Relationship-based Visualization of High-dimensional Data Clusters

Alexander Strehl
strehl@ece.utexas.edu

Joydeep Ghosh
ghosh@ece.utexas.edu

Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas 78712-1084
USA

Abstract

In several real-life data mining applications, data resides in very high ($>$ 1000) dimensional space, where both clustering techniques developed for low dimensional spaces (k-means, BIRCH, CLARANS, CURE, DBScan etc) as well as visualization methods such as parallel coordinates or projective visualizations, are rendered ineffective. This paper proposes a relationship based approach to clustering that alleviates both problems, side-stepping the "curse of dimensionality" issue by working in a suitable similarity space instead of the original high-dimensional attribute space. The similarity measure used can be tailored to satisfy business criteria such as obtaining user clusters representing comparable amounts of revenue. The clustering algorithm is used to re-order the data points so that the resulting (rearranged) similarity matrix can be readily visualized in two dimensions, with clusters showing up as bands. While such visualization is not novel, the two key contributions of our method are: (i) it leads to clusters of (approximately) equal importance, and (ii) related clusters show up adjacent to one another, further facilitating the visualization of results. Both properties arise from the efficient and scalable top-down graph-partitioning approach used for clustering in similarity space. The visualization is very helpful for assessing and improving clustering. For example, actionable recommendations for splitting or merging of clusters can be easily derived, and it also guides the user towards the right number of clusters. Results are presented on a real retail industry data-set of several thousand customers and products, as well as on clustering of web document collections.

Keywords: Clustering, graph partitioning, high dimensional data, visualization, customer segmentation, text mining

1 Introduction

Knowledge discovery in databases often requires clustering the data into a number of distinct segments or groups in an effective and efficient manner. Good clusters show high similarity within a group and low similarity between any two different groups. Automatically generated web page clusters, for example, can provide a structure for organizing large bodies of text for efficient browsing and searching of the web. Grouping customers based on buying behavior provides useful marketing decision support knowledge; especially in e-business applications where electronically observed behavioral data is readily available. Customer clusters can be used to identify up- and cross-selling opportunities with existing customers. While clustering is a classical and well studied area, it turns out that both the applications described above, as well as some other data mining applications, pose some unique challenges that severely test traditional techniques for clustering and cluster visualization.

To take a specific example, a large market-basket database may involve millions of customers and several thousand product-lines. The customer's interaction history is typically characterized by a *vector space* model. For each product a customer could potentially buy, a feature (attribute) is recorded in the data. In the most simple case, a feature is a binary value that indicates if the corresponding product was purchased or not within a given period of time. To model the different importance of various products better, our clustering uses *non-negative real* features such as quantity and price of the goods purchased. Most customers only buy a small subset of products. Thus the corresponding feature vector describing such a customer is (i) High-dimensional (large number of products), and (ii) Sparse (most features are zero for most samples). Also the dataset typically has significant outliers, such as a few, big corporate customers that appear in an otherwise small retail customer data. Filtering these outliers may not be easy, nor desirable since they could be very important (e.g., major revenue contributors). In addition, features are often neither nominal, nor continuous, but have discrete positive ordinal attribute values, with a strongly non-Gaussian distribution.

A key insight, obtained recently by us and others, is that is advantageous to work in similarity space instead of the original (high-dimensional) vector space in such cases [11, 15, 23, 22]. In this paper we focus on the visualization benefits of working in similarity space, which accrues as a useful by-product, specially when a top-down graph-partitioning method is used for clustering. This is a key aspect of the proposed technique since it helps in the design process and is also critical for acceptance of results by a non-technical person. After summarizing related work in the next section, we describe the domain-specific transformation into similarity space in Section 3, and show how a simple but effective visualization technique can be based on this in Section 4. Section 5 summarizes the specific clustering technique (OPOSSUM) based a multi-level implementation of the KL graph partitioning algorithm [17, 14], that we employed. The resulting clusters are visualized in Section 6.

2 Related Work

Clustering has been widely studied in several disciplines, specially since the early 60's [13, 12]. Some classic approaches include partitional methods such as k-means, k-medioids, hierarchical agglomerative clustering, unsupervised Bayes, and soft, statistical mechanics, or EM based techniques. Most classical techniques, and even fairly recent ones proposed in the data mining community (CLARANS, DBScan, BIRCH, CLIQUE, CURE, WaveCluster etc. [20]), are based on distances between the samples in the original vector space. Their emphasis is primarily on an efficient and scalable (w.r.t. number of records) implementation of approximate k-means or k-medioids. Thus they are faced with the "curse of dimensionality" [10] and the associated sparsity issues, when dealing with very high-dimensional data. Essentially the amount of data to sustain a given spatial density increases exponentially with the dimensionality of the input space, or alternatively, the sparsity increases exponentially given a constant amount of data. This directly affects any method based on spatial density. Moreover, if data is distributed at random, weird things happen, such as points becoming roughly equi-distant from one another. Consequently, distance or spatial density based techniques do not work well in general with high-dimensional data. Some other limitations of popular clustering methods are nicely illustrated in [15].

Recently, some innovative approaches that directly address high-dimensional data mining have emerged. ROCK (Robust Clustering using linKs) [11] is an agglomerative hierarchical clustering technique for categorical attributes. It uses the binary Jaccard coefficient and a thresholding criterion to establish links between samples. Common neighbors are used to define inter-connectivity of clusters which is used to merge clusters. CHAMELEON [15] starts with partitioning the data into a large number of clusters by partitioning the v-nearest neighbor graph. In the subsequent stage clusters are merged based on relative inter-connectivity and relative closeness measures. These localized measures lead to a dynamic adaption capability with spectacular results for 2–dimensional data. But its effectiveness and interpretability for higher dimensional data is not reported.

Ordering has also been used in conjunction with clustering. In OPTICS [1] instead of producing an explicit clustering, an augmented ordering of the database is produced. Subsequently, this ordering is used to display various metrics such as reachability values. In cluster analysis of genome data [7] re-ordering the primary data matrix and representing it graphically has been explored. The sparsity of the original data matrix makes working on it directly uninteresting in our domain. This visualization differs from our work since we work in the relationship-space and not the primary data space. Sparse primary data matrix reorderings have also been considered for browsing hypertext [2].

Visualization of high-dimensional data clusters can be largely divided into three popular approaches:

1. Dimensionality reduction by selection of 2 or 3 dimensions, or, more generally, projecting the data down to 2 or 3 dimensions. Often these dimensions correspond to principal components or an scalable approximation thereof (e.g., Fastmap [8]). Chen, for example, creates a browsable 2–dimensional space of authors through co-citations [5]. Another noteworthy method is CViz [6], which projects onto the plane that passes through three selected cluster centroids to yield a “discrimination optimal” 2–dimensional projection. These projections are useful for a medium number of dimensions, i.e., if d is not too large (< 100).[1] Nonlinear projections have also been studied [4]. Recreating a 2– or 3–dimensional space from a similarity graph can also be done through multi-dimensional scaling [24].

2. Parallel axis plots show each object as a line along d parallel axis. However, this technique is rendered ineffective if the number of dimensions d or the number of objects gets too high.

3. Kohonen’s Self Organizing Map (SOM) [18] provides an innovative and powerful way of clustering while enforcing constraints on a logical topology imposed on the cluster centers. If this topology is 2–dimensional, one can readily ”visualize” the clustering of data. Essentially a 2–dimensional manifold is mapped onto the (typically higher dimensional) feature space, trying to approximate data density while maintaining topological constraints. Since the mapping is not bijective, the quality can degrade very rapidly with increasing dimensionality of feature space, unless the data is largely confined to a much lower order manifold within this space [4].

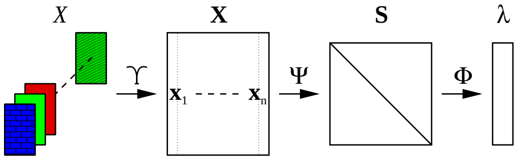


Figure 1: The relationship-based clustering framework.

A useful survey of visualization methods for data mining in general (not focussed on clustering) can be found in [16]. The popular books by E. Tufte [25] on visualizing information are also recommended.

3 Domain Specific Features and Similarity Space

Notation. Let n be the number of objects (e.g., customers, documents) in the data and d the number of features (e.g., products, words) for each sample $\mathbf{x}_j$ with $j \in \{1,\ldots,n\}$. Let k be the desired number of clusters. The input data can be represented by a $d \times n$ data matrix $\mathbf{X}$ with the j-th column vector representing the sample $\mathbf{x}_j$. $\mathbf{x}_j^\dagger$ denotes the transpose of $\mathbf{x}_j$. Hard clustering assigns a label $\lambda_j \in \{1,\ldots,k\}$ to each d–dimensional sample $\mathbf{x}_j$, such that similar samples get the same label. In general the labels are treated as nominals with no inherent order, though in some cases, such as 1–dimensional SOMs or top-down recursive graph-bisection, the labeling contains extra ordering information. Let $\mathcal{C}_\ell$ denote the set of all objects in the ℓ-th cluster ($\ell \in \{1,\ldots,k\}$), with $\mathbf{x}_j \in \mathcal{C}_\ell \Leftrightarrow \lambda_j = \ell$ and $n_\ell = |\mathcal{C}_\ell|$.

Fig. 1 gives an overview of our relationship-based clustering **process** from a set of raw object descriptions $\mathcal{X}$ via the vector space description $\mathbf{X}$ and similarity space description $\mathbf{S}$ to the cluster labels λ: $(\mathcal{X} \in \mathcal{I}^n) \xrightarrow{\Upsilon} (\mathbf{X} \in \mathcal{F}^n \subset \mathbb{R}^{d\times n}) \xrightarrow{\Psi} (\mathbf{S} \in \mathcal{S}^{n\times n} = [0,1]^{n\times n} \subset \mathbb{R}^{n\times n}) \xrightarrow{\Phi} (\lambda \in \mathcal{O}^n = \{1,\ldots,k\}^n)$. For example in web-page clustering, $\mathcal{X}$ is a collection of n web-pages x_j with $j \in \{1,\ldots,n\}$. Extracting features using Υ yields $\mathbf{X}$, the term frequencies of stemmed words, normalized such that for all documents $\mathbf{x} : \|\mathbf{x}\|_2 = 1$. Similarities are computed, using e.g., cosine based similarity Ψ yielding the $n \times n$ similarity matrix $\mathbf{S}$. Finally, the cluster label vector λ is computed using a clustering function Φ, such as graph-partitioning. In short, the basic process can be denoted as $\mathcal{X} \xrightarrow{\Upsilon} \mathbf{X} \xrightarrow{\Psi} \mathbf{S} \xrightarrow{\Phi} \lambda$.

Similarity Measures. The key idea behind dealing with very high-dimensional data is to work in similarity space rather than the original vector space in which the feature vectors reside. A similarity measure captures the relationship between two d-dimensional objects in a single number (using on the order of non-zeros or d, at worst, computations). Once this is done, the original high-dimensional space is not dealt with at all, we only work in the transformed similarity space, and subsequent processing is independent of d.

A similarity measure $\in [0,1]$ captures how related two data-points $\mathbf{x}_a$ and $\mathbf{x}_b$ are. It should be symmetric ($s(\mathbf{x}_a,\mathbf{x}_b) = s(\mathbf{x}_b,\mathbf{x}_a)$), with self-similarity $s(\mathbf{x}_a,\mathbf{x}_a) = 1$. However, in general, similarity functions (respectively their distance function equivalents) do *not* fulfill the triangular inequality.

A brute force implementation does involve $O(n^2 \times d)$ operations, since similarity needs to be computed between each pair of data points, and involve all the dimensions. Also, $O(n^2)$ storage is

[1] For text mining, projecting down to about 50 dimensions does not affect results much (e.g. latent semantic indexing). However, it is still too high to visualize. A projection to lower dimensions leads to substantial degradation so 3–dimensional projection becomes meaningless.

required for the similarity matrix. Computing the similarity matrix is the bottleneck; once that is done, any subsequent clustering routine does not depend on d at all, and also scales much better with n.

The computational complexity can be reduced in a variety of ways, from subsampling the data for seed clustering (particularly effective when the clusters are of comparable size), to rolling up the customer or product hierarchies. The storage requirements can be reduced by exploiting sparsity in the original data as well as the similarity graph. Both these issues are addressed in our subsequent work; for now we concentrate on dealing with high dimensionality issues and on visualization of results. For an analysis of how to parallelize our technique for near linear speedup on distributed memory multicomputers, see [21].

An obvious way to compute similarity is through a suitable monotonic and inverse function of a Minkowski (L_p) distance, d. Candidates include $s = 1/(1+d)$ and $s = e^{-d^2}$, the later being preferable due to maximum likelihood properties [23]. Similarity can also be defined by the cosine of the angle between two vectors:

$$s^{(\mathrm{C})}(\mathbf{x}_a, \mathbf{x}_b) = \frac{\mathbf{x}_a^\dagger \mathbf{x}_b}{\|\mathbf{x}_a\|_2 \cdot \|\mathbf{x}_b\|_2} \tag{1}$$

Cosine similarity is widely used in text clustering because two documents with the same proportions of term occurrences but different lengths are often considered identical. In retail data such normalization loses important information about the life-time customer value, and we have recently shown that the **extended Jaccard similarity** measure is more appropriate [23]. For binary features, the Jaccard coefficient [13] measures the ratio of the intersection of the product sets to the union of the product sets corresponding to transactions $\mathbf{x}_a$ and $\mathbf{x}_b$, each having binary (0/1) elements.

$$s^{(\mathrm{J})}(\mathbf{x}_a, \mathbf{x}_b) = \frac{\mathbf{x}_a^\dagger \mathbf{x}_b}{\|\mathbf{x}_a\|_2^2 + \|\mathbf{x}_b\|_2^2 - \mathbf{x}_a^\dagger \mathbf{x}_b} \tag{2}$$

The extended Jaccard coefficient is also given by equation 2, but allows elements of $\mathbf{x}_a$ and $\mathbf{x}_b$ to be arbitrary positive real numbers. This coefficient captures a vector-length-sensitive measure of similarity. However, it is still invariant to scale (dilating $\mathbf{x}_a$ and $\mathbf{x}_b$ by the same factor does not change $s(\mathbf{x}_a, \mathbf{x}_b)$). A detailed discussion of the properties of various similarity measures can be found in [23], where it is shown that the extended Jaccard coefficient is particularly well suited for market-basket data.

Clearly, for general data distributions, one cannot avoid the "curse of dimensionality". What we have achieved is to determine an appropriate measure for the given applications, which captures the essential aspects of the class of high-dimensional data distributions being considered. For other applications, one would have to determine what similarity measure is suitable.

4 CLUSION: Cluster Visualization

In this section, we present our visualization tool, highlight some of its properties and compare it with some popular visualization methods. Applications to visualizing high-dimensional data clusters are relegated to section 6.

4.1 Coarse Seriation

When data is limited to 2 or 3 dimensions, the most powerful tool for judging cluster quality is usually the human eye. CLUSION, our CLUSter visualizatION toolkit, allows us to convert high-dimensional data into a perceptually more suitable format, and employ the human vision system to explore the *relationships* in the data, *guide* the clustering process, and *verify* the quality of the results. In our experience with two years of Dell customer data, we found CLUSION effective for getting clusters balanced w.r.t. number of customers or net dollar amount, and even more so for conveying the results to upper management.

CLUSION looks at the output of a clustering routine, reorders the data points such that points with the same cluster label are contiguous, and then visualizes the resulting similarity matrix, $\mathbf{S}'$. More formally, the original $n \times n$ similarity matrix $\mathbf{S}$ is permuted with a $n \times n$ permutation matrix $\mathbf{P}$ which is defined as follows:

$$p_{i,j} = \begin{cases} 1 & \text{if } j = \sum_{a=1}^{i} l_{a,\lambda_i} + \sum_{\ell=1}^{\lambda_i - 1} n_\ell \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

l are entries in the binary $n \times k$ cluster membership indicator matrix $\mathbf{L}$:

$$l_{i,j} = \begin{cases} 1 & \text{if } \lambda_i = j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

In other words, $p_{i,j}$ is 1 if j is the sum of the number of points amongst the first i that belong to the same cluster and the number of points in the first $\lambda_i - 1$ clusters. Now, the permuted similarity matrix $\mathbf{S}'$ and the corresponding label vector λ' and data matrix $\mathbf{X}'$ are:

$$\mathbf{S}' = \mathbf{P}\mathbf{S}\mathbf{P}^\dagger \quad , \quad \lambda' = \mathbf{P}\lambda \quad , \quad \mathbf{X}' = \mathbf{P}\mathbf{X} \tag{5}$$

For a "good" clustering algorithm and $k \rightarrow n$ this is related to sparse matrix reordering, for this results in the generation of a "banded matrix" where high entries should all fall near the diagonal line from the upper left to the lower right of the matrix. Since equation 5 is essentially a partial ordering operation we also refer to it as coarse *seriation*, a phrase used in disciplines such as anthropology and archaeology to describe the reordering of the primary data matrix so that similar structures (e.g., genetic sequences) are brought closer [19, 7].

4.2 Visualization

The seriation of the similarity matrix, $\mathbf{S}'$, is very useful for visualization. Since the similarity matrix is 2–dimensional, it can be readily visualized as a gray-level image where a white (black) pixel corresponds to minimal (maximal) similarity of 0 (1). The darkness (gray level value) of the pixel at row a and column b increases with the similarity between the samples $\mathbf{x}_a$ and $\mathbf{x}_b$. When looking at the image it is useful to consider the similarity s as a random variable taking values from 0 to 1. The similarity *within* cluster ℓ is thus represented by the average intensity within a square region with side length n_ℓ, around the main diagonal of the matrix. The off-diagonal rectangular areas visualize the relationships *between* clusters. The brightness distribution in the rectangular areas yields insight towards the quality of the clustering and possible improvements. In order to make these regions apparent, thin red horizontal and vertical lines are used to show the divisions into the rectangular regions[2]. Visualizing similarity space in this way can help to quickly get a feel for the clusters in the data. Even for a large number of points, a sense for the intrinsic number of clusters k in a data-set can be gained.

Fig. 2 shows CLUSION output in four extreme scenarios to provide a feel for how data properties translate to the visual display. Without any loss of generality, we consider the partitioning of a set of objects into 2 clusters. For each scenario, on the left hand side the original similarity matrix $\mathbf{S}$ and the seriated version $\mathbf{S}'$ (CLUSION) for an optimal bipartitioning is shown. On the right hand side four histograms for the distribution of similarity values s, which range from 0 to 1, are shown. From left to right, we have plotted: distribution of s over the entire data, within the first cluster, within the

[2] This can be more clearly seen in the color pictures in the soft-copy.

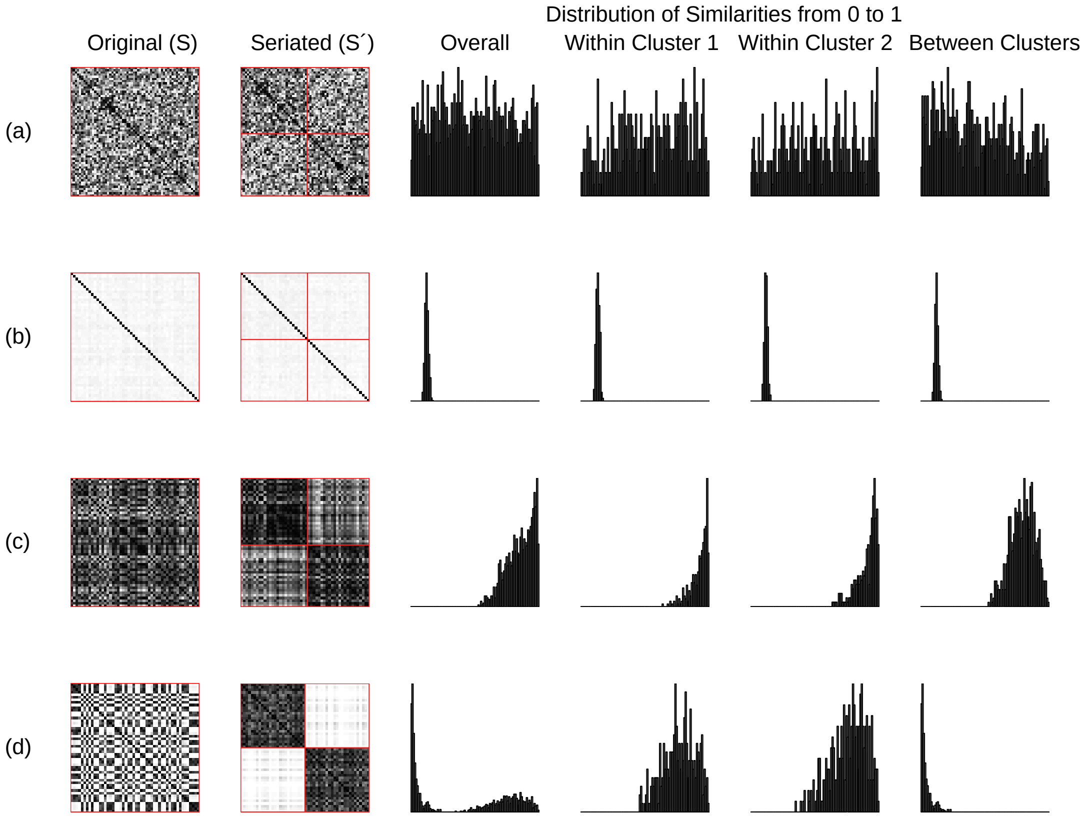


Figure 2: Illustrative CLUSION patterns in original order and seriated using optimal bipartitioning are shown in the left two columns. The right four columns show corresponding similarity distributions. In each example there are 50 objects: (a) no natural clusters (randomly related objects), (b) set of singletons (pairwise near orthogonal objects), (c) one natural cluster (uni-modal Gaussian), (d) two natural clusters (mixture of two Gaussians)

second cluster, and between first and second cluster. If the data is naturally clustered and the clustering algorithm is good, then the middle two columns of plots will be much more skewed to the right as compared to the first and fourth columns. In our visualization this corresponds to brighter off-diagonal regions and darker block diagonal regions in $\mathbf{S}'$ as compared to the original $\mathbf{S}$ matrix.

The proposed visualization technique is quite powerful and versatile. In Fig. 2(a) the chosen similarity behaves randomly. Consequently, no strong visual difference between on- and off-diagonal regions can be perceived with CLUSION in S'. It indicates clustering is ineffective which is expected since there is no structure in the similarity matrix. Fig. 2(b) is based on data consisting of pair-wise almost equi-distant singletons. Clustering into two groups still renderes the on-diagonal regions very bright suggesting more splits. In fact, this will remain unchanged until each data-point is a cluster by itself, thus, revealing the singleton character of the data. For monolithic data (Fig. 2(c)), many strong similarities are indicated by an almost uniformly dark similarity matrix $\mathbf{S}$. Splitting the data results in dark off-diagonal regions in $\mathbf{S}'$. A dark off-diagonal region suggests that the clusters in the corresponding rows and columns should be merged (or not be split in the first place). CLUSION indicates that this data is actually one large cluster. In Fig. 2(d), the gray-level distribution of $\mathbf{S}$ exposes bright as well as dark pixels, thereby recommending it should be split. In this case, $k = 2$ apparently is a very good choice (and the clustering algorithm worked well) because in $\mathbf{S}'$ on-diagonal regions are uniformly dark and off-diagonal regions are uniformly bright.

This induces an intuitive mining process that guides the user to the “right” number of clusters. Too small a k leaves the on-diagonal regions inhomogeneous. On the contrary, growing k beyond the natural number of clusters will introduce dark off-diagonal regions. Finally, CLUSION can be used to visually compare the appropriateness of different similarity measures.

4.3 Comparison

CLUSION gives a *relationship-centered* view, as contrasted with common projective techniques, such as the selection of dominant features or optimal linear projections (PCA), which are *object-centered*. In CLUSION, the actual features are transparent, instead, all pair-wise relationships, the relevant aspect for the purpose of clustering, are displayed.

Fig. 3 compares CLUSION with some other popular visualizations. In Fig. 3 parallel axis, PCA projection, CViz (projection through plane defined by centroids of clusters 1, 2, and 3) as well as CLUSION succeed in visualizing the IRIS data. Membership in cluster 1/2/3 is indicated by colors red/blue/green (parallel axis), colors red/blue/green and shapes ∘/×/+ (PCA and CViz), and position on diagonal from upper left to lower right corner (CLUSION), respectively. All four tools succeed in visualizing three clusters and making apparent that clusters 2 and 3 are closer than any other and cluster 1 is very compact.

Fig. 3(b) shows the same comparison for 293 documents from which 2903 word frequencies where extracted to be used as features. In fact this data consists of 5 clusters selected from 40 clusters extracted from a YAHOO news document collection which will

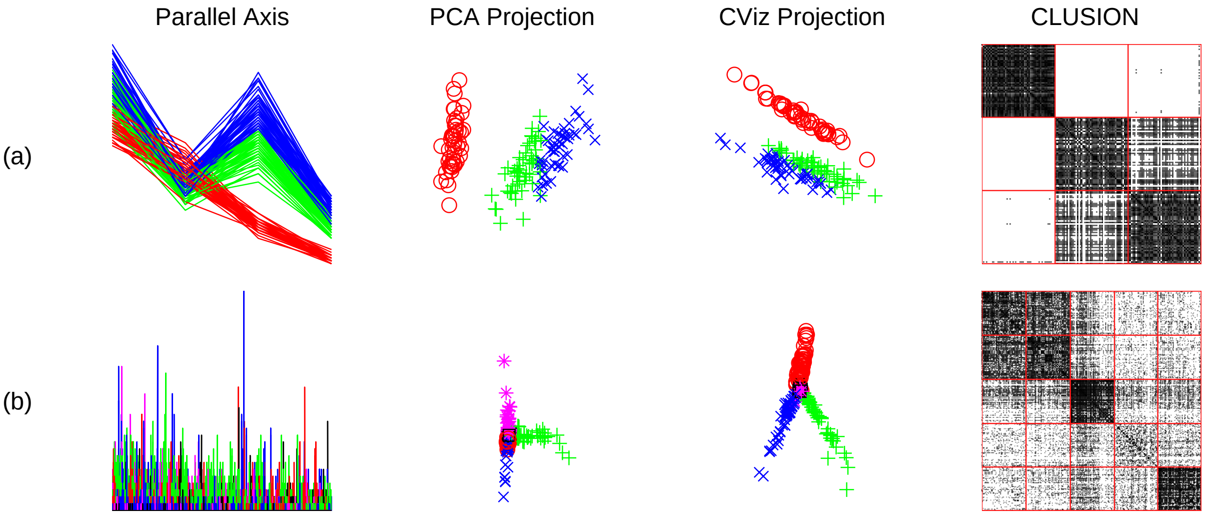


Figure 3: Comparison of Visualization Techniques. All tools work well on the IRIS data (a). On the high-dimensional document data (b), only CLUSION reveals that clusters 1 and 2 are actually highly related, cluster 3 is strong and interdisciplinary, 4 is weak, and 5 is strong.

be described in more detail in section 6. The colors black/magenta and the shapes □/$*$ have been added to indicate cluster 4/5, respectively. The parallel axis plot becomes useless clutter due to the high number of dimensions as well as the large number of objects. PCA and CViz succeed in separating three clusters each (2, 3, 5 and 1, 2, 3, respectively) and show all others superimposed on the axis origin. They give no suggestions towards which clusters are compact or which clusters are related. Only CLUSION suggests that clusters 1 and 2 are actually highly related, cluster 3 is interdisciplinary, 4 is weak, and 5 is a strong cluster. And indeed, when looking at the cluster descriptions (which might not be so easily available and understandable in all domains), the intuitive interpretations revealed by CLUSION are proven to be very true:

cluster	dominant category	purity	entropy	most frequent word stems
1	`health` (H)	100%	0.00	hiv, depress, immun
2	`health` (H)	100%	0.00	weight, infant, babi
3	`online` (o)	58%	0.43	apple, intel, electron
4	`film` (f)	38%	0.72	hbo, ali, alan
5	`television` (t)	83%	0.26	household, sitcom, timeslot

Note that the majority category, purity, and entropy are only available where a supervised categorization is given. Of course the categorization cannot be used to tune the clustering. Clusters 1 and 2 contains only documents from the `Health` category so they are highly related. The 4th cluster, which is indicated to be weak by CLUSION, has in fact the lowest purity in the group with 38% of documents from the most dominant category (`film`). CLUSION also suggests cluster 3 is not only strong, as indicated by the dark diagonal region, but also has distinctly above average relationships to *all* other 4 clusters. On inspecting the word stems typifying this cluster (Apple, Intel, and electron(ics)) it is apparent that this is because of the interdisciplinary appearance of technology savvy words in recent news releases. Since such cluster descriptions might not be so easily available or well understood in all domains, the intuitive display of CLUSION is very useful.

CLUSION has several other powerful properties. For example, it can be integrated with product hierarchies (meta-data) to provide simultaneous customer and product clustering, as well as multi-level views / summaries. It also has a graphical user interface so one can interactively browse / split / merge a data-set.

5 OPOSSUM

In this section, we summarize OPOSSUM (Optimal Partitioning of Sparse Similarities Using Metis), the clustering technique whose results will be used for visualization using CLUSION. Further details are given in [22], and the scalability and parallel processing aspects are analyzed in [21].

OPOSSUM differs from other graph-based clustering techniques by application-driven balancing of clusters, non-metric similarity measures, and visualization driven heuristics for finding an appropriate k. OPOSSUM strives to deliver "balanced" clusters using either of the following two criteria:

- *Sample balanced:* Each cluster should contain roughly the same number of samples, n/k. This allows, for example, retail marketers to obtain a customer segmentation with equally sized customer groups.

- *Value balanced:* Each cluster should contain roughly the same amount of feature values. In customer clustering, a cluster represents a k-th fraction of the total feature value $v = \sum_{j=1}^{n} \sum_{i=1}^{d} x_{i,j}$. If we use extended revenue per product (quantity × price) as value, then each cluster represents a roughly equal contribution to total revenue.

We formulate the desired balancing properties by assigning each sample (customer) a weight and then softly constrain the sum of weights in each cluster. For sample balanced clustering, we assign each sample $\mathbf{x}_j$ the same weight $w_j = 1/n$. To obtain value balancing properties, a sample $\mathbf{x}_j$'s weight is set to $w_j = \sum_{i=1}^{d} x_{i,j}/v$. Please note that the sum of weights for all samples is 1. Balancing avoids trivial clusterings (e.g., $k-1$ singletons and 1 big cluster). More importantly, the desired balancing properties have many application driven advantages. For example when each cluster contains the same number of customers, discovered phenomena (e.g. frequent products, co-purchases) have equal significance / support and are thus easier to evaluate. When each customer cluster equals the same revenue share, marketing can spend an equal amount of attention and budget to each of the groups.

We map the problem of clustering to partitioning a vertex weighted graph into k unconnected components by removing a minimal amount of edges while maintaining a balancing constraint.

The objects to be clustered can be viewed as a set of vertices. Two vertices $\mathbf{x}_a$ and $\mathbf{x}_b$ are connected with an undirected edge (a, b) of positive weight given by the similarity $s(\mathbf{x}_a, \mathbf{x}_b)$. The clustering task is then to find an edge separator with a minimum sum of edge weights, that partitions the graph into k disjoint pieces. Without loss of generality, we can assume that the vertex weights w_j are normalized to sum up to 1: $\sum_{j=1}^{n} w_j = 1$. While striving for the minimum cut objective, the balancing constraint $k \cdot \max_\ell \sum_{\lambda_j = \ell} w_j \leq t$ has to be fulfilled. The left hand side of the inequality is called the imbalance (the ratio of the biggest cluster in terms of cumulative normalized edge weight to the desired equal cluster-size $1/k$) and has a lower bound of 1. The balancing threshold t enforces perfectly balanced clusters for $t = 1$. In general t is slightly greater than 1 (e.g., we use $t = 1.05$ for all our experiments which allows at most 5% of imbalance).

After experimentation with several techniques for this, we decided to use the Metis multi-level multi-constraint graph partitioning package because it is very fast and scales well. A detailed description of the algorithms used in Metis can be found in Karypis et al. [14]. This is embellished by our heuristic [22] that obtains an appropriate value of k during the clustering process, and whose results are supported by the visualization technique. Among the alternatives considered were spectral bisectioning techniques. However these were not so efficient as METIS and also the number of clusters natural for such an approach was a power of 2, reducing some flexibility as compared to METIS.

6 Experiments

6.1 Retail Market-basket Clusters

First, we will show clusters in a real retail transaction database of 21672 customers of a drugstore[3]. For the illustrative purpose of this paper, we randomly selected 2500 customers. The total number of transactions (cash register scans) for these customers is 33814 over a time interval of three months. We rolled up the product hierarchy once to obtain 1236 different products purchased. 15% of the total revenue is contributed by the single item `Financial-Depts` (on site financial services such as check cashing and bill payment) which was removed because it was too common. 473 of these products accounted for less than $25 each in toto and were dropped. The remaining $d = 762$ features and $n = 2466$ customers (34 customers had empty baskets after removing the irrelevant products) were clustered using Opossum.

In this customer clustering case study we set $k = 20$. In this application domain, the number of clusters is often predetermined by marketing considerations such as advertising industry standards, marketing budgets, marketers ability to handle multiple groups, and the cost of personalization. In general, a reasonable value of k can be obtained using heuristics as in [21].

Opossum's results for this example are obtained with a 1.7 GHz Pentium 4 PC with 512 MB RAM in approximately 35 seconds (~30s file I/O, 2.5s similarity computation, 0.5s conversion to integer weighted graph, 0.5s graph partitioning). Fig. 4 shows the extended Jaccard similarity matrix (83% sparse) using Clusion: (a) originally (randomly) ordered data, (b) seriated using Euclidean k-means, (c) using SOM, (d) using standard Jaccard k-means, (e) using extended Jaccard sample balanced Opossum, (f) using value balanced Opossum clustering. Customer and revenue ranges are given below each image. In (a), (b), (c), and (d) clusters are neither compact nor balanced. In (e) and (f) clusters are much more compact, even though there is the additional constraint that they be balanced, based on equal number of customers and equal revenue metrics, respectively. Below each Clusion visualization, the ranges of numbers of customers and revenue totals in $ over all clusters are given to indicate balancedness. We also experimented with minimum distance agglomerative clustering but this resulted in 19 singletons and 1 cluster with 2447 customers so we did not bother including this approach. Clearly, k-means in the original feature space, the standard clustering algorithm, does not perform well at all (Fig. 4(b)). The SOM after 100000 epochs performs slightly better (Fig. 4(c)) but is outperformed by the standard Jaccard k-means (Fig. 4(d)) which is adopted to similarity space by using $\sqrt{-log(s^{(\mathrm{J})})}$ as distances [23]. As the relationship-based Clusion shows, Opossum (Fig. 4(e),(f)) gives more compact (better separation of on- and off-diagonal regions) and well balanced clusters as compared to all other techniques. For example, looking at standard Jaccard k-means, the clusters contain between 48 and 597 customers contributing between $608 and $70443 to revenue[4]. Thus the clusters may not be of comparable importance from a marketing standpoint. Moreover clusters are hardly compact: Darkness is only slightly stronger in the on-diagonal regions in Fig. 4(d). All visualizations have been histogram equalized for printing purposes. However, they are still much better observed by browsing interactively on a computer screen.

A very compact and useful way of profiling a cluster is to look at their most *descriptive* and their most *discriminative* features. For market-basket data, this can be done by looking at a cluster's highest revenue products and the most unusual revenue drivers (e.g., products with highest revenue lift). Revenue lift is the ratio of the average spending on a product in a particular cluster to the average spending in the entire data-set.

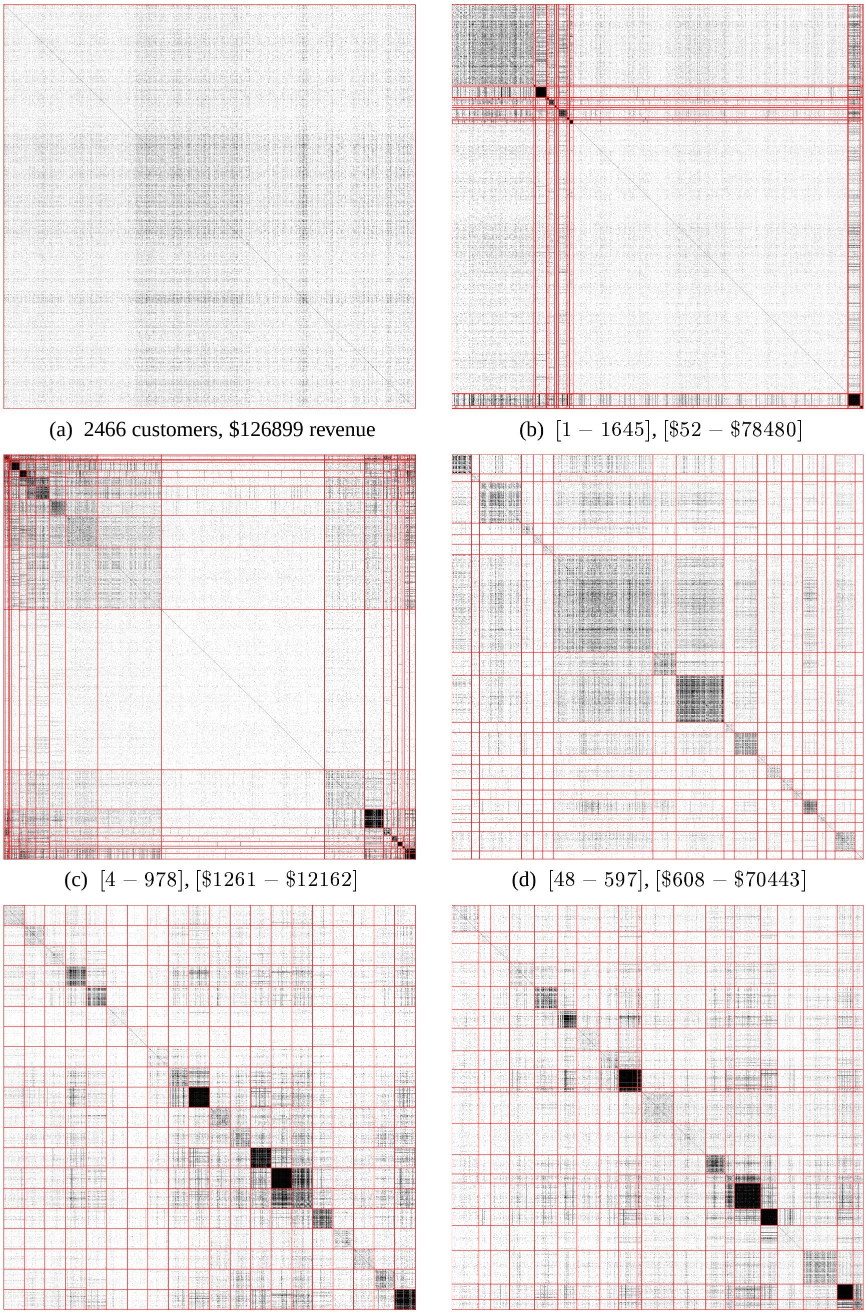
In Table 1 the top three descriptive and discriminative products for the customers in the 20 value balanced clusters are shown (see also Fig. 4(f)). Customers in cluster $\mathcal{C}_2$, for example, mostly spent their money on smoking cessation gum ($10.15 on average). Interestingly, while this is a 35-fold average spending on smoking cessation gum, these customers also spend 35 times more on blood pressure related items, peanuts and snacks. Do these customers lead an unhealthy lifestyle and are eager to change? Why not offer a special "quit smoking bundle" to increase revenue using the knowledge about this aspect of customer buying behavior? Cluster $\mathcal{C}_{15}$, which can be seen to be highly compact cluster of shoppers characterized by greeting card purchases (Maybe elderly people because they spend 7 times more money on hearing aid batteries?). Note that Opossum had an extra constraint that clusters should be of comparable value. This may force a larger natural cluster to split, as may be the case causing the similar clusters 9 and 10. Both are Christmas gift shoppers (Table 1(top)), cluster 9 are the moderate spenders and cluster 10 are the big spenders, as cluster 10 is much smaller with equal revenue contribution (Fig. 4(f)). Our hunch is reinforced by looking at Fig. 4(f).

6.2 Web-document Clusters

In this section, we present results on documents from the YAHOO news section. Each of the 2340 documents is characterized by a bag of words. The data is publicly available from `ftp://ftp.cs.umn.edu/dept/users/boley/` (K1 series) and was used in [3, 23]. The 20 original YAHOO news categories are `Business` (B), `Entertainment (no sub-category` (E), `art` (a), `cable` (c), `culture` (cu), `film` (f), `industry` (i), `media` (m), `multimedia` (mm), `music` (mu), `online` (o), `people` (p), `review` (r), `stage` (s), `television` (t), `variety` (v)), `Health` (H), `Politics` (P), `Sports` (S), `Technology` (T) and correspond to the category labels $1, \ldots, 20$, respectively. The raw 21839×2340 word-by-document matrix

[3] provided by Knowledge Discovery 1

[4] The solution for k-means depends on the initial choices for the means. A representative solution is shown here.



(a) 2466 customers, \$126899 revenue

(b) $[1 - 1645]$, $[\$52 - \$78480]$

(c) $[4 - 978]$, $[\$1261 - \$12162]$

(d) $[48 - 597]$, $[\$608 - \$70443]$

(e) $[122 - 125]$, $[\$1624 - \$14361]$

(f) $[28 - 203]$, $[\$6187 - \$6609]$

Figure 4: Visualizing clustering high-dimensional drugstore data into 20 clusters. Relationship visualizations using Clusion: (a) original (randomly) ordered data, (b) seriated or partially reordered using Euclidean k-means, (c) using SOM, (d) using standard Jaccard k-means, (e) using extended Jaccard sample balanced Opossum, (f) using value balanced Opossum clustering. Customer and revenue ranges are given below each image. In (a), (b), (c), and (d) clusters are neither compact nor balanced. In (e) and (f) clusters are much more compact, even though there is the additional constraint that they be balanced, based on equal number of customers and equal revenue metrics, respectively.

$\mathcal{C}_\ell$	top product	$	lift	sec. product	$	lift	third product	$	lift
1	bath gift packs	3.44	7.69	hair growth m	0.90	9.73	boutique island	0.81	2.61
2	smoking cessati	10.15	34.73	tp canning item	2.04	18.74	blood pressure	1.69	34.73
3	vitamins other	3.56	12.57	tp coffee maker	1.46	10.90	underpads hea	1.31	16.52
4	games items 180	3.10	7.32	facial moisturi	1.80	6.04	tp wine jug ite	1.25	8.01
5	batt alkaline i	4.37	7.27	appliances item	3.65	11.99	appliances appl	2.00	9.12
6	christmas light	8.11	12.22	appliances hair	1.61	7.23	tp toaster/oven	0.67	4.03
7	christmas food	3.42	7.35	christmas cards	1.99	6.19	cold bronchial	1.91	12.02
8	girl toys/dolls	4.13	12.51	boy toys items	3.42	8.20	everyday girls	1.85	6.46
9	christmas giftw	12.51	12.99	christmas home	1.24	3.92	christmas food	0.97	2.07
10	christmas giftw	19.94	20.71	christmas light	5.63	8.49	pers cd player	4.28	70.46
11	tp laundry soap	1.20	5.17	facial cleanser	1.11	4.15	hand&body thera	0.76	5.55
12	film cameras it	1.64	5.20	planners/calend	0.94	5.02	antacid h2 bloc	0.69	3.85
13	tools/accessori	4.46	11.17	binders items 2	3.59	10.16	drawing supplie	1.96	7.71
14	american greeti	4.42	5.34	paperback items	2.69	11.04	fragrances op	2.66	12.27
15	american greeti	5.56	6.72	christmas cards	0.45	2.12	basket candy it	0.44	1.45
16	tp seasonal boo	10.78	15.49	american greeti	0.98	1.18	valentine box c	0.71	4.08
17	vitamins e item	1.76	6.79	group stationer	1.01	11.55	tp seasonal boo	0.99	1.42
18	halloween bag c	2.11	6.06	basket candy it	1.23	4.07	cold cold items	1.17	4.24
19	hair clr perman	12.00	16.76	american greeti	1.11	1.34	revlon cls face	0.83	3.07
20	revlon cls face	7.05	26.06	hair clr perman	4.14	5.77	headache ibupro	2.37	12.65

$\mathcal{C}_\ell$	top product	$	lift	sec. product	$	lift	third product	$	lift
1	action items 30	0.26	15.13	tp video comedy	0.19	15.13	family items 30	0.14	11.41
2	smoking cessati	10.15	34.73	blood pressure	1.69	34.73	snacks/pnts nut	0.44	34.73
3	underpads hea	1.31	16.52	miscellaneous k	0.53	15.59	tp irons items	0.47	14.28
4	acrylics/gels/w	0.19	11.22	tp exercise ite	0.15	11.20	dental applianc	0.81	9.50
5	appliances item	3.65	11.99	housewares peg	0.13	9.92	tp tarps items	0.22	9.58
6	multiples packs	0.17	13.87	christmas light	8.11	12.22	tv's items 6	0.44	8.32
7	sleep aids item	0.31	14.61	kava kava items	0.51	14.21	tp beer super p	0.14	12.44
8	batt rechargeab	0.34	21.82	tp razors items	0.28	21.82	tp metal cookwa	0.39	12.77
9	tp furniture it	0.45	22.42	tp art&craft al	0.19	13.77	tp family plan,	0.15	13.76
10	pers cd player	4.28	70.46	tp plumbing ite	1.71	56.24	umbrellas adult	0.89	48.92
11	cat litter scoo	0.10	8.70	child acetamino	0.12	7.25	pro treatment i	0.07	6.78
12	heaters items 8	0.16	12.91	laverdiere ca	0.14	10.49	ginseng items 4	0.20	6.10
13	mop/broom lint	0.17	13.73	halloween cards	0.30	12.39	tools/accessori	4.46	11.17
14	dental repair k	0.80	38.17	tp lawn seed it	0.44	35.88	tp telephones/a	2.20	31.73
15	gift boxes item	0.10	8.18	hearing aid bat	0.08	7.25	american greeti	5.56	6.72
16	economy diapers	0.21	17.50	tp seasonal boo	10.78	15.49	girls socks ite	0.16	12.20
17	tp wine 1.5l va	0.17	15.91	group stationer	1.01	11.55	stereos items 2	0.13	10.61
18	tp med oint,liq	0.10	8.22	tp dinnerware i	0.32	7.70	tp bath towels	0.12	7.28
19	hair clr perman	12.00	16.76	covergirl imple	0.14	11.83	tp power tools	0.25	10.89
20	revlon cls face	7.05	26.06	telephones cord	0.56	25.92	ardell lashes i	0.59	21.87

Table 1: List of *descriptive* (top) and *discriminative* products (bottom) dominant in each of the 20 value balanced clusters obtained from the drugstore data (see also Fig. 4(f)). For each item the average amount of $ spent in this cluster and the corresponding lift is given.

consists of the non-normalized occurrence frequencies of stemmed words, using Porter's suffix stripping algorithm [9]. Pruning all words that occur less than 0.01 or more than 0.10 times on average because they are insignificant (e.g., `abdrazakof`) or too generic (e.g., `new`), respectively, results in $d = 2903$.

Let us point out some worthwhile differences between clustering market-baskets and documents. Firstly, discrimination of vector length is no longer desired since customer life-time value matters but document length does not. Consequently, we use cosine similarity $s^{(\mathrm{C})}$ instead of extended Jaccard similarity $s^{(\mathrm{J})}$. Also, in document clustering we are less concerned about balancing, since there are usually no direct monetary costs of the actions derived from the clustering involved. As a consequence of this, we over-cluster first with sample-balanced OPOSSUM and then allow user guided merging of clusters through CLUSION. The YAHOO news dataset is notorious for having some diffuse groups with overlaps among categories, a few categories with multi-modal distributions etc. These aspects can be easily explored by looking at the class labels within each cluster, merging some clusters and then again visualizing the results.

Fig. 5 shows clusterings with three settings of k. For $k = 10$ (Fig. 5(a)) most clusters are not dense enough, despite the fact that the first two clusters already seem like they should not have been split. After increasing to $k = 40$ (Fig. 5(b)), CLUSION indicates that the clustering now has sufficiently compact clusters. Now, we successively merge pairs of highly related clusters (by simply clicking on bright off-diagonal regions) until we obtain our final clustering with $k = 20$ (Fig. 5(c)).

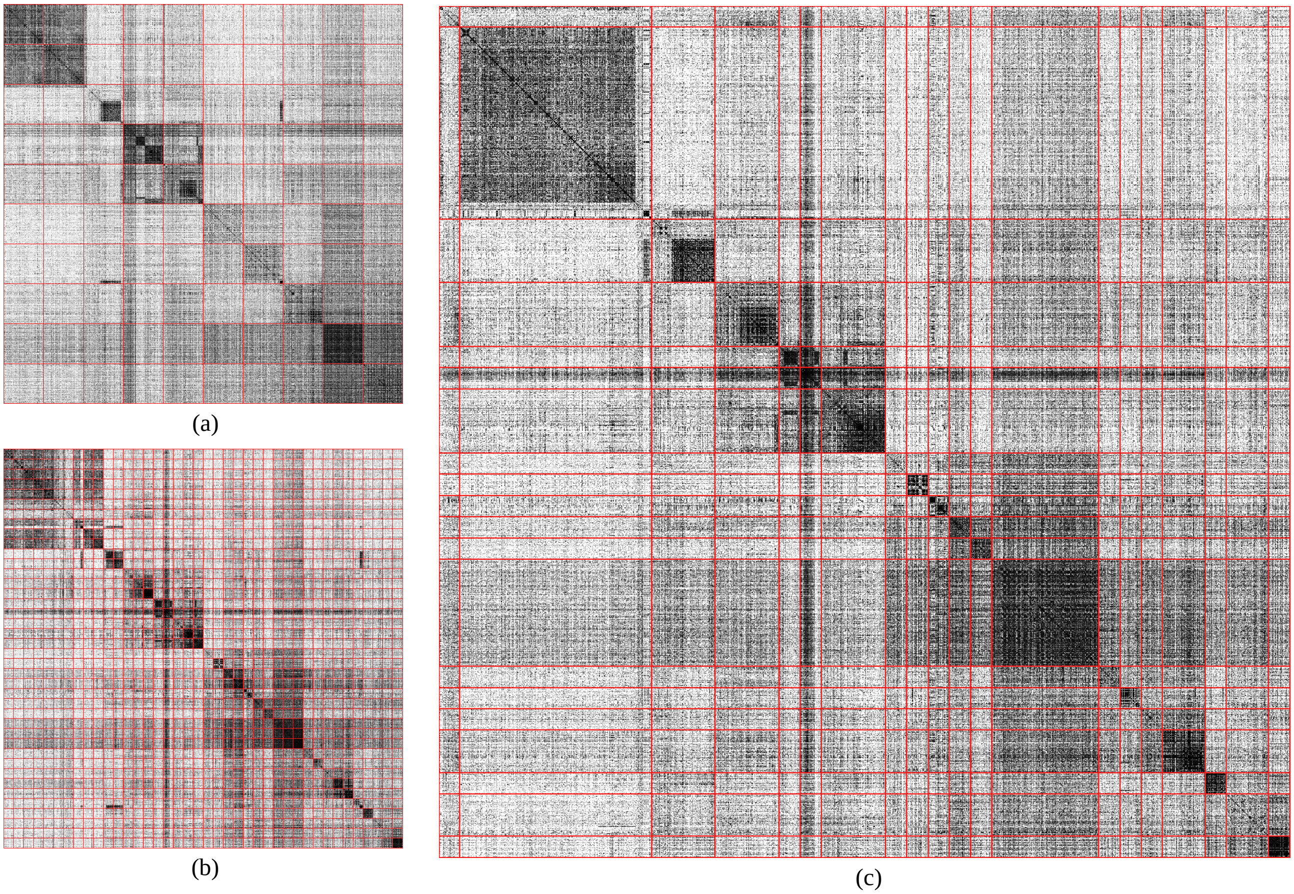
Table 2(left) shows cluster evaluations, their descriptive and discriminative word stems. Each cluster ($\mathcal{C}_\ell$) is evaluated using the dominant category ($\mathcal{K}_{\hat{h}}$), purity ($\Lambda^{(\mathrm{P})}$), and entropy ($\Lambda^{(\mathrm{E})}$). Let $n_\ell^{(h)}$ denote the number of objects in cluster $\mathcal{C}_\ell$ that are classified to be in category h as given by the original YAHOO categorization. Cluster $\mathcal{C}_\ell$'s *purity* can be defined as

$$\Lambda^{(\mathrm{P})}(\mathcal{C}_\ell) = \frac{1}{n_\ell} \max_h (n_\ell^{(h)}). \tag{6}$$

Purity can be interpreted as the classification rate under the assumption that all samples of a cluster are predicted to be members of the actual dominant class for that cluster. Alternatively, we also use $[0, 1]$ *entropy*, which is defined for a problem with g categories as

$$\Lambda^{(\mathrm{E})}(\mathcal{C}_\ell) = -\sum_{h=1}^{g} \frac{n_\ell^{(h)}}{n_\ell} \log \left(\frac{n_\ell^{(h)}}{n_\ell} \right) / \log(g). \tag{7}$$

Entropy is a more comprehensive measure than purity since rather than just considering the number of objects "in" and "not in" the most frequent category, it considers the entire distribution. Table 2(right) gives the complete confusion matrix, which indicates how clusters and categories are related. Note that neither category nor prior distribution information is used during the unsupervised clustering process. In fact, the clustering is very good. It is much better than the original categorization in terms of edge cut and similarity


(a)

(b)

(c)

Figure 5: Comparison of various number of clusters k for YAHOO news data: (a) under-clustering at $k = 10$, (b) over-clustering at $k = 40$, (c) good clustering through interactive split and merge using CLUSION at $k = 20$. See color pictures in soft copy for cluster boundaries.

lift, and it provides a much better grouping when only word frequencies are looked at. The evaluation metrics serve the purpose of validating our results capture relevant categorizations. However, their importance for our purpose is limited since we are solving a clustering problem and not a classification problem. The largest *and* best cluster is cluster 2 with 483 out of 528 documents being from the health cluster. Health related documents show a very distinct set of words and can, hence, be nicely separated. Small and not well distinguished categories have been put together with other documents (For example, the arts category has mostly been absorbed by the music category to form clusters 14 and 16.). This is inevitable since the 20 categories vary widely in size from 9 to 494 documents while the clusters OPOSSUM provides are much more balanced (at least 58 documents per cluster).

7 Conclusion

This paper presents a viable way of visualizing the results of clustering very high ($>$ 1000) dimensional data. The visualization technique is simple but particularly effective because it is applied to the output of a top-down graph partitioning algorithm, which is what the clustering algorithm is converted into via a suitable translation from metrical to similarity space. The visualization toolkit CLUSION allows even non-specialists to get an intuitive visual impression of the grouping nature of objects that may be originally defined in a high-dimensional space. This ability is very important if the tool is to be accepted and applied by a wider community. It also provides a powerful visual aid for assessing and improving clustering. For example, actionable recommendations for splitting or merging of clusters can be easily derived, and readily applied via a point-and-click user interface. It also guides the user towards the "right number" of clusters.

Acknowledgments. We want to express our gratitude to Mark Davis of Knowledge Discovery 1 (since then acquired by Net Perceptions) for providing the drugstore retail data set. This research was supported in part by the NSF under Grant ECS-9000353, and by gifts from Knowledge Discovery 1, Dell, Intel, and Accenture.

References

[1] Michael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: ordering points to identify the clustering structure. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Philadephia, Pennsylvania, USA*, pages 49–60, 1999.

[2] Michael W. Berry, Bruce Hendrickson, and Padma Raghavan. Sparse matrix reordering schemes for browsing hypertext. In *Lectures in Applied Mathematics (LAM)*, volume 32, pages 99–123. American Mathematical Society, 1996.

[3] D. Boley, M. Gini, R. Gross, E. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Partitioning-based

$\mathcal{C}_\ell$	$\mathcal{K}_{\hat{h}}$	$\Lambda^{(\mathrm{P})}$	$\Lambda^{(\mathrm{E})}$	top 3 descriptive terms	top 3 discriminative terms
1	P	21.05	0.72993	israel, teeth, dental	mckinnei, prostat, weizman
2	H	91.48	0.1543	breast, smok, surgeri	symptom, protein, vitamin
3	S	68.39	0.39872	smith, player, coach	hingi, touchdown, rodman
4	P	52.84	0.60362	republican, committe, reform	icke, veto, teamster
5	T	63.79	0.39092	java, sun, card	nader, wireless, lucent
6	o	57.63	0.4013	apple, intel, electron	pentium, ibm, compaq
7	B	60.23	0.47764	cent, quarter, rose	dow, ahmanson, greenspan
8	f	37.93	0.66234	hbo, ali, alan	phillip, lange, wendi
9	cu	50.85	0.47698	bestsell, weekli, hardcov	hardcov, chicken, bestsell
10	p	36.21	0.55982	albert, nomin, winner	forcibl, meredith, sportscast
11	f	67.80	0.32978	miramax, chri, novel	cusack, cameron, man
12	f	77.59	0.3057	cast, shoot, indie	juliett, showtim, cast
13	r	47.28	0.56121	showbiz, sound, band	dialogu, prodigi, submiss
14	mu	44.07	0.56411	concert, artist, miami	bing, calla, goethe
15	p	50.00	0.49932	notabl, venic, classic	stamp, skelton, espn
16	mu	18.97	0.71443	fashion, sold, bbc	poetri, versac, worn
17	p	55.08	0.54369	funer, crash, royal	spencer, funer, manslaught
18	t	82.76	0.2381	household, sitcom, timeslot	timeslot, slot, household
19	f	38.79	0.57772	king, japanes, movi	denot, winfrei, atop
20	f	69.49	0.36196	weekend, ticket, gross	weekend, gross, mimic

	B	E	a	c	cu	f	i	m	mm	mu	o	p	r	s	t	v	H	P	S	T
7	**106**	1	-	4	2	-	30	6	-	4	2	1	-	-	5	2	-	2	-	11
9	-	-	-	3	**30**	17	-	-	1	1	2	2	1	-	1	1	-	-	-	-
8	-	-	1	7	-	**22**	2	-	-	3	1	5	8	1	5	2	-	-	1	-
11	-	-	-	1	-	**40**	1	-	-	-	-	1	2	-	1	13	-	-	-	-
12	-	-	-	2	-	**45**	-	-	-	-	-	2	1	2	4	2	-	-	-	-
19	-	1	-	3	1	**45**	1	-	-	8	-	15	2	-	25	14	-	-	1	-
20	-	1	1	-	-	**41**	-	-	-	4	-	-	-	5	6	1	-	-	-	-
14	-	2	8	-	4	2	-	-	-	**26**	1	12	-	2	1	-	-	1	-	-
16	-	1	4	1	9	9	2	2	1	**11**	-	11	-	-	6	-	-	1	-	-
6	8	-	-	-	-	-	1	-	3	-	**34**	-	-	-	-	1	-	-	-	12
10	-	-	-	3	1	4	-	-	2	2	1	**21**	2	-	20	2	-	-	-	-
15	-	-	2	1	5	13	-	-	-	4	2	**29**	-	-	2	-	-	-	-	-
17	-	1	-	2	6	5	1	6	-	12	1	**65**	3	-	12	4	-	-	-	-
13	-	-	1	1	9	22	6	1	3	33	9	58	**139**	7	2	3	-	-	-	-
18	-	-	1	2	-	1	-	-	-	-	-	2	-	-	**48**	4	-	-	-	-
2	2	-	-	2	1	1	1	-	1	1	3	5	-	-	6	-	**483**	5	17	-
1	3	2	2	1	-	4	-	1	-	4	-	10	-	-	5	-	11	**12**	2	-
4	14	-	4	7	5	2	15	5	-	6	3	6	-	1	12	2	-	**93**	1	-
3	1	-	-	1	1	5	10	-	3	5	-	3	-	-	23	3	-	-	**119**	-
5	8	-	-	3	-	-	-	-	-	1	6	-	-	-	3	-	-	-	-	**37**

Table 2: Cluster evaluations, their descriptive and discriminative terms (left) as well as the confusion matrix (right) for the YAHOO news example (see also Fig. 5(c)). For each cluster number $\mathcal{C}_\ell$ the dominant category $\mathcal{K}_{\hat{h}}$, purity $\Lambda^{(\mathrm{P})}$, and entropy $\Lambda^{(\mathrm{E})}$ are shown.

clustering for web document categorization. *Decision Support Systems*, 27:329–341, 1999.

[4] K. Chang and J. Ghosh. A unified model for probabilistic principal surfaces. *IEEE Trans. PAMI*, 23(1):22–41, Jan 2001.

[5] Chaomei Chen. Visualising semantic spaces and author co-citation networks in digital libraries. *Information Processing and Management*, 35:401–420, 1999.

[6] I. Dhillon, D. Modha, and W. Spangler. Visualizing class structure of multidimensional data. In S. Weisberg, editor, *Proceedings of the 30th Symposium on the Interface: Computing Science and Statistics, Minneapolis, MN, May 13–16 1998*, 1998.

[7] Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95:14863–14868, December 1998.

[8] C. Faloutsos and K. Lin. Fastmap: a fast algorithm for indexing, data mining and visualization of traditional and multimedia datasets. In *Proc. ACM SIGMOD Int. Conf. on Management of Data, San Jose, CA*, pages 163–174. ACM, 1995.

[9] W. Frakes. Stemming algorithms. In W. Frakes and R. Baeza-Yates, editors, *Information Retrieval: Data Structures and Algorithms*, pages 131–160. Prentice Hall, New Jersey, 1992.

[10] J. H. Friedman. An overview of predictive learning and function approximation. In V. Cherkassky, J.H. Friedman, and H. Wechsler, editors, *From Statistics to Neural Networks, Proc. NATO/ASI Workshop*, pages 1–61. Springer Verlag, 1994.

[11] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: a robust clustering algorithm for categorical attributes. In *Proceedings of the 15th International Conference on Data Engineering*, 1999.

[12] John A. Hartigan. *Clustering Algorithms*. Wiley, New York, 1975.

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

[14] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal of Scientific Computing*, 20(1):359–392, 1998.

[15] George Karypis, Eui-Hong Han, and Vipin Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer*, 32(8):68–75, August 1999.

[16] Daniel A. Keim and Hans-Peter Kriegel. Visualization techniques for mining large databases: A comparison. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):932–938, 1996. Special Issue on Data Mining.

[17] B. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell Systems Technical Journal*, 49:291–307, 1970.

[18] T. Kohonen. The self-organizing map. *Proc. IEEE*, 78(9):1464–1480, Sept 1990.

[19] Fionn Murtagh. *Multidimensional Clustering Algorithms*. Physica-Verlag, Heidelberg and Vienna, 1985.

[20] Rajeev Rastogi and Kyuseok Shim. Scalable algorithms for mining large databases. In Jiawei Han, editor, *KDD-99 Tutorial Notes*. ACM, 1999.

[21] Alexander Strehl and Joydeep Ghosh. A scalable approach to balanced, high-dimensional clustering of market-baskets. In *Proc. HiPC 2000, Bangalore*, volume 1970 of *LNCS*, pages 525–536. Springer, December 2000.

[22] Alexander Strehl and Joydeep Ghosh. Value-based customer grouping from large retail data-sets. In *Proc. SPIE Conference on Data Mining and Knowledge Discovery, Orlando*, volume 4057, pages 33–42. SPIE, April 2000.

[23] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of similarity measures on web-page clustering. In *Proc. AAAI Workshop on AI for Web Search (AAAI 2000), Austin*, pages 58–64. AAAI, July 2000.

[24] W.S. Torgerson. Multidimensional scaling, i: theory and method. *Psychometrika*, 17:401–419, 1952.

[25] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, Connecticut, 1983.