Text Mining with Information–Theoretical Clustering

Jacob Kogan[1]
kogan@umbc.edu
Department of Mathematics and Statistics
UMBC, Baltimore, MD 21250
Charles Nicholas
nicholas@umbc.edu
Department of Computer Science and Electrical Engineering
UMBC, Baltimore, MD 21250
and
Vladimir Volkovich
zeev@actcom.co.il
Software Engineering Department
ORT Braude Academic College
Karmiel 21982, Israel

Version #2
May 6, 2003

Abstract

Hybrid information retrieval (IR) algorithms combine different normalization techniques and similarity functions. Hybrid algorithms provide an efficient technique to improve precision and recall (see [3]). Motivated by the success of hybrid IR algorithms we report a hybrid clustering scheme based on algorithms introduced recently in [5] and [7]. We report experiments with data in a vector space or *bag of words* model [2] of dimension 4000. Application of a feature selection technique described in [5] leads to 85% reduction in dimension of the vector space model. Applications of the reported clustering scheme to the data in the "reduced" vector space model outperforms a number of existing clustering algorithms.

[1]author to whom correspondence should be addressed

1 Introduction

A wide variety of clustering algorithms applicable for objects of general nature are already available in the literature (see e.g., [8]). The classical k–means algorithm is probably the most celebrated and widely used general clustering technique. In a number of application, including Text Mining and Information Retrieval, one is interested in clustering sets of normalized vectors. In this paper we suggest a two step clustering procedure that handles unit norm vectors. One of these steps is based on the k–means clustering algorithm.

It is known that the quality of the final partition generated by k–means type clustering algorithms depends on a good choice of the initial partition (see e.g., [5], [4]). The first step of the proposed procedure is the Spherical Principal Directions Partitioning (sPDDP) introduced recently by [5]. This singular value decomposition (SVD) based algorithm clusters l_2 unit vectors and generates "good" initial partitions for the second step of the procedure.

The second step is a modification of a new information–theoretic clustering algorithm recently suggested by [7]. The new algorithm uses the Kullback–Leibler (KL) divergence as a similarity measure between two unit l_1 vectors with non–negative coordinates (for the details we refer the reader to [7]). The algorithm resembles the classical batch k-means algorithm [8], and suffers from similar deficiencies. A modification of the algorithm that remedies some of the deficiencies is proposed in this paper. We shall call the proposed modification the Kullback–Leibler means algorithm, or KL–means.

The outline of the paper is the following. A brief description of the Spherical Principal Directions Partitioning algorithm is provided in Section 2. In Section 3 we describe two clustering algorithms. The first one is the information–theoretic clustering algorithm introduced in [7], and the second one is the KL–means algorithm which combines algorithms introduced independently by [7] and [1]. In Section 4 we describe the data, preprocessing, and relevant results already available in the literature. Section 5 contains results of numerical experiments.

2 Spherical Principal Directions Divisive Partitioning

In this section we briefly describe the singular value decomposition based algorithm. The algorithm approximates a finite set of unit norm vectors $\mathbf{X} \subset \mathbf{R}^n$ by a one dimensional great circle of the unit sphere $\mathbf{S}^{n-1}$. A great circle is represented by an intersection of $\mathbf{S}^{n-1}$ and a two dimensional subspace $\mathbf{P}$ of $\mathbf{R}^n$. The two dimensional approximation is easy to visualize (see Section 5). The proposed algorithm is the following:

Algorithm 2.1 *The Spherical Principal Directions Divisive Partitioning (sPDDP) clustering algorithm.*

1. *Given a set of unit vectors* $\mathbf{X}$ *in* $\mathbf{R}^n$ *determine the two dimensional plane* $\mathbf{P}$ *that approximates* $\mathbf{X}$ *in the "best possible way".*

2. *Project* $\mathbf{X}$ *onto* $\mathbf{P}$*. Denote the projection of the set* $\mathbf{X}$ *by* $\mathbf{Y}$*, and the projection of a vector* $\mathbf{x}$ *by* $\mathbf{y}$ *(note that* $\mathbf{y}$ *is two dimensional).*

3. *If* $\mathbf{y} \neq 0$ *"push"* $\mathbf{y} \in \mathbf{Y}$ *to the great circle, and denote the corresponding vector by* $\mathbf{z} = \dfrac{\mathbf{y}}{\|\mathbf{y}\|}$*. Denote the constructed set by* $\mathbf{Z}$*.*

4. *Partition* $\mathbf{Z}$ *into two clusters* $\mathbf{Z}_1$ *and* $\mathbf{Z}_2$*. Assign projections* $\mathbf{y}$ *with* $\|\mathbf{y}\| = 0$ *to* $\mathbf{Z}_1$*.*

5. *Generate the induced partition* $\{\mathbf{X}_1, \mathbf{X}_2\}$ *of* $\mathbf{X}$ *as follows:*

$$\mathbf{X}_1 = \{\mathbf{x} \ : \ \mathbf{z} \in \mathbf{Z}_1\}, \ \textit{and} \ \mathbf{X}_2 = \{\mathbf{x} \ : \ \mathbf{z} \in \mathbf{Z}_2\}. \tag{1}$$

For technical details concerning sPDDP we refer the reader to [5].

3 Information-theoretic clustering

The method described below follows the ideas introduced by [12]: "Given the empirical joint distribution of two variables, one variable is compressed so that the mutual information about another variable is preserved as much as possible. The method can be considered as finding a minimal sufficient partition or efficient relevant coding of one variable with respect to the other one."

Specifically, let Y be a discrete random variable that takes values from the set $\mathcal{Y} = \{y_1, \ldots, y_m\}$, and let X be a discrete random variable that takes values from the set $\mathcal{X} = \{x_1, \ldots, x_n\}$ (in Text Mining applications $\mathcal{X}$ may denote documents, or phrases in a document collection, $\mathcal{Y}$ may denote words, or terms in the document collection). Let $\Pi_k = \{\pi_1, \ldots, \pi_k\}$, $k \leq n$, be a partition of $\mathcal{X}$. The information about Y captured by Π_k can be measured by the mutual information $I(Y, \Pi_k)$, where the mutual information $I(Y, X)$ between two random variables Y and X with probability distributions $p(y)$ and $p(x)$, and the joint probability distribution $p(y, x)$ is given by

$$I(Y, X) = \sum_{y,x} p(y, x) \log \frac{p(y, x)}{p(x)p(y)}.$$

The inequality

$$I(Y, \Pi_n) - I(Y, \Pi_k) \geq 0 \tag{2}$$

always holds (for details see [7]). We define the quality of a cluster π by

$$q(\pi) = \sum_{x \in \pi} p(x) KL(p(Y|x), p(Y|\pi)).$$

Here the relative entropy, or Kullback-Leibler divergence [10] between two probability distributions $p(x)$ and $q(x)$ is defined as $KL(p,q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$, and the probability distribution $p(Y|\pi)$ is given by

$$p(Y|\pi) = \sum_{x \in \pi} \frac{p(x)}{p(\pi)} p(Y|x) \text{ where } p(\pi) = \sum_{x \in \pi} p(x). \tag{3}$$

The quality of the partition Π_k is given by

$$\mathcal{Q}(\Pi_k) = \sum_{j=1}^{k} q(\pi_j) = \sum_{\pi \in \Pi_k} \sum_{x \in \pi} p(x) KL(p(Y|x), p(Y|\pi)) = I(Y, \Pi_n) - I(Y, \Pi_k). \tag{4}$$

The optimal partition Π_k^o is the one that minimizes (4). The $k-$means type clustering framework for building Π_k^o was introduced independently by [1] and [7]. The iterative process for the batch KL–means proposed in [7] is the following: Let $\Pi_k = \{\pi_1, \ldots, \pi_k\}$ be a partition of $\mathcal{X}$. For $x \in \pi_i \subset \mathcal{X}$ denote by $\texttt{min}(x)$ the index j of the cluster with the "nearest" probability distribution i.e.,

$$KL(p(Y|x), p(Y|\pi_{\texttt{min}(x)})) \leq KL(p(Y|x), p(Y|\pi_l)),\ l = 1, \ldots, k.$$

Define the "next Batch Kullback-Leibler Means" partition $\Pi_k' = \texttt{nextBKLM}(\Pi_k)$ as follows

$$\Pi_k' = \{\pi_1', \ldots, \pi_k'\},\ \pi_j' = \{x\ :\ j = \texttt{min}(x)\}.$$

The algorithm reported in [7] is given next.

Algorithm 3.1 Batch Kullback-Leibler means algorithm.
Given a user supplied tolerance $\texttt{tol} < 0$ *do the following:*

1. *Set* $t = 0$.
2. *Start with an arbitrary partitioning* $\Pi_k^t = \left\{\pi_1^t, \ldots, \pi_k^t\right\}$.
3. *Generate the partition* $\texttt{nextBKLM}(\Pi_k^t)$.
4. *If* $(\mathcal{Q}(\texttt{nextBKLM}(\Pi_k^t)) - \mathcal{Q}(\Pi_k) < \texttt{tol})$
   set $\Pi_k^{t+1} = \texttt{nextBKLM}(\Pi_k^t)$
   increment t *by 1*
   go to Step 3.
5. *Stop.*

It is shown in [7] that Algorithm 3.1 outperforms the agglomerative algorithms given in [12].

The classical batch $k-$means algorithm is prone to local minima [8]. In Section 5 we show that the presented information–theoretical version of the algorithm experiences the same problem. Next we list the reasons that may cause the algorithm to get trapped in a local minimum. Let $\{\pi_1, \pi_2\}$ be a two cluster partition. Assume that $x_1 \in \pi_1$.

1. The decision whether x_1 should be removed from the cluster π_1 and assigned to the cluster π_2 made by Algorithm 3.1 is based on the sign of

$$\Delta = p(x_1)\left[KL(p(Y|x_1), p(Y|\pi_2)) - KL(p(Y|x_1), p(Y|\pi_1))\right]. \tag{5}$$

The variable x_1 should be moved to π_2 if and only if $\Delta \geq 0$.

To compute the change Δ_{exact} in the value of the objective function $\mathcal{Q}$ caused by removal of x_1 from π_1 and assigning it to π_2 we need an additional definition.

Definition 3.1 *For $\pi \subset \mathcal{X}$, and $x_1 \in \pi$, we denote $\pi - \{x_1\}$ by $\pi^-(x_1)$. For $\pi \subset \mathcal{X}$, and $x_1 \notin \pi$, we denote $\pi \bigcup \{x_1\}$ by $\pi^+(x_1)$. When it does not lead to ambiguity we shall abuse notations and denote the sets $\pi^-(x_1)$ and $\pi^+(x_1)$ just by π^- and π^+ respectively.*

The change in the value of the objective function $\mathcal{Q}$ caused by removal of x_1 from π_1 and assigning it to π_2 is

$$\begin{aligned}
\Delta_{\text{exact}} &= \sum_{x\in\pi_1^-} p(x)KL(p(Y|x), p(Y|\pi_1^-)) - \sum_{x\in\pi_1^-} p(x)KL(p(Y|x), p(Y|\pi_1)) \\
&+ \sum_{x\in\pi_2} p(x)KL(p(Y|x), p(Y|\pi_2^+)) - \sum_{x\in\pi_2} p(x)KL(p(Y|x), p(Y|\pi_2)) \\
&+ p(x_1)\left[KL(p(Y|x_1), p(Y|\pi_2^+)) - KL(p(Y|x_1), p(Y|\pi_1))\right].
\end{aligned}$$

In the next paragraph we will need the following "convexity" inequality

$$\sum_{i=1}^{n} \alpha_i KL(p_i, q) \geq \sum_{i=1}^{n} \alpha_i KL\left(p_i, \sum_{i=1}^{n} \alpha_i p_i\right) \tag{6}$$

that holds for a set of probability distributions $p_1, \ldots, p_n, q$ and non–negative weights $\alpha_1, \ldots \alpha_n$, $\sum_{i=1}^{n} \alpha_i = 1$ (see [7] for details).

The difference

$$\begin{aligned}
\Delta_{\text{exact}} - \Delta &= \sum_{x\in\pi_1^-} p(x)\left[KL(p(C|x), p(C|\pi_1^-)) - KL(p(C|x), p(C|\pi_1))\right] \\
&+ \sum_{x\in\pi_2^+} p(x)\left[KL(p(Y|x), p(Y|\pi_2^+)) - KL(p(Y|x), p(Y|\pi_2))\right].
\end{aligned}$$

Due to (6) both terms on the right hand side are non-positive, and one has

$$\Delta \geq \Delta_{\text{exact}}. \tag{7}$$

It is, therefore, possible that while $\Delta \geq 0$, and the batch $k-$means stops, a change of cluster affiliation for x_1 decreases the value of the objective function. While the magnitude

of $\Delta_{\text{exact}} - \Delta$ is usually very small for large clusters, it becomes significant when the clusters are small (see [9], [4] where the problem is analyzed in detail and a remedy is suggested for the classical quadratic and the "spherical" versions of the $k-$means algorithm). It is shown in [4] that the modification leads to a significant improvement of clustering results in the case of small clusters (up to 100 vectors per cluster).

2. The following scenario is an additional reason for a possible "failure" of the algorithm: Suppose that a document vector x_1 contains 10 first document collection terms only. That is, the probability distribution $(p(y_1|x_1), \ldots, p(y_m|x_1))^T$ has 10 first nonzero entries only, i.e.,

$$p(y_i|x_1) > 0,\ i = 1, \ldots, 10,\ \text{and}\ p(y_j|x_1) = 0,\ j = 11, \ldots, m.$$

Suppose also that no document in cluster π_2 contains the first term of x_1, i.e.,

$$p(y_1|x) = 0 \text{ for each } x \in \pi_2,\ \text{and, due to (3), } p(y_1|\pi_2) = 0.$$

In such a case $KL\left(p(Y|x_1), p(Y|\pi_2)\right) = \infty$, and Algorithm 3.1 does **not** move x_1 from π_1 to π_2. If, however, the only document in π_1 containing any of the 10 words of x_1 is x_1 itself (i.e., $p(y_i|x) = 0$, $i = 1, \ldots, 10$, $x \in \pi_1$, and $x \neq x_1$), and the last 9 words of x_1 are contained in documents of π_2 (i.e., $p(y_j|\pi_2) > 0$, $j = 2, \ldots, 10$), then, in these writers' opinion, π_2 is a more appropriate host cluster for x_1 than π_1 is. (See also [11] for discussion of the Kullback-Leibler divergence and its applications to Information Retrieval problems).

The incremental version of the Kullback-Leibler means algorithm has the ability to rectify the drawbacks listed above. To present the algorithm we need additional definitions.

Definition 3.2 *A first variation of a partition $\Pi_k = \{\pi_l\}_{l=1}^k$ is a partition $\Pi'_k = \{\pi'_l\}_{l=1}^k$ obtained from $\{\pi_l\}_{l=1}^k$ by removing a single vector x from a cluster π_i of $\{\pi_l\}_{l=1}^k$ and assigning this vector to an existing cluster π_j of $\{\pi_l\}_{l=1}^k$.*

Next we look for the "steepest descent" first variation, i.e., a first variation that leads to the maximal decrease in the objective function. The formal definition follows:

Definition 3.3 *The partition* $\texttt{nextFV}\left(\Pi_k\right)$ *is a first variation of* Π_k *so that for each first variation* Π'_k *one has* $\mathcal{Q}\left(\texttt{nextFV}\left(\Pi_k\right)\right) \leq \mathcal{Q}\left(\Pi'_k\right)$.

The algorithm presented next is a slight modification of the one described in [1].

Algorithm 3.2 Incremental Kullback-Leibler means algorithm.
Given a user supplied tolerance $\texttt{tol} < 0$ *do the following:*

1. *Set* $t = 0$.

2. *Start with an arbitrary partitioning* $\Pi_k^t = \left\{\pi_1^t, \ldots, \pi_k^t\right\}$.

3. *Generate* $\texttt{nextFV}\,(\Pi_k)$.

4. *If* ($\mathcal{Q}\,(\texttt{nextFV}\,(\Pi_k)) - \mathcal{Q}\,(\Pi_k) < \texttt{tol}$)
   set $\Pi_k^{t+1} = \texttt{nextFV}\,(\Pi_k^t)$
   increment t *by 1*
   go to Step 3.

5. *Stop.*

Excellent clustering results for a small data set (a data matrix of size 197×203) are reported in [1]. While computationally more accurate, Algorithm 3.2 is slower than Algorithm 3.1. Each iteration of the incremental algorithm changes cluster affiliation of a single vector only. The computational complexity of the Algorithm 3.2 is shown to be the same as that of Algorithm 3.1 (see [1], where generation of first variation partitions and associated computations are described).

We next describe an algorithm that benefits from the speed of Algorithm 3.1 and the accuracy of Algorithm 3.2.

Algorithm 3.3 KL–means clustering algorithm.
Given user supplied tolerances $\texttt{tol}_\texttt{B} \leq 0$ *and* $\texttt{tol}_\texttt{I} \leq 0$

1. *Set* $t = 0$.

2. *Start with an arbitrary partitioning* $\Pi_k^t = \left\{\pi_1^t, \ldots, \pi_k^t\right\}$.

3. *while* $(\mathcal{Q}\,(\texttt{nextBKLM}\,(\Pi_k^t)) - \mathcal{Q}\,(\Pi_k^t) < \texttt{tol}_\texttt{B})$
   set $\Pi_k^{t+1} = \texttt{nextBKLM}\,(\Pi_k^t)$
   increment t *by 1*

4. *If* $(\mathcal{Q}\,(\texttt{nextFV}\,(\Pi_k^t)) - \mathcal{Q}\,(\Pi_k^t) < \texttt{tol}_\texttt{I})$
   set $\Pi_k^{t+1} = \texttt{nextFV}\,(\Pi_k^t)$
   increment t *by 1*
   go to Step 3.

5. *Stop.*

In the next section we present the clustering results already available in the literature, and describe the hybrid clustering procedure which is the main contribution of this paper.

4 Text data and vector space model

Dhillon and Modha [6] have recently used the spherical k–means algorithm for clustering text data. In one of their experiments the algorithm was applied to a data set containing 3893 documents [6]. The data set contains the following three document collections (available from `ftp://ftp.cs.cornell.edu/pub/smart`):

- Medlars Collection (1033 medical abstracts),
- CISI Collection (1460 information science abstracts),
- Cranfield Collection (1400 aerodynamics abstracts).

Partitioning the entire collection into 3 clusters generates the "confusion" matrix reported in Table 1 (see [6]). The entry ij in the Table 1 is the number of documents that belong to cluster i

	Medlars	CISI	Cranfield
cluster 0	1004	5	4
cluster 1	18	1440	16
cluster 2	11	15	1380

Table 1: spherical k–means generated "confusion" matrix with **69** "misclassified" documents using 4099 terms

and document collection j. The "confusion" matrix shows that only 69 documents (i.e., less that 2% of the entire collection) have been "misclassified" by the algorithm. After removing stopwords Dhillon and Modha [6] reported 24,574 unique words, and after eliminating low–frequency and high–frequency words they selected **4,099** words to construct the vector space model.

Our data set is a merger of the three document collections:

- DC0 (Medlars Collection 1033 medical abstracts)
- DC1 (CISI Collection 1460 information science abstracts)
- DC2 (Cranfield Collection 1398 aerodynamics abstracts)

(the documents are available from `http://www.cs.utk.edu/~lsi/`). The Cranfield collection tackled by Dhillon and Modha contained two empty documents. These two documents have been removed from DC2. The other document collections are identical.

We denote the overall collection of 3891 documents by DC. After stopword removal (see `ftp://ftp.cs.cornell.edu/pub/smart/english.stop`), and Porter stemming the data set contains 15,864 unique terms. (No stemming was applied to the 24,574 unique words reported in [6]).

To exploit statistics of term occurrence throughout the corpus we remove terms that occur in less than r sentences across the collection, and denote the remaining terms by slice(r). (Although r should be collection dependent, the experiments in this chapter are performed with $r = 20$). The first l best quality terms that belong to slice(r) define the dimension of the vector space model.

We denote the frequency of a term $\mathbf{t}$ in the document $\mathbf{d}_j$ by f_j. Following an approach outlined in [5] we measure the quality of the term $\mathbf{t}$ by

$$\sum_{j=1}^{n_0} f_j^2 - \frac{1}{n_0}\left[\sum_{j=1}^{n_0} f_j\right]^2, \tag{8}$$

where n_0 is the total number of documents in the collection. Note that the quality of a term is proportional to the term frequency variance.

To evaluate the impact of feature selection based on the quality function (8) on clustering we conducted the following experiment. The best quality **600** terms are selected, and l_2 unit norm vectors for the 3891 documents are built (we use the `tfn` scheme to construct document vectors, for details see [6]). A two step procedure is employed to partition the 3891 vectors into 3 clusters:

1. the sPDDP algorithm (see Section 2) is applied to generate 3 clusters (the obtained clusters are used as an initial partition in the next step),

2. the vectors are normalized in l_1 norm, and the KM–means algorithm (see Section 3) is applied to the partition obtained in the previous step.

The results of the experiment with $\texttt{tol}_\texttt{B} = \texttt{tol}_\texttt{I} = 0$ are provided in the next section.

5 Clustering results

In what follows we display clustering results for the document collection DC described in Section 4 and compare the results with those reported by [6] and [5]. An application of sPDDP to the entire data set generates the first two clusters (see Figure 1, where two dimensional projections of Medlars, CISI and Cranfield document vectors are denoted by red '*', green 'o', and blue '+' respectively). Projections of the clusters on the two dimensional plane $\mathbf{P}$ (the left part of Figure 1) are "pushed" to the unit circle (see Algorithm 2.1) and partitioned into two clusters (see the right part of Figure 1). The black solid line indicates the partition of the points on the unit circle. To illustrate the induced partition of the two dimensional projections we also show this line on the left part of the figure. One can clearly identify three clusters, and the first application of sPDDP separates one cluster from the other two. The second application of sPDDP to the

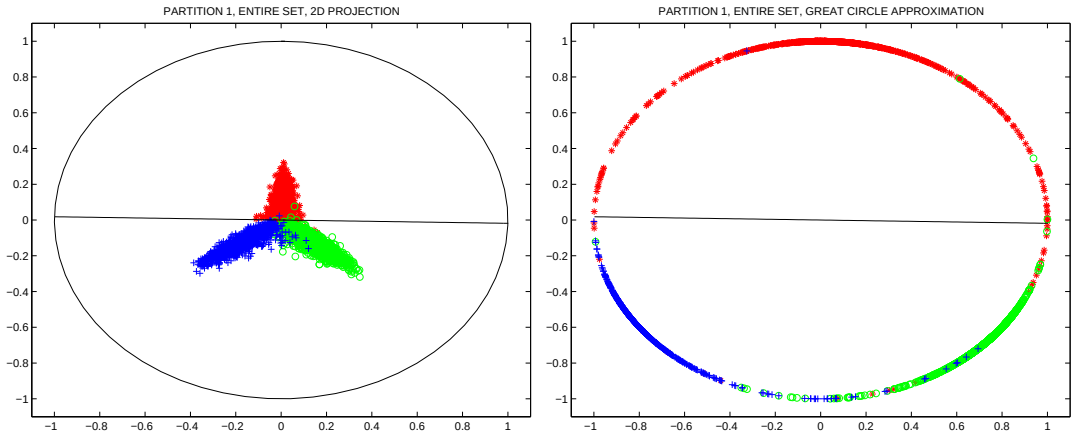


Figure 1: sPDDP generated two cluster partition of the entire data set

largest cluster generated by the first application recomputes the plane $\mathbf{P}$ and splits the cluster (see Figure 2). The confusion matrix for the three cluster partition generated by sPDDP is

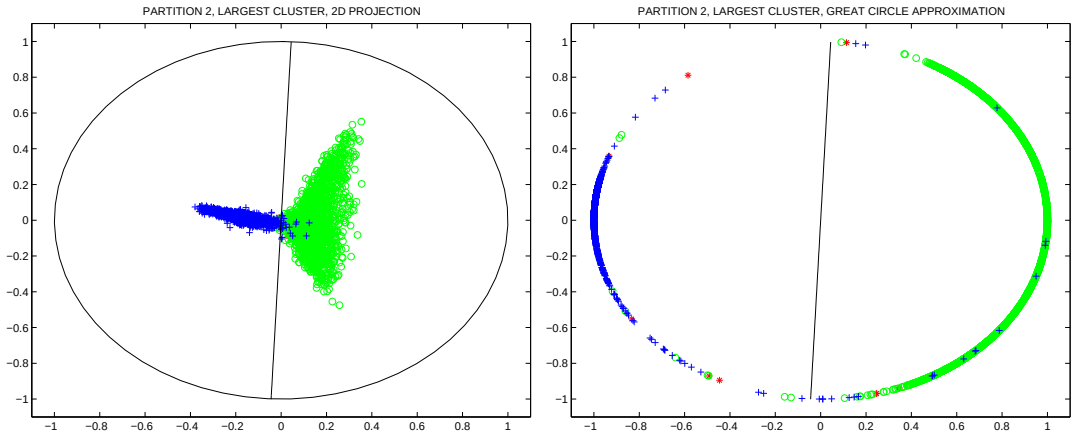


Figure 2: sPDDP generated two cluster partition of the largest cluster

given in Table 2 below. Note that there is no *a priori* connection between document collection i and cluster i. Hence, one can not expect the "confusion" matrix to have diagonal structure unless rows (or columns) of the matrix are suitably permuted. A good clustering procedure should be able to produce a "confusion" matrix with a single "dominant" entry in each row. The "confusion" matrices for the three clusters provided in Tables 2 and 3 illustrate this remark.

When the number of terms is relatively small, some documents may contain no selected terms, and their corresponding vectors are zeros. We always remove these vectors ahead of clustering and assign the "empty" documents into a special cluster. This cluster is the last row in the "confusion" matrix (and is empty in the experiment reported in Tables 2 and 3).

While the number of selected terms is only 600 (i.e., only about 15% of the number of terms

	DC0	DC1	DC2
cluster 0	1000	3	1
cluster 1	8	10	1376
cluster 2	25	1447	21
"empty" documents			
cluster 3	0	0	0

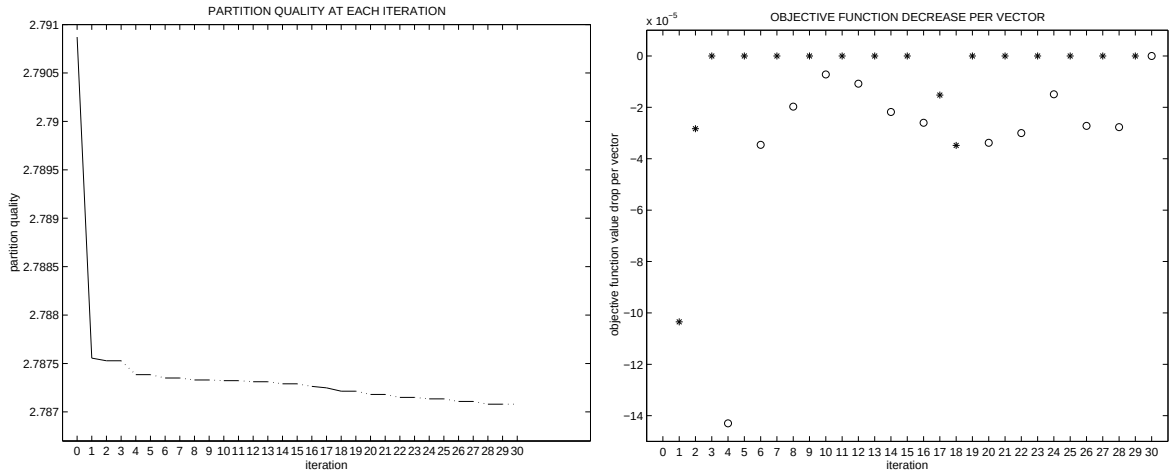
Table 2: sPDDP generated initial "confusion" matrix with **68** "misclassified" documents using best 600 terms

reported in [6]) the quality of sPDDP generated confusion matrix is comparable with that of the confusion matrix generated by the spherical k–means algorithm (see Table 1). A subsequent application of the KL–means algorithm to the partition generated by sPDDP further improves the confusion matrix (see Table 3).

	DC0	DC1	DC2
cluster 0	1010	6	0
cluster 1	3	4	1387
cluster 2	20	1450	11
"empty" documents			
cluster 3	0	0	0

Table 3: KL–means generated final "confusion" matrix with **44** "misclassified" documents using best 600 terms

We pause briefly to analyze performance of the KL–means algorithm. In the experiment described above the algorithm performs 30 iterations.



The left figure shows the values of $\mathcal{Q}(\Pi_k^t)$, $t = 0, 1, \ldots, 30$. The solid line shows changes in the objective function caused by batch KL-means iterations, and the dotted line does the same for incremental KL-means iterations. The figure shows that the lion's share of the work is done by the first 3 batch KL-means iterations. From iteration 4 to iteration 16 values of the objective function drop due to incremental KL-means iterations only. At iterations 17, 18 and 19 the batch KL-means kicks in. For the rest of the run the objective function changes are due to incremental KL-means iterations only.

The right figure shows the drop of the objective function values per number of vectors reshuffled between clusters due to an iteration (we shall call it a "relative drop", and when no vector changes its cluster affiliation we report relative drop 0). Stars indicate relative drop due to batch KL-means iterations, and circles show relative drop due to incremental KL-means iterations. We observe that the relative drop due to batch KL-means becomes 0 at iteration 3, and the maximal relative drop occurs at iteration 4 due to an incremental KL-means iteration. An inspection reveals that at this iteration a vector x was moved from cluster π_3 to cluster π_2, and x was "missed" by the batch KL-means iteration 3 because

$$KL(p(Y|x), p(Y|\pi_1)) = \infty, \text{ and } KL(p(Y|x), p(Y|\pi_2)) = \infty.$$

For the very same reason vectors were "missed" by the batch KL-means algorithm at iterations 5 and 19.

Table 4 summarizes clustering results for sPDDP algorithm and the combinations "sPDDP+k–means", and "sPDDP+ KL–means" algorithms for different choices of index terms. Note that zero document vectors are created when the number of selected terms is less than 300. Table 4

		documents misclassified by sPDDP		
# of terms	zero vectors	alone	and k–means	and KL–means
100	12	383	258	168
200	3	277	133	116
300	0	228	100	81
400	0	88	80	56
500	0	76	62	40
600	0	68	62	44

Table 4: Number of documents "misclassified" by sPDDP, "sPDDP+k–means", and "sPDDP+ KL–means".

indicates consistent superiority of the "sPDDP+ KL–means" algorithm.

Acknowledgments. The authors thank Jim X. Chen and Penny Rheingans for valuable suggestions that improved exposition of the results. The work of Kogan and Nicholas was supported in part by the US Department of Defense.

References

[1] P. Berkhin and Becher J. D. Learning simple relations: Theory and applications. In *Proc. Second SIAM International Conference on Data Mining*, pages 420–436, Arlington, April 2002.

[2] M. Berry and M. Browne. *Understanding Search Engines.* SIAM, 1999.

[3] M. Cornelson, E. Greengrass, R. L. Grossman, R. Karidi, and D. Shnidman. Combining information retrieval algorithms using machine learning. In *Proceedings of the SIAM KDD Workshop on Text Mining*, to appear.

[4] I. S. Dhillon, Y. Guan, and J. Kogan. Refining clusters in high-dimensional text data. In I. S. Dhillon and J. Kogan, editors, *Proceedings of the Workshop on Clustering High Dimensional Data and its Applications at the Second SIAM International Conference on Data Mining*, pages 71–82. SIAM, 2002.

[5] I. S. Dhillon, J. Kogan, and C. Nicholas. Feature selection and document clustering. In M.W. Berry, editor, *A Comprehensive Survey of Text Mining.* Springer-Verlag, 2003, to appear.

[6] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, January 2001. Also appears as IBM Research Report RJ 10147, July 1999.

[7] Inderjit S. Dhillon, Subramanyam Mallela, and Rahul Kumar. Enhanced word clustering for hierarchical text classification. In *Proceedings of the The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD-2002)*, pages 191–200, 2002.

[8] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley & Sons, second edition, 2000.

[9] J. Kogan. Means clustering for text data. In M.W.Berry, editor, *Proceedings of the Workshop on Text Mining at the First SIAM International Conference on Data Mining*, pages 47–54, 2001.

[10] S. Kullback and Leibler R.A. On information and sufficiency. *Journal of Mathematical Analysis and Applications*, 22:79–86, 1951.

[11] M. Larkey, L. Connell and J. Callan. Collection selection and results merging with topically organized U.S. patents and TREC data. In *Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM)*, pages 282–289, McLean, VA, 2000. ACM.

[12] N. Slonim and N. Tishby. The power of word clusters for text classification. In *23rd European Colloquium on Information Retrieval Research (ECIR)*, Darmstadt, 2001.