Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

ISSN - 0103-2569

Visual Mapping of Text Collections using an Approximation of Kolmogorov Complexity

Guilheme P. Telles
Rosane Minghim
Fernando Vieira Paulovich

Nº 262

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Jun./2005

Visual Mapping of Text Collections using an Approximation of Kolmogorov Complexity

Guilheme P. Telles, Rosane Minghim, and Fernando Vieira Paulovich

Instituto de Ciências Matemáticas e de Computação
CP 668, São Carlos 13560-970, São Paulo, Brazil
{gpt,rminghim,paulovic}@icmc.usp.br

Abstract. The generation of content-based text maps is an important issue to support exploration of information and to help find relevant reading material in increasingly complex document databases. Most techniques that help relate or visualize texts rely on a vector representation that is, at its best, ad-hoc as to its parameterization. This paper presents a novel approach capable of generating a map of documents without the painstaking pre-processing steps, by comparing text against text through an approximation of the Kolmogorov complexity. The similarity measure taken from that analysis is then used to map data in 2D by applying fast multidimensional projection techniques (instead of dimensionality reduction or random initial point placement). The resulting maps show a high degree of content separation and good grouping of similar documents. The approach can be used to map text collections in a variety of applications and the map can be interacted with to further explore text groups. By avoiding vector representation our technique decreases the bias characteristic of that approach and the need for user knowledge of the process. The approach also lends itself to incremental processing for reduction of computational costs.

1 Introduction

In a set up of document collections, such as text databases or Internet search results, it is not easy to locate reading material even with an effective ranking system. In the applications targeted here the goal of a user's analysis is to locate relevant documents to examine or study amongst a considerable number of recovered texts. We approach that by studying effective mapping strategies that are capable of coding relationships amongst documents geometrically and visually. An interactive map based on document content can be explored to locate a text relevant to a query or to another target text and to find groups of similar or related documents. This is usually done after a pre-filtering process that has narrowed the collection down to a range of few hundred to a couple of thousand texts. Typical applications are research, education and training.

Most text mapping strategies are based on clustering or dimensional reduction that rely on text vector space representations whereby, after a considerably lengthy pre-processing step, texts in a collection are represented as a vector of

many dimensions. Each dimension is a relevant term in the text set. One of the problems with this largely used vector approach is that the dimension of the final data set can reach the thousands easily. It is a well know fact that, as the dimension gets larger, the ability to properly infer vector distances gets impaired, prompting the need for some sort of attribute selection.

As far as processing speed is concerned, the transformation of that representation into a map of some sort may involve either dimensional reduction or clustering techniques, which can be rather slow themselves, or on faster projections, without significant quality loss ([22]). The fact remains, though, that the pre-processing step involved in all of them include various procedures, such as stopwords elimination and stemming, that can be affected by various parameter settings. Other adjustments for feature selection and frequency analysis are often necessary. These adjustments have large influence on the outcome, sometimes prompting more than a few iterations before the result is satisfactory, and making the process too sensitive to change in the subject target texts. In most real cases, therefore, it is not possible to tell in advance what the right tuning for the pre-processing is.

This paper proposes a novel approach to text mapping based on direct comparison between texts contents without the need for any preprocessing, by overriding the vector representation altogether. We calculate an approximation of Kolmogorov complexity ([19]) as a similarity measure between texts. That is used as a distance value to generate multi-dimensional projections into 2D space by means of a fast approach [30]. The resulting maps can be interacted with in a form that allows further exploration.

The following section offers a review of relevant text mapping literature an the role of projections in its context. Section 3 describes the multi-dimensional projection procedure employed here step by step. A summary of the theory and implementation of the Kolmogorov 'distance' is given in Section 4. Results and comparison to other maps are given in Section 5, which is followed by analysis, conclusion and further work discussion.

2 Previous Work

Due to the complexity and variety of the information and scenarios involved in text examination, alternative means of mappings text sets must be sought. Here we review the works in the literature that deal with this problem that, in our view, cover the main issues relating to visual mapping techniques for documents.

A number of different techniques for visualization of textual results from Web and other searches have being deployed ([1], [17], [29], [2], [7]). While these techniques are capable of displaying large text bodies, they tend to make location of relevant reading material more troublesome. Our focus in this work is to provide complementary tools to support mapping of documents in a way that helps locate neighboring similarities between texts and groups of texts. So we assume a pre-filtering task that reduces the universe of targeted documents to a few

hundreds (maybe thousands) of texts in a few areas of interest (not necessarily pre-determined).

Many techniques for text visualization exist that search for a representation of the content of an individual text (e.g. [21], [27]), of text collections (e.g. [17], [3], [31]), or of themes approached in texts (e.g. [13], [33], [34]) in order to meet the above mentioned targets.

Usually text processing tasks employ the vector space model [28] whereby texts are represented as points in a vector space. In this representation each text is a vector with dimensions represented by terms (n-grams). The vector coordinates are the weights of the terms based on their frequency. Typically, dimensions reach the thousands even for small to medium databases. Transformation of a text collection into a vector space is preceded by elimination of non-influential words (such as stopwords), reduction of words to their radicals (stemming), and frequency counting of some sort (various exist). The initial representation is followed by reduction in space dimensions, typically involving cutting off words that are too frequent or too rare in that particular collection, and clustering dimensions to generate new 'combined' attributes, in an attempt to overcome the dimensionality curse.

The most common way to extract structure from a text collection is by applying some sort of dimensional reduction technique over the resulting vector representation. This is the case of systems based on Multi-dimensional Scaling (MDS), Principal Component Analysis (PCA) or Latent Semantic Analysis (LSA), that work with statistical measures for subspace reduction, and Self-Organizing Maps (SOM), that employ neural computation ([31], [33], [3], [16], [34]). Those techniques can be used to plot the original data in bidimensional (2D) space, when dimension is reduced to 2.

Although dimensionality reduction is a natural processing trend for texts, these types of techniques have high computational costs and low adaptability to incremental processing. Multidimensional reduction techniques also cause other difficulties, such as [14]: high information loss when applied directly to two dimensions (for display); reduction in input dimensions do not seem to affect greatly the outcome; and there is an inherent discretization problem associated with techniques such as SOM, by which individual documents in groups are not distinguishable. For the target of this work, dimension reduction poses and additional problem: when used to display the results in 2D, the mappings to subspaces may define groups of 'similar documents', but locally it is not possible to relate neighboring texts. In a previous work [20] LSA has been successfully applied for the generation of document maps with high local content relationships, but the high computational cost remains a problem as well as the handling of vector transformations.

Another recurring strategy for dealing with the organization of information from a text collection is document clustering ([5], [26]), many times employed in combination with dimensional reduction and SOM ([15], [31], [17]). They provide a way of relating documents with varying success rates. When clustering techniques are applied, here too the intracluster relations are not given as a

result. However, they are very useful to provide general overviews of large collections, although they usually have to be interpreted by users with certain level of expertise.

Point placement strategies and force-based point placement improvement have been used before to generate document displays [7, 12]. Although still based on vector representation, they can avoid partly or completely the extensive calculations needed in dimensional reduction techniques by starting with a semi-random point placement and re-adjusting their position based on attraction by similarity.

There are approaches that completely avoid the problem of high dimensionality by simply ordering the most used terms in the text and employing the first N terms [27]. These strategies work well for single text representation and for association of a limited number of texts, and even for some degree of clustering. However, it also lacks a way of clearly relating different documents and displaying levels of similarity. Other approaches (such as the one by Carey and others [5]) combine a number of different strategies to allow various views of the same document set, potentially improving focusing and analysis tasks.

Final maps resulting from the techniques mentioned above are meant to analyze a number of properties of documents, including similarity, co-citation, term co-occurrence and various others. We refer to the work of Katy Borner and others [4] for a detailed description of the available techniques for text mapping and its applications, systems and challenges. A few systems are being developed dedicated to viewing maps from multi-dimensional data and some of them are particularly dedicated to text collections. One recently published system [12] adds representational power to the conventional ways of plotting text as points in 2D by separating their contents in thematic areas and handling levels of interaction by hierarchical organization.

In general the methods discussed above lack the ability to determine levels of associations between texts contents. Others are computationally expensive. Faster mapping approaches with the ability of associating texts by similarity have been put forward [22]. Their gain in processing time is attained by using projection techniques, which are faster compared to dimension reduction and also provide an initial point placement prone to speed up force-based improvement schemes. But those too are based on the vector representations.

Text vector representations, although very useful and largely employed, many times are cause for concern. They tend to impose bias and be difficult to tune. Depending on the choice of pre-processing parametrizations (such as Luhn's cut, vocabulary, types of stemming, types of frequency count, type of feature clustering or selection), the outcome of the analysis and displays of text collections can be highly affected. That also makes it difficult for lay users to employ the representation inside its usual context without knowledge of the processing, pre-processing and visualization techniques.

Another issue that impairs general use of vector-based techniques is its adaptability to incremental processing. Adding new texts to a previously existing col-

lection mostly implies in rebuilding the visualization (including pre-processing) almost from start, propagating possible limitations to every map formation.

The technique presented here is based on projection techniques that have been proven useful to group and separate data when the similarity measure is sufficiently powerful [30, 22]. To be able to function, these projections need only the distance between data points (in our case, texts), and not the original data themselves. The contribution of our technique is in producing such a distance between texts through a similarity measure that completely avoids the conventional vector representation. This way it eliminates the need for the ad-hoc pre-processing steps necessary to build that representation and avoids the problem of treating high dimensionality, a real trouble for texts. This measure is based on an approximation of the Kolmogorov complexity [19] and is calculated by comparing text against text. This can be done using the texts in their original form or, in some cases, in part of their original form. The results have reproduced separation and grouping advantages of previous methods and is not sensitive to tuning or to text dimensionality. These features also make the method easier to use. Opposed to most vector based mappings, it is naturally adaptable to incremental processing, with affordable storage overhead.

The visual representation adopted here is the landscape-type of display, which is very useful due to its ability to reveal information without resorting to highly attentive perceptual processes. Additionally, surfaces are highly interactive and familiar to most users. Landscape plots have been the choice of many useful presentations of texts before ([31], [33], [6], [8]). This feature combined with the absence of pre-processing allows interpretation of mappings even by users with little expertise in the field. The surface representation of our technique is enriched by mapping further significant information to visual attributes (such as lines, colors and height) and aural attributes (such as pitch and timbre). The final map can be explored to the advantage of users interested in having an overview of a set of texts, locating important texts in corpora, or finding useful associations between texts, thus selecting material to read or study.

3 Projection techniques for text visualization

Methods of data projection into lower dimensions have the advantage over dimensionality reduction that they are much faster and, depending on the type of projection, good for incremental processing. A previous work [30] has shown the advantages of projection techniques based on distance metrics to obtain useful views of multi-dimensional data sets. When the distance calculation captures significant data set features, they are capable of separating the data collection into groups and result in good association between neighboring individuals. Additionally, those techniques lend themselves to landscape plots. Their application for mapping texts represented in the vector space analogy was tested before with satisfactory results [22]. In this work the same types of projections are used to map points into a plane except the distance between texts is now calculated without generating the vector representation (see Section 4).

Different from other techniques that can be used to map data into 2D or 3D, such as dimensional reduction, clustering, or point placement strategies that start from a random or semi-random 2D display, the goal of distance-based projection techniques – e.g. Fastmap [11] and NNP [30] – is to place a set of points defined in multi-dimensional space in another space such that the relative distances between points are preserved as much as possible. The degree to which that distance cannot be preserved is called the error of the projection. For projections into a bidimensional plane, this problem can be stated as:

Let X be a set of points in $\mathbb{R}^n$ and $d : \mathbb{R}^n \rightarrow \mathbb{R}$ be a criterion of proximity between points in $\mathbb{R}^n$. Find a set of points P in $\mathbb{R}^2$ such that if $\alpha : X \rightarrow P$ is a bijective relation and $d_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a proximity criterion in $\mathbb{R}^2$, then $|d(x_i, x_j) - d_2(\alpha(x_i), \alpha(x_j))|$ is minimum for every pair of points $x_i, x_j \in X$.

The set P is called a projection. In this type of projection, it is of great importance the definition of a proper proximity criterion, calculated in our case from the Kolmogorov complexity estimation.

We have used two projection algorithms in our experiments, with similar results: Fastmap and NNP. The first realizes hyperplane projection and the second performs geometric placement in neighborhoods. Each projection is improved by a projection improvement scheme called Force [30], that enhances those projections by recovering part of the information lost during the mapping process, in a similar procedure as that adopted before by other researchers, such as Chalmers [7]. What the Force scheme does is iteratively approach points projected too far and repel points that were projected closer than they should have. It does that by a fraction of the ideal distance at each iteration. The computational cost of such improvement is minimum due to the initial point positioning by projection. The Force scheme also provides an overall measurement of the projection error (see [30]).

The resulting points, now in 2D space, are connected by triangulation, thus to produce a surface, allowing interaction for exploration of text content. On top of that surface, text properties can be mapped to display attributes to support exploration. In our case, color, height and sound were employed to complete the map. Each attribute (color, height or sound) can map text clustering, or text category (in case it is pre-determined), as well as additional information such as year of publication, number of citations, rank, and so on.

The complete set of steps taken to build a map based on projection from Kolmogorov distance (or any other distance measure for that matter) is the following:

1. calculation of a triangular distance matrix comparing all texts and judging their similarity;
2. projection the points (texts) onto bidimensional space using a fast algorithm, followed by an improvement strategy;
3. triangulation of the data.

Figure 1 illustrates the complete process, and shows one possible resulting map. On top of the map, color (as well as height) were used to show clustering of the projected texts.

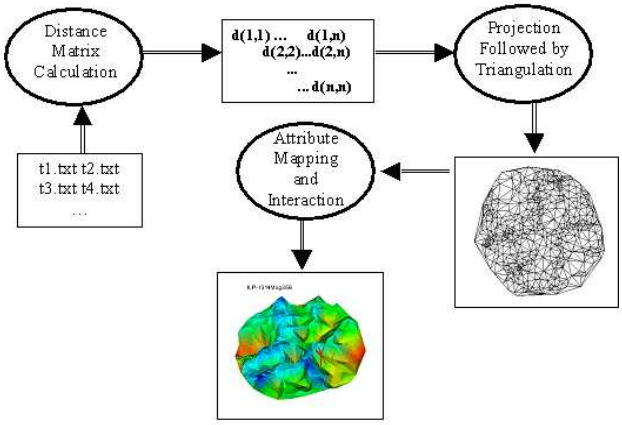


Fig. 1. The whole mapping process. No pre-processing actually needed.

The calculation of the distance between texts is presented in the next section.

4 Kolmogorov Complexity as a means to define distance between texts

Intuitively, the Kolmogorov complexity is a measure of the amount of information that a message contains. It can also be seen as a measure of randomness of a string or as the length of a string that results after perfect compression. An extensive treatment on the Kolmogorov complexity appears in Li and Vitányi's book [19]. A text can be seen as a string, so the discussion that follows applies to texts directly.

Formally, the Kolmogorov complexity of a string y, $K(y)$, is the size of the smallest algorithm that outputs y. Any formal notion of algorithm can be applied, such as Turing machines. The conditional Kolmogorov complexity of a string y given a string x, $K(y|x)$ is the size of the smallest algorithm that outputs y when x is given as input. Intuitively the conditional Kolmogorov complexity is the amount of information in y that is not known by x. The Kolmogorov complexity considered here is the prefix version, where algorithms are considered to be prefix-free, that is, no algorithm is a proper prefix of another.

The Kolmogorov complexity is not computable but it can be approximated using compression. Let xy denote the concatenation of strings x and y. Li and coworkers [18] defined the normalized distance between x and y

$$d(x, y) = 1 - \frac{K(x) - K(x|y)}{K(xy)} \tag{1}$$

and showed that $d(x, y)$ is a metric up to logarithmic additive terms.

Let $C(x)$ denote the length of the compressed version of a string x. Cilibrasi and Vitányi have shown that the normalized compression distance

$$NCD(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{max\{C(x), C(y)\}} \quad (2)$$

is a quasi-universal metric when the compressor in use is normal.

Kolmogorov complexity have been applied theoretical computer science [25], clustering [10], plagiarism detection [9], phylogeny [18] and others [10].

In this work we used the LZW compression algorithm [32] to evaluate an approximation of $d(x,y)$ denoted $d'(x,y)$:

$$d'(x,y) = 1 - \frac{C(y) - C(y|x)}{C(xy)} \quad (3)$$

We have evaluated the terms $C(y)$, $C(y|x)$ and $C(xy)$ as follows. The term $C(y)$ is the number of table positions that LZW would output while compressing y minus the number of table positions that LZW would output when the input is a string of $|y|$ equal symbols plus one. The subtraction is a tentative to overcome the limitations imposed by the very nature of LZW. The term $C(xy)$ is evaluated in the same way. The term $C(y|x)$ is the number of table positions that LZW would output while compressing y, without counting the positions that belong to the compression table for x. That is, a table position is counted only if it does not belong to the table constructed during the compression of x. Our algorithm was implemented in Perl, using a hash for the table.

In our tests, we have also used the CompLearn [23] package together with gzip [24] (chosen for speed) to evaluate NCD. We have noticed slightly better results using $d'(x,y)$ (see Section 5), although with poorer performance.

Generating a distance table for n texts $t_1, t_2, \ldots, t_n$ with lengths $l_1, l_2, \ldots, l_n$ requires computing $C(t_i)$ for every text, and $C(t_i|t_j)$ and $C(t_i t_j)$ for every pair such that $i \neq j$. Then the cost of the distance table construction is $O(s^2)$ on the average, where $s = \sum_{i=1}^{n} l_i$ and with average cost $O(1)$ for a hash operation.

Adding a new text t_{n+1} to the set does not require recomputing the whole matrix. If we store the values of $C(t_i)$ and the LZW table produced for every t_i, it is enough to calculate $C(t_i|t_{n+1})$ and $C(t_i t_{n+1})$ for $1 \leq i \leq n$, at cost $O(s + l_{i+1})$ on the average. The length of LZW table for a text of length l is $O(l)$, so storing them requires an affordable amount of space.

5 Results

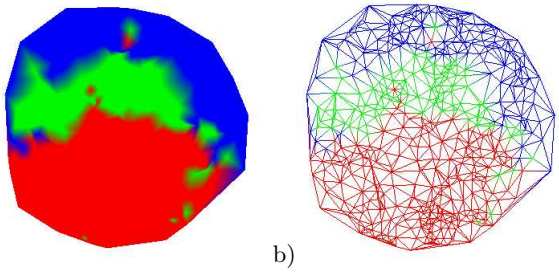
In the remaining of this text we shall refer to the similarity calculation based on NCD (Equation 2) as NCD and we shall refer to our similarity calculation based on compression by LZW (Equation 3) as k-lzw. In general, we refer to distance calculation based on Kolmogorov complexity approximation Kolmogorov distances or k-distances.

One of the goals of our maps is being able to distribute the files onto a surface automatically, allowing proximity of closely related contents, association between documents and groups of documents, and display of additional information related to them on the same map. By doing projection based on distance

metrics it has been possible to obtain results that match those goals, provided the distance measure is capable of coding well content relationships.

To evaluate the ability of a particular distance measure to map corpus content, we have used text collections previously classified as belonging to general areas of knowledge. This classification is based purely on the source of the papers, and therefore some degree of overlapping is expected due to the presence of similar concepts or techniques cross-areas.

Pseudo-classes on the maps were colored to reflect mapping results visually, showing how those documents of the same class were distributed over the surface. Figure 2 shows the result of one final map from k-lzw for a corpus made out of papers from three basic areas: Inductive Logic Programming (ILP), Case-based Reasoning (CBR) and Information Retrieval (IR). It shows that this mapping is capable of keeping most of the documents of the same class in the same region of the map.



a) b)

Fig. 2. Mapping of scientific documents related to three different primary sources (CBR is red. ILP is green. IR is blue.).a) flat surface b) wireframe model

Processing times for calculation of k-distances for a whole data set are quite high. The set in Figure 2, comprising 574 files took 2h30min to process from scratch in a Pentium IV processor of 3GHz. Opposite to conventional techniques based on frequency count, though, this type of processing is incremental and adding new documents does not require recalculation of previously obtained results. Only the new distances must be processed as databases increase in size, as mentioned in Section 4.

Table 1 gives details of the documents used to process the remaining tests. The first scientific papers data set (corpus1) included title, authors, abstract and references from a number of texts. CBR and ILP subsets were taken from journals on those subjects. The IR and SON (sonification) subsets were articles obtained as a result of Internet searches pre-filtered to comply to those pseudo-classes. Those were all collected by members of our team. The corpus2 set was recovered from an Internet repository and comprehends files in the ISI format on the subjects of Bibliographic Coupling (BC), Cocitation Analysis (SC), Milgrams

(MG) and Information Visualization (IV)[1]. The remaining sets are messages from news discussion groups recovered from an internet repository[2].

Table 1. Datasets used in the tests

Set	Areas	General Content	Files
corpus1	CBR+IR+ILP+SON	Scientific Documents	675
corpus2	SC+BC+MG+IV	ISI Files	1624
message1	atheism+graphics	discussion group messages	200
message2	atheism+graphics +baseball	discussion groups messages	300
message3	seven varied subjects	discussion group messages	700
message4	ten varied subjects	discussion group messages	1000

Previous results have shown that distance based projections of text collections based on vector representation with conventional distance metrics (such as cosine) can provide a point placement with good separation between general subject areas as well as good grouping of similar documents. This process, that entails vector representation followed by k-means attribute combination, followed by projection with improvement scheme, has been called IDMAP (Interactive Document Map)[22]. We call the mapping realized here (projection with improvement scheme based on k-distances) Kolmomap. We compare these two approaches for visual result.

Figures 3 and 4 show the visual results of some of the data sets in both IDMAP and Kolmomap. It can be seen that the result is comparable to that obtained with IDMAP in terms of region placement and sub-grouping. Separation is better with IDMAP in some cases, but it tends to jam similar documents into pockets, making it more difficult to interpret neighboring relationships in dense regions.

In 'more generic texts' the intrinsic content relationship is a lot less obvious than in academic or scientific papers. Figure 4 shows the generation of Kolmomaps and IDMAPs from messages in discussion groups (data sets message1 and message2 in Table 1). Pseudo-class in this case is the theme of the discussion

[1] ella.slis.indiana.edu/~ katy/outgoing/hitcite/{bc,sc,mb,iv}.txt

[2] Hettich, S. and Bay, S. D. (1999). The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science

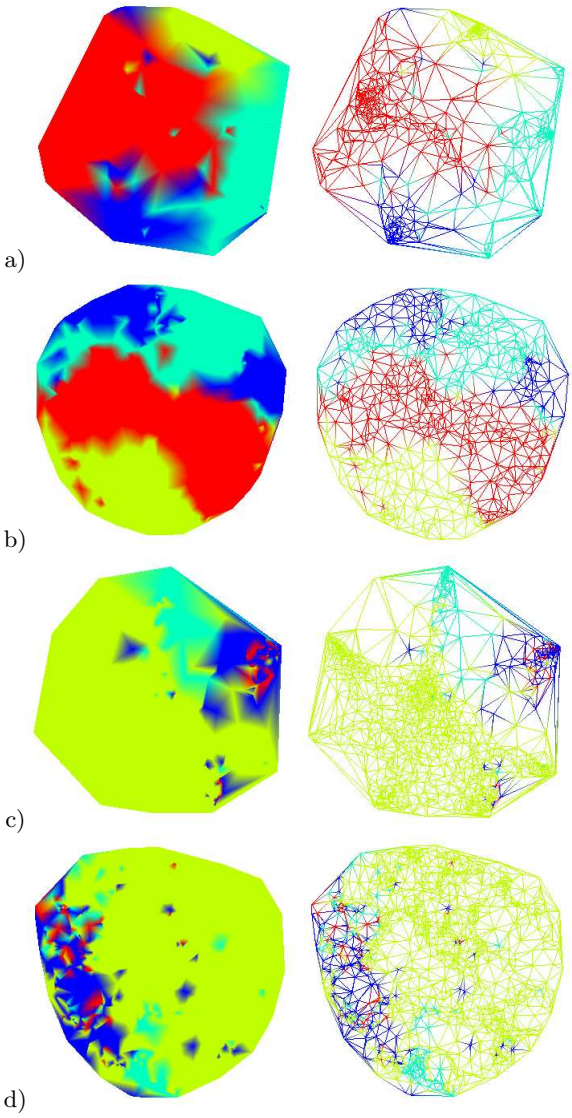


Fig. 3. IDMAP and Kolmomap of the two bibliographic data sets. a) IDMAP results from corpus1 set. b) Kolmomap results from corpus1 set. c) IDMAP results from corpus2 set. d) Kolmomap results from corpus2 set. In dataset corpus2, one part of the corpus - in yellow - 1236 files - is a lot larger than any of the other three. See Table 1.

groups. The maps show that Kolmomaps are capable of distinguishing subjects in that context much better than IDMAPs.

Once the range of themes starts to broaden and specialize, Kolmomaps tend to mix messages coming from distinct sub-groups. This can be understood in the light of the fact that discourse tends to be full of common expressions and terms, regardless of the subject under discussion. Figure 5 shows that fact.

Observing the pictures in Figure 5, it can be noted that Kolmomaps can still determine various 'pockets' of similar documents and also that the mixture is not uniform, that is, at least in the body with 700 messages, there is overall mixture of 'reds' and 'blues' as well as 'greens' and 'yellows'. Blues and reds are sci.space and alt.religion, and greens and yellows are various 'comp' subjects and 'forsale' subjects. Therefore even with lesser distinction between pseudo-classes than the 3-theme case, there is an underlying pattern reflected by the Kolmogorov distances. IDMAP, as might be expected from the previous message tests, was not capable of any significant separation between subjects.

Detailed exploration of various maps have shown a close relationship between proximity in the map and the expected proximity of document content with few exceptions.

In order to test the capability of content based point placement, two tests were performed. In one of them 5 'intruders' (documents not previously processed in the set) were added to the map. They belonged to the general class of sonification, but were not in the initial set. All five had at least two common authors and represented and evolution of the same sonification system over the years. Additionally to those, two other papers were added. Again two of the authors at least were repeated, but the main subject of the paper was not sonification. One of them mentioned sonification incidentally (paper A), the other didn't (paper B). Figure 6 shows the map for that test. In that case the Kolmomaps projected those related papers together (Figures 6a and 6b). The papers that were not primarily on sonification were still mapped within the general sonification pseudo-class, indicating the importance of author's names (and most likely some self-reference too) in the context. However, it can be seen that, for the k-lzw implementation, the fact that the content itself was diverse from that of the neighbors, made paper B push them away and form an isle within the group. NCD also provided approximation between most of the related papers, but provided less distinction for that text with content diverse from sonification. IDMAP did not perform as well (Figure 6a), placing the files within the scope of the general sonification groups but making less distinction of those without direct relation with sonification.

The second test meant to compare the two different approximations of Kolmogorov distances (k-lzw and NCD) further. In this test, we added intruders (also 5) in the messages test set message1 that were not related to any of the previous two newsgroups. The two groups were comp.graphics and alt.atheism and the intruders (in blue) belonged to the theme of rec.sports.baseball. Figure 7 presents the results of that placement. It shows that for the k-lzw map the new points are pushed away from the previously existing sets of messages (either to

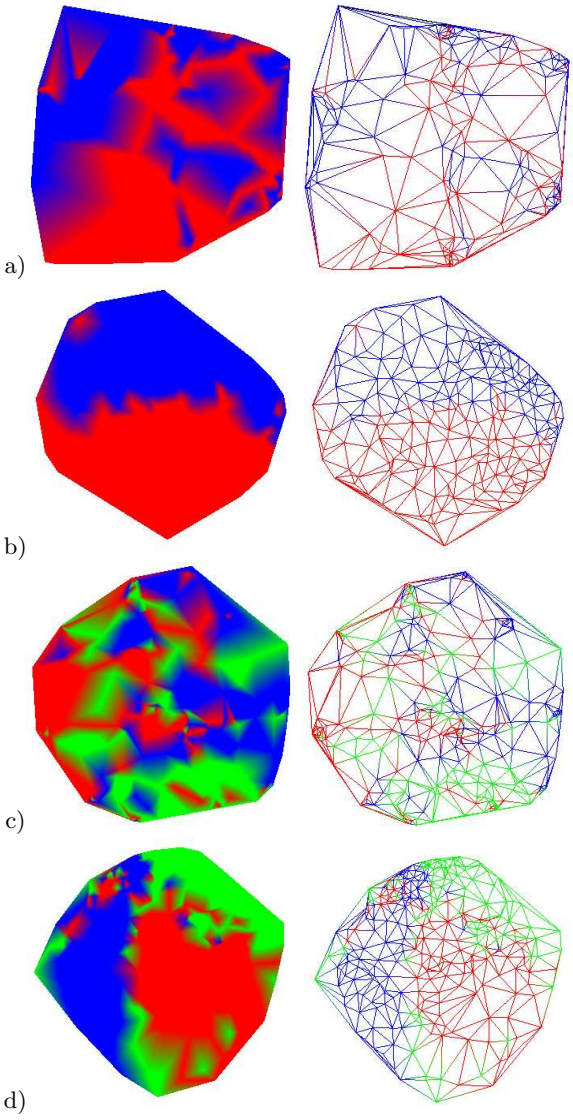


Fig. 4. IDMAP and Kolmomap of the newsgroups data sets. a) IDMAPs of message1. b) Kolmomaps of message1. c) IDMAPs of message2. d) Kolmomaps of message2.

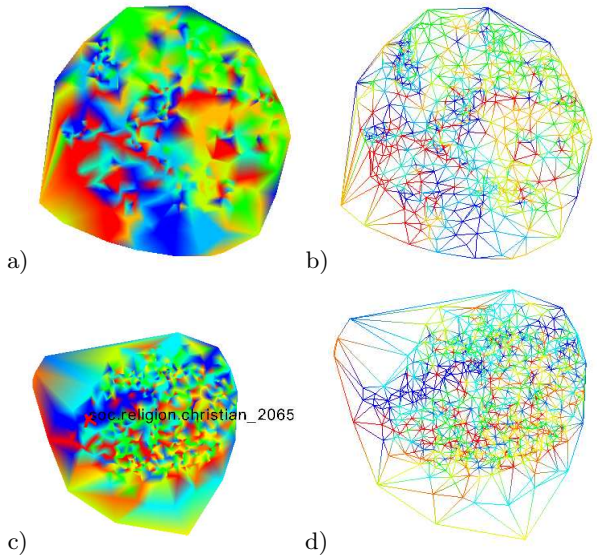


Fig. 5. Kolmomaps of message3 and message4 data sets.

the border of the map, or, in one case, between the two groups). NCD did not perform as well, mixing half of the intruders within either one of the previously formed groups.

Projection errors according to calculations published in a previous work [30] are presented in Table 2. Those values support the evidence that the mixture in maps of larger human communication files (news groups with larger number of subjects and messages) is caused by message content instead of the projection itself.

Table 2. Approximate Projection Errors

Set	k-lzw	NCD	IDMAP
corpus 1	0.25	0.33	0.2
corpus 2	0.25	0.32	0.17
message1	0.17	0.25	0.21
message2	0.17	0.25	0.21
message3	0.23	0.3	0.18
message4	0.17	0.25	0.17

Assigning attributes to the map components (vertices, for instance) can show one or more additional degrees of information on top of the landscape map. Various attributes can be visually mapped to color, height, isocurves, etc., allowing

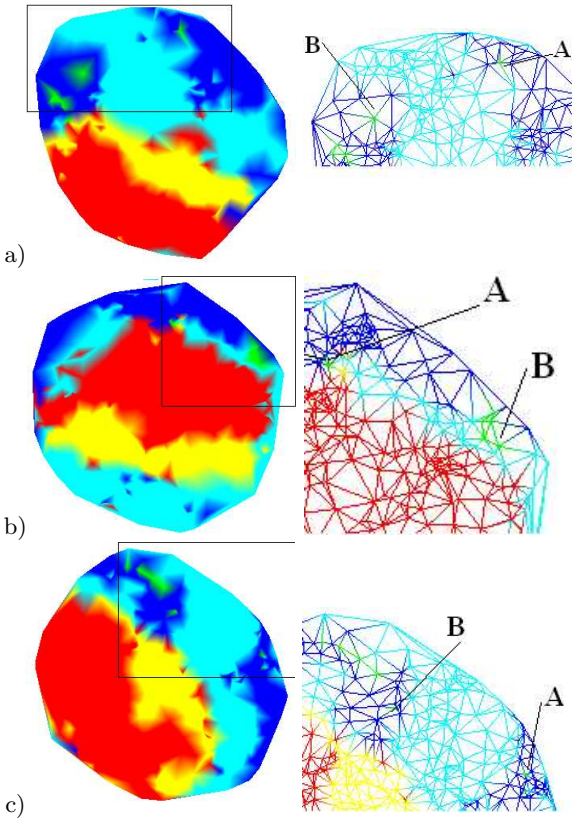


Fig. 6. Maps with highly correlated 'intruders' (in green)a) Kolmomap. b) maps based on NCD estimation. c) IDMAPs.

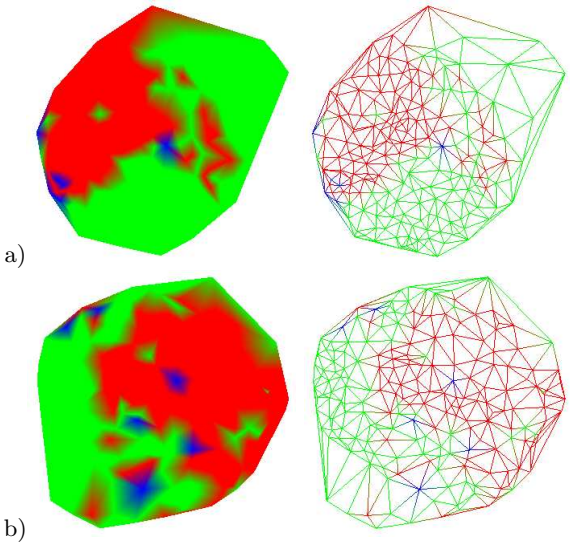


Fig. 7. News-maps with uncorrelated 'intruders' (in blue).

further analysis of relationships between the documents represented in the map. Previous maps in this text have shown pseudo-class by using color. Figure 8a shows the same data mapped redundantly to height for the corpus1 dataset. In that case, different classes are 'plateau layers' in the map and misplaced (or pseudo-misplaced as the case may be) points can be easily located as peaks or depressions in the topography of the maps.

To highlight proximity relationships further, one attribute mapping found useful was to perform a hierarchical clustering of the projected data. A hierarchical clustering can group points closer together in various levels. By doing that, and then showing the depth of the cluster on the maps, it is possible to have visual aid to locate pockets of closely related documents. Figure 8b illustrates the corpus1 data set after submission to single-link hierarchical clustering. Both color and height are mapping the depth of the document in the clustering tree. The dark blue (and highest) areas show where the points are closer together; the following levels or grouping are lighter blue, green, orange, yellow, and red (lowest), the latter showing the points (documents) that are most isolated from their 'neighbors'.

The former mappings can be combined to produce composed visualizations of multiple attributes. Figure 8c shows clustering mapped to height and class mapped to color for the same dataset, and Figure 8d shows the mapping inverse to that (height is class, color is cluster).

The map can be interacted with for exploration. In our case, a tool for exploration of that type of representation, called Spider Cursor, is under development. The current version allows walking over the surface using a cursor that highlights the edges to neighbors (see Figure 8e). It also allows the user to choose how nu-

meric attributes will be mapped to visual attributes such as color, height, level curves, glyphs and sound. Paths can be marked, selected and cut to extract parts of the map (Figure 8f). One attribute, possibly literal, is shown on a window on the map. In our examples, they were the labels of the files containing the document.

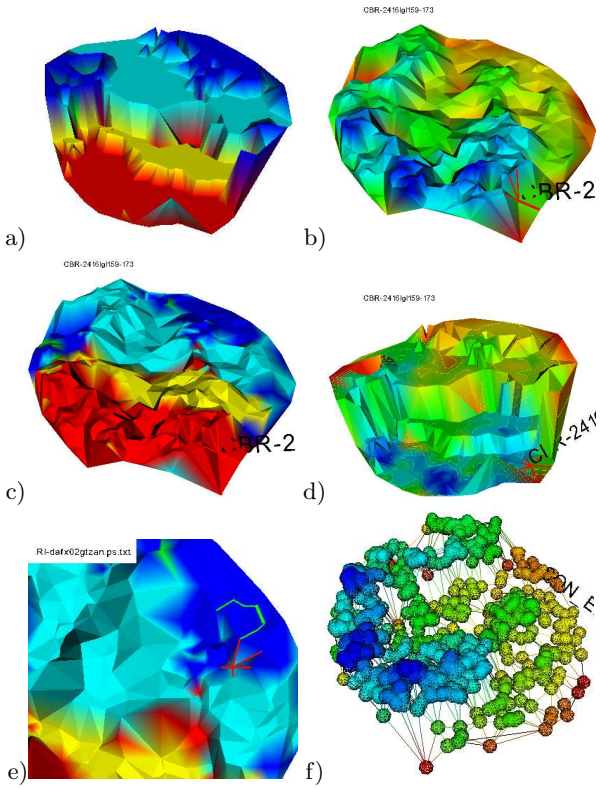


Fig. 8. Various possibilities of attribute mapping and interaction with the corpus1 map.

In visual terms, Kolmomaps were very successful both in distributing documents in larger subject areas and in grouping files with similar contents. The fact that it does not require term analysis makes it a good option to process (and organize) corpora of documents to be analyzed in an interactive session.

The process of forming the distance matrix for a whole corpus is costly, particularly if processed from scratch. Table 3 shows the times to process each data set in full at once, with no intermediate storage (that is, the matrix was not built incrementally). The projection itself, including hierarchical clustering, is done in a matter of few seconds.

Table 3. Times for distance calculation (h - hours, m - minutes)

Processor Pentium IV - 3GHz		
Set	k-lzw	NCD
corpus 2	11.3 h	5.5h
message3	2h	17 m
message4	4.2h	39 m

6 Conclusions

The technique presented here of similarity calculation via Kolmogorov complexity taken from raw texts has proven useful in conjunction with distance-based projections to build maps of texts. Those distances have shown to separate general content areas and to generate proximity between similar texts, facts reflected by the projection based on the distances.

Kolmomaps have shown to work well for texts in a variety of contexts. It is a good step towards helping text organization by content in groups of documents in a context, where selection of pre-filtered reading material is necessary (such as research, education, training in technology etc.).

Kolmomaps have high computational cost in it distance calculation stage. However, some facts about the time to generate maps of documents based on distances by Kolmogorov complexity estimation should be phrased:

- No pre-processing is needed. The approach allows immediate application to text documents without the need to realize (or understand) vector transformation. No lengthy editions are necessary either. That in itself saves time in the whole analysis process.
- Opposed to vector representation schemes, this process is incremental. For data sets that grow – and they usually do –, it is possible to store intermediate distances and symbol tables, speeding up the calculation for a new text to be added.
- The projections are actually very fast, in the order of a few seconds (including improvement, hierarchical clustering and display).
- Although NCD is faster, k-lzw estimation has resulted in more consistent map behavior. It was also noted that k-lzw time increase showed more regular behavior than NCD as the document bodies grew larger.

The maps using k-distances compared well with IDMAPs. Interpretation of the resulting maps is easy to learn. In a short time, users learn that being close means related content, and longer edges mean that the distance is larger than to those with smaller edges.

Further work is planned in analising a number of text corpora, in data structures for storing and recovering map results, and in adding extra semantic layers on top of this map to reflect other dimensions, such as cocitation and relevance.

To our knowledge, the conditional Kolmogorov complexity was not evaluated our way before, so it remains to show that this measure is a (quasi-)metric, as the experimental results suggest.

Scalability to larger data sets was not an issue here. Rather, we have been researching methods to help distinguish important relationships in text data sets of a manageable size but still too large to have the user make sense of it on his/her own efficiently. However, we believe that with that the approach can be scalable a further level, by developing an incremental data organization approach to go with the mappings.

7 Acknowledgments

The authors want to acknowledge the support of FAPESP Brazilian financial agency. We acknowledge prof. Alneu de Andrade Lopes and his students for the CBR+ corpora. Additionally, the work of Lionis Watanabe and Renato Oliveira in tuning up the interaction tool is much appreciated.

References

1. O. Alonso and R. Baeza-Yates. Alternative implementation techniques for web text visualization. In *Proc. of the First Latin American Web Congress*, pages 202–203, Santiago, Chile, November 2003. IEEE Computer Society, IEEE Press.
2. R. Baeza-Yates. Visualizing large answers in text databases. In *Int. Workshop on Advanced User Interfaces (AVI'96)*, pages 101–107, Ubbio, Italy, 1996. ACM Press.
3. A. Booker, M. Condliff, M. Greaves, F.B. Holt, A.Kao, D.J. Pierce, S. Poteet, and Y.-J.J. Wu. Visualizing text data sets. *Computing in Science and Eng.*, 1(4):26–35, 1999.
4. K. Borner, C. Chen, and K. Boyack. Visualizing knowledge domains. *Annual Review of Informtion Science & Technology*, 37:1–51, 2003.
5. M. Carey, D.C. Heesch, and S.M. Ruger. A visualization tool for document searching and browsing. In *Proc. of Intl. Conf on Distributed Multimedia Systems*, 2003.
6. M. Chalmers. Using a landscape methaphor to represent a corpus of documents. In A.U. Frank and I. Campari, editors, *Proc. of COSIT '93*, volume 716 of *Lecture Notes in Computer Science*, pages 377–390. Springer, 1993.
7. M. Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *Information Visualization 1996*, pages 127–132, San Francisco - CA, USA, 1996. IEEE CS Press.
8. M. Chalmers and P. Chitson. Bead: Explorations in information visualization. In *Proc. of ACM SIGIR*, pages 330–337. ACM Press, 1992.
9. X. Chen, M. Li, B. Mckinnon, and A. Seker. A theory of uncheatable program plagiarism detection and its practical implementation. Technical report, UCSB, 2002.
10. R. Cilibrasi and P.M.B. Vitányi. Clustering by compression. *IEEE Trans. Information Theory*, 51(4):1546–1555, 2005.

11. C. Faloutsos and K. Lin. Fastmap: A fast algorithm for indexing, datamining and visualization of traditional and multimedia databases. In *Proc. of International Conference on Management of Data*, pages 163–174, San Jose-CA, USA, 1995. ACM Press: New York.
12. M. Granitzer, W. Kienreichand V. Sabol, K. Andrews, and W. Klieber. Evaluating a system for interactive exploration of large, hierarchically structured document repositories. In *Information Visualization 2004*, pages 127–132, Austing- TX, USA, 2004. IEEE CS Press.
13. S. Havre, E. Hetzler, P. Whitney, and L. Nowell. ThemeRiver: Visualizing thematic changes in large document collections. *IEEE Transactions On Visualization And Computer Graphics,*, 8(1):9–20, Jan-Mar 2002.
14. S. Huang, M. Ward, and E. Rundensteiner. Exploration of dimensionality reduction for text visualization. Technical Report TR - 03–14, Worcester Polytechnic Institute, Computer Science Department, 2003.
15. S. Iritano and M. Ruffolo. Managing the knowledge contained in electronic documents: a clustering method for text mining. In *12th International Workshop on Database and Expert Systems Applications (DEXA) Workshop*, pages 454–458. IEEE Computer Society Press, 2001.
16. S. Kaski, T. Honkela, K. Lagus, and T. Kohonen. WEBSOM - self-organizing maps of document collections. *Neurocomputing*, 1(1-3):110–117, 1998.
17. A. Leuski and J. Allan. Lighthouse: Showing the way to relevant information. In *InfoVIS*, pages 125–130. IEEE Computer Society Press, 2000.
18. M. Li, J. H. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang. An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, 17(2):149–154, 2001.
19. M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Verlag, 2nd edition, 1997.
20. A.A. Lopes, R. Minghim, and Vinicius Melo. Creating interactive document maps through dimensionality reduction and visualization techniques. Technical report, USP - Instituto de Ciências Matemáticas e de Computação, São Carlos-SP, Brazil, 2005.
21. N.E. Miller, P.C. Wong, M. Brewster, and H. Foote. Topic islands - a wavelet-based text visualization system. In *Proc. of the conference on Visualization '98*, pages 189–196, Research Triangle Park, North Carolina, United States, 1998. IEEE Computer Society, IEEE Computer Society Press.
22. R. Minghim, F.V. Paulovich, and A.A. Lopes. Fast content-based map generation for interactive exploration of document collections. In *Technical Report*. ICMC - University of São Paulo, 2005.
23. CompLearn Home Page. http://complearn.sourceforge.net/.
24. GZIP Home Page. www.gzip.org.
25. W.J. Paul, J.I. Seiferas, and J. Simon. An information-theoretic approach to time bounds for on-line computation. *J. Comput. Syst. Sci.*, 23(2):108–126, 1981.
26. M. Rasmussen and G. Karypis. gCLUTO - an interactive clustering, visualization, and analysis system. Technical Report CSE/UMN TR 04-021, Univ. of Minnesota, Dep. of Computer Science and Engineering, 2004.
27. R.M. Rohrer, D.S. Ebert, and J.L. Sibert. The shape of shakespeare: Visualizing text using implicit surfaces. In *Proc. the IEEE Symposium on Information Visualization*, pages 121–129. IEEE Press, 1998.
28. G. Salton. Developments in automatic text retrieval. *Science*, 253:974–980, 1991.

29. M.M. Sebrechts, J. Cugini, S.J. Laskowski, J. Vasilakis, and M.S. Miller. Visualization of search results: A comparative evaluation of text, 2d, and 3d interfaces. In *22nd ACM-SIGIR Conf. Research and Development in Information Retrieval*, pages 3–10. ACM Press, 1999.
30. E. Tejada, R. Minghim, and L.G. Nonato. On improved projection techniques to support visual exploration of multi-dimensional data sets. *Information Visualization Journal*, 2(4):218–231, 2003.
31. E. Weippl. Visualizing content based relations in texts. In *Proc. of the 2nd Australian conference on User interface*, pages 34–41, Queensland, Australia, 2001. IEEE Computer Society, IEEE Computer Society.
32. T.A. Welch. A technique for high-performance data compression. *IEEE Computer*, 17(6):8–19, 1984.
33. J.A. Wise. The ecological approach to text visualization. *Journal of the American Society for Information Science*, 50(13):1224–1233, November 1999.
34. J.A. Wise, J.J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, and V. Crow. Visualizing the non-visual: spatial analysis and interaction with information for text documents. In *Readings in information visualization: using vision to think*, pages 442–450, San Francisco, CA - USA, 1995. Morgan Kaufmann Publishers Inc.