can be proved that the closed form solution to this optimization problem is $\mathbf{e}_2$, the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix $L = D^{-1/2}(D - S)D^{-1/2}$ (Shi & Malik, 2000).[3] Given that f=$\mathbf{e}_2$, we discretize f to produce a 2-way clustering of X by applying 2-means to the n data points represented by $\mathbf{e}_2$ (Ng et al., 2001). Note that $\mathbf{e}_2$ is only an approximation to the (discrete) normalized cut solution. Our definition of optimal and suboptimal clusterings refers to the continuous normalized cut objective as defined in (1).

Learning suboptimal partitioning functions. Suboptimal clusterings would be useful if they can reveal distinct clustering dimensions of the data. Our algorithm for producing multiple suboptimal partitioning functions is simple: we put progressively more constraints on the solution space in our constrained optimization problem. For example, if we add the constraint $f \perp \mathbf{e_2}$, we obtain a new partitioning function f', which captures the normalized cut partition that is orthogonal to $\mathbf{e}_1$ and $\mathbf{e}_2$.[4] Similar to the derivation of the optimal partitioning function, one can show that f'=$\mathbf{e}_3$. 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 (Shi & Malik, 2000). In other words, except $\mathbf{e}_2$, all eigenvectors of L are *suboptimal* solutions to the optimization problem, with $\mathbf{e}_n$ being more suboptimal as n increases. Hence, we can produce a suboptimal clustering by applying 2-means to each $\mathbf{e}_n$ separately. 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.

To put it in a nutshell, our algorithm produces multiple clusterings as follows: given data X and a similarity function s, we form the Laplacian L, compute the second through $(m + 1)$-th eigenvectors of L, and apply 2-means to each of these m eigenvectors to produce m different clusterings. Interestingly, spectral learning naturally ensures that each of these m clusterings are distinctively different and qualitatively strong:

Distinctivity: Note that we employ the principal eigenvectors of L as real-valued partitioning functions. If $f_i, i = 1 : m$ is our set of m partitioning functions where f_1 is the most optimal and f_m is the least optimal, then f_i=$\mathbf{e}_{i+1}$. 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 employ a symmetric similarity measure to compute the similarity between two data points, S and L are symmetric. As a result, the eigenvectors of L are orthogonal to each other. This gives us direct proof of the distinctivity of each partitioning function. For example, if we use the squared dot product, ψ, to compute the similarity between two partitioning functions, then we can show that $\forall_{i,j} \ \psi(f_i, f_j) = (f_i^T f_j)^2 = (\mathbf{e}_{i+1}^T \mathbf{e}_{j+1})^2 = 0$. Hence, our algorithm satisfies the distinctivity constraint.[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 the partitioning function corresponding to $\mathbf{e}_3$ is the next optimal solution that is orthogonal to $\mathbf{e}_2$, and the partitioning function corresponding to $\mathbf{e}_4$ is the next optimal solution that is orthogonal to $\mathbf{e}_2$ and

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

[4] Note that the constraint $f \perp D^{1/2}\mathbf{1}$ in the problem ensures that the solution is always orthogonal to $\mathbf{e}_1$.

[5] Note that we compare two partitioning functions in the continuous space. Their similarity might be different in the discrete space.

$\mathbf{e}_3$. Hence, each of the $m-1$ suboptimal partitioning functions is the "next best" orthogonal solution that can be achieved by a spectral system. In other words, they are the closest to optimal partitioning function w.r.t. the objective function. Hence, if m is reasonably small, then each of the $m-1$ suboptimal clusterings are qualitative strong. 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, producing one optimal and three suboptimal clusterings for each dataset.[6]

From a modeling point of view, it is not easy to design a clustering algorithm that can produce multiple clusterings of a dataset and satisfy both distinctivity and quality, as also demonstrated by the related work discussed in Section 2. For example, Jain et al. (2008) learn two clusterings C^1 and C^2 with k_1 and k_2 clusters respectively in a "decorrelated" k-means framework, by proposing the following objective function:

$$\sum_{i=1}^{k_1} \sum_{x \in C_i^1} ||x-\mu_i||^2 + \sum_{j=1}^{k_2} \sum_{x \in C_j^2} ||x-\nu_j||^2$$

$$+\lambda \sum_{i,j} (\beta_j^T \mu_i)^2 + \lambda \sum_{i,j} (\alpha_i^T \nu_j)^2$$

where α_i, β_j are the mean vectors of C_i^1 and C_j^2 respectively; μ_i, ν_j are the representative vectors of C_i^1 and C_j^2 respectively; and λ is a regularization parameter. The first two terms in the above objective function correspond to typical k-means type error terms, whereas the last two terms ensure that two clusterings are distinctively different. Note that to generate two distinct clusterings, the objective function needs to have four terms. To generate m distinct clusterings, it needs to have $(m + \binom{m}{2})$ terms, which make the objective function highly convoluted. On the other hand, producing m distinct clusterings in our spectral framework is relatively straightforward.

3.3. Visualizing Clustering Dimensions

So far, we have shown how to produce multiple clusterings ($C^i, i = 1\!:\!m$) of a dataset. Next, we identify the most informative unigrams characterizing each clustering so that the corresponding clustering dimension is visualizable to the user. To select 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, w_i will have a high rank w.r.t. 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 partition, 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 can then visualize each induced clustering dimension by inspecting the features in the corresponding ranked lists.

4. Evaluation

We perform evaluations on document clustering tasks.

4.1. Experimental Setup

Datasets. We employ five text datasets. **Two Newsgroups** (TNG) consists of all the documents from two sections of 20 Newsgroups, `talks.politics` and `sci.crypt`. Blitzer et al.'s (2007) **book** (BOO) and **DVD** datasets each contains 2000 customer reviews of books and DVDs from Amazon. The **MIX** dataset is a 4000-document dataset consisting of the 2000 BOO reviews and the 2000 DVD reviews, as described above. Finally, our **POA** dataset contains 2000 political articles written by columnists who identified themselves as either Republicans or Democrats.[7]

Gold-standard creation. We asked five graduate students not affiliated with this research to annotate each dataset with different clustering dimensions. They first independently proposed plausible 2-way clustering dimensions for a dataset after reading its documents, and then agreed on a set of clustering dimensions for the dataset through discussion. As seen in Table 3, seven distinct clustering dimensions were proposed for the five datasets, including: (1) Sentiment (whether the sentiment expressed in a review is *positive* or *negative*); (2) Subjectivity (whether a review contains mostly *objective* material (e.g., description of a product) or mostly *subjective* material (e.g., the author's opinion about the product)); (3) Topic1 (whether a review was written for a *book* or a *movie*); (4) Topic2 (whether a document is about *science* or *politics*); (5) Strength (whether the opinion expressed in a review is *strong* or *weak*); (6) Political affiliation (whether a political article was written by a *Democrat* or a *Republican*); and (7) Policy (whether a political article describes a *domestic* or *foreign* policy).[8]

Next, we asked the same group of people to annotate

[6] 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] These political articles were selected randomly from http://www.commondreams.org/archives.

[8] Our annotated datasets are available at http://www.utdallas.edu/∼sajib/multi-clusterings.html.

TNG	*Topic2*
BOO	*Sentiment*, Subjectivity, Strength
DVD	*Sentiment*, Subjectivity, Strength
MIX	*Topic1*, *Sentiment*, Subjectivity, Strength
POA	*Political Affiliation*, Policy

Table 3. Clustering dimensions for the five datasets.

each dataset along each of its clustering dimensions. As these datasets have been annotated w.r.t. some of the clustering dimensions (i.e., the italicized ones in Table 3) when we collected them, the annotators only need to annotate w.r.t. the non-italicized dimensions.

Preprocessing. To preprocess a document, we follow Dasgupta & Ng (2009): 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. Moreover, we remove from the vector punctuation, numbers, words of length one, and words that occur in only a single document. Finally, we exclude words with high document frequency, many of which are stopwords or domain-specific general-purpose words. We compute the similarity between two documents by taking the dot product of their feature vectors.

4.2. Interpretability of Clustering Dimensions

To investigate (1) whether an induced clustering dimension is human-interpretable when represented as two ranked lists of features, and (2) how well our algorithm can recover the clustering dimensions manually identified for each dataset (see Table 3), we performed the following human experiment independently with ten graduate students, none of whom were involved in the human annotation process described previously.

Specifically, for each clustering produced by our algorithm, we showed each human judge the top 100 features selected for each cluster according to WLLR, (see Table 6 for a snippet), and asked her to determine whether the resulting dimension can be *labeled* (e.g., with a dimension label such as Sentiment). If so, she would assign a label to the dimension as well as a label to each of the two clusters involved in the dimension. In addition, she was told that the same dimension label can be assigned to more than one induced clustering dimension for each dataset. Note that she was *not* informed of the set of possible dimension labels and cluster labels , although she had some knowledge about each dataset. For instance, she knew that BOO is composed of book reviews, and MIX is a collection of book and DVD reviews.

Results of this experiment are shown in Table 4. For each of the four dimensions (induced using $\mathbf{e}_2$ through $\mathbf{e}_5$) for each dataset, we show (1) the fraction of judges who determine that the dimension is interpretable (and therefore can be labeled), and (2) the dimension label assigned by the majority of these judges. As we can see, the ten judges achieved high consistency in terms of whether a dimension can be labeled. In fact, in all cases where a dimension was determined to be interpretable, an agreement rate of $\geq 70\%$ was achieved on *which* label should be assigned to the dimension. Considering the fact that the judges were not informed of the possible set of labels *a priori*, this is a fairly high agreement rate. More importantly, the judges recovered almost all of the dimensions shown in Table 3 for each dataset, with the exception of Strength, which was not identified for any of the three datasets that contain this dimension. It is also worth noting that some of the judges labeled the Policy dimension as "War vs. Non-War", which we considered correct as the two roughly refer to the same dimension. Hence, the human judges recovered 10 of the 13 dimensions in Table 3, yielding a recall of 77%.

4.3. Clustering Quality

Next, we examine the quality of the clusterings induced by our algorithm. To gauge the performance of our algorithm, we will first report the results of four baseline systems below. We use accuracy and Adjusted Rand Index (ARI) to evaluate the clusterings produced by each system against the gold clusterings.

4.3.1. Baseline Systems

Traditional clustering algorithms. We use Ng et al.'s (2001) spectral clustering algorithm and Nonnegative Matrix Factorization (NMF) (Xu et al., 2003) as our first two baselines. Since these methods can propose only one clustering per dataset but most of our datasets contain at least two gold clusterings (one for each clustering dimension), we compare this proposal clustering against each of the gold clusterings to obtain the accuracy results in rows 1 and 2 of Table 5.[9]

Meta clustering. Since our clustering algorithm produces multiple clusterings, it is desirable to have a baseline that also produces multiple clusterings. However, as mentioned before, many of the existing algorithms that produce multiple clusterings work in a semi-supervised setting (Gondek & Hofmann, 2004; Davidson & Qi, 2007). The only notable exceptions are Caruana et al. (2006) and Jain et al. (2008) [see Section 2 for a discussion]. Since Jain et al.'s approach produces two clusterings but some of our datasets can

[9] The ARI results exhibit the same trend as those of accuracy and are omitted here due to space limitations.

	2nd eigenvector		3rd eigenvector		4th eigenvector		5th eigenvector	
TNG	1.0	Topic2	1.0	Topic2	1.0	Topic2	0.0	–
BOO	0.0	–	0.8	Subjectivity	1.0	Sentiment	0.4	–
DVD	0.8	Subjectivity	1.0	Sentiment	0.0	–	0.2	–
MIX	1.0	Topic1	0.7	Subjectivity	1.0	Sentiment	1.0	Sentiment
POA	0.7	Political Aff.	1.0	Policy	1.0	Policy	0.0	–

Table 4. Human interpretability results. Shown for each eigenvector are: (1) the fraction of the judges that believe the corresponding dimension is human-interpretable; and (2) the label assigned by the majority of these judges if at least five judges believe that the dimension is interpretable. A '–' is used to indicate a non-interpretable dimension.

		TNG	BOO			DVD			MIX				POA	
	System	Dim1	Dim1	Dim2	Dim3	Dim1	Dim2	Dim3	Dim1	Dim2	Dim3	Dim4	Dim1	Dim2
1	Ng et al.	89.8	58.9	58.8	51.5	54.9	61.5	54.9	77.9	52.9	68.5	51.8	54.3	67.6
2	NMF	85.2	52.1	57.8	50.7	50.3	60.5	51.9	69.2	51.7	58.6	52.9	53.0	61.1
3	META	76.2	50.8	51.2	51.5	53.9	71.0	52.9	50.2	50.2	58.6	50.1	59.4	61.6
4	IFR	83.8	58.9	63.2	50.2	51.2	60.5	50.1	77.1	50.0	51.0	50.1	57.8	61.6
5	Ours	83.8	69.5	63.8	56.7	70.7	60.5	55.4	77.1	68.9	59.7	54.2	69.7	70.2

Table 5. Results in terms of accuracy for the five datasets. Dim$n$ of a dataset refers to the nth dimension listed for the dataset in Table 3. For instance, Dim1 and Dim2 of BOO correspond to Sentiment and Subjectivity, respectively.

be clustered in three different ways, we evaluate meta clustering (Caruana et al., 2006) only. We produce multiple clusterings for each dataset by running this algorithm 100 times and report in row 3 of Table 5 the *best* result obtained for each dimension of each dataset. Although the best results are reported, meta clustering underperforms the first two baselines for all but two dimensions (DVD/Dim2 and POA/Dim1).

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 the feature space the top informative features that are identified using WLLR for each cluster. To produce another clustering, we repeat these two steps with the reduced feature space. To obtain the results of this algorithm in row 4 of Table 5, we (1) run it for m iterations to produce m clusterings, where m is the number of dimensions the dataset has, and (2) find the bipartite matching between the proposal clusterings and the gold standard clusterings that has the highest average accuracy. 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. Except for BOO/Dim2 and POA/Dim1, this algorithm never outperforms the first baseline.

4.3.2. Our Clustering Algorithm

Results of our algorithm are shown in row 5 of Table 5 and are computed as follows. For each dataset, we (1) find the one-to-one mapping between the m proposal clusterings and the gold standard clusterings that yields the highest average accuracy, and (2) compute the accuracy of a proposal clustering against the mapped gold standard clustering. Note that the clustering accuracy along the Strength dimension for all three sentiment datasets is low, which suggests that our algorithm fails to induce a clustering along Strength. Nevertheless, our algorithm frequently outperforms all of the baselines, and its clustering accuracies along all other dimensions are reasonably good (59.7% to 83.8%). This substantiates our claim that our algorithm can induce multiple meaningful clusterings of a dataset along distinct dimensions. As we can see, the accuracies are generally higher for the topic-related dimensions (e.g., Politics vs. Science and Domestic vs. Foreign) than the other dimensions (e.g., Sentiment, Subjectivity). This should not be surprising: learning non-topical classification tasks is difficult even for supervised systems that are trained on a large amount of labeled data (e.g., Thomas et al. (2006)).

In Table 6, a shaded column corresponds to an eigenvector (i.e., a clustering) that achieves the best accuracy along the dimension it is labeled with for the DVD and POA datasets. As we can see, the eigenvector that achieves the best accuracy along Political Affiliation is $\mathbf{e}_5$. Interestingly, according to Table 4, the eigenvector that was labeled as Political Affiliation by the human judges was $\mathbf{e}_2$, not $\mathbf{e}_5$. This discrepancy is perhaps not surprising, as the informative features for Democrats and Republicans are highly overlapping, which complicates the recognition of the Political Affiliation dimension. Nevertheless, for each of the remaining clustering dimensions, the human-selected eigenvector is also the one that achieves the best accuracy. This provides suggestive evidence that it is possible to visualize a dimension based on the informative features.

DVD				POA			
$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
Subjective	**Positive**	$\mathbf{C}_1$	$\mathbf{C}_1$	$\mathbf{C}_1$	**Foreign**	$\mathbf{C}_1$	**Republican**
fan	music	video	saw	israel	iran	countries	voters
bought	wonderful	music	watched	islamic	iraqi	israel	conservative
video	collection	found	fan	violence	forces	muslim	gop
money	cast	workout	loved	muslim	nuclear	iran	win
series	quality	bought	series	islam	countries	oil	polls
waste	video	videos	comedy	god	israeli	god	poll
dvds	excellent	times	enjoy	peace	saddam	living	candidates
videos	enjoy	children	season	soldiers	strategic	peace	hilary
season	family	watched	whole	saddam	east	western	kerry
workout	must	kids	liked	enemy	iraqis	east	clinton
Objective	**Negative**	$\mathbf{C}_2$	$\mathbf{C}_2$	$\mathbf{C}_2$	**Domestic**	$\mathbf{C}_2$	**Democrat**
role	money	series	money	tax	family	nsa	agency
young	thought	cast	quality	economy	love	court	information
cast	waste	fan	video	budget	person	constitutional	companies
actors	worst	money	director	companies	someone	judiciary	department
men	nothing	stars	found	income	parents	surveillance	justice
us	actually	actors	version	taxes	church	committee	warrant
world	maybe	comedy	sound	spending	book	sen	criminal
played	boring	original	waste	cuts	young	democrat	investigation
performance	read	worst	special	billion	woman	nomination	legal
script	down	action	picture	prices	guy	alito	documents
Subjectivity	**Sentiment**				**Policy**		**Political Aff.**

Table 6. Top ten features induced for each dimension for the DVD and POA datasets. The dimension/cluster labels are taken from the gold clustering to which an eigenvector ($\mathbf{e}_2$, . . ., $\mathbf{e}_5$) is mapped; $\mathbf{C}_1$ and $\mathbf{C}_2$ are the unlabeled clusters.

5. Conclusions

We presented an algorithm for producing multiple clusterings of a text collection along its important dimensions without using any labeled data. This contrasts with the majority of existing clustering algorithms, which can only produce a single clustering of a dataset along its most prominent dimension.

In addition, our work has led to a better understanding of spectral clustering. To our knowledge, we are the first to employ spectral clustering to produce multiple clusterings of a dataset, and show in the context of text clustering that a dimension induced by spectral clustering can be human-interpretable.

Finally, we have contributed to text visualization and summarization. By representing an induced clustering dimension using words that are representative of the dimension, our algorithm offers humans a convenient means to visualize a dimension, facilitating exploratory text analysis.

References

Bilenko, M., Basu, S., and Mooney R. J. Integrating constraints and machine learning in semi-supervised clustering. In *ICML*, pp. 81–88, 2004.

Blei, D. M., Ng, A. Y., and Jordon, M. I. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3: 993–1022, 2003.

Blitzer, J., Dredze, M., and Pereira, F. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, pp. 440–447, 2007.

Caruana, R., Elhawary, M. F., Nguyen, N., and Smith, C. Meta clustering. In *ICDM*, pp. 107–118, 2006.

Dasgupta, S. and Ng, V. Topic-wise, sentiment-wise, or otherwise? Identifying the hidden dimension for unsupervised text classification. In *EMNLP*, 2009.

Davidson, I. and Qi, Z. Finding alternative clusterings using constraints. In *ICDM*, pp. 240–249, 2007.

Gondek, D. and Hofmann, T. Non-redundant data clustering. In *ICDM*, pp. 75–82, 2004.

Jain, P., Meka, R., and D., Inderjit S. Simultaneous unsupervised learning of disparate clusterings. In *SDM*, pp. 858–869, 2008.

Liu, B., Li, X., Lee, W. S., and Yu, P. S. Text classification by labeling words. In *AAAI*, pp. 425–430, 2004.

Ng, A., Jordan, M., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

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

Thomas, M., Pang, B., and Lee, L. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *EMNLP*, pp. 327–335, 2006.

Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. Constrained k-means clustering with background knowledge. In *ICML*, pp. 577–584, 2001.

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

Xu, W., Liu, X., and Gong, Y. Document clustering based on non-negative matrix factorization. In *SIGIR*, 2003.