Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 01-040

Criterion Functions for Document Clustering: Experiments and Analysis

Ying Zhao and George Karypis

November 29, 2001

Criterion Functions for Document Clustering*
Experiments and Analysis

Ying Zhao and George Karypis

University of Minnesota, Department of Computer Science / Army HPC Research Center
Minneapolis, MN 55455
Technical Report #01-40

{yzhao, karypis}@cs.umn.edu

Last updated on November 29, 2001 at 12:33am

Abstract

In recent years, we have witnessed a tremendous growth in the volume of text documents available on the Internet, digital libraries, news sources, and company-wide intranets. This has led to an increased interest in developing methods that can help users to effectively navigate, summarize, and organize this information with the ultimate goal of helping them to find what they are looking for. Fast and high-quality document clustering algorithms play an important role towards this goal as they have been shown to provide both an intuitive navigation/browsing mechanism by organizing large amounts of information into a small number of meaningful clusters as well as to greatly improve the retrieval performance either via cluster-driven dimensionality reduction, term-weighting, or query expansion. This ever-increasing importance of document clustering and the expanded range of its applications led to the development of a number of new and novel algorithms with different complexity-quality trade-offs. Among them, a class of clustering algorithms that have relatively low computational requirements are those that treat the clustering problem as an optimization process which seeks to maximize or minimize a particular *clustering criterion function* defined over the entire clustering solution.

The focus of this paper is to evaluate the performance of different criterion functions for the problem of clustering documents. Our study involves a total of eight different criterion functions, three of which are introduced in this paper and five that have been proposed in the past. Our evaluation consists of both a comprehensive experimental evaluation involving fifteen different datasets, as well as an analysis of the characteristics of the various criterion functions and their effect on the clusters they produce. Our experimental results show that there are a set of criterion functions that consistently outperform the rest, and that some of the newly proposed criterion function lead to the best overall results. Our theoretical analysis of the criterion function shows that their relative performance depends on (i) the degree to which they can correctly operate when the clusters are of different tightness, and (ii) the degree to which they can lead to reasonably balanced clusters.

1 Introduction

The topic of clustering has been extensively studied in many scientific disciplines and over the years a variety of different algorithms have been developed [31, 22, 6, 27, 20, 35, 2, 48, 13, 43, 14, 15, 24]. Two recent surveys on

*This work was supported by NSF CCR-9972519, EIA-9986042, ACI-9982274, by Army Research Office contract DA/DAAG55-98-1-0441, by the DOE ASCI program, and by Army High Performance Computing Research Center contract number DAAH04-95-C-0008. Related papers are available via WWW at URL: **http://www.cs.umn.edu/~karypis**

the topics [21, 18] offer a comprehensive summary of the different applications and algorithms. These algorithms can be categorized along different dimensions based either on the underlying methodology of the algorithm, leading to *agglomerative* or *partitional* approaches, or based on the structure of the final solution, leading to *hierarchical* or *non-hierarchical* solutions.

Agglomerative algorithms find the clusters by initially assigning each object to its own cluster and then repeatedly merging pairs of clusters until a certain stopping criterion is met. A number of different methods have been proposed for determining the next pair of clusters to be merged, such as group average (UPGMA) [22], single-link [38], complete link [28], CURE [14], ROCK [15], and CHAMELEON [24]. Hierarchical algorithms produce a clustering that forms a dendrogram, with a single all inclusive cluster at the top and single-point clusters at the leaves. On the other hand, partitional algorithms, such as K-means [33, 22], K-medoids [22, 27, 35], Autoclass [8, 6], graph-partitioning-based [45, 22, 17, 40], or spectral-partitioning-based [5, 11], find the clusters by partitioning the entire dataset into either a predetermined or an automatically derived number of clusters. Depending on the particular algorithm, a k-way clustering solution can be obtained either directly, or via a sequence of repeated bisections. In the former case, there is in general no relation between the clustering solutions produced at different levels of granularity, whereas the later case gives rise to hierarchical solutions.

In recent years, various researchers have recognized that partitional clustering algorithms are well-suited for clustering large document datasets due to their relatively low computational requirements [7, 30, 1, 39]. A key characteristic of many partitional clustering algorithms is that they use a global criterion function whose optimization drives the entire clustering process[1]. For some of these algorithms the criterion function is implicit (*e.g.*, PDDP), whereas for other algorithms (*e.g*, K-means and Autoclass) the criterion function is explicit and can be easily stated. This later class of algorithms can be thought of as consisting of two key components. First is the criterion function that needs to be optimized by the clustering solution, and second is the actual algorithm that achieves this optimization. These two components are largely independent of each other.

The focus of this paper is to study the suitability of different criterion functions to the problem of clustering document datasets. In particular, we evaluate a total of eight different criterion functions that measure various aspects of intra-cluster similarity, inter-cluster dissimilarity, and their combinations. These criterion functions utilize different views of the underlying collection, by either modeling the documents as vectors in a high dimensional space, or by modeling the collection as a graph. We experimentally evaluated the performance of these criterion functions using 15 different data sets obtained from various sources. Our experiments showed that different criterion functions do lead to substantially different results, and that there are a set of criterion functions that produce the best clustering solutions.

Our analysis of the different criterion functions shows that their overall performance depends on the degree to which they can correctly operate when the dataset contains clusters of different densities (*i.e.*, they contain documents whose pairwise similarities are different) and the degree to which they can produce balanced clusters. Moreover, our analysis also shows that the sensitivity to the difference in the cluster densities can also explain an outcome of our study (that was also observed in earlier results reported in [39]), that for some clustering algorithms the solution obtained by performing a sequence of repeated bisections is better (and for some criterion functions by a considerable amount) than the solution obtained by computing the clustering directly. When the solution is computed via repeated bisections, the difference in density between the two clusters that are discovered is in general smaller than the density differences between all the clusters. As a result, clustering algorithms that cannot handle well variation in cluster density tend to perform substantially better when used to compute the clustering via repeated bisections.

The rest this paper is organized as follows. Section 2 provides some information on how documents are represented and how the similarity or distance between documents is computed. Section 3 describes the different criterion functions as well as the algorithms used to optimize them. Section 4 provides the detailed experimental evaluation of the various criterion functions. Section 5 analyzes the different criterion functions and explains their performance. Finally, Section 6 provides some concluding remarks and directions of future research.

[1] Global clustering criterion functions are not an inherent feature of partitional clustering algorithms but they can also be used in the context of agglomerative algorithms.

2 Preliminaries

Document Representation The various clustering algorithms that are described in this paper use the vector-space model [37] to represent each document. In this model, each document d is considered to be a vector in the term-space. In its simplest form, each document is represented by the *term-frequency* (TF) vector

$$d_{tf} = (tf_1, tf_2, \ldots, tf_m),$$

where tf_i is the frequency of the ith term in the document. A widely used refinement to this model is to weight each term based on its *inverse document frequency* (IDF) in the document collection. The motivation behind this weighting is that terms appearing frequently in many documents have limited discrimination power, and for this reason they need to be de-emphasized. This is commonly done [37] by multiplying the frequency of each term i by $\log(N/df_i)$, where N is the total number of documents in the collection, and df_i is the number of documents that contain the ith term (*i.e.*, document frequency). This leads to the *tf-idf* representation of the document, *i.e.*,

$$d_{tfidf} = (tf_1 \log(N/df_1), tf_2 \log(N/df_2), \ldots, tf_m \log(N/df_m)).$$

To account for documents of different lengths, the length of each document vector is normalized so that it is of unit length ($\|d_{tfidf}\| = 1$), that is each document is a vector in the unit hypersphere. In the rest of the paper, we will assume that the vector representation for each document has been weighted using *tf-idf* and it has been normalized so that it is of unit length.

Similarity Measures Over the years, two prominent ways have been proposed to compute the similarity between two documents d_i and d_j. The first method is based on the commonly used cosine function [37] given by

$$\cos(d_i, d_j) = \frac{{d_i}^t d_j}{\|d_i\| \|d_j\|}, \tag{1}$$

and since the document vectors are of unit length, the above formula simplifies to $\cos(d_i, d_j) = {d_i}^t d_j$. This measure becomes one if the documents are identical, and zero if there is nothing in common between them (*i.e.*, the vectors are orthogonal to each other). The second method computes the similarity between the documents using the Euclidean distance, give by

$$\text{dis}(d_i, d_j) = \sqrt{(d_i - d_j)^t (d_i - d_j)} = \|d_i - d_j\|. \tag{2}$$

If the distance is zero, then the documents are identical, and if there is nothing in common between their distance is $\sqrt{2}$. Note that besides the fact that one measures similarity and the other measures distance, these measures are quite similar to each other because the document vectors are of unit length.

Definitions Through-out this paper we will use the symbols n, m, and k to denote the number of documents, the number of terms, and the number of clusters, respectively. We will use the symbol S to denote the set of n documents that we want to cluster, $S_1, S_2, \ldots, S_k$ to denote each one of the k clusters, and $n_1, n_2, \ldots, n_k$ to denote the sizes of the corresponding clusters.

Given a set A of documents and their corresponding vector representations, we define the ***composite*** vector D_A to be

$$D_A = \sum_{d \in A} d, \tag{3}$$

and the ***centroid*** vector C_A to be

$$C_A = \frac{D_A}{|A|}. \tag{4}$$

The composite vector D_A is nothing more than the sum of all documents vectors in A, and the centroid C_A is nothing more than the vector obtained by averaging the weights of the various terms present in the documents of A. Note that

even though the document vectors are of length one, the centroid vectors will not necessarily be of unit length.

Vector Properties By using the cosine function as the measure of similarity between documents we can take advantage of a number of properties involving the composite and centroid vectors of a set of documents. In particular, if S_i and S_j are two sets of unit-length documents containing n_i and n_j documents respectively, and D_i, D_j and C_i, C_j are their corresponding composite and centroid vectors then the following is true:

1. The sum of the pair-wise similarities between the documents in S_i and the document in S_j is equal to ${D_i}^t D_j$. That is,
$$\sum_{d_q \in D_i, d_r \in D_j} \cos(d_q, d_r) = \sum_{d_q \in D_i, d_r \in D_j} {d_q}^t d_r = {D_i}^t D_j. \tag{5}$$

2. The sum of the pair-wise similarities between the documents in S_i is equal to $\|D_i\|^2$. That is,
$$\sum_{d_q, d_r \in D_i} \cos(d_q, d_r) = \sum_{d_q, d_r \in D_i} {d_q}^t d_r = {D_i}^t D_i = \|D_i\|^2. \tag{6}$$

   Note that this equation includes the pairwise similarities involving the same pairs of vectors.

3 Document Clustering

At a high-level the problem of of clustering is defined as follows. Given a set S of n documents, we would like to partition them into a pre-determined number of k subsets $S_1, S_2, \ldots, S_k$, such that the documents assigned to each subset are more similar to each other than the documents assigned to different subsets.

As discussed in the introduction, our focus in this paper is to study the suitability of various clustering criterion functions in the context of partitional document clustering algorithms. Consequently, the clustering problem becomes that of given a particular clustering criterion function $\mathcal{C}$, compute a k-way clustering solution such that the value of $\mathcal{C}$ is optimized. In the rest of this section we first present a number of different criterion functions that can be used to both evaluate and drive the clustering process, followed by a description of our optimization algorithms.

3.1 Clustering Criterion Functions

3.1.1 Internal Criterion Functions

This class of clustering criterion functions focuses on producing a clustering solution that optimizes a particular criterion function that is defined over the documents that are part of each cluster and does not take into account the documents assigned to different clusters. Due to this intra-cluster view of the clustering process we will refer to these criterion functions as ***internal***.

The first internal criterion function that we will study maximizes the sum of the average pairwise similarities between the documents assigned to each cluster, weighted according to the size of each cluster. Specifically, if we use the cosine function to measure the similarity between documents, then we want the clustering solution to optimize the following criterion function:

$$\text{maximize } \mathcal{I}_1 = \sum_{r=1}^{k} n_r \left(\frac{1}{n_r^2} \sum_{d_i, d_j \in S_r} \cos(d_i, d_j) \right). \tag{7}$$

By using Equation 6, the above formula can be re-written as:

$$\mathcal{I}_1 = \sum_{r=1}^{k} \frac{\|D_r\|^2}{n_r}.$$

Note that our definition of $\mathcal{I}_1$ includes the self-similarities between the documents of each cluster. The $\mathcal{I}_1$ criterion

function is similar to that used in the context of hierarchical agglomerative clustering that uses the group-average heuristic to determine which pair of clusters to merge next.

The second criterion function that we will study is used by the popular vector-space variant of the K-means algorithm [7, 30, 10, 39, 23]. In this algorithm each cluster is represented by its centroid vector and the goal is to find the clustering solution that maximizes the similarity between each document and the centroid of the cluster that is assigned to. Specifically, if we use the cosine function to measure the similarity between a document and a centroid, then the criterion function becomes the following:

$$\text{maximize } \mathcal{I}_2 = \sum_{r=1}^{k} \sum_{d_i \in S_r} \cos(d_i, C_r). \tag{8}$$

This formula can be re-written as follows:

$$\mathcal{I}_2 = \sum_{r=1}^{k} \sum_{d_i \in S_r} \frac{{d_i}^t C_r}{\|C_r\|} = \sum_{r=1}^{k} \frac{{D_r}^t C_r}{\|C_r\|} = \sum_{r=1}^{k} \frac{{D_r}^t D_r}{\|D_r\|} = \sum_{r=1}^{k} \|D_r\|.$$

Comparing the $\mathcal{I}_2$ criterion function with $\mathcal{I}_1$ we can see that the essential difference between these criterion functions is that $\mathcal{I}_2$ scales the within-cluster similarity by the $\|D_r\|$ term as opposed to n_r term used by $\mathcal{I}_1$. The term $\|D_r\|$ is nothing more than the square-root of the pairwise similarity between all the document in S_r, and will tend to emphasize the importance of clusters (beyond the $\|D_r\|^2$ term) whose documents have smaller pairwise similarities compared to clusters with higher pair-wise similarities. Also note that if the similarity between a document and the centroid vector of its cluster is defined as just the dot-product of these vectors, then we will get back the $\mathcal{I}_1$ criterion function.

Finally, the last internal criterion function that we will study is that used by the traditional K-means algorithm. This criterion function uses the Euclidean distance to determine which documents should be clustered together, and determines the overall quality of the clustering solution by using the *sum-of-squared-errors* function. In particular, this criterion is defined as follows:

$$\text{minimize } \mathcal{I}_3 = \sum_{r=1}^{k} \sum_{d_i \in S_r} \|d_i - C_r\|^2. \tag{9}$$

Note that by some simple algebraic manipulations [12], the above equation can be rewritten as:

$$\mathcal{I}_3 = \sum_{r=1}^{k} \frac{1}{n_r} \sum_{d_i, d_j \in S_r} \|d_i - d_j\|^2, \tag{10}$$

that is similar in nature to the $\mathcal{I}_1$ criterion function but instead of using similarities it is expressed in terms of squared distances.

3.1.2 External Criterion Functions

Unlike internal criterion functions, external criterion functions derive the clustering solution by focusing on optimizing a function that is based on how the various clusters are different from each other. Due to this inter-cluster view of the clustering process we will refer to these criterion functions as ***external***.

It is quite hard to define external criterion functions that lead to meaningful clustering solutions. For example, it may appear that an intuitive external function may be derived by requiring that the centroid vectors of the different clusters are as mutually orthogonal as possible, *i.e.*, they contain documents that share very few terms across the different clusters. However, for many problems this criterion function has trivial solutions that can be achieved by assigning to the first $k-1$ clusters a single document that shares very few terms with the rest, and then assigning the rest of the documents to the kth cluster.

For this reason, the external function that we will study tries to separate the documents of each cluster from the entire collection, as opposed trying to separate the documents among the different clusters. In particular, our external

criterion function is defined as

$$\text{minimize} \quad \sum_{r=1}^{k} n_r \cos(C_r, C), \tag{11}$$

where C is the centroid vector of the entire collection. From this equation we can see that we try to minimize the cosine between the centroid vector of each cluster to the centroid vector of the entire collection. By minimizing the cosine we essentially try to increase the angle between them as much as possible. Also note that the contribution of each cluster is weighted based on the cluster size, so that larger clusters will weight heavier in the overall clustering solution. This external criterion function was motivated by multiple discriminant analysis and is similar to minimizing the trace of the between-cluster scatter matrix [12, 41]. Equation 11 can be re-written as

$$\sum_{r=1}^{k} n_r \cos(C_r, C) = \sum_{r=1}^{k} n_r \frac{{C_r}^t C}{\|C_r\|\|C\|} = \sum_{r=1}^{k} n_r \frac{{D_r}^t D}{\|D_r\|\|D\|} = \frac{1}{\|D\|} \left(\sum_{r=1}^{k} n_r \frac{{D_r}^t D}{\|D_r\|} \right),$$

where D is the composite vector of the entire document collection. Note that since $1/\|D\|$ is constant irrespective of the clustering solution the criterion function can be re-stated as:

$$\text{minimize} \quad \mathcal{E}_1 = \sum_{r=1}^{k} n_r \frac{{D_r}^t D}{\|D_r\|}. \tag{12}$$

As we can see from Equation 12, even-though our initial motivation was to define an external criterion function, because we used the cosine function to measure the separation between the cluster and the entire collection, the criterion function does take into account the within-cluster similarity of the documents (due to the $\|D_r\|$ term). Thus, $\mathcal{E}_1$ is actually a hybrid criterion function that combines both external as well as internal characteristics of the clusters.

Another external criterion function can be defined with respect to the Euclidean distance function and the squared-errors of the centroid vectors as follows:

$$\text{maximize} \quad \mathcal{E}_2 = \sum_{r=1}^{k} n_r \|C_r - C\|^2. \tag{13}$$

However, it can be shown that maximizing $\mathcal{E}_2$ is identical to minimizing $\mathcal{I}_3$ [12], and we will not consider it any further.

3.1.3 Hybrid Criterion Functions

The various criterion functions we described so far focused only on optimizing a single criterion function the was either defined in terms on how documents assigned to each cluster are related together, or on how the documents assigned to each cluster are related with the entire collection. In the first case, they tried to maximize various measures of similarity over the documents in each cluster, and in the second case, they tried to minimize the similarity between the cluster's documents and the collection. However, the various clustering criterion function can be combined to define a set of ***hybrid*** criterion functions that simultaneously optimize multiple individual criterion functions.

In our study, we will focus on two hybrid criterion function that are obtained by combining criterion $\mathcal{I}_1$ with $\mathcal{E}_1$, and $\mathcal{I}_2$ with $\mathcal{E}_1$, respectively. Formally, the first criterion function is

$$\text{maximize} \quad \mathcal{H}_1 = \frac{\mathcal{I}_1}{\mathcal{E}_1} = \frac{\sum_{r=1}^{k} \|D_r\|^2/n_r}{\sum_{r=1}^{k} n_r {D_r}^t D/\|D_r\|}, \tag{14}$$

and the second is

$$\text{maximize} \quad \mathcal{H}_2 = \frac{\mathcal{I}_2}{\mathcal{E}_1} = \frac{\sum_{r=1}^{k} \|D_r\|}{\sum_{r=1}^{k} n_r {D_r}^t D/\|D_r\|}. \tag{15}$$

Note that since $\mathcal{E}_1$ is minimized, both $\mathcal{H}_1$ and $\mathcal{H}_2$ need to be maximized as they are inversely related to $\mathcal{E}_1$.

3.1.4 Graph Based Criterion Functions

The various criterion functions that we described so far, view each document as a multidimensional vector. An alternate way of viewing the relations between the documents is to use graphs. In particular, two types of graphs have been proposed for modeling the document in the context of clustering. The first graph is nothing more than the graph obtained by computing the pair-wise similarities between the documents, and the second graph is obtained by viewing the documents and the terms as a bipartite graph.

Given a collection of n documents S, the similarity graph G_s is obtained by modeling each document as a vertex, and having an edge between each pair of vertices whose weight is equal to the similarity between the corresponding documents. Viewing the documents in this fashion, a number of internal, external, or combined criterion functions can be defined that measure the overall clustering quality. In our study we will investigate one such criterion function called MinMaxCut, that was proposed recently [11]. MinMaxCut falls under the category of criterion functions that combine both the internal and external views of the clustering process and is defined as [11]

$$\text{minimize} \quad \sum_{r=1}^{k} \frac{\text{cut}(S_r, S - S_r)}{\sum_{d_i, d_j \in S_r} \text{sim}(d_i, d_j)},$$

where $\text{cut}(S_r, S - S_r)$ is the edge-cut between the vertices in S_r to the rest of the vertices in the graph $S - S_r$. The edge-cut between two sets of vertices A and B is defined to be the sum of the edges connecting vertices in A to vertices in B. The motivation behind this criterion function is that the clustering process can be viewed as that of partitioning the documents into groups by minimizing the edge-cut of each partition. However, for reasons similar to those discussed in Section 3.1.2, such an external criterion may have trivial solutions, and for this reason each edge-cut is scaled by the sum of the internal edges. As shown in [11], this scaling leads to better balanced clustering solutions.

If we use the cosine function to measure the similarity between the documents, and Equations 5 and 6, then the above criterion function can be re-written as

$$\sum_{r=1}^{k} \frac{\sum_{d_i \in S_r, d_j \in S - S_r} \cos(d_i, d_j)}{\sum_{d_i, d_j \in S_r} \cos(d_i, d_j)} = \sum_{r=1}^{k} \frac{{D_r}^t (D - D_r)}{\|D_r\|^2} = \left(\sum_{r=1}^{k} \frac{{D_r}^t D}{\|D_r\|^2} \right) - k,$$

and since k is constant, the criterion function can be simplified to

$$\text{minimize} \quad \mathcal{G}_1 = \sum_{r=1}^{k} \frac{{D_r}^t D}{\|D_r\|^2}. \tag{16}$$

An alternate graph model views the various documents and their terms as a bipartite graph $G_b = (V, E)$, where V consists of two sets V_d and V_t. The vertex set V_d corresponds to the documents whereas the vertex set V_t corresponds to the terms. In this model, if the ith document contains the jth term, there is an edge connecting the corresponding ith vertex of V_d to the jth vertex of V_t. The weights of these edges are set using the *tf-idf* model discussed in Section 2. Given such a bipartite graph, the problem of clustering can be viewed as that of computing a simultaneous partitioning of the documents and the terms so that a criterion function defined on the edge-cut is optimized. In our study we will focus on a particular edge-cut based criterion function called the normalized cut, which was recently used in the context of this bipartite graph model for document clustering [46, 9]. The normalized cut criterion function is defined as

$$\text{minimize} \quad \mathcal{G}_2 = \sum_{r=1}^{k} \frac{\text{cut}(V_r, V - V_r)}{W(V_r)}, \tag{17}$$

where V_r is the set of vertices assigned to the rth cluster, and $W(V_r)$ is the sum of the weights of the adjacency lists of the vertices assigned to the rth cluster. Note that the rth cluster will contain vertices from both the V_d and V_t, *i.e.*, both documents as well as terms. The key motivation behind this representation and criterion function is to compute a clustering that groups together documents as well as the terms associated with these documents. Also, note that the various $W(V_r)$ quantities are used primarily as normalization factors, to ensure that the optimization of the criterion

function does not lead to trivial solutions. Its purpose is similar to the $\|D_r\|^2$ factor used in $\mathcal{G}_1$ (Equation 16).

3.2 Criterion Function Optimization

There are many ways that the various criterion functions described in the previous section can be optimized. A common way of performing this optimization is to use a greedy strategy. Such greedy approaches are commonly used in the context of partitional clustering algorithms (*e.g.*, K-means), and for many criterion functions it has been shown that they converge to a local minima. An alternate way is to use more powerful optimizers such as those based on the spectral properties of the document's similarity matrix [47] or document-term matrix [46, 9], or various multilevel optimization methods [26, 25]. However, such optimization methods have only been developed for a subset of the various criterion functions that are used in our study. For this reason, in our study, the various criterion functions were optimized using a greedy strategy. This was done primarily to ensure that the optimizer was equally powerful (or weak), regardless of the particular criterion function.

Our greedy optimizer consists of two phases: (i) ***initial clustering***, and (ii) ***cluster refinement***. In the initial clustering phase, a clustering solution is computed as follows. If k is the number of desired clusters, k documents are randomly selected to form the *seeds* of these clusters. The similarity of each document to each of these k seeds is computed, and each document is assigned to the cluster corresponding to its most similar seed. The similarity between documents and seeds is determined using the cosine measure of the corresponding document vectors. This approach leads to an initial clustering solution for all but the $\mathcal{G}_2$ criterion function. For $\mathcal{G}_2$ the above approach will only produce an initial partitioning of V_d (*i.e.*, the document vertices) and does not produce an initial partitioning of V_t (*i.e.*, the term vertices). Our algorithm obtains an initial partitioning of V_t by inducing it from the partitioning of V_d. This is done as follows. For each term-vertex v, we compute the edge-cut of v to each one of the k partitions of V_d, and assign v to the partition the corresponds to the highest cut. In other words, if we look at the column corresponding to v in the document-term matrix, and sum-up the various weights of this column according to the partitioning of the rows, then v is assigned to the partition that has the highest sum. Note that by assigning v to that partition, the total edge-cut due to v is minimized.

The goal of the cluster refinement phase is to take the initial clustering solution and iteratively refine it. Since the various criterion functions have different characteristics, depending on the particular criterion function we use three different refinement strategies.

The refinement strategy that we used for $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{E}_1$, $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{G}_1$ is the following. It consists of a number of iterations. During each iteration, the documents are visited in a random order. For each document, d_i, we compute the change in the value of the criterion function obtained by moving d_i to one of the other $k-1$ clusters. If there exist some moves that lead to an improvement in the overall value of the criterion function, then d_i is moved to the cluster that leads to the highest improvement. If no such cluster exists, d_i remains in the cluster that it already belongs to. The refinement phase ends, as soon as we perform an iteration in which no documents moved between clusters. Note that unlike the traditional refinement approach used by K-means type of algorithms, the above algorithm moves a document as soon as it is determined that it will lead to an improvement in the value of the criterion function. This type of refinement algorithms are often called *incremental* [12]. Since each move directly optimizes the particular criterion function, this refinement strategy always converges to a local minima. Furthermore, because the various criterion functions that use this refinement strategy are defined in terms of cluster composite and centroid vectors, the change in the value of the criterion functions as a result of single document moves can be computed efficiently.

The refinement strategy that we used for the $\mathcal{I}_3$ criterion function is identical to that of K-means, that has been shown to converge to a local minima [33]. It consists of a number of iterations. During each iteration, the documents are visited in a random order. For each document d_i we compute its distance to the k cluster centroids and assign d_i to the cluster that corresponds to the closest centroid. Once all the documents have been assigned to the different clusters, the centroids of the clusters are recomputed. The refinement phase ends as soon as we perform an iteration in which no documents moved between clusters.

The refinement strategy that we used for the $\mathcal{G}_2$ criterion function is based on alternating the cluster refinement between document-vertices and term-vertices, that was used in the past for partitioning bipartite graphs [29]. Similarly

to the other two refinement strategies, it consists of a number of iterations but each iteration consists of two steps. In the first step, the documents are visited in a random order. For each document, d_i, we compute the change in $\mathcal{G}_2$ that is obtained by moving d_i to one of the other $k-1$ clusters. If there exist some moves that decrease $\mathcal{G}_2$, then d_i is moved to the cluster that leads to the highest reduction. If no such cluster exists, d_i remains in the cluster that it already belongs to. In the second step, the terms are visited in a random order. For each term, t_j, we compute the change in $\mathcal{G}_2$ that is obtained by moving t_i to one of the other $k-1$ clusters. If there exist some moves that decrease $\mathcal{G}_2$, then t_j is moved to the cluster that leads to the highest reduction. If no such cluster exists, t_j remains in the cluster that it already belongs to. The refinement phase ends, as soon as we perform an iteration in which no documents and terms are moved between clusters. As it was with the first refinement strategy, this approach will also converge to a local minima.

The algorithms used during the refinement phase are greedy in nature, they are not guaranteed to converge to a global minima, and the local minima solution they obtain depends on the particular set of seed documents that were selected to obtain the initial clustering. To eliminate some of this sensitivity, the overall process is repeated a number of times. That is, we compute N different clustering solutions (*i.e.*, initial clustering followed by cluster refinement), and the one that achieves the best value for the particular criterion function is kept. In all of our experiments, we used $N = 10$. For the rest of this discussion when we refer to the clustering solution we will mean the solution that was obtained by selecting the best out of these N potentially different solutions.

4 Experimental Results

We experimentally evaluated the performance of the different clustering criterion functions on a number of different datasets. In the rest of this section we first describe the various datasets and our experimental methodology, followed by a description of the experimental results.

4.1 Document Collections

In our experiments, we used a total of fifteen different datasets, whose general characteristics are summarized in Table 1. The smallest of these datasets contained 878 documents and the largest contained 11,162 documents. To ensure diversity in the datasets, we obtained them from different sources. For all data sets, we used a stop-list to remove common words, and the words were stemmed using Porter's suffix-stripping algorithm [36]. Moreover, any term that occurs in fewer than two documents was eliminated.

Data	Source	# of documents	# of terms	# of classes
classic	CACM/CISI/CRANFIELD/MEDLINE	7089	12009	4
fbis	FBIS (TREC)	2463	12674	17
hitech	San Jose Mercury (TREC)	2301	13170	6
reviews	San Jose Mercury (TREC)	4069	23220	5
sports	San Jose Mercury (TREC)	8580	18324	7
la12	LA Times (TREC)	6279	21604	6
new3	TREC	9558	36306	44
tr31	TREC	927	10128	7
tr41	TREC	878	7454	10
ohscal	OHSUMED-233445	11162	11465	10
re0	Reuters-21578	1504	2886	13
re1	Reuters-21578	1657	3758	25
k1a	WebACE	2340	13879	20
k1b	WebACE	2340	13879	6
wap	WebACE	1560	8460	20

Table 1: Summary of data sets used to evaluate the various clustering criterion functions.

The *classic* dataset was obtained by combining the CACM, CISI, CRANFIELD, and MEDLINE abstracts that

were used in the past to evaluate various information retrieval systems[2]. In this data set, each individual set of abstracts formed one of the four classes. The *fbis* dataset is from the Foreign Broadcast Information Service data of TREC-5 [42], and the classes correspond to the categorization used in that collection. The *hitech*, *reviews*, and *sports* datasets were derived from the San Jose Mercury newspaper articles that are distributed as part of the TREC collection (TIPSTER Vol. 3). Each one of these datasets were constructed by selecting documents that are part of certain topics in which the various articles were categorized (based on the *DESCRIPT* tag). The *hitech* dataset contained documents about computers, electronics, health, medical, research, and technology; the *reviews* dataset contained documents about food, movies, music, radio, and restaurants; and the *sports* dataset contained documents about baseball, basketball, bicycling, boxing, football, golfing, and hockey. In selecting these documents we ensured that no two documents share the same *DESCRIPT* tag (which can contain multiple categories). The *la12* dataset was obtained from articles of the Los Angeles Times that was used in TREC-5 [42]. The categories correspond to the *desk* of the paper that each article appeared and include documents from the entertainment, financial, foreign, metro, national, and sports desks. Datasets *new3*, *tr31*, and *tr41* are derived from TREC-5 [42], TREC-6 [42], and TREC-7 [42] collections. The classes of these datasets correspond to the documents that were judged relevant to particular queries. The *ohscal* dataset was obtained from the OHSUMED collection [19], which contains 233,445 documents indexed using 14,321 unique categories. Our dataset contained documents from the antibodies, carcinoma, DNA, in-vitro, molecular sequence data, pregnancy, prognosis, receptors, risk factors, and tomography categories. The datasets *re0* and *re1* are from Reuters-21578 text categorization test collection Distribution 1.0 [32]. We divided the labels into two sets and constructed data sets accordingly. For each data set, we selected documents that have a single label. Finally, the datasets *k1a*, *k1b*, and *wap* are from the WebACE project [34, 16, 3, 4]. Each document corresponds to a web page listed in the subject hierarchy of Yahoo! [44]. The datasets *k1a* and *k1b* contain exactly the same set of documents but they differ in how the documents were assigned to different classes. In particular, *k1a* contains a finer-grain categorization than that contained in *k1b*.

4.2 Experimental Methodology and Metrics

For each one of the different datasets we obtained a 5-, 10-, 15-, and 20-way clustering solution that optimized the various clustering criterion functions. The quality of a clustering solution was measured by using two different metrics that look at the class labels of the documents assigned to each cluster. The first metric is the widely used *entropy* measure that looks are how the various classes of documents are distributed within each cluster, and the second measure is the *purity* that measures the extend to which each cluster contained documents from primarily one class.

Given a particular cluster S_r of size n_r, the entropy of this cluster is defined to be

$$E(S_r) = -\frac{1}{\log q} \sum_{i=1}^{q} \frac{n_r^i}{n_r} \log \frac{n_r^i}{n_r},$$

where q is the number of classes in the dataset, and n_r^i is the number of documents of the ith class that were assigned to the rth cluster. The entropy of the entire clustering solution is then defined to be the sum of the individual cluster entropies weighted according to the cluster size. That is,

$$\mathit{Entropy} = \sum_{r=1}^{k} \frac{n_r}{n} E(S_r).$$

A perfect clustering solution will be the one that leads to clusters that contain documents from only a single class, in which case the entropy will be zero. In general, the smaller the entropy values, the better the clustering solution is. In

[2] They are are available from ftp://ftp.cs.cornell.edu/pub/smart.

a similar fashion, the purity of this cluster is defined to be

$$P(S_r) = \frac{1}{n_r} \max_i(n_r^i),$$

which is nothing more than the fraction of the overall cluster size that the largest class of documents assigned to that cluster represents. The overall purity of the clustering solution is obtained as a weighted sum of the individual cluster purities and is given by

$$Purity = \sum_{r=1}^{k} \frac{n_r}{n} P(S_r).$$

In general, the larger the values of purity, the better the clustering solution is.

To eliminate any instances that a particular clustering solution for a particular criterion function got trapped into a bad local minima, in all of our experiments we actually found ten different clustering solutions. The various entropy and purity values that are reported in the rest of this section correspond to the average entropy and purity over these ten different solutions. As discussed in Section 3.2 each of the ten clustering solutions corresponds to the best solution out of ten different initial partitioning and refinement phases. As a result, for each particular value of k and criterion function we computed 100 clustering solutions. The overall number of experiments that we performed was 3*100*4*8*15 = 144,000, that were completed in about 8 days on a Pentium III@600MHz workstation.

4.3 Evaluation of Direct k-way Clustering

Our first set of experiments was focused on evaluating the quality of the clustering solutions produced by the various criterion functions when they were used directly to compute a k-way clustering solution. The results for the various datasets and criterion functions for 5-, 10-, 15-, and 20-way clustering solutions are shown in Table 2, which shows both the entropy and the purity results for the entire set of experiments. The results in this table are provided primarily for completeness and in order to evaluate the various criterion functions we actually summarized these results by looking at the average performance of each criterion function over the entire set of datasets.

One way of summarizing the results is to average the entropies (or purities) for each criterion function over the fifteen different datasets. However, since the clustering quality for different datasets is quite different and since the quality tends to improve as we increase the number of clusters, we felt that such simple averaging may distort the overall results. For this reason, our summarization is based on averaging relative entropies, as follows. For each dataset and value of k, we divided the entropy obtained by a particular criterion function by the smallest entropy obtained for that particular dataset and value of k over the different criterion functions. These ratios represent the degree to which a particular criterion function performed worse than the best criterion function for that particular series of experiments. Note that for different datasets and values of k, the criterion function that achieved the best solution as measured by entropy may be different. These ratios are less sensitive to the actual entropy values and the particular value of k. We will refer to these ratios as *relative entropies*. Now, for each criterion function and value of k we averaged these relative entropies over the various datasets. A criterion function that has an ***average relative entropy*** close to 1.0 will indicate that this function did the best for most of the datasets. On the other hand, if the average relative entropy is high, then this criterion function performed poorly. We performed a similar transformation for the various purity functions. However, since higher values of purity are better, instead of dividing a particular purity value with the best-achieved purity (*i.e.*, higher purity), we took the opposite ratios. That is, we divided the best-achieved purity with that achieved by a particular criterion function, and then averaged them over the various datasets. In this way, the values for the ***average relative purity*** can be interpreted in a similar manner as those of the average relative entropy (they are good if they are close to 1.0 and they are getting worse as they become greater than 1.0).

The values for the average relative entropies and purities for the 5-, 10-, 15-, and 20-way clustering solutions are shown in Table 3. Furthermore, the rows labeled "Avg" contain the average of these averages over the four sets of clustering solutions. The entries that are underlined correspond to the criterion functions that performed the best, whereas the boldfaced entries correspond to the criterion functions that performed within 2% of the best.

Entropy

	5-way Clustering								10-way Clustering							
DataSet	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
classic	0.41	0.23	0.39	0.21	0.24	0.22	0.29	0.49	0.29	0.20	0.25	0.16	0.17	0.17	0.24	0.30
fbis	0.53	0.51	0.53	0.51	0.51	0.51	0.53	0.56	0.43	0.40	0.42	0.40	0.40	0.40	0.45	0.48
hitech	0.76	0.67	0.72	0.64	0.67	0.66	0.68	0.70	0.68	0.61	0.65	0.63	0.63	0.63	0.60	0.69
k1a	0.57	0.48	0.53	0.50	0.49	0.49	0.48	0.52	0.49	0.39	0.47	0.40	0.39	0.39	0.41	0.46
k1b	0.27	0.20	0.25	0.23	0.20	0.22	0.19	0.18	0.20	0.12	0.20	0.17	0.14	0.15	0.14	0.16
la12	0.65	0.39	0.59	0.38	0.41	0.39	0.47	0.47	0.48	0.38	0.44	0.40	0.38	0.39	0.42	0.47
new3	0.75	0.69	0.75	0.70	0.70	0.69	0.70	0.76	0.67	0.59	0.66	0.60	0.60	0.59	0.62	0.69
ohscal	0.74	0.66	0.71	0.63	0.66	0.65	0.65	0.69	0.63	0.55	0.62	0.54	0.55	0.54	0.58	0.64
re0	0.55	0.51	0.53	0.48	0.51	0.49	0.51	0.56	0.41	0.40	0.40	0.38	0.37	0.38	0.40	0.51
re1	0.56	0.49	0.54	0.49	0.50	0.48	0.51	0.64	0.49	0.41	0.45	0.42	0.41	0.42	0.44	0.56
reviews	0.54	0.32	0.56	0.36	0.34	0.34	0.31	0.58	0.30	0.27	0.30	0.29	0.28	0.28	0.31	0.42
sports	0.42	0.22	0.42	0.22	0.25	0.20	0.23	0.45	0.35	0.21	0.38	0.20	0.23	0.18	0.29	0.35
tr31	0.45	0.39	0.40	0.40	0.39	0.38	0.47	0.36	0.29	0.22	0.23	0.23	0.22	0.21	0.22	0.32
tr41	0.40	0.36	0.39	0.34	0.38	0.35	0.36	0.46	0.29	0.24	0.26	0.26	0.22	0.25	0.29	0.34
wap	0.55	0.48	0.52	0.49	0.49	0.49	0.48	0.54	0.46	0.39	0.43	0.42	0.39	0.41	0.41	0.47
	15-way Clustering								20-way Clustering							
DataSet	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
classic	0.25	0.19	0.26	0.17	0.17	0.17	0.24	0.23	0.28	0.19	0.25	0.18	0.16	0.17	0.23	0.27
fbis	0.36	0.34	0.38	0.35	0.35	0.34	0.38	0.46	0.34	0.33	0.33	0.33	0.32	0.33	0.34	0.44
hitech	0.63	0.60	0.62	0.62	0.59	0.61	0.60	0.69	0.61	0.57	0.59	0.60	0.58	0.59	0.59	0.68
k1a	0.43	0.35	0.40	0.36	0.34	0.35	0.37	0.45	0.38	0.32	0.37	0.33	0.31	0.33	0.35	0.44
k1b	0.19	0.12	0.16	0.15	0.13	0.13	0.13	0.16	0.17	0.12	0.16	0.14	0.13	0.13	0.14	0.17
la12	0.44	0.38	0.43	0.38	0.37	0.38	0.39	0.49	0.44	0.37	0.40	0.38	0.37	0.38	0.40	0.50
new3	0.60	0.53	0.61	0.53	0.53	0.52	0.56	0.65	0.57	0.49	0.55	0.49	0.48	0.48	0.52	0.61
ohscal	0.60	0.54	0.59	0.54	0.54	0.54	0.56	0.66	0.58	0.53	0.58	0.54	0.53	0.54	0.55	0.66
re0	0.38	0.37	0.38	0.37	0.36	0.37	0.37	0.49	0.36	0.34	0.35	0.36	0.34	0.35	0.35	0.45
re1	0.43	0.37	0.41	0.37	0.36	0.36	0.42	0.53	0.40	0.32	0.37	0.33	0.33	0.33	0.38	0.49
reviews	0.28	0.24	0.25	0.28	0.25	0.25	0.26	0.37	0.28	0.24	0.26	0.26	0.24	0.24	0.25	0.37
sports	0.28	0.18	0.27	0.20	0.20	0.19	0.21	0.33	0.24	0.15	0.24	0.19	0.17	0.16	0.19	0.33
tr31	0.25	0.18	0.21	0.22	0.19	0.19	0.22	0.28	0.20	0.17	0.19	0.20	0.17	0.16	0.20	0.27
tr41	0.22	0.18	0.20	0.20	0.20	0.19	0.22	0.31	0.18	0.15	0.17	0.17	0.15	0.17	0.18	0.28
wap	0.42	0.35	0.39	0.36	0.34	0.35	0.38	0.45	0.37	0.33	0.36	0.33	0.32	0.33	0.35	0.44

Purity

	5-way Clustering								10-way Clustering							
DataSet	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
classic	0.77	0.89	0.78	0.92	0.89	0.91	0.82	0.66	0.85	0.91	0.87	0.93	0.93	0.93	0.87	0.83
fbis	0.50	0.53	0.51	0.51	0.52	0.51	0.50	0.52	0.60	0.62	0.60	0.61	0.62	0.62	0.58	0.59
hitech	0.43	0.54	0.47	0.56	0.52	0.54	0.49	0.50	0.50	0.58	0.55	0.57	0.58	0.57	0.58	0.51
k1a	0.42	0.49	0.46	0.49	0.49	0.49	0.50	0.47	0.52	0.61	0.53	0.62	0.62	0.63	0.59	0.54
k1b	0.84	0.83	0.83	0.81	0.84	0.82	0.85	0.86	0.90	0.91	0.88	0.87	0.89	0.89	0.91	0.89
la12	0.53	0.79	0.59	0.78	0.76	0.78	0.67	0.73	0.68	0.78	0.73	0.77	0.78	0.78	0.74	0.73
new3	0.21	0.24	0.21	0.23	0.23	0.24	0.23	0.21	0.28	0.31	0.28	0.31	0.31	0.31	0.29	0.26
ohscal	0.34	0.41	0.36	0.45	0.41	0.43	0.44	0.39	0.47	0.55	0.48	0.56	0.56	0.56	0.52	0.47
re0	0.43	0.48	0.45	0.56	0.47	0.54	0.52	0.46	0.61	0.63	0.62	0.66	0.66	0.66	0.61	0.50
re1	0.47	0.53	0.50	0.53	0.53	0.53	0.52	0.38	0.55	0.61	0.59	0.59	0.62	0.61	0.60	0.48
reviews	0.62	0.81	0.61	0.79	0.80	0.81	0.82	0.58	0.84	0.84	0.85	0.83	0.84	0.84	0.82	0.74
sports	0.66	0.87	0.65	0.87	0.80	0.87	0.87	0.65	0.74	0.85	0.70	0.86	0.83	0.87	0.77	0.72
tr31	0.64	0.69	0.69	0.68	0.69	0.70	0.64	0.72	0.79	0.85	0.84	0.83	0.85	0.86	0.86	0.77
tr41	0.67	0.71	0.68	0.71	0.69	0.71	0.71	0.63	0.77	0.79	0.79	0.75	0.81	0.76	0.76	0.73
wap	0.44	0.50	0.46	0.49	0.49	0.50	0.51	0.46	0.54	0.60	0.57	0.60	0.62	0.60	0.59	0.53
	15-way Clustering								20-way Clustering							
DataSet	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
classic	0.88	0.91	0.87	0.93	0.93	0.93	0.87	0.88	0.85	0.92	0.87	0.92	0.93	0.93	0.88	0.86
fbis	0.66	0.68	0.65	0.66	0.68	0.67	0.63	0.61	0.69	0.69	0.69	0.69	0.69	0.69	0.68	0.62
hitech	0.53	0.58	0.57	0.57	0.58	0.58	0.57	0.51	0.56	0.59	0.57	0.58	0.59	0.58	0.57	0.51
k1a	0.58	0.66	0.61	0.67	0.69	0.68	0.63	0.56	0.63	0.70	0.65	0.68	0.71	0.70	0.66	0.58
k1b	0.89	0.92	0.91	0.90	0.91	0.91	0.92	0.90	0.89	0.92	0.90	0.90	0.91	0.91	0.90	0.90
la12	0.72	0.78	0.72	0.78	0.79	0.78	0.77	0.70	0.71	0.78	0.75	0.78	0.78	0.78	0.76	0.68
new3	0.34	0.38	0.33	0.38	0.37	0.39	0.35	0.30	0.38	0.42	0.39	0.43	0.43	0.43	0.39	0.35
ohscal	0.49	0.55	0.50	0.57	0.55	0.56	0.52	0.46	0.52	0.57	0.52	0.57	0.58	0.57	0.56	0.46
re0	0.64	0.66	0.65	0.67	0.67	0.67	0.62	0.53	0.65	0.68	0.67	0.68	0.68	0.68	0.65	0.57
re1	0.58	0.64	0.60	0.63	0.64	0.65	0.60	0.49	0.60	0.67	0.63	0.67	0.66	0.67	0.62	0.53
reviews	0.85	0.86	0.86	0.83	0.85	0.85	0.85	0.79	0.83	0.87	0.85	0.85	0.86	0.86	0.85	0.79
sports	0.79	0.87	0.80	0.85	0.85	0.86	0.84	0.73	0.81	0.89	0.82	0.87	0.87	0.88	0.86	0.74
tr31	0.82	0.87	0.85	0.83	0.85	0.86	0.85	0.81	0.85	0.88	0.86	0.85	0.87	0.88	0.86	0.82
tr41	0.83	0.85	0.83	0.83	0.82	0.83	0.81	0.74	0.85	0.87	0.86	0.85	0.87	0.86	0.85	0.77
wap	0.59	0.66	0.62	0.67	0.67	0.67	0.62	0.56	0.64	0.68	0.65	0.70	0.70	0.69	0.66	0.57

Table 2: Entropy and Purity for the various datasets and criterion functions for the clustering solutions obtained via direct k-way clustering.

Average Relative Entropy

k	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
5	1.361	**1.041**	1.312	**1.044**	1.069	**1.033**	1.092	1.333
10	1.312	**1.042**	1.246	1.069	**1.035**	**1.040**	1.148	1.380
15	1.252	**1.019**	1.190	1.071	**1.029**	**1.029**	1.132	1.402
20	1.236	**1.018**	1.180	1.086	**1.022**	**1.035**	1.139	1.486
Avg	1.290	**1.030**	1.232	1.068	**1.039**	**1.034**	1.128	1.400

Average Relative Purity

k	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
5	1.209	**1.034**	1.163	**1.018**	1.051	**1.021**	1.054	1.173
10	1.112	**1.017**	1.082	**1.024**	**1.008**	**1.013**	1.054	1.161
15	1.087	**1.012**	1.068	**1.019**	**1.012**	**1.009**	1.057	1.163
20	1.076	**1.007**	1.054	**1.017**	**1.006**	**1.009**	1.047	1.165
Avg	1.121	**1.018**	1.092	**1.019**	**1.019**	**1.013**	1.053	1.166

Table 3: Relative entropies and purities averaged over the different datasets for different criterion functions for the clustering solutions obtained via direct k-way clustering. Underlined entries represent the best performing scheme, and boldfaced entries correspond to schemes that performed within 2% of the best.

A number of observations can be made by analyzing the results in Table 3. First, the $\mathcal{I}_1$, $\mathcal{I}_3$, and the $\mathcal{G}_2$ criterion functions lead to clustering solutions that are consistently worse than the solutions obtained using the other criterion functions. This is true both when the quality of the clustering solution was evaluated using the entropy as well as the purity measures. They lead to solutions that are 19%–35% worse in terms of entropy and 8%–15% worse in terms of purity than the best solution. Second, the $\mathcal{I}_2$ and the $\mathcal{H}_2$ criterion functions lead to the best solutions irrespective of the number of clusters or the measure used to evaluate the clustering quality. Over the entire set of experiments, these methods are either the best or always within 2% of the best solution. Third, the $\mathcal{H}_1$ criterion function performs the next best and overall is within 2% of the best solution for both entropy and purity. Fourth, the $\mathcal{E}_1$ criterion function also performs quite well when the quality is evaluated using purity. Finally, the $\mathcal{G}_1$ criterion function always performs somewhere in the middle of the road. It is on the average 9% worse in terms of entropy and 4% worse in terms of purity when compared to the best scheme. Also note that the relative performance of the various criterion functions remains more-or-less the same for both the entropy- and the purity-based evaluation methods. The only change is that the relative differences between the various criterion functions as measured by entropy are somewhat greater when compared to those measured by purity. This should not be surprising, as the entropy measure takes into account the entire distribution of the documents in a particular cluster and not just the largest class as it is done by the purity measure.

4.4 Evaluation of k-way Clustering via Repeated Bisections

Our second set of experiments was focused on evaluating the clustering solutions produced by the various criterion functions when the overall k-way clustering solution was obtained via a sequence of cluster bisections (RB). In this approach, a k-way solution is obtained by first bisecting the entire collection. Then, one of the two clusters is selected and it is further bisected, leading to a total of three clusters. The process of selecting and bisecting a particular cluster continues until k clusters are obtained. Each of these bisections is performed so that the resulting two-way clustering solution optimizes a particular criterion function. However, the overall k-way clustering solution will not necessarily be at a local minima with respect to the criterion function. Obtaining a k-way clustering solution in this fashion may be desirable because the resulting solution is hierarchical, and thus it can be easily visualized. The key step in this algorithm is the method used to select which cluster to bisect next, and a number of different approaches were described in [39, 23]. In all of our experiments, we chose to select the largest cluster, as this approach lead to reasonably good and balanced clustering solutions [39].

Table 4 shows the quality of the clustering solutions produced by the various criterion functions for 5-, 10-, 15-,

Entropy

	5-way Clustering								10-way Clustering							
DataSet	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
classic	0.38	0.25	0.36	0.29	0.32	0.34	0.26	0.32	0.19	0.22	0.18	0.17	0.17	0.21	0.23	0.15
fbis	0.50	0.50	0.50	0.54	0.49	0.52	0.49	0.53	0.40	0.40	0.41	0.44	0.40	0.44	0.41	0.43
hitech	0.71	0.63	0.66	0.63	0.63	0.62	0.67	0.71	0.66	0.59	0.63	0.58	0.60	0.57	0.60	0.66
k1a	0.55	0.50	0.52	0.49	0.52	0.49	0.50	0.52	0.43	0.42	0.43	0.41	0.42	0.40	0.41	0.42
k1b	0.23	0.23	0.22	0.23	0.24	0.22	0.22	0.22	0.20	0.18	0.19	0.15	0.18	0.14	0.15	0.16
la12	0.61	0.47	0.54	0.49	0.44	0.41	0.48	0.42	0.43	0.40	0.42	0.42	0.40	0.38	0.41	0.39
new3	0.76	0.70	0.75	0.69	0.72	0.69	0.70	0.74	0.64	0.59	0.63	0.59	0.61	0.59	0.58	0.64
ohscal	0.72	0.63	0.72	0.64	0.64	0.64	0.63	0.67	0.62	0.54	0.62	0.54	0.56	0.54	0.56	0.60
re0	0.54	0.51	0.52	0.49	0.49	0.49	0.51	0.55	0.41	0.39	0.41	0.39	0.39	0.39	0.40	0.44
re1	0.55	0.49	0.53	0.52	0.51	0.52	0.50	0.62	0.45	0.42	0.44	0.41	0.42	0.41	0.42	0.54
reviews	0.39	0.33	0.37	0.31	0.34	0.31	0.35	0.55	0.35	0.30	0.33	0.24	0.31	0.26	0.29	0.36
sports	0.39	0.27	0.38	0.24	0.30	0.24	0.28	0.36	0.28	0.16	0.27	0.15	0.21	0.13	0.15	0.27
tr31	0.35	0.32	0.37	0.29	0.32	0.32	0.31	0.35	0.22	0.17	0.27	0.21	0.18	0.18	0.18	0.23
tr41	0.36	0.38	0.38	0.43	0.39	0.37	0.36	0.38	0.27	0.28	0.27	0.30	0.26	0.28	0.27	0.24
wap	0.53	0.50	0.51	0.49	0.51	0.48	0.49	0.53	0.43	0.42	0.44	0.40	0.43	0.39	0.41	0.44
	15-way Clustering								20-way Clustering							
DataSet	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
classic	0.18	0.21	0.18	0.17	0.16	0.20	0.22	0.13	0.18	0.20	0.17	0.15	0.16	0.19	0.21	0.13
fbis	0.36	0.37	0.36	0.40	0.35	0.38	0.34	0.39	0.34	0.34	0.33	0.38	0.33	0.36	0.33	0.37
hitech	0.64	0.56	0.61	0.56	0.57	0.55	0.57	0.64	0.61	0.53	0.59	0.54	0.56	0.53	0.55	0.62
k1a	0.38	0.36	0.36	0.35	0.35	0.34	0.37	0.38	0.35	0.32	0.33	0.32	0.31	0.32	0.33	0.35
k1b	0.17	0.13	0.14	0.14	0.13	0.13	0.14	0.14	0.15	0.12	0.13	0.12	0.12	0.12	0.12	0.13
la12	0.41	0.38	0.39	0.39	0.38	0.38	0.38	0.37	0.37	0.37	0.38	0.38	0.37	0.36	0.38	0.36
new3	0.57	0.54	0.57	0.53	0.55	0.53	0.53	0.58	0.52	0.49	0.51	0.49	0.50	0.49	0.49	0.53
ohscal	0.57	0.52	0.57	0.52	0.54	0.52	0.53	0.57	0.56	0.51	0.56	0.51	0.53	0.51	0.52	0.56
re0	0.36	0.35	0.36	0.37	0.36	0.37	0.35	0.39	0.32	0.33	0.32	0.32	0.33	0.32	0.33	0.36
re1	0.40	0.36	0.39	0.35	0.37	0.35	0.37	0.49	0.37	0.33	0.35	0.33	0.33	0.32	0.34	0.44
reviews	0.29	0.25	0.27	0.23	0.27	0.23	0.24	0.33	0.26	0.24	0.26	0.20	0.25	0.21	0.24	0.32
sports	0.20	0.12	0.19	0.12	0.16	0.12	0.12	0.23	0.19	0.12	0.19	0.12	0.14	0.12	0.11	0.21
tr31	0.18	0.15	0.22	0.16	0.17	0.17	0.15	0.19	0.16	0.14	0.18	0.14	0.15	0.15	0.14	0.17
tr41	0.20	0.22	0.23	0.23	0.24	0.23	0.23	0.19	0.18	0.16	0.18	0.19	0.17	0.18	0.16	0.16
wap	0.38	0.34	0.37	0.35	0.35	0.34	0.35	0.39	0.35	0.31	0.34	0.32	0.32	0.32	0.32	0.35

Purity

	5-way Clustering								10-way Clustering							
DataSet	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
classic	0.79	0.88	0.79	0.85	0.81	0.77	0.87	0.80	0.92	0.88	0.93	0.93	0.93	0.90	0.88	0.94
fbis	0.54	0.51	0.54	0.50	0.53	0.50	0.50	0.51	0.64	0.59	0.63	0.58	0.62	0.55	0.57	0.60
hitech	0.48	0.59	0.53	0.55	0.55	0.58	0.48	0.48	0.53	0.61	0.56	0.61	0.56	0.62	0.59	0.56
k1a	0.44	0.49	0.47	0.48	0.47	0.48	0.48	0.47	0.57	0.59	0.58	0.60	0.57	0.61	0.60	0.59
k1b	0.85	0.80	0.82	0.81	0.80	0.82	0.84	0.83	0.86	0.85	0.84	0.88	0.85	0.90	0.88	0.88
la12	0.57	0.71	0.64	0.69	0.75	0.76	0.70	0.78	0.73	0.77	0.75	0.74	0.77	0.78	0.75	0.79
new3	0.20	0.24	0.20	0.24	0.22	0.24	0.23	0.22	0.28	0.31	0.27	0.32	0.30	0.31	0.32	0.28
ohscal	0.36	0.47	0.36	0.45	0.47	0.46	0.47	0.42	0.46	0.60	0.47	0.59	0.58	0.59	0.56	0.51
re0	0.50	0.53	0.50	0.55	0.52	0.53	0.52	0.47	0.60	0.64	0.60	0.64	0.62	0.63	0.63	0.60
re1	0.48	0.53	0.50	0.50	0.53	0.50	0.52	0.40	0.56	0.61	0.58	0.62	0.60	0.62	0.62	0.47
reviews	0.75	0.80	0.77	0.83	0.79	0.82	0.78	0.58	0.78	0.80	0.79	0.87	0.81	0.86	0.81	0.79
sports	0.69	0.76	0.70	0.85	0.75	0.79	0.79	0.71	0.77	0.88	0.80	0.91	0.83	0.92	0.90	0.79
tr31	0.73	0.75	0.71	0.78	0.74	0.74	0.75	0.74	0.85	0.89	0.80	0.84	0.88	0.89	0.89	0.85
tr41	0.70	0.69	0.69	0.61	0.69	0.73	0.73	0.71	0.79	0.75	0.79	0.73	0.77	0.73	0.74	0.80
wap	0.45	0.49	0.48	0.49	0.47	0.49	0.49	0.46	0.56	0.59	0.57	0.61	0.56	0.62	0.59	0.57
	15-way Clustering								20-way Clustering							
DataSet	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
classic	0.92	0.89	0.93	0.93	0.93	0.90	0.88	0.95	0.92	0.90	0.93	0.94	0.93	0.91	0.89	0.95
fbis	0.67	0.62	0.67	0.63	0.65	0.63	0.68	0.63	0.69	0.67	0.69	0.63	0.68	0.64	0.68	0.65
hitech	0.55	0.62	0.57	0.62	0.60	0.62	0.61	0.57	0.56	0.63	0.58	0.62	0.60	0.64	0.61	0.58
k1a	0.63	0.66	0.65	0.68	0.67	0.69	0.66	0.64	0.66	0.69	0.68	0.70	0.70	0.70	0.69	0.66
k1b	0.89	0.91	0.90	0.89	0.90	0.91	0.89	0.90	0.90	0.91	0.90	0.91	0.91	0.92	0.92	0.91
la12	0.74	0.78	0.76	0.77	0.78	0.78	0.76	0.79	0.77	0.78	0.77	0.77	0.78	0.79	0.76	0.79
new3	0.33	0.37	0.33	0.38	0.36	0.38	0.37	0.33	0.38	0.42	0.39	0.41	0.42	0.42	0.41	0.38
ohscal	0.52	0.61	0.52	0.62	0.59	0.62	0.60	0.52	0.52	0.62	0.53	0.62	0.59	0.62	0.60	0.54
re0	0.64	0.67	0.64	0.64	0.64	0.64	0.66	0.64	0.67	0.69	0.67	0.68	0.66	0.69	0.68	0.66
re1	0.60	0.65	0.61	0.65	0.63	0.66	0.65	0.51	0.62	0.68	0.64	0.67	0.67	0.68	0.67	0.55
reviews	0.83	0.86	0.84	0.87	0.84	0.87	0.87	0.81	0.84	0.87	0.84	0.89	0.85	0.88	0.87	0.81
sports	0.85	0.90	0.85	0.93	0.89	0.92	0.92	0.83	0.86	0.90	0.86	0.93	0.90	0.92	0.93	0.85
tr31	0.87	0.89	0.83	0.89	0.88	0.89	0.90	0.87	0.88	0.90	0.86	0.90	0.89	0.89	0.90	0.88
tr41	0.83	0.79	0.80	0.78	0.78	0.78	0.78	0.85	0.84	0.86	0.85	0.82	0.85	0.83	0.86	0.87
wap	0.62	0.67	0.63	0.68	0.67	0.68	0.66	0.63	0.66	0.69	0.67	0.70	0.69	0.70	0.69	0.66

Table 4: Entropy and Purity for the various datasets and criterion functions for the clustering solutions obtained via repeated bisections.

and 20-way clustering, when these solutions were obtained via repeated bisections. Again, these results are primarily provided for completeness and our discussion will focus on the average relative entropies and purities for the various clustering solutions shown in Table 5. The values in this table were obtained by using exactly the same procedure discussed in Section 4.3 for averaging the results of Table 4.

Average Relative Entropy

k	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
5	1.207	**1.050**	1.168	**1.060**	1.083	**1.049**	**1.053**	1.191
10	1.243	1.112	1.239	1.083	1.129	**1.056**	1.106	1.221
15	1.190	**1.085**	1.183	**1.077**	1.102	**1.079**	**1.085**	1.205
20	1.183	**1.070**	1.169	**1.057**	1.085	**1.072**	**1.075**	1.209
Avg	1.206	**1.079**	1.190	**1.069**	1.100	**1.064**	**1.080**	1.207

Average Relative Purity

k	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
5	1.137	**1.035**	1.110	**1.047**	**1.055**	**1.041**	**1.050**	1.127
10	1.099	**1.039**	1.093	**1.030**	1.051	**1.024**	**1.043**	1.089
15	1.077	**1.029**	1.070	**1.022**	**1.038**	**1.021**	**1.029**	1.081
20	1.063	**1.016**	1.053	**1.018**	**1.025**	**1.014**	**1.021**	1.068
Avg	1.094	**1.030**	1.082	**1.030**	**1.042**	**1.025**	**1.036**	1.091

Table 5: Relative entropies and purities averaged over the different datasets for different criterion functions for the clustering solutions obtained via repeated bisections. Underlined entries represent the best performing scheme, and boldfaced entries correspond to schemes that performed within 2% of the best.

A number of observations can be made by analyzing these results. First, the $\mathcal{I}_1$, $\mathcal{I}_3$, and $\mathcal{G}_2$ criterion functions lead to the worse clustering solutions, both in terms of entropy and in terms of purity. Second, the $\mathcal{H}_2$ criterion function leads to the best overall solutions, whereas the $\mathcal{I}_2$, $\mathcal{E}_1$, and $\mathcal{G}_1$ criterion functions are within 2% of the best. The $\mathcal{H}_1$ criterion function performs within 2% of the best solution when the quality is measured using purity, and it is about 3.3% from the best when the quality is measured using entropy. These results are in general consistent with those obtained for direct k-way clustering but in the case of repeated bisections, there is a reduction in the relative difference between the best and the worst schemes. For example, in terms of entropy, $\mathcal{G}_2$ is only 13% worse than the best (compared to 35% for direct k-way). Similar trends can be observed for the other criterion functions and for purity. This relative improvement becomes most apparent for the $\mathcal{G}_1$ criterion function that now almost always performs within 2% of the best. The reason for these improvements will be discussed in Section 5. Also, another interesting observation is that the average relative entropies (and purities) for repeated bisections are higher than the corresponding results obtained for direct k-way. This indicates that there is a higher degree of variation between the relative performance of the various criterion functions for the different data sets.

Finally, Figure 1 compares the quality of the clustering solutions obtained via direct k-way clustering to those obtained via repeated bisections. These plots were obtained by dividing the entropy (or purity) achieved by the direct k-way approach (Table 2) with that of the entropy (or purity) achieved by the RB approach, and then averaging these ratios over the fifteen data sets for each one of the criterion functions and number of clusters. Since lower entropy values are better, ratios that are greater than one indicate that the RB approach leads to better solutions than direct k-way and vice versa. Similarly, since higher purity values are better, ratios that are smaller than one indicate the RB approach leads to better solutions than direct k-way.

Looking at the plots in Figure 1 we can make a number of observations. First, in terms of both entropy and purity, the $\mathcal{I}_1$, $\mathcal{I}_3$, $\mathcal{G}_1$, and $\mathcal{G}_2$ criterion functions lead to worse solutions with direct k-way than with RB clustering. Second, for the remaining criterion functions, the relative performance appears to be sensitive on the number of clusters. For small number of clusters, the direct k-way approach tends to lead to better solutions; however, as the number of clusters increases the RB approach tends to outperform direct k-way. In fact, this sensitivity on the number of clusters appears to be true for all eight clustering criterion functions, and the main difference has to do with how quickly the quality

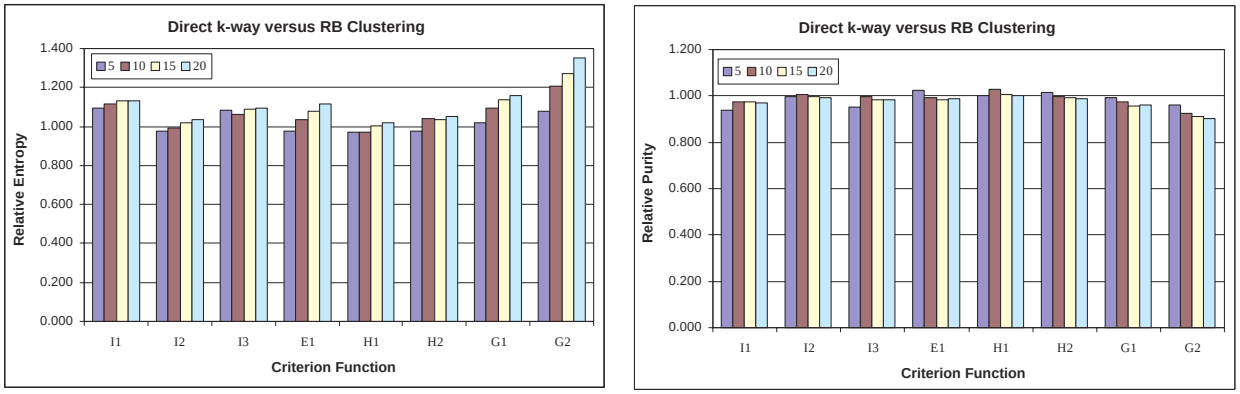

Figure 1: The relative performance of direct *k*-way clustering over that of repeated bisections (RB) averaged over the different datasets, for the entropy and purity measures.

of the direct *k*-way clustering solution degrades. Third, the $\mathcal{I}_2$, $\mathcal{H}_1$, and $\mathcal{H}_2$ criterion functions appear to be the least sensitive, as their relative performance does not change significantly between direct *k*-way and RB.

The fact that for many of the clustering criterion functions the quality of the solutions obtained via repeated bisections is better than that achieved by direct *k*-way clustering is both surprising and alarming. This is because, even-though the solution obtained by the RB approach is not even at a local minima with respect to the particular criterion function, it leads to qualitatively better clusters. Intuitively, we expected that direct *k*-way will be strictly better than RB and the fact that this does not happen suggests that there may be some problems with some of the criterion functions. This will be further discussed and analyzed in Section 5.

4.5 Evaluation of k-way Clustering via Repeated Bisections followed by k-way Refinement

To further investigate the surprising behavior of the RB-based clustering approach we performed a sequence of experiments in which the final solution obtained by the RB-approach for a particular criterion functions, was further refined using a greedy *k*-way refinement algorithm whose goal was to optimize the particular criterion function. The *k*-way refinement algorithm that we used is identical to that described in Section 3.2. We will refer to this scheme as *RB-k-way*. The detailed experimental results from this sequence of experiments is shown in Table 6, and the summary of these results in terms of average relative entropies and purities is shown in Table 7.

Comparing the relative performance of the various criterion functions we can see that they are more similar to those of direct *k*-way (Table 3) than those of the RB-based approach (Table 5). In particular, $\mathcal{I}_2$, $\mathcal{E}_1$, $\mathcal{H}_1$, and $\mathcal{H}_2$ tend to outperform the rest, with $\mathcal{I}_2$ doing the best in terms of entropy and $\mathcal{H}_2$ doing the best in terms of purity. Also, we can see that both $\mathcal{I}_1$, $\mathcal{I}_3$, $\mathcal{G}_1$, and $\mathcal{G}_2$ are considerably worse than the best scheme. Figure 2 compares the relative quality of the RB-*k*-way solutions to the solutions obtained by the RB-based scheme. These plots were generated using the same method for generating the plots in Figure 1. Looking at these results we can see that by optimizing the $\mathcal{I}_1$, $\mathcal{E}_1$, $\mathcal{G}_1$, and $\mathcal{G}_2$ criterion functions, the quality of the solutions become worse, especially for large number of clusters. The largest degradation happens for $\mathcal{G}_1$ and $\mathcal{G}_2$. On the other hand, as we optimize either $\mathcal{I}_2$, $\mathcal{H}_1$, or $\mathcal{H}_2$, the overall cluster quality changes only slightly (sometimes it gets better and sometimes it gets worse). These results verify the observations we made in Section 4.4 that suggest that the optimization of some of the criterion functions does not necessarily lead to better quality clusters, especially for large values of *k*.

5 Discussion & Analysis

The experimental evaluation of the various criterion functions presented in Section 4 show two interesting trends. First, the quality of the clustering solutions produced by some seemingly similar criterion functions is often substantially different. For instance, all three internal criterion functions, $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$, try to produce a clustering solution

Entropy

	5-way Clustering								10-way Clustering							
DataSet	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
classic	0.40	0.23	0.38	0.21	0.30	0.33	0.26	0.32	0.27	0.23	0.23	0.17	0.16	0.18	0.24	0.17
fbis	0.49	0.50	0.49	0.51	0.48	0.50	0.53	0.55	0.39	0.39	0.40	0.42	0.40	0.42	0.43	0.48
hitech	0.75	0.64	0.69	0.64	0.66	0.62	0.67	0.72	0.67	0.59	0.63	0.61	0.61	0.59	0.62	0.67
k1a	0.55	0.49	0.52	0.50	0.51	0.49	0.48	0.52	0.44	0.40	0.43	0.42	0.40	0.40	0.41	0.42
k1b	0.23	0.22	0.22	0.24	0.23	0.23	0.19	0.21	0.22	0.15	0.20	0.17	0.17	0.16	0.13	0.15
la12	0.65	0.47	0.57	0.46	0.41	0.40	0.53	0.44	0.47	0.39	0.45	0.40	0.37	0.38	0.41	0.43
new3	0.76	0.69	0.76	0.69	0.71	0.69	0.71	0.75	0.65	0.58	0.64	0.59	0.60	0.59	0.61	0.68
ohscal	0.75	0.61	0.74	0.62	0.61	0.61	0.61	0.68	0.63	0.54	0.62	0.53	0.56	0.53	0.59	0.63
re0	0.54	0.50	0.53	0.48	0.49	0.48	0.50	0.56	0.42	0.39	0.41	0.38	0.37	0.37	0.40	0.46
re1	0.58	0.48	0.55	0.50	0.50	0.50	0.50	0.64	0.47	0.42	0.44	0.42	0.41	0.41	0.44	0.57
reviews	0.39	0.35	0.37	0.35	0.35	0.29	0.30	0.58	0.32	0.30	0.31	0.28	0.30	0.29	0.31	0.38
sports	0.44	0.28	0.43	0.24	0.33	0.27	0.34	0.40	0.33	0.17	0.31	0.19	0.22	0.16	0.19	0.34
tr31	0.35	0.31	0.39	0.29	0.31	0.30	0.33	0.36	0.20	0.17	0.26	0.20	0.17	0.18	0.19	0.23
tr41	0.37	0.35	0.37	0.39	0.36	0.33	0.30	0.38	0.28	0.26	0.26	0.28	0.23	0.27	0.28	0.25
wap	0.54	0.49	0.51	0.50	0.51	0.48	0.48	0.53	0.43	0.39	0.42	0.41	0.42	0.40	0.42	0.44
	15-way Clustering								20-way Clustering							
DataSet	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
classic	0.23	0.20	0.23	0.15	0.16	0.17	0.23	0.17	0.23	0.19	0.23	0.17	0.16	0.17	0.21	0.20
fbis	0.36	0.36	0.36	0.37	0.34	0.36	0.38	0.46	0.35	0.33	0.33	0.36	0.32	0.35	0.33	0.45
hitech	0.65	0.58	0.61	0.59	0.60	0.58	0.59	0.66	0.62	0.56	0.59	0.58	0.57	0.57	0.58	0.66
k1a	0.40	0.33	0.36	0.36	0.34	0.34	0.37	0.40	0.38	0.30	0.33	0.33	0.30	0.31	0.33	0.39
k1b	0.19	0.11	0.16	0.17	0.13	0.13	0.13	0.14	0.17	0.11	0.14	0.14	0.12	0.12	0.14	0.14
la12	0.44	0.37	0.41	0.39	0.36	0.38	0.41	0.43	0.40	0.36	0.39	0.38	0.35	0.37	0.40	0.44
new3	0.58	0.52	0.57	0.52	0.53	0.51	0.55	0.64	0.52	0.47	0.51	0.48	0.47	0.46	0.51	0.60
ohscal	0.59	0.51	0.59	0.53	0.53	0.52	0.54	0.64	0.58	0.52	0.57	0.53	0.52	0.52	0.56	0.64
re0	0.37	0.34	0.35	0.36	0.36	0.35	0.33	0.41	0.34	0.32	0.32	0.33	0.34	0.32	0.31	0.39
re1	0.40	0.35	0.39	0.36	0.36	0.34	0.41	0.53	0.38	0.32	0.35	0.33	0.32	0.32	0.39	0.50
reviews	0.28	0.25	0.25	0.25	0.25	0.24	0.25	0.36	0.26	0.24	0.24	0.25	0.25	0.22	0.24	0.37
sports	0.25	0.17	0.23	0.16	0.17	0.17	0.18	0.31	0.23	0.16	0.22	0.17	0.17	0.16	0.19	0.32
tr31	0.19	0.17	0.22	0.20	0.18	0.17	0.20	0.26	0.19	0.16	0.18	0.18	0.15	0.17	0.22	0.25
tr41	0.23	0.21	0.22	0.20	0.21	0.21	0.23	0.24	0.21	0.15	0.17	0.17	0.15	0.17	0.19	0.24
wap	0.38	0.32	0.37	0.35	0.35	0.34	0.36	0.40	0.36	0.29	0.33	0.32	0.32	0.33	0.34	0.39

Purity

	5-way Clustering								10-way Clustering							
DataSet	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
classic	0.78	0.89	0.78	0.92	0.81	0.80	0.86	0.80	0.86	0.87	0.89	0.92	0.94	0.92	0.86	0.93
fbis	0.54	0.51	0.54	0.52	0.53	0.50	0.48	0.50	0.64	0.61	0.63	0.61	0.63	0.56	0.60	0.57
hitech	0.44	0.59	0.50	0.57	0.55	0.58	0.50	0.48	0.51	0.60	0.56	0.60	0.57	0.60	0.57	0.54
k1a	0.44	0.49	0.47	0.48	0.47	0.48	0.49	0.47	0.55	0.60	0.57	0.60	0.60	0.61	0.59	0.60
k1b	0.86	0.80	0.82	0.80	0.80	0.80	0.84	0.84	0.85	0.87	0.85	0.85	0.86	0.88	0.92	0.89
la12	0.53	0.72	0.60	0.72	0.76	0.77	0.65	0.76	0.68	0.78	0.70	0.77	0.79	0.78	0.77	0.74
new3	0.20	0.24	0.20	0.23	0.22	0.24	0.22	0.21	0.29	0.32	0.28	0.32	0.30	0.32	0.29	0.26
ohscal	0.35	0.48	0.35	0.47	0.48	0.47	0.48	0.42	0.48	0.59	0.50	0.59	0.57	0.59	0.51	0.48
re0	0.51	0.53	0.49	0.56	0.52	0.54	0.52	0.46	0.59	0.62	0.59	0.66	0.63	0.66	0.63	0.59
re1	0.43	0.53	0.47	0.53	0.53	0.53	0.53	0.38	0.55	0.61	0.57	0.61	0.61	0.62	0.59	0.45
reviews	0.76	0.79	0.78	0.81	0.79	0.81	0.83	0.56	0.83	0.84	0.83	0.85	0.82	0.83	0.81	0.78
sports	0.64	0.75	0.66	0.81	0.72	0.76	0.72	0.67	0.76	0.88	0.77	0.87	0.83	0.90	0.87	0.74
tr31	0.72	0.76	0.70	0.79	0.76	0.77	0.75	0.72	0.87	0.88	0.80	0.87	0.89	0.88	0.90	0.85
tr41	0.71	0.71	0.70	0.62	0.71	0.75	0.78	0.71	0.77	0.76	0.79	0.73	0.78	0.74	0.73	0.80
wap	0.45	0.49	0.48	0.49	0.47	0.49	0.50	0.46	0.56	0.61	0.57	0.60	0.58	0.62	0.57	0.57
	15-way Clustering								20-way Clustering							
DataSet	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
classic	0.89	0.90	0.89	0.94	0.94	0.93	0.89	0.93	0.89	0.91	0.89	0.93	0.94	0.93	0.89	0.90
fbis	0.66	0.64	0.67	0.64	0.66	0.65	0.65	0.58	0.68	0.69	0.70	0.65	0.69	0.66	0.69	0.59
hitech	0.53	0.60	0.58	0.60	0.57	0.61	0.59	0.54	0.55	0.61	0.58	0.60	0.60	0.61	0.59	0.54
k1a	0.58	0.69	0.64	0.69	0.68	0.69	0.62	0.63	0.62	0.71	0.68	0.70	0.72	0.71	0.69	0.64
k1b	0.88	0.93	0.90	0.88	0.91	0.91	0.92	0.92	0.89	0.92	0.89	0.90	0.91	0.92	0.91	0.92
la12	0.70	0.79	0.74	0.77	0.79	0.78	0.73	0.73	0.74	0.79	0.76	0.78	0.80	0.79	0.75	0.72
new3	0.35	0.38	0.34	0.39	0.37	0.39	0.35	0.31	0.40	0.43	0.41	0.42	0.43	0.44	0.40	0.34
ohscal	0.52	0.60	0.51	0.59	0.58	0.60	0.58	0.47	0.53	0.59	0.54	0.59	0.59	0.59	0.53	0.47
re0	0.61	0.67	0.64	0.67	0.64	0.68	0.66	0.63	0.66	0.68	0.67	0.68	0.67	0.70	0.69	0.64
re1	0.60	0.66	0.61	0.64	0.63	0.66	0.60	0.47	0.61	0.68	0.64	0.67	0.67	0.68	0.61	0.49
reviews	0.82	0.86	0.85	0.86	0.86	0.87	0.86	0.79	0.83	0.87	0.86	0.87	0.85	0.88	0.86	0.77
sports	0.80	0.88	0.81	0.90	0.87	0.89	0.88	0.76	0.82	0.88	0.82	0.89	0.87	0.89	0.85	0.75
tr31	0.87	0.88	0.83	0.87	0.86	0.89	0.86	0.83	0.86	0.88	0.87	0.87	0.89	0.88	0.84	0.83
tr41	0.80	0.81	0.81	0.82	0.79	0.81	0.79	0.81	0.82	0.88	0.85	0.85	0.86	0.85	0.85	0.80
wap	0.62	0.68	0.64	0.69	0.67	0.69	0.64	0.61	0.65	0.71	0.67	0.71	0.70	0.70	0.67	0.63

Table 6: Entropy and Purity for the various datasets and criterion functions for the clustering solutions obtained via repeated bisections followed by k-way refinement.

Average Relative Entropy

k	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
5	1.304	**1.081**	1.256	**1.077**	1.121	**1.076**	**1.097**	1.273
10	1.278	**1.065**	1.240	1.088	**1.063**	**1.051**	1.127	1.255
15	1.234	**1.037**	1.182	1.089	**1.057**	**1.046**	1.140	1.334
20	1.248	**1.030**	1.157	1.098	**1.041**	**1.051**	1.164	1.426
Avg	1.266	**1.053**	1.209	1.088	**1.070**	**1.056**	1.132	1.322

Average Relative Purity

k	$\mathcal{I}_1$	$\mathcal{I}_2$	$\mathcal{I}_3$	$\mathcal{E}_1$	$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{G}_1$	$\mathcal{G}_2$
5	1.181	**1.040**	1.145	**1.039**	1.063	**1.040**	1.063	1.160
10	1.105	**1.026**	1.090	**1.027**	**1.035**	**1.023**	1.058	1.115
15	1.092	**1.015**	1.069	**1.017**	**1.027**	**1.008**	1.051	1.133
20	1.079	**1.010**	1.049	**1.020**	**1.014**	**1.009**	1.049	1.148
Avg	1.114	**1.023**	1.088	**1.026**	**1.035**	**1.020**	1.055	1.139

Table 7: Relative entropies and purities averaged over the different datasets for different criterion functions for the clustering solutions obtained via repeated bisections followed by k-way refinement. Underlined entries represent the best performing scheme, and boldfaced entries correspond to schemes that performed within 2% of the best.

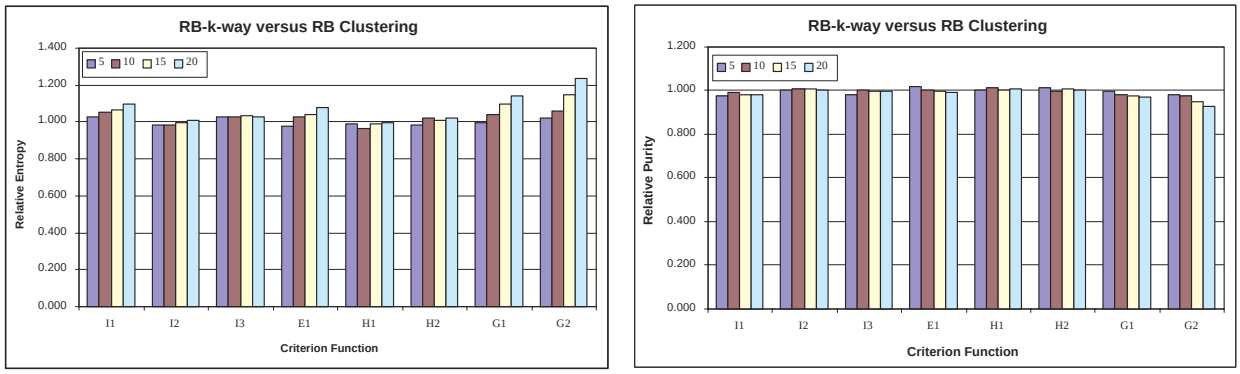

Figure 2: The relative performance of repeated bisections-based clustering followed by k-way refinement over that of repeated bisections alone. The results are averaged over the different datasets, for the entropy and purity measures.

that maximizes a particular within cluster similarity function. However, out of these three criterion functions, $\mathcal{I}_2$ performs substantially better than the rest. This is also true for the $\mathcal{E}_1$ and $\mathcal{G}_1$ criterion functions, that attempt to minimize a function that takes into account both the within cluster similarity and the across cluster dissimilarity. However, in most of the experiments, $\mathcal{E}_1$ tends to perform consistently better than $\mathcal{G}_1$. The second trend is that for many criterion functions, the quality of the solutions produced via repeated bisections is in general better than the corresponding solution produced either via direct k-way clustering or after performing k-way refinement. Furthermore, this performance gap seems to increase with the number of clusters k. In the remaining of this section we analyze the different criterion functions and explain the cause of these trends.

5.1 Analysis of the $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$ Criterion Functions

As a starting point for analyzing the performance of the three internal criterion functions it is important to qualitatively understand how they fail. Table 8 shows the 10-way clustering solutions obtained for the *sports* dataset using each one of the three internal criterion functions. The row of each subtable represents a particular cluster, and it shows the class-distribution of the documents assigned to it. For example, the first cluster for $\mathcal{I}_1$ contains 1034 documents from the "baseball" category and a single document from the "football" category. The columns labeled "Size" show the number of documents assigned to each cluster, whereas the column labeled "Sim" shows the average similarity

between any two documents in each cluster. The last row of each subtable shows the values for the entropy and purity measures for the particular clustering solution. Note that these clusterings were computed using the direct k-way clustering approach.

$\mathcal{I}_1$ Criterion

cid	Size	Sim	baseball	basketball	football	hockey	boxing	bicycling	golfing
1	1035	0.098	1034	1					
2	594	0.125		1	592				1
3	322	0.191		321	1				
4	653	0.127		1		652			
5	413	0.163	413						
6	1041	0.058			1041				
7	465	0.166	464		1				
8	296	0.172		296					
9	3634	0.020	1393	789	694	157	121	145	335
10	127	0.268	108	1	17		1		
Entropy=0.357, Purity=0.736									

$\mathcal{I}_2$ Criterion

cid	Size	Sim	baseball	basketball	football	hockey	boxing	bicycling	golfing
1	475	0.087	97	35	143	8	112	64	16
2	384	0.129	1	1		381			1
3	1508	0.032	310	58	1055	11	5	59	10
4	844	0.094	1	1	841				1
5	400	0.163		1		399			
6	835	0.097	829		6				
7	1492	0.067	1489	1	2				
8	756	0.099	2	752	1	1			
9	621	0.108	618	1	2				
10	1265	0.036	65	560	296	9	5	22	308
Entropy=0.240, Purity=0.824									

$\mathcal{I}_3$ Criterion

cid	Size	Sim	baseball	basketball	football	hockey	boxing	bicycling	golfing
1	2762	0.020	1445	263	348	144	119	123	320
2	569	0.130			568				1
3	334	0.185		333	1				
4	570	0.113		272		298			
5	1144	0.055	67	20	1018	8	2	22	7
6	354	0.188				354			0
7	1009	0.099	1009						
8	991	0.047	45	522	410	5	1		8
9	391	0.171	391						
10	456	0.168	455		1				
Entropy=0.362, Purity=0.745									

Table 8: The cluster-class distribution of the clustering solutions for the $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$ criterion functions for the *sports* dataset.

As shown in Table 8, all three internal criterion functions produce unbalanced clustering solutions, *i.e.* mixtures of large, loose clusters and small, tight clusters. However, $\mathcal{I}_1$ and $\mathcal{I}_3$ behave differently from $\mathcal{I}_2$ in two ways. Both $\mathcal{I}_1$ and $\mathcal{I}_3$ produce solutions in which at least one cluster contains a very large number of documents from different categories with very low pairwise similarities. In the case of $\mathcal{I}_1$, this large *poor* cluster is the ninth, whereas in the case of $\mathcal{I}_3$, the poor cluster is the first one. One the other hand, $\mathcal{I}_2$ does not produce a single very large cluster of very poor quality. That is why $\mathcal{I}_1$ and $\mathcal{I}_3$ produce clustering solutions that have a considerably higher entropy (or lower purity) than the $\mathcal{I}_2$ criterion function. The second qualitative difference between the clustering solutions produced by $\mathcal{I}_1$ and $\mathcal{I}_3$ over that of $\mathcal{I}_2$ is that if we exclude the large poor clusters, the remaining of the clusters tend to be quite pure as well as relatively *tight* (*i.e.*, the average similarity between their documents is high). The $\mathcal{I}_2$ criterion function also produces fairly pure clusters, but they tend to contain somewhat more noise and be less tight. These observations on the nature of the clustering solutions produced by the three criterion functions also hold for the remaining of the datasets and

they are not specific to the *sports* dataset.

To analyze this behavior we will focus on the conditions under which the movement of a particular document from one cluster to another will lead to an improvement in the overall criterion function. Consider a k-way clustering solution, let S_i and S_j be two of these clusters, and d be a particular document that is initially part of S_i. Furthermore, let μ_i and μ_j be the average similarity between the documents in S_i and S_j, respectively (*i.e.*, $\mu_i = {C_i}^t C_i$, and $\mu_j = {C_j}^t C_j$), and let δ_i and δ_j be the average similarity between d and the documents in S_i and S_j, respectively (*i.e.*, $\delta_i = d^t C_i$, and $\delta_j = d^t C_j$).

It is shown in Appendix A that according to the $\mathcal{I}_1$ criterion function the document d will be moved from S_i to S_j iff

$$\delta_i - \delta_j < \frac{\mu_i - \mu_j}{2}, \tag{18}$$

and in the case of the $\mathcal{I}_2$ criterion function this move will happen iff

$$\frac{\delta_i}{\delta_j} < \sqrt{\frac{\mu_i}{\mu_j}}. \tag{19}$$

According to Equation 18, $\mathcal{I}_1$ will move the document d from the S_i cluster to the S_j cluster, iff the difference between the average similarities of d to the documents of each cluster ($\delta_i - \delta_j$) is less than half of the difference between the average similarity among the documents in S_i and S_j ($(\mu_i - \mu_j)/2$). In addition, as long as μ_i is much larger than μ_j, then even if δ_j is zero, $\mathcal{I}_1$ will still move to S_j the documents of S_i that have low average similarity with the other documents of that cluster, *i.e.*, δ_i is small. Thus, if two clusters contain documents that have substantially different average pairwise similarities, the $\mathcal{I}_1$ criterion function will tend to move some of the *peripheral* documents from the tight cluster to the loose cluster. This observation explains the results shown in Table 8, in which $\mathcal{I}_1$'s clustering solution contains nine fairly pure and tight clusters, and a single large and poor quality cluster. That single cluster acts almost like a *garbage collector* which attracts all the peripheral documents of the other clusters.

According to Equation 19, $\mathcal{I}_2$ will move the document d from the S_i cluster to the S_j cluster, iff the ratio between the average similarities of d to the documents of each cluster (δ_i/δ_j) is less than the square root of the ratio between the average similarity among the documents in S_i and S_j ($\sqrt{\mu_i/\mu_j}$). $\mathcal{I}_2$ will also lead unbalanced solutions by noticing that when δ_i and δ_j are similar and relatively small, $\mathcal{I}_2$ will assign the document d to the looser cluster, which makes the tight cluster tighter, and the loose cluster looser.

Comparing Equation 18 with Equation 19, we can make two observations that explain the different behavior of $\mathcal{I}_1$ and $\mathcal{I}_2$, under the condition $\mu_i > \mu_j$. First, unlike $\mathcal{I}_1$ that will move a document to cluster S_j even if δ_j is zero as long as $(\mu_i - \mu_j)/2 > \delta_i$ (*i.e.*, it is a peripheral document of a relatively tight cluster), $\mathcal{I}_2$ will move that document only if it has a non-trivial average similarity to the documents of S_j. If this is not true, δ_i/δ_j will be fairly large, potentially violating the required condition (Equation 19). Second, when δ_i and δ_j are relatively small, that is

$$\delta_j < \mu_j \frac{\alpha - 1}{2(\sqrt{\alpha} - 1)} \quad \text{and} \quad \delta_i < \mu_i \frac{\sqrt{\alpha}(\alpha - 1)}{2(\sqrt{\alpha} - 1)}, \quad \text{where} \quad \alpha = \frac{\mu_i}{\mu_j},$$

the move condition of $\mathcal{I}_1$ can be satisfied more easily than that of $\mathcal{I}_2$, (*i.e.*, the range of δ_i and δ_j values to meet the move condition of $\mathcal{I}_1$ is larger than that of $\mathcal{I}_2$). Given the same δ_j, $\mathcal{I}_1$ can move documents with higher δ_i than $\mathcal{I}_2$. To this extend, $\mathcal{I}_1$ is more powerful to pull the peripheral documents of the tight cluster towards the loose cluster. For these two reasons, $\mathcal{I}_2$ does not lead to clustering solutions in which there exist one single large cluster that contains peripheral documents from the rest of the clusters and makes those clusters very pure and tight.

Moreover, when documents have relatively high degree of similarity to other documents in S_j and S_i, that is

$$\delta_j > \mu_j \frac{\alpha - 1}{2(\sqrt{\alpha} - 1)} \quad \text{and} \quad \delta_i > \mu_i \frac{\sqrt{\alpha}(\alpha - 1)}{2(\sqrt{\alpha} - 1)}, \quad \text{where} \quad \alpha = \frac{\mu_i}{\mu_j},$$

$\mathcal{I}_2$ tends to more frequently move them from the tight cluster to the loose cluster compared to the $\mathcal{I}_1$ criterion function, as long as $\delta_j/\sqrt{\mu_j} > \delta_i/\sqrt{\mu_i}$.

To graphically illustrate this Figure 3 shows the range of δ_i and δ_j values for which the movement of a particular document d from the ith to the jth cluster leads to an improvement in either the $\mathcal{I}_1$ or $\mathcal{I}_2$ criterion function. The plots in Figure 3(a) were obtained using $\mu_i = .10$, $\mu_j = 0.05$, whereas the plot in Figure 3(b) were obtained using $\mu_i = .20$ and $\mu_j = 0.05$. For both sets of plots was used $n_i = n_j = 400$. The x-axis of the plots in Figure 3 correspond to δ_j, whereas the y-axis corresponds to δ_i. For both cases, we let these average similarities take values between zero and one. The various regions in the plots of Figure 3 are labeled based on whether or not any of the criterion functions will move d to the other cluster, based on the particular set of δ_i and δ_j values.

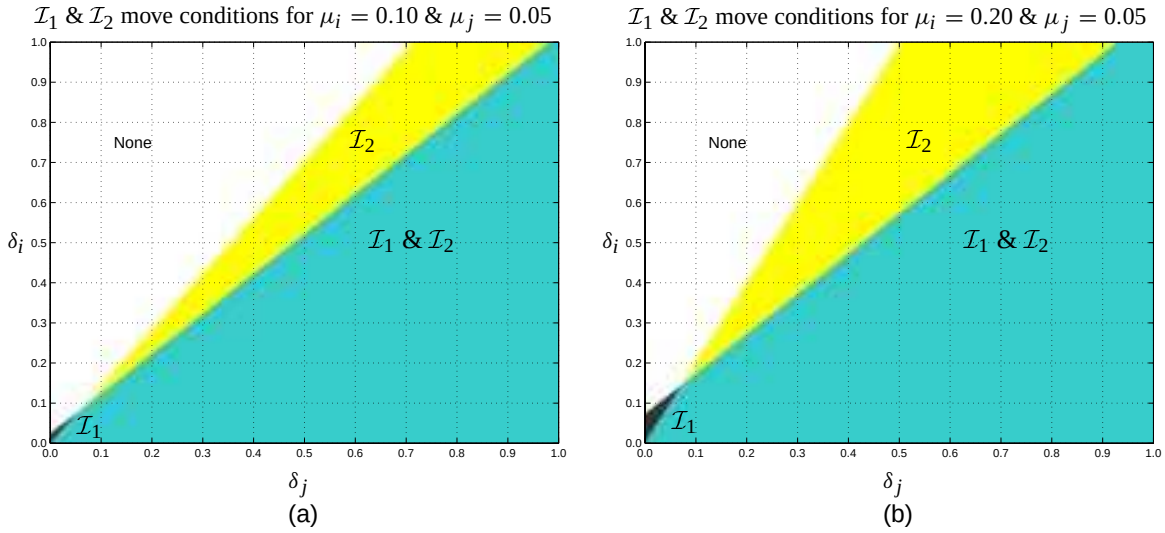

Figure 3: The range of values of δ_i and δ_j for which a particular document d will move from the ith to the jth cluster. The first plot (a) shows the ranges when the average similarity of the documents in the ith and jth cluster are 0.10 and 0.05, respectively. The second plot (b) shows the ranges when the respective similarities are 0.20 and 0.05. For both cases each of the clusters was assumed to have 400 documents.

Looking at these plots we can see that there is a region of small δ_i and δ_j values for which $\mathcal{I}_1$ will perform the move where $\mathcal{I}_2$ will not. These conditions are the ones that we already discuss and are the main reason why $\mathcal{I}_1$ tends to create a large poor quality cluster and $\mathcal{I}_2$ does not. There is also a region for which $\mathcal{I}_2$ will perform the move but $\mathcal{I}_1$ will not. This is the region for which $\delta_i > \delta_j + (\mu_i - \mu_j)/2$ but $\delta_j/\sqrt{\mu_j} > \delta_i/\sqrt{\mu_i}$. That is the average similarity between document d and cluster S_j relative to the square-root of the internal similarity of S_j is greater than the corresponding quantity of S_i. Moreover, as the plots illustrated, the size of this region increases as the difference between the tightness of the two clusters increases.

The justification for this type of moves is that d behaves more like the documents in S_j (as measured by $\sqrt{\mu_j}$) than the documents in S_i. To that extend, $\mathcal{I}_2$ exhibits some dynamic modeling characteristics [24], in the sense that its move is based both on how close it is to a particular cluster as well as on the properties of the cluster itself. However, even though the principle of dynamic modeling has been shown to be useful for clustering, it may sometimes lead to errors as primary evidence of cluster membership (*i.e.*, the actual δ_i & δ_j values) are second guessed. This may be one of the reasons why the $\mathcal{I}_2$ criterion function leads to clusters that in general are more noisy than the corresponding clusters of $\mathcal{I}_1$, as the example in Table 8 illustrates.

Our discussion so far focused primarily on the $\mathcal{I}_1$ and $\mathcal{I}_2$ criterion functions. However, since the cosine and Euclidean distance functions are similar to each other, and because $\mathcal{I}_3$ is identical in nature to $\mathcal{I}_1$, $\mathcal{I}_3$ exhibits similar characteristics with $\mathcal{I}_1$. To see this, recall from Equation 10 that the $\mathcal{I}_3$ criterion function can be re-written as:

$$\mathcal{I}_3 = \sum_{r=1}^{k} \frac{1}{n_r} \sum_{d_i, d_j \in S_r} \|d_i - d_j\|^2.$$

Now, using some basic trigonometric manipulations we have that

$$\|d_i - d_j\|^2 = \sin^2(d_i, d_j) + (1 - \cos(d_i, d_j))^2 = 2(1 - \cos(d_i, d_j)).$$

Using this relation, Equation 10 can be re-written as:

$$\mathcal{I}_3 = \sum_{r=1}^{k} \frac{1}{n_r} \sum_{d_i, d_j \in S_r} 2(1 - \cos(d_i, d_j)) = 2\left(\sum_{r=1}^{k} n_r - \sum_{r=1}^{k} \frac{1}{n_r} \sum_{d_i, d_j \in S_r} \cos(d_i, d_j)\right) = 2(n - \mathcal{I}_1).$$

Thus, minimizing $\mathcal{I}_3$ is the same as maximizing $\mathcal{I}_1$.

5.2 Analysis of the $\mathcal{E}_1$ and $\mathcal{G}_1$ Criterion Functions

The $\mathcal{E}_1$ and $\mathcal{G}_1$ criterion functions both measure the quality of the overall clustering solution by taking into account both the separation between clusters and the *tightness* of each cluster. However, as the experiments presented in Section 4 show $\mathcal{E}_1$ leads to better clustering solutions that $\mathcal{G}_1$ for all three sets of experiments. Furthermore, the highest performance difference between these two criterion functions occurs during the direct k-way clustering. Table 9 shows the 10-way clustering solutions for the *sports* data set produced by $\mathcal{E}_1$ and $\mathcal{G}_1$ that illustrate this difference in the overall clustering quality. As we can see from these results the $\mathcal{E}_1$ criterion function leads to clustering solutions that are considerably more balanced than those produced by the $\mathcal{G}_1$ criterion function. In fact, the solution obtained by the $\mathcal{G}_1$ criterion function exhibits similar characteristics (but to a lesser extend) with the corresponding solutions obtained by the $\mathcal{I}_1$ and $\mathcal{I}_3$ criterion functions described in the previous section. It tends to produce a mixture of large and small clusters, with the smaller clusters being quite tight and the larger clusters being quite loose.

$\mathcal{E}_1$ Criterion

cid	Size	Sim	baseball	basketball	football	hockey	boxing	bicycling	golfing
1	1330	0.076	1327	2	1				
2	975	0.080	3	5	966				1
3	742	0.072	15	703	24				
4	922	0.079	84	8	32	797			1
5	768	0.078	760	1	6		1		
6	897	0.054	6	2	889				
7	861	0.091	845	0	15				1
8	565	0.079	24	525	13	1			2
9	878	0.034	93	128	114	4	97	121	321
10	642	0.068	255	36	286	7	24	24	10
Entropy=0.203, Purity=0.865									

$\mathcal{G}_1$ Criterion

cid	Size	Sim	baseball	basketball	football	hockey	boxing	bicycling	golfing
1	519	0.146	516		3				
2	597	0.118	1		595				1
3	1436	0.033	53	580	357	13	100	20	313
4	720	0.105		718	1	1			
5	1664	0.032	1387	73	77	49	7	63	8
6	871	0.101	871						
7	1178	0.049	6	5	1167				
8	728	0.111		1		727			
9	499	0.133	498		1				
10	368	0.122	80	33	145	19	15	62	14
Entropy=0.239, Purity=0.840									

Table 9: The cluster-class distribution of the clustering solutions for the $\mathcal{E}_1$ and $\mathcal{G}_1$ criterion functions for the *sports* dataset.

In order to compare the $\mathcal{E}_1$ and $\mathcal{G}_1$ criterion functions it is important to rewrite them in a way that makes their similarities and dissimilarities apparent. To this end, let μ_r be the average similarity between the documents of the rth

cluster S_r, and let ξ_r be the average similarity between the documents in S_r to the entire set of documents S. Using these definitions, the $\mathcal{E}_1$ criterion function (Equation 12) can be rewritten as

$$\mathcal{E}_1 = \sum_{r=1}^{k} n_r \frac{{D_r}^t D}{\|D_r\|} = \sum_{r=1}^{k} n_r \frac{n_r n \xi_r}{n_r \sqrt{\mu_r}} = n \sum_{r=1}^{k} n_r \frac{\xi_r}{\sqrt{\mu_r}}, \tag{20}$$

and the $\mathcal{G}_1$ criterion function (Equation 16) can be rewritten as

$$\mathcal{G}_1 = \sum_{r=1}^{k} \frac{{D_r}^t D}{\|D_r\|^2} = \sum_{r=1}^{k} \frac{n_r n \xi_r}{n_r^2 \mu_r} = n \sum_{r=1}^{k} \frac{1}{n_r} \frac{\xi_r}{\mu_r}. \tag{21}$$

Comparing Equations 20 and 21 we can see that they differ in two ways. The first difference has to do with the way they measure the quality of a particular cluster, and the second has to do with the way they combine the individual cluster quality measures to derive the overall quality of the clustering solution.

In the case of $\mathcal{E}_1$, the quality of the rth cluster is given by $\xi_r / \sqrt{\mu_r}$ whereas in the case of $\mathcal{G}_1$ the quality is measured as ξ_r / μ_r. Since for both $\mathcal{E}_1$ and $\mathcal{G}_1$, the quality of each cluster is inversely related to either μ_r or $\sqrt{\mu_r}$, both of them will prefer clustering solutions in which there are no clusters that are extremely loose (*i.e.*, they have small μ_r values). Now, because large clusters tend to have small μ_r values, both of the cluster quality measures will tend to produce solutions that contain reasonably balanced clusters. Furthermore, because $\mu_r \leq 1$, we have that $\mu_r \leq \sqrt{\mu_r}$, which in turn implies that the sensitivity of $\mathcal{G}_1$'s cluster quality measure on clusters with small μ_r values is higher than the corresponding sensitivity of $\mathcal{E}_1$. Consequently, due to the way $\mathcal{G}_1$ measures the quality of a cluster, we would have expected it to lead to more balanced clustering solutions than $\mathcal{E}_1$, which as the results in Table 9 show it does not happen. As a result, the unbalanced clusters produced by $\mathcal{G}_1$ cannot be attributed to this difference.

This suggest that the second difference between $\mathcal{E}_1$ and $\mathcal{G}_1$, that is, the way they combine the individual cluster quality measures to derive the overall quality of the clustering solution, is the reason for the unbalanced clusters. The $\mathcal{E}_1$ criterion function sums the individual cluster qualities weighting them proportionally to the size of each cluster. $\mathcal{G}_1$ performs a similar summation but each cluster quality is weighted proportionally to the *inverse* of the size of the cluster. This weighting scheme is similar in nature to that used in the *ratio-cut* objective—used widely in graph partitioning. This difference in how the individual cluster qualities are weighted is the reason why $\mathcal{G}_1$ leads to significantly more unbalanced clustering solutions than $\mathcal{I}_1$.

This is because of the following reason. Recall from our previous discussion that since the quality measure of each cluster is inversely related on μ_r, the quality measure of large clusters will have large values, as these clusters will tend to be loose (*i.e.*, μ_r will be small). Now, in the case of $\mathcal{E}_1$, by multiplying the quality measure of a cluster by its size, it ensures that these large loose clusters contribute a lot to the overall value of $\mathcal{E}_1$'s criterion function. As a result, $\mathcal{E}_1$ will tend to be optimized when there are no large loose clusters. On the other hand, in the case of $\mathcal{G}_1$, by dividing the quality measure of a large loose cluster by its size, it has the net effect of decreasing the contribution of this cluster to the overall value of $\mathcal{G}_1$'s criterion function. As a result, $\mathcal{G}_1$ can be optimized at a point in which there exist some large and loose clusters.

To illustrate this, we created a new criterion function that is derived from $\mathcal{G}_1$'s cluster quality measure but uses $\mathcal{E}_1$'s combining mechanism. That is, this new criterion function $\mathcal{G}_1'$ is defined as follows:

$$\text{minimize} \;\; \mathcal{G}_1' = n \sum_{r=1}^{k} n_r \frac{\xi_r}{\mu_r} = \sum_{r=1}^{k} n_r^2 \frac{{D_r}^t D}{\|D_r\|^2} \tag{22}$$

We used $\mathcal{G}_1'$ to find a 10-way clustering solution of the *sports* dataset which is shown in Table 10. Comparing the clustering solution produced by $\mathcal{G}_1'$ to that produced by $\mathcal{G}_1$ (Table 9) we can see that $\mathcal{G}_1'$'s solution is more balanced and it achieves substantially lower entropy.

$\mathcal{G}'_1$ Criterion

cid	Size	Sim	baseball	basketball	football	hockey	boxing	bicycling	golfing
1	972	0.081	948		24				
2	948	0.075	81	13	61	792			1
3	528	0.051	3	107	11	2	86	1	318
4	898	0.079	19	861	17	1			
5	806	0.076	795	1	9		1		
6	988	0.077	5	2	980				1
7	793	0.058	2		791				
8	713	0.053	46	388	272	2			5
9	586	0.061	167	37	180	12	35	144	11
10	1348	0.075	1346	1	1				
Entropy=0.189, Purity=0.862									

Table 10: The cluster-class distribution of the clustering solutions for the $\mathcal{G}'_1$ criterion function for the *sports* dataset.

5.3 Analysis of the $\mathcal{G}_2$ Criterion Function

The various experiments presented in Section 4 showed that the $\mathcal{G}_2$ criterion function consistently led to clustering solutions that were among the worst over the solutions produced by the various criterion functions that were considered in this study. To illustrate how the $\mathcal{G}_2$ criterion function fails, Table 11 shows the 10-way clustering solution produced via direct *k*-way clustering on the *sports* dataset.

$\mathcal{G}_2$ Criterion

cid	Size	Sim	baseball	basketball	football	hockey	boxing	bicycling	golfing
1	491	0.096	1	5	485				
2	1267	0.056	8	5	1244	10			
3	42	0.293	2	1	3		1	35	
4	630	0.113	0	627	2	1			
5	463	0.126	462		1				
6	2596	0.027	1407	283	486	184	42	107	87
7	998	0.040	49	486	124	8	79	3	249
8	602	0.120		1		601			
9	1202	0.081	1194	2	1	5			
10	289	0.198	289						
Entropy=0.315, Purity=0.796									

Table 11: The cluster-class distribution of the clustering solutions for the $\mathcal{G}_2$ criterion function for the *sports* dataset.

Looking at this solution we can see that $\mathcal{G}_2$ produces solutions that are highly unbalanced. For example, the sixth cluster contains over 2500 documents from many difference categories, whereas the third cluster contains only 42 documents that are primarily from a single category. Note that, the clustering solution produced by $\mathcal{G}_2$ is very similar to that produced by the $\mathcal{I}_1$ criterion function (Table 8). In fact, for most of the clusters we can find a good one-to-one mapping between the two schemes.

The nature of $\mathcal{G}_2$'s criterion function makes it extremely hard to analyze it. However, one reason that can potentially explain the unbalanced clusters produced by $\mathcal{G}_2$ is the fact that it uses a normalized-cut inspired approach to trade-off separation between the clusters (as measured by the cut) versus the size of the respective clusters. It has been shown in [11] that when the normalized cut approach is used in the context of traditional graph partitioning, it leads to a solution that is considerably more unbalanced than that obtained by the $\mathcal{G}_1$ criterion function. However, as our discussion in Section 5.2 showed, even $\mathcal{G}_1$'s balancing mechanism often leads to quite unbalanced clustering solutions.

5.4 Analysis of the $\mathcal{H}_1$ and $\mathcal{H}_2$ Criterion Functions

The last set of criterion function that we will focus on are the hybrid criterion functions $\mathcal{H}_1$ and $\mathcal{H}_2$ that were derived by combining the $\mathcal{I}_1$ and $\mathcal{E}_1$ and the $\mathcal{I}_2$ and $\mathcal{E}_1$ criterion functions, respectively. The 10-way clustering solutions

produced by these criterion functions on the *sports* dataset are shown in Table 12. Looking at the results in this table and comparing them against the results produced by the $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{E}_1$, criterion functions we can see that $\mathcal{H}_1$ and $\mathcal{H}_2$ lead to clustering solutions that combine the characteristics of their respective pairs of individual criterion functions. In particular, the $\mathcal{H}_1$ criterion function leads to a solution that is considerably more balanced than that of $\mathcal{I}_1$ and somewhat more unbalanced than that of $\mathcal{E}_1$. Similarly, $\mathcal{H}_2$'s solution is also more balanced than $\mathcal{I}_2$ and somewhat less balanced than $\mathcal{E}_1$.

$\mathcal{H}_1$ Criterion

cid	Size	Sim	baseball	basketball	football	hockey	boxing	bicycling	golfing
1	1220	0.049	60	20	1131	5	2		2
2	724	0.106		722	1	1			
3	696	0.111		1	694				1
4	1469	0.070	1468	1					
5	562	0.138	560		2				
6	576	0.118	574	1	1				
7	764	0.108	1	1		762			
8	1000	0.045	63	554	370	5	1		7
9	1261	0.023	397	109	130	36	118	145	326
10	308	0.116	289	1	17		1		
Entropy=0.221, Purity=0.833									

$\mathcal{H}_2$ Criterion

cid	Size	Sim	baseball	basketball	football	hockey	boxing	bicycling	golfing
1	1462	0.997	1457	2	3				
2	908	0.994	2	2	903				1
3	707	0.960	11	679	17				
4	831	0.957	23	4	8	795			1
5	701	0.989	693	1	6		1		
6	999	0.978	15	7	977				
7	830	0.986	818		11				1
8	526	0.949	17	499	7	1			2
9	997	0.321	128	181	149	5	101	113	320
10	619	0.428	248	35	265	8	20	32	11
Entropy=0.196, Purity=0.863									

Table 12: The cluster-class distribution of the clustering solutions for the $\mathcal{H}_1$ and $\mathcal{H}_2$ criterion functions for the *sports* dataset.

Overall, from the experiments in Section 4 we can see that the quality of the solutions (as measured by entropy) produced by $\mathcal{H}_1$ tends to be between that of $\mathcal{I}_1$ and $\mathcal{E}_1$—but closer to that of $\mathcal{E}_1$'s; and the solution produced by $\mathcal{H}_2$ tends to be between that of $\mathcal{I}_2$ and $\mathcal{E}_1$—but closer to that of $\mathcal{I}_2$'s. If the quality is measured in terms of purity, the performance of $\mathcal{H}_1$ relative to $\mathcal{I}_1$ and $\mathcal{E}_1$ remains the same, whereas $\mathcal{H}_2$ tends to outperform both $\mathcal{I}_2$ and $\mathcal{E}_1$.

To understand how these criterion functions consider the conditions under which a particular document d will move from its current cluster S_i to another cluster S_j. This document will always be moved (or stay where it is), if each one of the two criterion functions used to define either $\mathcal{H}_1$ or $\mathcal{H}_2$ would improve (or degrade) by performing such a move. The interesting case happens when according to one criterion function d should be moved and according to the other one d should remain where it is. In that case, the overall decision will depend at how much a particular criterion function improves relative to the degradation of the other function. In general, if such a move leads to a large improvement and a small degradation, it is performed. In order to make such trade-offs possible it is important for the pair of criterion functions involved to take roughly the same range of values (*i.e.*, be of the same order). If that is not true, then improvements in one criterion function will not be comparable to degradations in the other.

In the case of the $\mathcal{H}_1$ and $\mathcal{H}_2$ criterion functions, our studies showed that as long as k is sufficiently large, both the $\mathcal{I}_1$ and $\mathcal{I}_2$ criterion functions are of the same order than $\mathcal{E}_1$. However, in most cases $\mathcal{I}_2$ is closer to $\mathcal{E}_1$ that $\mathcal{I}_1$. This better match between the $\mathcal{I}_2$ and $\mathcal{E}_1$ criterion functions may explain why $\mathcal{H}_2$ seems to perform better than $\mathcal{H}_1$ relative to their respective pairs of criterion functions, and why $\mathcal{H}_1$'s solutions are much closer to those of $\mathcal{E}_1$ instead of $\mathcal{I}_1$.

5.5 Analysis of Direct k-way Clustering versus Repeated Bisections

As discussed in the beginning of this section, the experiments presented in Section 4 show that for most criterion functions, for sufficiently large values of k, the clustering solutions produced by repeated bisections are better than the solutions obtained via direct k-way clustering. We believe this is because of the following reason.

From our analysis of the $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_3$, and $\mathcal{G}_1$ criterion functions we know that based on the difference between the tightness (*i.e.*, the average pairwise similarity between the documents in the cluster) of the two clusters, documents that are naturally part of the tighter cluster will end up moving to the looser cluster. In other words, the various criterion functions will tend to produce incorrect clustering results when clusters have different degrees of tightness. Of course, the degree to which a particular criterion function is sensitive to tightness differences will be different for the various criterion functions.

Now, when the clustering solution is obtained via repeated bisections, the difference in tightness between each pair of clusters in successive bisections will tend to be relatively small. This is because, each cluster to be bisected, will tend to be relatively homogeneous (due to the way it was discovered), resulting in a pair of subclusters with small density differences. On the other hand, when the clustering is computed directly or when the final k-way clustering obtained via a sequence of repeated bisections is refined, there can exist clusters that have significant differences in tightness. Whenever there exist such pairs of clusters, most of the criterion function will end up moving some of their documents of the tighter cluster (that are weakly connected to the rest of the documents in that cluster) to the looser cluster. Consequently, the final clustering solution can potentially be worse than that obtained via repeated bisections.

To illustrate this behavior we used the $\mathcal{I}_2$ criterion function and computed a 15-way clustering solution using repeated bisections, and then refined this solution by performing a 15-way refinement. These results are shown in Table 13. The repeated-bisections solution contains some clusters that are quite loose as well as some clusters that are quite tight. Comparing this solution against the one obtained after performing k-way refinement we can see that the size of cluster 6 and 8 (which are among the looser clusters) increased substantially, whereas the size of some of the tighter clusters decreased (*e.g.*, cluster 5, 10, and 14).

Finally, in the case of $\mathcal{E}_1$, the reason that the clusters produced by direct k-way clustering are worse than the corresponding clusters produced via repeated bisections has to do with the tendency of $\mathcal{E}_1$ to produce solutions that are balanced. As a result, the degree of cluster size imbalance is greater when the clusters are obtained via repeated bisections, than the corresponding imbalance of direct k-way clustering. We believe that this additional constraint of the k-way clustering is the reason for the somewhat worse performance observed for direct k-way clustering.

6 Concluding Remarks

In this paper we studied eight different global criterion functions for clustering large documents datasets. Five of these functions ($\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_3$, $\mathcal{G}_1$, and $\mathcal{G}_2$) have been previously proposed for document clustering, whereas the remaining three ($\mathcal{E}_1$, $\mathcal{H}_1$, and $\mathcal{H}_2$) were introduced by us. Our study consisted of a detailed experimental evaluation using fifteen different datasets and three different approaches to find the desired clusters, followed by a theoretical analysis of the characteristics of the various criterion functions. Our analysis showed that the performance difference observed by the various criterion functions can be attributed to the extend to which the criterion functions are sensitive to clusters of different degrees of tightness, and the extend to which they can lead to reasonably balanced solutions. Moreover, our analysis was able to identify the deficiencies of the $\mathcal{G}_1$ criterion function and provide guidance on how to improve it ($\mathcal{G}'_1$).

Our experiments showed that the traditional criterion function used by the vector-space K-means ($\mathcal{I}_2$) lead to reasonably good results, and that some of the recently proposed criterion functions ($\mathcal{G}_1$ and $\mathcal{G}_2$) perform worse. Our three new criterion functions performed reasonably well, with the $\mathcal{H}_2$ criterion function achieving the best overall results.

$\mathcal{I}_2$ Criterion - Repeated Bisections

cid	Size	Sim	baseball	basketball	football	hockey	boxing	bicycling	golfing
1	245	0.121	243	0	2				
2	596	0.067	2	1	593				
3	485	0.097	1	480	3	1			
4	333	0.080	3	6	3		2	1	318
5	643	0.104	642		1				
6	674	0.047	669	2	1	1	1		
7	762	0.099		1	760				1
8	826	0.045	42	525	247	6			6
9	833	0.105	832	1					
10	795	0.102	1	1	1	791			1
11	579	0.061	6		573				
12	647	0.034	174	34	156	10	119	144	10
13	191	0.110	189		2				
14	611	0.125	608		3				
15	360	0.168		359	1				
Entropy=0.125, Purity=0.904									

$\mathcal{I}_2$ Criterion — After k-way Refinement

cid	Size	Sim	baseball	basketball	football	hockey	boxing	bicycling	golfing
1	292	0.120	280		11		1		
2	471	0.080	1	2	468				
3	468	0.100	1	464	2	1			
4	363	0.072	3	7	5	1	6	20	321
5	545	0.123	542	1	2				
6	1030	0.033	832	36	73	18	4	65	2
7	661	0.110	1	0	660				
8	914	0.046	52	514	334	8	1		5
9	822	0.105	822						
10	771	0.105	1	1		769			
11	641	0.052	2		639				
12	447	0.091	89	30	139	11	110	60	8
13	250	0.105	244		5	1			
14	545	0.138	540		5				
15	360	0.168	2	355	3				
Entropy=0.168, Purity=0.884									

Table 13: The cluster-class distribution of the clustering solutions for the $\mathcal{I}_2$ criterion function for the *sports* dataset, for the repeated-bisections solution and the repeated-bisections followed by k-way refinement.

References

[1] Charu C. Aggarwal, Stephen C. Gates, and Philip S. Yu. On the merits of building categorization systems by supervised clustering. In *Proc. of the Fifth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pages 352–356, 1999.

[2] M.W. Berry, S.T. Dumais, and G.W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37:573–595, 1995.

[3] D. Boley, M. Gini, R. Gross, E.H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Document categorization and query generation on the world wide web using WebACE. *AI Review)*, 11:365–391, 1999.

[4] D. Boley, M. Gini, R. Gross, E.H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Partitioning-based clustering for web document categorization. *Decision Support Systems (accepted for publication)*, 1999.

[5] Daniel Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4), 1998.

[6] P. Cheeseman and J. Stutz. Baysian classification (autoclass): Theory and results. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI/MIT Press, 1996.

[7] D.R. Cutting, J.O. Pedersen, D.R. Karger, and J.W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the ACM SIGIR*, pages pages 318–329, Copenhagen, 1992.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39, 1977.

[9] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. Technical Report TR-2001-05, Department of Computer Science, University of Texas, Austin, 2001.

[10] I.S. Dhillon and D.S. Modha. Concept decomposition for large sparse text data using clustering. Technical Report Research Report RJ 10147, IBM Almadan Research Center, 1999.

[11] Chris Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst Simon. Spectral min-max cut for graph partitioning and data clustering. Technical Report TR-2001-XX, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, 2001.

[12] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

[13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the Second Int'l Conference on Knowledge Discovery and Data Mining*, Portland, OR, 1996.

[14] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: An efficient clustering algorithm for large databases. In *Proc. of 1998 ACM-SIGMOD Int. Conf. on Management of Data*, 1998.

[15] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. ROCK: a robust clustering algorithm for categorical attributes. In *Proc. of the 15th Int'l Conf. on Data Eng.*, 1999.

[16] E.H. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. WebACE: A web agent for document categorization and exploartion. In *Proc. of the 2nd International Conference on Autonomous Agents*, May 1998.

[17] E.H. Han, G. Karypis, V. Kumar, and B. Mobasher. Hypergraph based clustering in high-dimensional data sets: A summary of results. *Bulletin of the Technical Committee on Data Engineering*, 21(1), 1998.

[18] J. Han, M. Kamber, and A. K. H. Tung. Spatial clustering methods in data mining: A survey. In H. Miller and J. Han, editors, *Geographic Data Mining and Knowledge Discovery*. Taylor and Francis, 2001.

[19] W. Hersh, C. Buckley, T.J. Leone, and D. Hickam. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *SIGIR-94*, pages 192–201, 1994.

[20] J. E. Jackson. *A User's Guide To Principal Components*. John Wiley & Sons, 1991.

[21] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[22] A.K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[23] G. Karypis and E.H. Han. Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval & categorization. Technical Report TR-00-016, Department of Computer Science, University of Minnesota, Minneapolis, 2000. Available on the WWW at URL *http://www.cs.umn.edu/˜karypis*.

[24] G. Karypis, E.H. Han, and V. Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer*, 32(8):68–75, 1999.

[25] G. Karypis, E.H. Han, and V. Kumar. Multilevel refinement for hierarchical clustering. Technical Report TR-99-020, Department of Computer Science, University of Minnesota, Minneapolis, 1999.

[26] G. Karypis and V. Kumar. A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1), 1999. Also available on WWW at URL http://www.cs.umn.edu/˜karypis. A short version appears in Intl. Conf. on Parallel Processing 1995.

[27] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons, 1990.

[28] B. King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 69:86–101, 1967.

[29] T. Kolda and B. Hendrickson. Partitioning sparse rectangular and structurally nonsymmetric matrices for parallel computation. *SIAM Journal on Scientific Computing*, 21(6):2048–2072, 2000.

[30] Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proc. of the Fifth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pages 16–22, 1999.

[31] R.C.T. Lee. Clustering analysis and its applications. In J.T. Toum, editor, *Advances in Information Systems Science*. Plenum Press, New York, 1981.

[32] D. D. Lewis. Reuters-21578 text categorization test collection distribution 1.0. *http://www.research.att.com/∼lewis*, 1999.

[33] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Symp. Math. Statist, Prob.*, pages 281–297, 1967.

[34] J. Moore, E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, and B. Mobasher. Web page categorization and feature selection using association rule and principal component clustering. In *7th Workshop on Information Technologies and Systems*, Dec. 1997.

[35] R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proc. of the 20th VLDB Conference*, pages 144–155, Santiago, Chile, 1994.

[36] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[37] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.

[38] P. H. Sneath and R. R. Sokal. *Numerical Taxonomy*. Freeman, London, UK, 1973.

[39] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

[40] A. Strehl and J. Ghosh. Scalable approach to balanced, high-dimensional clustering of market-baskets. In *Proceedings of HiPC*, 2000.

[41] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 1999.

[42] TREC. Text REtrieval conference. *http://trec.nist.gov*, 1999.

[43] Xiong Wang, Jason T. L. Wang, Dennis Shasha, Bruce Shapiro, Sitaram Dikshitulu, Isidore Rigoutsos, and Kaizhong Zhang. Automated discovery of active motifs in three dimensional molecules. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 89–95, 1997.

[44] Yahoo! Yahoo! http://www.yahoo.com.

[45] K. Zahn. Graph-tehoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, (C-20):68–86, 1971.

[46] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Bipartite graph partitioning and data clustering. In *CIKM*, 2001.

[47] H. Zha, X. He, C. Ding, H. Simon, and M. Gu. Spectral relaxation for k-means clustering. Technical Report TR-2001-XX, Pennsylvania State University, University Park, PA, 2001.

[48] T. Zhang, R. Ramakrishnan, and M. Linvy. Birch: an efficient data clustering method for large databases. In *Proc. of 1996 ACM-SIGMOD Int. Conf. on Management of Data*, Montreal, Quebec, 1996.

A Analysis of $\mathcal{I}_1$'s and $\mathcal{I}_2$'s Document Move Condition

Consider a k-way clustering solution, let S_i and S_j be two of these clusters, and d be a particular document that is initially part of S_i, and let D_i, C_i, and D_j, C_j be the composite and centroid vectors of these two clusters, such that D_i and C_i contain all the documents of S_i except d. According to the $\mathcal{I}_1$ criterion function (Equation 7) the move of d from S_i to S_j will reduce the overall value of the criterion function if and only if

$$\frac{\|D_i + d\|^2}{n_i + 1} + \frac{\|D_j\|^2}{n_j} < \frac{\|D_i\|^2}{n_i} + \frac{\|D_j + d\|^2}{n_j + 1}.$$

This equation can be rewritten as:

$$\begin{aligned}
\frac{\|D_i + d\|^2}{n_i + 1} - \frac{\|D_i\|^2}{n_i} &< \frac{\|D_j + d\|^2}{n_j + 1} - \frac{\|D_j\|^2}{n_j} \\
\frac{(D_i + d)^t (D_i + d)}{n_i + 1} - \frac{{D_i}^t D_i}{n_i} &< \frac{(D_j + d)^t (D_j + d)}{n_j + 1} - \frac{{D_j}^t D_j}{n_j} \\
\frac{{D_i}^t D_i + 1 + 2d^t D_i}{n_i + 1} - \frac{{D_i}^t D_i}{n_i} &< \frac{{D_j}^t D_j + 1 + 2d^t D_j}{n_j + 1} - \frac{{D_j}^t D_j}{n_j} \\
\frac{2n_i d^t D_i + n_i - {D_i}^t D_i}{n_i(n_i + 1)} &< \frac{2n_j d^t D_j + n_j - {D_j}^t D_j}{n_j(n_j + 1)} \\
2\frac{n_i}{n_i + 1} d^t C_i + \frac{1}{n_i + 1} - \frac{n_i}{n_i + 1} {C_i}^t C_i &< 2\frac{n_j}{n_j + 1} d^t C_j + \frac{1}{n_j + 1} - \frac{n_j}{n_j + 1} {C_j}^t C_j.
\end{aligned}$$

Now, if we assume that both n_i and n_j are sufficiently large, then $n_i/(n_i+1)$ and $n_j/(n_j+1)$ will be close to one, and $1/(n_i+1)$, $1/(n_j+1)$ will be close to zero. Under these assumptions, the various factors involving n_i and n_j can be eliminated leading to

$$2d^t C_i - {C_i}^t C_i < 2d^t C_j - {C_j}^t C_j.$$

Now, if μ_i and μ_j is the average similarity between the documents in S_i and S_j, respectively (*i.e.*, $\mu_i = {C_i}^t C_i$, and $\mu_j = {C_j}^t C_j$), and δ_i and δ_j is the average similarity between d and the documents in S_i and S_j, respectively (*i.e.*, $\delta_i = d^t C_i$, and $\delta_j = d^t C_j$), the above equation can be rewritten as

$$\delta_i - \delta_j < \frac{\mu_i - \mu_j}{2}. \tag{23}$$

That is, the document d will be moved to the S_j cluster as long as the difference between the average similarities of d to the documents of each cluster ($\delta_i - \delta_j$) is less than half of the difference between the average similarity among the documents in S_i and S_j ($(\mu_i - \mu_j)/2$).

On the other hand, the $\mathcal{I}_2$ criterion function will move d from S_i to S_j if and only if

$$\|D_i + d\| + \|D_j\| < \|D_i\| + \|D_j + d\|.$$

In a similar fashion with $\mathcal{I}_1$'s condition, the above equation can be rewritten as:

$$\|D_i + d\| - \|D_i\| < \|D_j + d\| - \|D_j\|$$

$$\sqrt{{D_i}^t D_i + 1 + 2d^t D_i} - \sqrt{{D_i}^t D_i} < \sqrt{{D_j}^t D_j + 1 + 2d^t D_j} - \sqrt{{D_j}^t D_j}. \tag{24}$$

Now, for sufficiently large clusters, we have that ${D_i}^t D_i + 2d^t D_i >> 1$, and thus

$${D_i}^t D_i + 1 + 2d^t D_i \approx {D_i}^t D_i + 2d^t D_i. \tag{25}$$

Furthermore, the following holds

$$\left(\sqrt{{D_i}^t D_i} + \frac{d^t D_i}{\sqrt{{D_i}^t D_i}}\right)^2 = {D_i}^t D_i + \frac{(d^t D_i)^2}{{D_i}^t D_i} + 2d^t D_i \approx {D_i}^t D_i + 2d^t D_i, \tag{26}$$

as long as

$$\frac{(d^t D_i)^2}{{D_i}^t D_i} = \frac{\delta_i^2}{\mu_i} = o(1),$$

that is, it is not significantly larger than one. This condition is fairly mild as it essentially requires that μ_i is sufficiently large relative to δ_i^2, which is always true for sets of documents that form clusters.

Now, using Equations 25 and 26 for both clusters, Equation 24 can be rewritten as

$$\begin{aligned}
\sqrt{\left(\sqrt{{D_i}^t D_i} + \frac{d^t D_i}{\sqrt{{D_i}^t D_i}}\right)^2} - \sqrt{{D_i}^t D_i} &< \sqrt{\left(\sqrt{{D_j}^t D_j} + \frac{d^t D_j}{\sqrt{{D_j}^t D_j}}\right)^2} - \sqrt{{D_j}^t D_j} \\
\frac{d^t D_i}{\sqrt{{D_i}^t D_i}} &< \frac{d^t D_j}{\sqrt{{D_j}^t D_j}}.
\end{aligned}$$

Finally, using the μ_i, μ_j, and δ_i, δ_j notation, from the above equation we get that $\mathcal{I}_2$ will move document d as long as

$$\frac{\delta_i}{\delta_j} < \sqrt{\frac{\mu_i}{\mu_j}}. \tag{27}$$