

Document Clustering via Adaptive Subspace Iteration

Tao Li
Computer Science Dept.
University of Rochester
Rochester, NY 14627-0226
taoli@cs.rochester.edu

Sheng Ma
IBM T.J. Watson Research
Center
Hawthorne, NY 10532
shengma@us.ibm.com

Mitsunori Ogihara
Computer Science Dept.
University of Rochester
Rochester, NY 14627-0226
ogihara@cs.rochester.edu

ABSTRACT

Document clustering has long been an important problem in information retrieval. In this paper, we present a new clustering algorithm *ASI*¹, which uses explicitly modeling of the subspace structure associated with each cluster. *ASI* simultaneously performs data reduction and subspace identification via an iterative alternating optimization procedure. Motivated from the optimization procedure, we then provide a novel method to determine the number of clusters. We also discuss the connections of *ASI* with various existential clustering approaches. Finally, extensive experimental results on real data sets show the effectiveness of *ASI* algorithm.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Clustering; I.2 [Artificial Intelligence]: Learning; I.5 [Pattern Recognition]: Applications

General Terms

Algorithms, Experimentation, Measurement, Performance, Theory, Verification

Keywords

document clustering, adaptive subspace identification, alternating optimization, factor analysis

1. INTRODUCTION

As a fundamental and effective tool for efficient document organization, summarization, navigation and retrieval of large amount of documents, document clustering has been very active and enjoying a growing amount of attention with the ever-increasing growth of the on-line information. A document clustering problem can be intuitively described as the problem of finding, given a set W of some n data points in a multi-dimensional space, a partition of W into classes such that the points within each class are *similar* to

¹*ASI* stands for Adaptive Subspace Iteration.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGIR '04, July 25–29, 2004, Sheffield, South Yorkshire, UK.
Copyright 2004 ACM 1-58113-881-4/04/0007 ...\$5.00.

each other. Good document clustering enables better information services by browsing and organizing documents into meaningful cluster hierarchies and provides a useful complement for traditional text search engines when key-word based search returns too many documents.

The more general problem of clustering has been studied extensively in machine learning [8, 29], information theory [32, 10], databases [19, 44], and statistics [4, 6] with various approaches and focuses. Unfortunately, many methods fail to produce satisfactory results because they do not validate or offer interpretation of the clusters produced, because they make some simple but critical assumptions on the data distributions (e.g., that they are Gaussian), or because the criterion/objective function used is based on a distance function between sample points. The last property is critical. Many of the existing clustering algorithms do not work efficiently in high dimensional spaces (*curse of dimensionality*). For document clustering, the data are indeed of high dimensions. It has been shown that in a high dimensional space the distance between every pair of points is almost the same for a wide variety of data distributions and distance functions [5]. This justifies the attempt of reducing dimension of the input data. Many feature selection techniques have been applied in that regard. However, as demonstrated in [1], the correlations among the dimensions are often specific to data locality, in the sense that some data points are correlated with a given set of features and others are correlated with respect to different features. In other words, in high dimensional space, each cluster usually has its own subspace structure. As pointed out in [24], all methods that overcome the dimensionality problems use a metric for measuring neighborhoods, which is often implicit and/or adaptive.

In this paper, we propose a new clustering algorithm, *ASI*, which explicitly models the subspace structure in the feature space associated with each cluster². The clustering algorithm simultaneously performs two tasks: data reduction (assigning data points into clusters) and subspace identification (identifying subspace structure associated with each cluster). The tasks are carried out by an iterative optimization procedure that alternates between identification of the subspace structure from current cluster partitions and updating of the clusters based on the identified new subspace structures. By explicitly modeling the subspace structure, *ASI* produces interpretable descriptions of the resulting clusters as an added bonus. In addition, through the iterative subspace identification, *ASI* performs implicit adaptive feature selection at each iteration and flexibly measures the distances between data points. Thus, it works well for high-dimensional data. We have shown the convergence property of *ASI* algorithm and conducted extensive experiments to show its effectiveness. The rest of the paper is organized as fol-

²By subspace structure, we mean the linear combinations of the original feature space.

lows: Section 2 introduces the clustering models of *ASI*, describes its optimization procedure and provides the method for deciding the number of clusters; Section 3 shows our experimental results; Section 4 surveys the related work. Finally, our conclusions are presented in Section 5.

2. ASI CLUSTERING

2.1 The Cluster Model

Due to space limitation, the notations used in the paper are introduced in Table 1. Note that $D^T D = \begin{bmatrix} n_1 & 0 & \dots & 0 \\ 0 & n_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & n_k \end{bmatrix}$ is

$W = (w_{ij})_{n \times m}$	The Data set
n	Number of data points
m	Number of features
$p_i = (w_{i1}, \dots, w_{im})$	The i -th data points
k	Number of clusters
n_1, n_2, \dots, n_k	Number of points for each cluster
$C = \{C_1, \dots, C_k\}$	The clusters
$D = (d_{ij})_{n \times k}$	A $n \times k$ matrix specifying cluster partitions $d_{ij} = 1$ if $p_i \in C_j$ and 0 otherwise
$F = (f_{ij})_{m \times k}$	F specifies the subspace structure for each cluster, f_{ij} 's are coefficients of each feature associated with each cluster.
$WF = (\sum_{i=1}^m w_{it} f_{ij})_{n \times k}$	The projection of the data points into the subspaces defined by F .
$S = (D^T D)^{-1} D^T WF$	The projection of the centroids into the subspaces defined by F .

Table 1: Notation

a diagonal matrix with the cluster size on the diagonal. The inverse of $D^T D$ serves as a weight matrix to compute the projection.

ASI clustering explicitly models the subspace structure associated with each cluster via linear combinations of the original features. As described in Table 1, F specifies the subspace structure for each cluster, i.e., the coefficients of the linear combination of the features and $S = (D^T D)^{-1} D^T WF$ shows the projection of the centroids into the subspaces. The clustering is then sought such that the points of a cluster are close to each other within the subspaces. Formally this can be described as

$$\arg \min_{D, F, S} O = \frac{1}{2} \|WF - DS\|_F^2, \quad (1)$$

where $\|X\|_F$ is the Frobenius norm of the matrix X , i.e., $\sqrt{\sum_{i,j} x_{ij}^2}$. Note that WF is the projection of the data points into the subspaces, D is a binary matrix specifying the cluster partitions and DS gives the approximation of the projection by the centroids. The above criterion is equivalent to the within-class distance of the partition induced by D .

Let $\text{tr}(A)$ be the trace of A . Noting that $\|A\|_F^2 = \text{tr}(A^T A)$ and $\text{tr}(AB) = \text{tr}(BA)$ we have:

$$\begin{aligned} O(D, F, S) &= \frac{1}{2} \|WF - DS\|_F^2 \\ &= \frac{1}{2} \text{tr}((WF - DS)(WF - DS)^T) \\ &= \frac{1}{2} (\text{tr}(W^T F^T FW) - 2\text{tr}(WFS^T D^T) + \text{tr}(DSS^T D)) \\ &= \text{tr}(W^T F^T FW) - \text{tr}(F^T W^T D(D^T D)^{-1} D^T WF) \\ &= \text{tr}(W^T F^T FW) - \text{tr}(F^T W^T DS) \end{aligned}$$

It is easy to see that $\text{tr}(W^T F^T FW)$ is total deviance of WF and $\text{tr}(F^T W^T D(D^T D)^{-1} D^T WF)$ is the between-class distance of the partition induced by D .

2.2 Optimization

The objective function can be minimized (local minima) by alternatively optimize one of D or F (and S) while fixing the other.

First, given F and S , we will minimize $O(D, F, S)$ with respect to D . By Equation (2), for fixed F and S , minimizing $O(D, F, S)$ reduces to maximizing $\text{tr}(F^T W^T DS)$. The problem is solved for different rows of D independently. Since D is a binary matrix, we have

$$d_{ij} = \begin{cases} 1 & \text{if } (WF)_i - S_j = \min\{(WF)_i - S_r, r = 1, \dots, k\} \\ 0 & \text{Otherwise} \end{cases} \quad (2)$$

Second, given D , update F by minimizing

$$\begin{aligned} O(D, F, S) &= \frac{1}{2} \|WF - DS\|_F^2 \\ &= \frac{1}{2} \|WF - D(D^T D)^{-1} D^T WF\|_F^2 \\ &= \frac{1}{2} \|(W - D(D^T D)^{-1} D^T W)F\|_F^2 \end{aligned}$$

The columns of F are the coefficients of the features associated with different clusters, and are usually orthogonal. So, the objective criterion is minimized by taking the smallest k eigenvectors of $W^T (I_n D (D^T D)^{-1} D^T) W$, or equivalently, the first k eigenvectors of $W^T (D (D^T D)^{-1} D^T - I_n) W$, where I_n is the identity matrix (see [17]). After F is updated, we also obtain a new $S = (D^T D)^{-1} D^T WF$.

These two steps are executed iteratively. After each iteration, we compute the value of the objective criterion $O(D, F, S)$. If it is decreased, we then repeat the process; otherwise, the process has converged to a local minima. Since the *ASI* procedure monotonically decreases the objective criterion, it converges to a local optima. The clustering procedure is described as Algorithm 1.

Algorithm 1 *ASI*: clustering procedure

Input: (data points: $W_{n \times m}$, # of classes: k)

Output: D : cluster assignment;

begin

1. **Initialization:**

1.1 Initialize D and F ;

1.2 Compute $S = (D^T D)^{-1} D^T WF$;

1.3 Compute $O_0 = O(D, F, S)$

2.

Iteration:

begin

2.1 Update D given F and S

2.2 Update F given D

2.3 Compute S

2.4 Compute the value of $O_1 = O(D, F, S)$;

2.5 If $O_1 < O_0$

2.5.1 Set O_0 to O_1

2.5.2 Repeat from 2.1

2.6 else

2.6.1 break from the loop; (Convergence)

end

3. **Return** D, F ;

end

The initialization step set the initial values for D and F . Since D is a binary matrix having at most one occurrence of 1 in each row, the local minimum identified is very sensitive to initialization. To overcome the sensitivity of initialization, a refining procedure, whose idea is to use mutual information to measure the similarity between a pair of clustering results, is employed. ASI attempts to find a best clustering result having the largest average mutual information against all others. In our experiments, ten runs of clustering are performed for refining.

A special case for ASI clustering occurs when we require that each row of F has at most one entry is 1 and all the rest is 0. In this case, it is assumed that different clusters associated with disjoint sets of features and F actually generates an implicit feature assignments into clusters. In other words, F induces the partitions of features and ASI implements a form of co-clustering of both data and features. We have developed a **mutually reinforcing** optimization procedure to exploit the duality of the data and features [30]: if a feature is shared by many points associated with a cluster, then feature has a high weight associated with the cluster. On the other hand, if a data point is shared by many features associated with a cluster, then the data point has a high weight associated with the cluster.

2.3 Deciding the Number of Clusters

ASI explicitly models the subspace structure in cluster analysis, and, as seen in Section 2.2, at each iteration, the subspace structure F is determined by the first k eigenvector of $W^T(D(D^T D)^{-1}D^T - I_n)W$. Note that each column of F corresponds to a cluster subspace and k is the number of clusters. Also observe that $(D(D^T D)^{-1}D^T - I_n)$ serves as a weight matrix since the diagonal entries of $(D^T D)^{-1}$ are cluster sizes. Hence it is natural to relate the number of clusters with the spectrum of $W^T W$.

Let $q = \min(m, n)$, WW^T and $W^T W$ share the first q eigenvalues. Note that WW^T is a $n \times n$ matrix and the ij -th entry of WW^T computes the inner product of the i -th row and the j -th row of W . If W is normalized, then each entry of WW^T shows the cosine similarity between corresponding data points. For simplicity, suppose that the points in W are ordered according to their cluster memberships, that is, the points in the first cluster appear first and the point in the k th cluster appears at the end. Such permutation does not change the spectral properties.

Since data points inside each cluster are similar to each other while they are quite different from those in other clusters, WW^T can be regarded as the sum of two matrices: $WW^T = L + E$ where $L = \begin{pmatrix} X_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & X_k \end{pmatrix} \in R^{n \times n}$, $X_i = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \vdots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \in R^{n_i \times n_i}$ and $E \in R^{n \times n}$ is a matrix with a small value in each entry (if W is normalized, the cosine measure is very similar to the Euclidean distance [45]).

LEMMA 1. Let $X = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \vdots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$, i.e., all entries in the matrix $X \in R^{n \times n}$ are 1. Then the only nonzero eigenvalue of X is n .

PROOF. Since the rank of matrix X is 1, then the dimension of the null space of X is $n - 1$. Hence there is only one nonzero eigenvalue of X . It is clear the summation of all the eigenvalues of X equals to the trace of the matrix X , which is n . Hence the nonzero eigenvalue is n . \square

LEMMA 2. Let $L = \begin{pmatrix} X_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & X_k \end{pmatrix}$. That is, L is

a block diagonal matrix, with each block matrix formed as in Lemma 1. Let n_i be the size of the matrix X_i , for $i = 1, \dots, k$. Then the only nonzero eigenvalue of L are n_1, n_2, \dots, n_k .

PROOF. Since L is block diagonal, its eigenvalues and eigenvectors are the union of the eigenvalues and eigenvectors of its blocks (the latter padded approximately with zeros). \square

LEMMA 3. Let A and E be two symmetric matrices with the same dimensions. Then

$$|\lambda_i(A) - \lambda_i(A + E)| \leq \|E\|_2, \text{ for } i = 1, \dots, n$$

where $\lambda_i(A)$ denotes the i -th largest eigenvalue of the matrix A , similarly, $\lambda_i(A + E)$ for matrix $(A + E)$.

PROOF. The lemma follows from the standard results in matrix perturbation theory. See [17, Theorem 8.3.4. on page 429]. \square

THEOREM 4. Let $M = L + E$ where L has the form as in Lemma 2 and E is a matrix with a small value in each entry. Then M has dominant k eigenvalues, which are close to n_1, n_2, \dots, n_k .

PROOF. The proof directly follows from Lemma 2 and Lemma 3. \square

Since WW^T can be regarded as the sum of two matrices, L and E , Theorem 4 shows that WW^T will have k dominant eigenvalues that are distinguished from the rest. Using these theoretical results, we then obtain a novel method to determine the number of clusters based on eigenvalues of WW^T .

We note here that the above conclusion can also be derived from matrix perturbation theory [27, 11]. Denote $S = WW^T$, set:

$$S(\epsilon) = S(0) + \epsilon S^{(1)} + \epsilon^2 S^{(2)} + \dots,$$

where $S(0) = L$ is the unperturbed part of S . It can then follows from Perron cluster analysis that the spectrum of $S(\epsilon)$ can be divided into two parts: k dominant eigenvalues and the remaining part of the spectrum bounded away from the dominant eigenvalues.

3. EXPERIMENTS

In this section, we apply our ASI algorithm to document clustering and compare its performance with other standard clustering algorithms. In our experiments, documents are represented using binary vector-space model where each document is a binary vector in the term space and each element of the vector indicates the presence/absence of the corresponding term.

3.1 Evaluation Measures

There are many ways to measure how accurately ASI performs. One is the *confusion matrix* which is described in [1]. Entry (o, i) of a confusion matrix is the number of data points assigned to output class o and generated from input class i . Another is the *purity* [45], which measures the extent to which each cluster contained data points from primarily one class. The purity of a clustering solution is calculated as a weighted sum of individual cluster purities: $Purity = \sum_{i=1}^K \frac{n_i}{n} P(S_i)$. Here S_i is a particular cluster of size n_i , K is the number of clusters, and n is the total number of points. $P(S_i)$, the individual cluster purity of cluster S_i , is defined to be

$P(S_i) = \frac{1}{n_i} \max_j(n_j^i)$, where n_j^i is the number of documents of the i -th input class that were assigned to the j -th cluster. A high purity value implies that the clusters are “pure” subsets of the input classes. In general, the larger the values of purity, the better the clustering solution is.

3.2 Document Datasets

To measure the effectiveness of *ASI* on document clustering, we use standard labeled corpora widely used in the information retrieval literature. We view the labels of the dataset as the objective knowledge on the structure of the datasets. We use the purity as the performance measure.

We use a variety of datasets, most of which are frequently used in the information retrieval research. The number of classes ranges from four to twenty, and the number of documents ranges from 476 to 8,280. This is varied enough to obtain good insights on how well *ASI* performs. Table 2 summarizes the characteristics of the datasets.

CSTR This is the dataset of the abstracts of technical reports published in the Department of Computer Science at the University of Rochester between 1991 and 2002. The TRs are available at <http://www.cs.rochester.edu/trs>. It has been used in [31] for text categorization. The dataset contained 476 abstracts, which were divided into four research areas: Natural Language Processing (NLP), Robotics/Vision, Systems, and Theory.

WebKB The WebKB dataset contains webpages gathered from university computer science departments. There are about 8,300 documents and they are divided into seven categories: student, faculty, staff, course, project, department and other. The raw text is about 27MB. Among these seven categories, student, faculty, course and project are four most populous entity-representing categories. The associated subset is typically called **WebKB4**. In this paper, we did experiments on both seven- and four-category datasets.

Reuters The Reuters-21578 Text Categorization Test collection contains documents collected from the Reuters newswire in 1987. It is a standard text categorization benchmark and contains 135 categories. In our experiments, we use a subsets of the data collection which include the ten frequent categories among the 135 topics and we call it **Reuters-top 10**.

K-dataset The K-dataset was from WebACE project [21] and it was used in [7] for document clustering. The K-dataset contains 2,340 documents consisting news articles from Reuters new service via the Web in October 1997. These documents are divided into twenty classes.

To pre-process the datasets, we remove the stop words using a standard stop list, all HTML tags are skipped and all header fields except subject and organization of the posted article are ignored. In all our experiments, we first select the top 1,000 words by mutual information with class labels. The feature selection is done with the rainbow package [33].

Datasets	# documents	# class
CSTR	476	4
WebKB4	4,199	4
WebKB	8,280	7
Reuters-top 10	2,900	10
K-dataset	2,340	20

Table 2: Document DataSets Descriptions.

3.3 Experimental Results

A recent comparative study on document clustering [47] showed that the performance of clustering based on graph partitioning package (CLUTO) is very good. The CLUTO package is built on a sophisticated multi-level graph partitioning engine and it provides many different criterion functions that can be used to drive both partitional and agglomerative clustering algorithms. Here, in our experiments, we compare the performance of *ASI* on the datasets with the algorithms provided in the CLUTO package [45, 46]. We choose several partitional criteria and several agglomerative clustering algorithms for comparison. We also provide the experiment results of traditional K-means algorithm.

The comparisons are shown in Table 3. Each entry is the purity of the corresponding column algorithm on the row dataset. P1 is a multi-level partitioning method which tries to maximize the cosine similarity between each document and the cluster centroid. Slink, Clink and UPGMA columns are different hierarchical clustering algorithms using single-linkage, complete-linkage and UPGMA aggregating policies. Single-linkage and complete-linkage use the maximum and the minimum distance between the two clusters, respectively, while UPGMA - Unweighted Pair-Groups Method Average uses the distance of the cluster centers to define the similarity of two clusters for aggregating. In our experiments, we use the cosine function measure of the two document vectors as their similarity.

Datasets	ASI	K-means	P1	Slink	Clink	UPGMA
CSTR	0.889	0.744	0.782	0.380	0.422	0.597
WebKB4	0.638	0.534	0.639	0.392	0.446	0.395
WebKB	0.501	0.423	0.489	N/A	N/A	N/A
Reuters	0.728	0.643	0.717	0.393	0.531	0.498
K-dataset	0.719	0.627	0.644	0.220	0.514	0.551

Table 3: Comparison of performance of clustering algorithms. Each entry is the purity of the corresponding column algorithm on the row dataset.

From Table 3, we observe that *ASI* achieves the best performance on all datasets except WebKB4. In addition, *ASI* outperforms K-means and the hierarchical methods on all datasets. On WebKB4, P1 has the best performance of 0.639 and *ASI* gives the result of 0.638. Figure 1 shows the graphical comparison. In total, the performance of *ASI* is always either the winner or very close to the winner. The comparison shows that, although there is no single winner on all the datasets, *ASI* is a viable and competitive algorithm in document clustering domain.

Table 4 shows the confusion matrices built from the clustering results on CSTR dataset. The columns of the confusion matrix are NLP, Robotics/Vision, Systems and Theory, respectively. The result shows that Systems and Theory are much different from each other and each different from NLP and from Robotics/Vision, that NLP and Robotics/Vision are similar to each other, and that NLP is more similar to Systems than Robotics/Vision.

Output	Input			
	1	2	3	4
A	68	0	8	0
B	8	1	4	120
C	0	0	160	0
D	25	70	6	6

Table 4: The confusion matrix of *ASI* on the CSTR dataset.

We now try to visualize the cluster structure that might be dis-

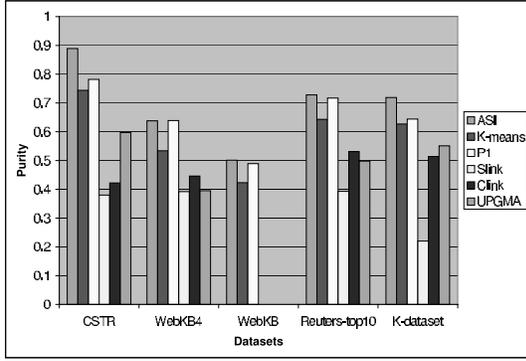


Figure 1: Comparisons of the purity values.

covered by ASI algorithm. We restrict that each row of the F has at most one entry is 1 and assigns each feature into clusters. Figure 2 shows the original word-document matrix of CSTR and the reordered matrix obtained by arranging rows and columns based on the cluster assignments. The figure reveals the hidden sparsity structure of both the document and word clusters. The four block diagonals in Figure 2(b) correspond to the four clusters and the dense region at the bottom of the figure identifies the feature outliers (which are distributed uniformly across the technical reports). The rough block diagonal sub-structures observed indicate the cluster structure relations between documents and words. ASI tends to yield better clustering solutions by explicitly identifying the subspace structure, especially for high dimensional sparse datasets. The dense region (corresponding to feature outliers) also reflects the feature selection ability of ASI.

A nice property of ASI is that the resulting classes can be easily described in terms of features, since the algorithm explicitly exploit the subspace structure in the feature space. In Table 5, we show the four word clusters obtained when applying ASI to the CSTR dataset. We see that these words are meaningful and are often representatives of the associated document cluster. For example, *shared* and *multiprocessor* are representatives of the *Systems* cluster³. Similarly, *Turing* and *reduction* are strongly related to the research efforts in the Theory group at Rochester. An interesting and also important implication is the interpretability of the clustering results. The document clusters could be well explained using its associated feature (word) clusters.

3.4 Number of Clusters

The ability of ASI to estimate the number of clusters can be demonstrated using the CSTR dataset. In this experiment, we normalize the entries of WW^T so that that the sum of each row equals to 1. The normalization makes the largest eigenvalue equal to 1 and all the others are less than 1. As discussed in Section 2.3, the number of clusters then corresponds to the largest eigenvalues of WW^T . Figure 3 shows the top eigenvalues of WW^T .

Note that the n -th eigenvalue is less than $1/2$ when $n > 4$. This is a strong indicator on the number of clusters in the dataset.

³This conforms with the fact that system research at Rochester’s computer science has traditionally focused on shared and parallel system processing. See <http://www.cs.rochester.edu/dept/systems/>.

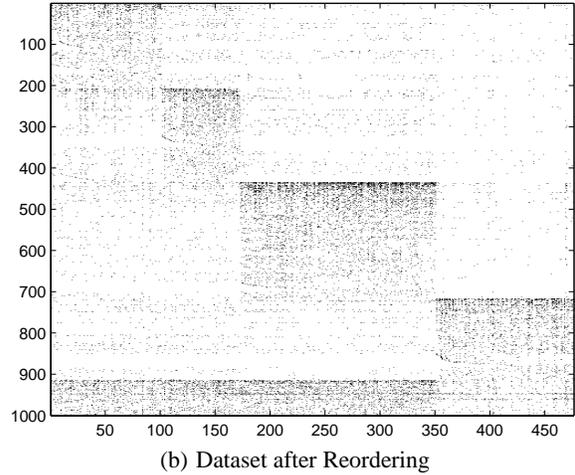
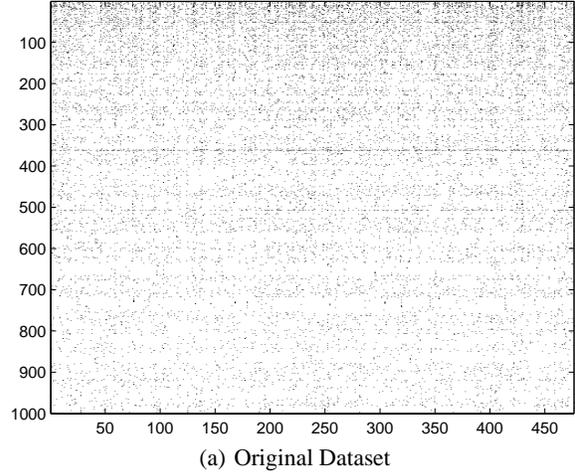


Figure 2: Visualization of the original document-data matrix and the reordered document-data matrix. The shaded region represents non-zero entries.

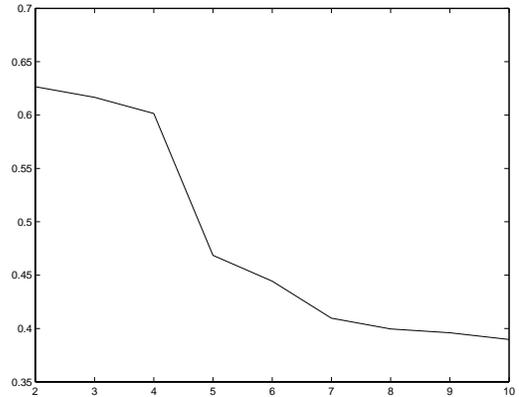


Figure 3: The top eigenvalues of normalized XX^T . The Y-axis indicates the eigenvalues and the X-axis indicates the order of the eigenvalues. Note that the largest eigenvalue is 1.

Cluster 1	Cluster 2	Cluster 3	Cluster 4
<i>share</i>	<i>train</i>	<i>track</i>	<i>turing</i>
<i>multiprocessor</i>	<i>spoken</i>	<i>freedom</i>	<i>reduction</i>
<i>cache</i>	<i>dialogue</i>	<i>movement</i>	<i>nondeterministic</i>
<i>synchronization</i>	<i>discourse</i>	<i>perception</i>	<i>collapse</i>
<i>local</i>	<i>plan</i>	<i>calibration</i>	<i>boolean</i>
<i>remote</i>	<i>speaker</i>	<i>target</i>	<i>oracle</i>
<i>load</i>	<i>utterance</i>	<i>sensor</i>	<i>bound</i>
<i>latency</i>	<i>corpus</i>	<i>eye</i>	<i>prove</i>
<i>contention</i>	<i>conversation</i>	<i>filter</i>	<i>downward</i>
<i>lock</i>	<i>parser</i>	<i>camera</i>	<i>count</i>
<i>schedule</i>	<i>act</i>	<i>behavior</i>	<i>circuit</i>
<i>block</i>	<i>inference</i>	<i>manipulator</i>	<i>fewp</i>
<i>message</i>	<i>semantic</i>	<i>motor</i>	<i>pspace</i>
<i>butterfly</i>	<i>disambiguation</i>	<i>robot</i>	<i>relativized</i>
<i>migration</i>	<i>linguistic</i>	<i>arm</i>	<i>string</i>
<i>policy</i>	<i>reason</i>	<i>stage</i>	<i>membership</i>
<i>page</i>	<i>lexicon</i>	<i>reconstruct</i>	<i>sat</i>
<i>busy</i>	<i>phrase</i>	<i>indoor</i>	<i>equivalent</i>
<i>wait</i>	<i>coverage</i>	<i>acquire</i>	<i>automata</i>
<i>multiprogramming</i>	<i>deductive</i>	<i>geometry</i>	<i>polynomial</i>

Table 5: The four word clusters obtained using ASI on the CSTR dataset. Cluster 1, 2, 3, and 4 represent Systems, Natural Language Processing, Robotics/Vision, and Theory, respectively. For each cluster, only top twenty words based on the associated degree in the final feature coefficients are included.

4. RELATED WORK

Traditional clustering techniques focus on one-sided clustering and they can be classified into partitional, hierarchical, density-based, and grid-based [25, 22]. Partitional clustering attempts to directly decompose the data set into k disjoint classes such that the data points in a class are nearer to one another than the data points in other classes. For example, the traditional K -means method tries to minimize the *sum-of-squared-errors* criterion function. Hierarchical clustering proceeds successively by building a tree of clusters. Density-based clustering is to group the neighboring points of a data set into classes based on density conditions. Grid-based clustering quantizes the object space into a finite number of cells that form a grid-structure and then performs clustering on the grid structure. Most of these algorithms use distance functions as objective criteria and are not effective in high dimensional spaces.

In what follows, we review the work that are closely related to our approach (see Figure 4 for a summary).

First of all, the ASI clustering can be regarded as an integration of K -means and eigenvector analysis [26]. ASI shares the alternating optimization procedure common to K -means type algorithms and performs factor analysis to identify the subspace structures for the clusters.

The simultaneous optimization in both directions of data and feature used in ASI is similar to the optimization procedure in co-clustering. The idea of co-clustering of data points and attributes dates back to [3, 23, 35]. Co-clustering is simultaneous clustering of both points and their attributes by way of utilizing the canonical duality contained in the point-by-attribute data representation. Govaert [18] studies simultaneous block clustering of the rows and columns of contingency tables. The idea of co-clustering has been also applied to the problem of clustering gene and tissue types based on gene expression [9]. Dhillon [12] presents a co-clustering algorithm for documents and words using bipartite

graph formulation and a spectral heuristic. More recently, Dhillon et al. [13] propose an information-theoretic co-clustering method for two-dimensional contingency table. By viewing a non-negative contingency table as a joint probability distribution between two discrete random variables, the optimal co-clustering is obtained by maximizing the mutual information between the clustered random variables. In fact, as demonstrated in Section 3, if we put restrictions on F , the subspace structure determined by F induces clustering on feature space. In other words, with restrictions on F , ASI enables an iterative co-clustering procedure for both the data and the feature assignments. However, unlike previously proposed co-clustering approaches, ASI performs explicit subspace optimization using least square minimization via matrix operations.

Also, ASI can be thought of as an approximate iterative information bottleneck method. The information bottleneck (IB) framework is first introduced for one-sided clustering [41]. The core idea of IB is as follows: given the empirical joint distribution of two variables (X, Y) , one variable is compressed so that the mutual information about the other is preserved as much as possible. The IB algorithm in [41] tries to minimize the quantity $I(X; \hat{X})$ while maximizing $I(\hat{X}; Y)$, where I is the mutual information and \hat{X} is the cluster representation of X . The overall criterion there is formulated as $I(X; \hat{X}) - \beta I(\hat{X}; Y)$, where β is a Lagrange multiplier determining the trade-off between compression and precision. Agglomerative versions of the IB method are used in [39, 38] to cluster documents after clustering individual words. ASI is similar to IB method in that the feature functions in both are restricted to clustering. The subspace structure induced by F captures most of the information about the original dataset and thus is a compact representation. The optimization procedure is then a procedure of searching for the concise representation.

The cluster model in ASI is similar to that the data and feature maps in [48]. The data and feature maps of [48] are defined as two functions mapping from the data and from the feature set to the number of clusters. The clustering algorithm of [48] is based on *Maximum Likelihood Principle* via a co-learning process between the data and feature maps. The maps can be viewed as an extremal case of the model in ASI when F is restricted to a binary function.

The optimization procedure for subspace structure identification in ASI utilizes the dominant eigenvectors of $W^T (D(D^T D)^{-1} D^T - I_n) W$. This carries the spirit of spectral clustering. Spectral methods have been successfully in many applications, including computer vision [37, 36], VLSI design [20] and graph partitioning [40]. Basically spectral methods use selected eigenvectors of the data affinity matrix to obtain a data representation that can be easily clustered or embedded in a low-dimensional space [42, 34]. The $W^T (D(D^T D)^{-1} D^T - I_n) W$ used in ASI can be thought of as an affinity matrix with some weighting schemes based on cluster size.

By iteratively updating, ASI performs an implicit adaptive feature selection at each iteration and has some common ideas with adaptive feature selection methods. Ding et al. [14] propose an adaptive dimension reduction clustering algorithm. The basic idea is to adaptively update the initial feature selection based on intermediate results during the clustering process and the process is repeated until the best results are obtained. Domeniconi et al. [15] use a Chi-squared distance analysis to compute a flexible metric for producing neighborhoods that are highly adaptive to query locations. Neighborhoods are elongated along less relevant feature dimensions and constricted along most influential ones.

Since ASI explicitly models the subspace structure at each iteration, it is viewed as an adaptive subspace clustering. CLIQUE [2] is an automatic subspace clustering algorithm for high dimensional spaces. It uses equal-size cells and cell density to find dense re-

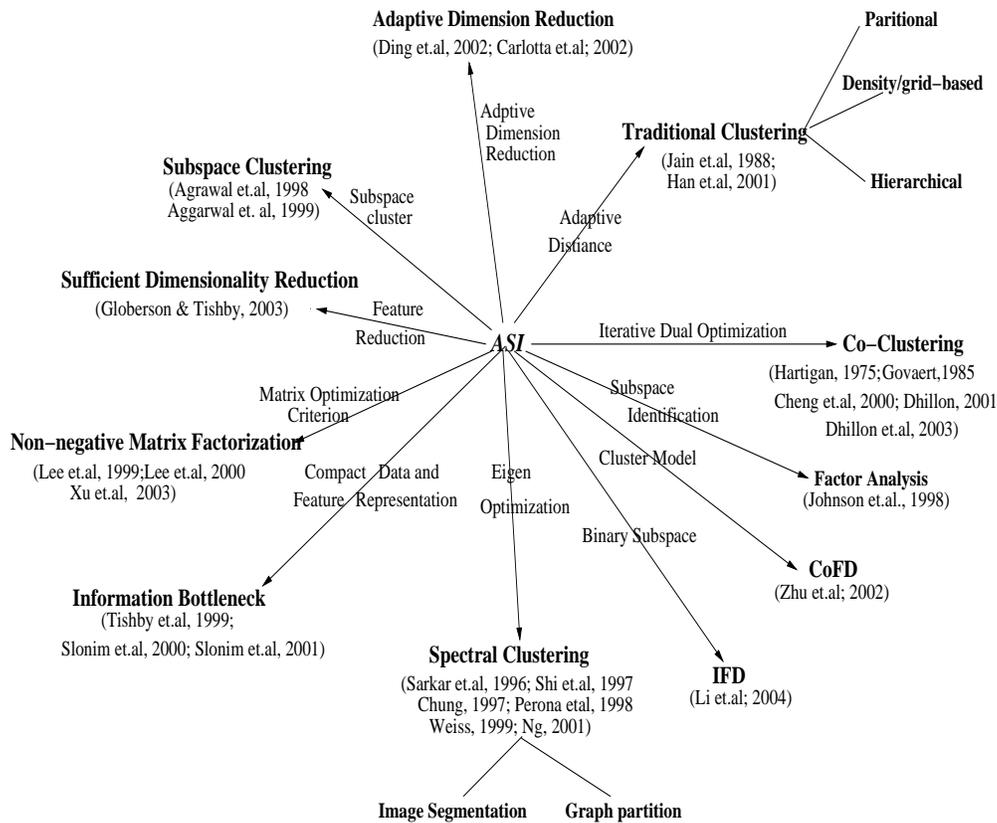


Figure 4: Summary of related work. The arrows show connections.

gions in each subspace in a high dimensional space, where cell size and the density threshold are given as a part of the input. Aggarwal et al. [1] introduce projected clustering and present algorithms for discovering interesting patterns in subspaces of high dimensional spaces. The core idea is a generalization of feature selection which enables selecting different sets of dimensions for different subsets of the data sets. *ASI* adaptively computes the distance measures and the number of dimensions for each class. It also does not require all projected classes to have the same number of dimensions.

ASI also shares many properties with sufficient dimensionality reduction [16] and with non-negative matrix factorization [28, 43].

5. CONCLUSIONS

In this paper, we introduced a new clustering algorithm that allows explicit modeling of the subspace structure associated with each cluster. A key idea of the algorithm is to iteratively perform two alternating procedures: optimization of the subspace structure and updating of the clusters. This is somewhat reminiscent of EM. Experimental results suggested that *ASI* is a viable and competitive clustering algorithm.

Acknowledgments

The authors want to thank Mr. Jieping Ye for providing useful insights on Section 2.3. We are also grateful to the conference reviewers for their helpful comments and suggestions. The first and the third authors are supported in part by NSF grants EIA-0080124 and EIA-0205061 and in part by NIH grant P30-AG18254.

6. REFERENCES

- [1] Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., & Park, J. S. (1999). Fast algorithms for projected clustering. *ACM SIGMOD Conference* (pp. 61–72).
- [2] Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. *ACM SIGMOD Conference* (pp. 94–105).
- [3] Anderberg, M. R. (1973). *Cluster analysis for applications*. Academic Press Inc.
- [4] Berger, M., & Rigoutsos, I. (1991). An algorithm for point clustering and grid generation. *IEEE Trans. on Systems, Man and Cybernetics*, 21, 1278–1286.
- [5] Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is nearest neighbor meaningful? *Proceedings of 7th International Conference on Database Theory(ICDT'99)* (pp. 217–235).
- [6] Bock, H.-H. (1989). Probabilistic aspects in cluster analysis. In O. Opitz (Ed.), *Conceptual and numerical analysis of data*, 12–44. Berlin: Springer-verlag.
- [7] Boley, D., Gini, M., Gross, R., Han, E.-H., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., & Moore, J. (1999). Document categorization and query generation on the world wide web using webace. *AI Review*, 13, 365–391.
- [8] Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., & Freeman, D. (1988). Autoclass: a Bayesian classification system. *Proceedings of the Fifteenth International Conference on Machine Learning(ICML'88)*.
- [9] Cheng, Y., & Church, G. M. Biclustering of expression data.

- In Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)* (pp. 93–103).
- [10] Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. John Wiley and Sons.
- [11] Deuffhard, P., Huisinga, W., Fischer, A., & Schutte, C. (2000). Identification of almost invariant aggregates in reversible nearly coupled markov chain. *Linear Algebra and Its Applications*, 315, 39–59.
- [12] Dhillon, I. (2001). *Co-clustering documents and words using bipartite spectral graph partitioning* (Technical Report 2001-05). Department of Computer Science, University of Texas at Austin.
- [13] Dhillon, I. S., Mallela, S., & Modha, S. S. (2003). Information-theoretic co-clustering. *ACM SIGKDD Conference* (pp. 89–98).
- [14] Ding, C., He, X., Zha, H., & Simon, H. (2002). Adaptive dimension reduction for clustering high dimensional data. *IEEE International Conference on Data Mining (ICDM 2002)* (pp. 107–114).
- [15] Domeniconi, C., Peng, J., & Gunopulos, D. (2002). Locally adaptive metric nearest-neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 1281–1285.
- [16] Globerson, A., & Tishby, N. (2003). Sufficient dimensionality reduction. *J. Mach. Learn. Res.*, 3, 1307–1331.
- [17] Golub, G. H., & Loan, C. F. V. (1991). *Matrix computations*. The Johns Hopkins University Press.
- [18] Govaert, G. (1985). Simultaneous clustering of rows and columns. *Control and Cybernetics*, 437–458.
- [19] Guha, S., Rastogi, R., & Shim, K. (1998). CURE: an efficient clustering algorithm for large databases. *ACM SIGMOD Conference* (pp. 73–84).
- [20] Hagen, L., & Kahng, A. B. (1992). New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. Computer-Aided Design*, 11, 1074–1085.
- [21] Han, E.-H., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., & Moore, J. (1998). WebACE: A web agent for document categorization and exploration. *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)* (pp. 408–415).
- [22] Han, J., & Kamber, M. (2000). *Data mining: Concepts and techniques*. Morgan Kaufmann Publishers.
- [23] Hartigan, J. (1975). *Clustering algorithms*. Wiley.
- [24] Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, prediction*. Springer.
- [25] Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice Hall.
- [26] Johnson, R., & Wichern, D. (1998). *Applied multivariate statistical analysis*. New York: Prentice-Hall.
- [27] Kato, L. (1995). *Perturbation theory for linear operators*. Springer.
- [28] Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 788–791.
- [29] Leung, Y., she Zhang, J., & Xu, Z.-B. (2000). Clustering by scale-space filtering. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 1396–1410.
- [30] Li, T., & Ma, S. (2004). IFD: iterative feature and data clustering. *Proceedings of the 2004 SIAM International conference on Data Mining (SDM 2004)*.
- [31] Li, T., Zhu, S., & Ogihara, M. (2003). Efficient multi-way text categorization via generalized discriminant analysis. *Proceedings of the Twelfth Conference on Information and Knowledge Management (CIKM 2003)* (pp. 317–324).
- [32] Linde, Y., Buzo, A., & Gray, R. (1980). An algorithm for vector quantization design. *IEEE Transactions on Communications*, 28, 84–95.
- [33] McCallum, A. K. (1996). Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. <http://www.cs.cmu.edu/mccallum/bow>.
- [34] Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems 14 (NIPS'01)*.
- [35] Nishisato, S. (1980). *Analysis of categorical data: Dual scaling and its applications*. Toronto: University of Toronto Press.
- [36] Perona, P., & Freeman, W. (1998). A factorization approach to grouping. *Lecture Notes in Computer Science*, 1406, 655–670.
- [37] Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 888–905.
- [38] Slonim, N., & Tishby, N. (1999). Agglomerative information bottleneck. *Advances in Neural Information Processing Systems 12 (NIPS'99)*.
- [39] Slonim, N., & Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. *ACM SIGIR 2000* (pp. 208–215).
- [40] Spielman, D. A., & Teng, S.-H. (1996). Spectral partitioning works: Planar graphs and finite element meshes. *In IEEE Symposium on Foundations of Computer Science* (pp. 96–105).
- [41] Tishby, N., Pereira, F. C., & Bialek, W. The information bottleneck method. *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing* (pp. 368–377).
- [42] Weiss, Y. (1999). Segmentation using eigenvectors: A unifying view. *Proceedings of International Conference on Computer Vision-Volume 2 ICCV (2)* (pp. 975–982).
- [43] Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. *ACM SIGIR 2003* (pp. 267–273).
- [44] Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. *ACM SIGMOD Conference* (pp. 103–114).
- [45] Zhao, Y., & Karypis, G. (2001). *Criterion functions for document clustering: Experiments and analysis* (Technical Report). Department of Computer Science, University of Minnesota.
- [46] Zhao, Y., & Karypis, G. (2002). *Evaluation of hierarchical clustering algorithms for document datasets* (Technical Report). Department of Computer Science, University of Minnesota.
- [47] Zhong, S., & Ghosh, J. (2003). A comparative study of generative models for document clustering. *Proceedings of the workshop on Clustering High Dimensional Data and Its Applications in SIAM Data Mining Conference*.
- [48] Zhu, S., Li, T., & Ogihara, M. (2002). CoFD: An algorithm for non-distance based clustering in high dimensional spaces. *4th International Conference on Data Warehousing and Knowledge Discovery (Dawak 2002)* (pp. 52–62).