Unsupervised Sparsification of Similarity Graphs

Tim Gollub and Benno Stein

Faculty of Media / Media Systems, Bauhaus-Universität Weimar, Germany
<firstname.lastname>@uni-weimar.de

Summary. Cluster analysis technology often grapples with high-dimensional and noisy data. The paper in hand identifies sparsification as an approach to address this problem. Sparsification improves both the runtime and the quality of cluster algorithms that exploit pairwise object similarities, i.e., that rely on similarity graphs. Sparsification has been addressed in the field of graphical cluster algorithms in the past, but the developed approaches leave the burden of parameter tuning to the user. Our new approach to sparsification relies on the inherent characteristics of the data and is completely unsupervised. It leads to significant improvements in the cluster quality and outperforms even the optimum supervised approaches to sparsification that rely on a single global threshold.

Key words: Cluster Analysis, Sparsification, Document Categorization

1 Introduction and Related Work

Cluster analysis deals with the problem of finding natural groups in large sets of data. Extensive discourses on clustering techniques are given in [3, 6, 8, 15]. For the purpose of this paper it is sufficient to distinguish between clustering techniques that are based on a similarity graph versus techniques that are exemplar-based. The contribution of our research is to the former class of algorithms. Figure 1 provides an overview of algorithms that are based on similarity graphs.

To motivate sparsification as a vital part of cluster analysis, consider the conceptual model of a cluster analysis process shown in Figure 2. The similarity graph G of a set of objects $O = \{o_1, o_2, \ldots, o_m\}$ is derived by estimating the similarities between all pairs of objects. Similarities between real-world objects such as documents cannot be assessed directly (unless done by human) but require a model formation or feature extraction step, resulting in a set of object *representations* $X = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_m}\}$. A vector $\mathbf{x_i} \in X$ corresponds to n features of an object o_i and comprises the respective feature weights, i.e., $\mathbf{x_i} = (w_{i_1}, w_{i_2}, \ldots, w_{i_n})^T$. A similarity function $s(\mathbf{x_i}, \mathbf{x_j}) \rightarrow [0, 1]$ is applied to all pairs in X to construct the raw similarity graph G'. If the model formation step is adequate,[1] G' resembles

[1] In the sense of Minsky [11]: $\mathbf{x_i}$ can answer the interesting question about o_i.

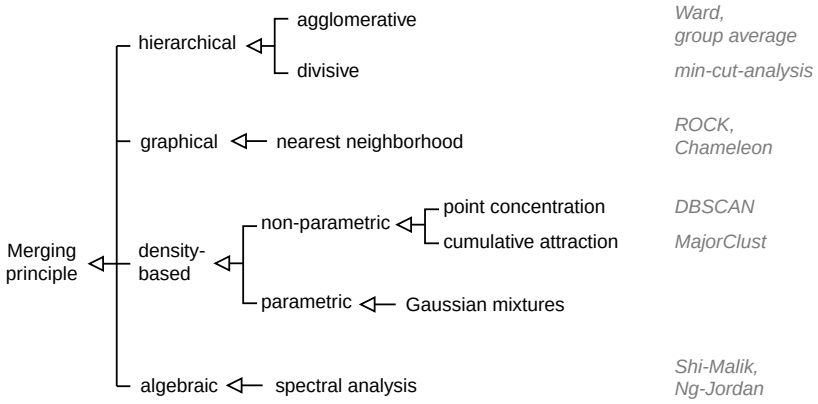

Fig. 1. Taxonomy of cluster algorithms using similarity graphs.

the similarity graph G of the real-world objects O. However, the indirection shown in the upper part of Figure 2 introduces undesired imprecision. For example, imprecision is introduced in the course of feature selection, feature computation, or similarity quantification. Hence the raw similarity graph G' models the similarities between the real-world objects O only approximately. Note that a cluster algorithm takes the similarity scores in G' at face value and runs the risk to make wrong decisions, especially in tie situations. Here sparsification comes into play. By modifying the raw similarity graph G', a smart sparsification obtains a more veritable similarity graph G.

In [9, 10], Kumar and Luxburg report on two major approaches to sparsification. The first one uses a global threshold τ to eliminate all edges with a similarity score below this value. As will be discussed in greater detail in Chapter 2, this approach has its major drawback in disregarding regions of variable density in the object space. The second approach to sparsification is more sensible. It discards all edges of G' that are not among the k strongest edges of a node. Several variants of this nearest neighbor sparsification are discussed in [2, 5, 7]. While nearest neighbor sparsifica-

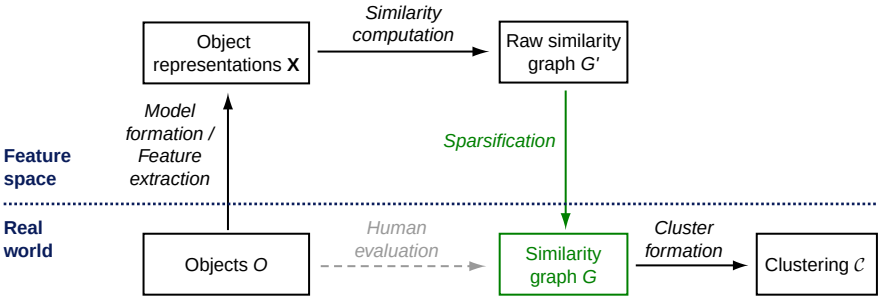

Fig. 2. Cluster analysis. A 4-step conceptual model.

tion longs for different regions in the document space, it comes at the price that the parameter k is application-dependent and has to be chosen carefully.

Our new approach to sparsification adapts itself in an unsupervised manner. It computes an expected similarity score for every edge in the graph G', and only similarity scores surpassing this expectation remain in the thinned-out graph G. To examine the potential of our idea, we conduct two experiments on four collections of text documents. In the first experiment the accuracy of our sparsification technique is compared to the best performing approach that uses a global threshold. In the second experiment raw and thinned-out similarity graphs are analyzed by a density-based cluster algorithm in order to evaluate the gain in clustering quality achieved by sparsification. The results of our experiments are promising in every respect: sparsification increases the quality of the clusterings. Even more, our approach excels in every experiment even the optimum sparsification that relies on a global threshold.

The organization of the paper is as follows. Section 2 gives a definition of sparsification in the context of cluster analysis and presents the new unsupervised sparsification approach. Section 3 reports on the experiments.

2 Sparsification

In the field of computational theory, sparsification is understood as a technique to guarantee a desired time bound when designing dynamic algorithms [1]. In cluster analysis research, improving the efficiency of an approach is of interest as well,[2] but sparsification is also used to enhance the cluster *quality*. Kumar states the goal of sparsification as the "efficient and effective identification of the core points belonging to clusters and subclusters" [9]. This definition, though reasonable, is closely related to the author's approach to graphical clustering. Here we propose a more general definition in the context of cluster analysis:

> *Sparsification is the interpretation of the similarity scores in the feature space in order to enhance the quality and the effort of the cluster formation task.*

Ideally, sparsification switches the similarity scores of edges between two clusters (inter-class edges) to zero, while setting the edge scores within clusters (intra-class edges) to 1. Let $c(o_i) \rightarrow \{1, \ldots, l\}$ assign the true class label to each object $o \in O$. Then, the optimum sparse similarity graph G fulfills the following condition:

$$\varphi(o_i, o_j) = \begin{cases} 1, & \text{if } c(o_i) = c(o_j) \\ 0 & \text{otherwise,} \end{cases}$$

where $\varphi(o_i, o_j)$ denotes the similarity between two real-world objects. The optimum similarity graph is only of theoretical interest since it requires unavailable knowledge about the true class labels. Existing approaches to sparsification work out a notion

[2] E.g., spectral clustering is efficient only with sparse matrices [10].

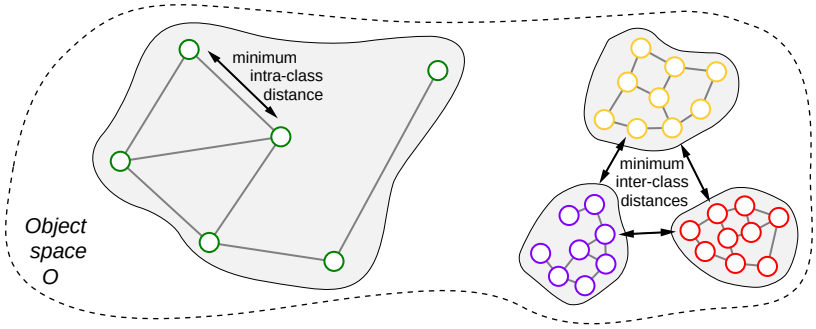

Fig. 3. Two different regions in the object space. The dense clusters on the right-hand side could be thinned-out effectively by applying a threshold reflecting the minimum inter-class distance. Within the cluster on the left, however, this threshold would eliminate all intra-class edges. A convincing result is obtained by constructing the mutual 3-nearest neighbor graph (illustrated by the indicated edges). Every node has at least one intra-class edge; all inter-class edges are discarded.

of probability that two objects belong to the same class. The underlying principle is the nearest neighbor principle. It states that if an object representation $\mathbf{x_1}$ is more similar to a representation $\mathbf{x_2}$ than to another representation $\mathbf{x_3}$, then the probability that $\mathbf{x_2}$ belongs to the same class as $\mathbf{x_1}$ should be higher than the probability that $\mathbf{x_3}$ belongs to the same class as $\mathbf{x_1}$:

$$s(\mathbf{x_1}, \mathbf{x_2}) > s(\mathbf{x_1}, \mathbf{x_3}) \quad \Leftrightarrow \quad P(c(o_1) = c(o_2)) > P(c(o_1) = c(o_3))$$

Upon this supposition several approaches, including ours, have been suggested.

2.1 Existing Approaches

The most common approach to sparsification is the use of a global threshold τ. Every similarity score below the threshold is discarded from the similarity graph. While this approach can be applied efficiently it has two serious drawbacks. First, the threshold's optimum value varies under different sets of objects and has to be found empirically. Second, applying one global threshold does not account for different regions in the object space. As illustrated in Figure 3 one has to cope with clusters where objects are connected much loser compared to other clusters. In such a situation the upper bound for the threshold is determined by the cluster of the lowest density.

The second approach to sparsification relies on the construction of a k-nearest neighbor graph of G'. The k-nearest neighbor graph retains those edges which are among the heaviest k edges of a node (= link to the k nearest neighbors). Several variants of this algorithm exist. The mutual k-nearest neighbor graph is constructed by discarding each edge for which the incident nodes are not among the k nearest neighbors of each other. Another interesting variant is called shared nearest neighbor graph, where the edges of an ordinary k-nearest neighbor graph are weighted according to the number of neighbors the incident nodes have in common. As illustrated

in Figure 3 a k-nearest neighbor graph is able to retain regions of different density in the object space. The main problem is the proper adjustment of the parameter k. And, since the optimum k heavily depends on the (unknown) number and size of the classes, even finding a limiting range of promising choices is difficult. Ertöz et al. state [2]:

> *"The neighborhood list size, k, is the most important parameter as it determines the granularity of the clusters. If k is too small, even a uniform cluster will be broken up into pieces due to the local variations in the similarity [...]. On the other hand, if k is too large, then the algorithm will tend to find only a few large, well-separated clusters, and small local variations in similarity will not have an impact."*

Hence, the construction of a suitable sparse similarity graph requires the generation and evaluation of a huge number of candidates. Note that, more than the runtime, the identification of a sensible internal evaluation measure is the limiting factor in this connection.

2.2 An Object-specific, Unsupervised Approach to Sparsification

Our goal is to provide a completely unsupervised approach to sparsification, while striving for the performance of the existing supervised approaches.To achieve this we claim that two objects in the thinned-out graph G are only allowed to share an edge, if the probability that they belong to the same cluster is high. In particular we propose that the following relation must hold:

$$P(c(o_1) = c(o_2)) \;>\; \max\{P(c(o_1) = c(o_{rand})),\; P(c(o_2) = c(o_{rand}))\},$$

with $o_{rand} \in O \setminus \{o_1, o_2\}$. I.e., the probability that two objects, o_1 and o_2, belong to the same cluster must exceed the probabilities that some randomly drawn object from O belongs to the same cluster as o_1 or o_2. Given this postulation, the nearest neighbor principle is used to establish a relation concerning the similarity scores of the corresponding object representations:

$$s(\mathbf{x_1}, \mathbf{x_2}) \;>\; \max\{s(\mathbf{x_1}, \overline{\mathbf{x}}),\; s(\mathbf{x_2}, \overline{\mathbf{x}})\},$$

where $\overline{\mathbf{x}}$ is a virtual object representation reflecting the characteristics of the object set. It comprises the average weights of all object representations in X:

$$\overline{\mathbf{x}} = (\overline{w}_1, \ldots, \overline{w}_n)^T \quad \text{with } \overline{w}_j = \frac{\sum_{i=0}^{m} w_{i,j}}{m}.$$

If the similarity score of two object representations does not exceed the postulated score, the respective edge is classified as an inter-class edge and is discarded. Altogether, the decision rule $\overline{\varphi}$ for unsupervised sparsification reads as follows:

$$\overline{\varphi}(o_1, o_2) := \begin{cases} s(\mathbf{x_1}, \mathbf{x_2}), & \text{if } s(\mathbf{x_1}, \mathbf{x_2}) \;>\; \max\{s(\mathbf{x_1}, \overline{\mathbf{x}}),\; s(\mathbf{x_2}, \overline{\mathbf{x}})\} \\ 0 & \text{otherwise.} \end{cases}$$

The decision rule above yields convincing results in our sparsification experiments. Nevertheless, cluster algorithms that are extremely sensitive to noise benefit from a more exhaustive sparsification. To account for this, the notion of *significance* is introduced into the formula by modifying the virtual object representation $\overline{\mathbf{x}}$. In the following formula the maximum weight of each feature w_i^* is considered as an upper bound, and the harmonic mean between this bound and the averaged feature weight is computed:

$$\widehat{\mathbf{x}} = (\widehat{w}_1, \ldots, \widehat{w}_n)^T \quad \text{with } \widehat{w}_i = \frac{2 \cdot w_i^* \cdot \overline{w_i}}{w_i^* + \overline{w_i}}.$$

The corresponding stricter decision rule $\widehat{\varphi}$, which accounts for significance, is derived by substituting $\widehat{\mathbf{x}}$ for $\overline{\mathbf{x}}$ in the decision rule $\overline{\varphi}$.

3 Evaluation

To evaluate the performance of our unsupervised approach to sparsification, 4 test collections were constructed from the Reuters news corpus RCV1 [12]. The collections vary with respect to the number of documents, the number of categories, as well as by the way the documents are distributed across the classes (cf. Table 1).

Table 1. Properties of the 4 test collections. Based on the first collection, one attribute at a time is altered in the subsequent collections.

Collection	Categories	Documents	Distribution
1	4	10.000	random
2	4	10.000	uniform
3	4	2.000	random
4	10	10.000	random

The documents are represented using the vector space model with normalized tf-feature-weights [14], having applied Porter stemming and stopword elimination. The similarity between two documents is computed as the dot product of their representations. The nonzero similarity scores are manually divided into intra-class and inter-class scores.

In the first experiment we are interested in the accuracy of our approach. It is specified in terms of the F-measure, $F = \frac{2 \times precision \times recall}{precision + recall}$. While *precision* denotes the proportion of intra-class edges in the thinned-out graph, *recall* is determined by the proportion of intra-class edges retained. The global threshold sparsification that classifies the edges best (= highest F-measure) is identified by an exhaustive search and is compared to the results obtained by our unsupervised approach. The average results of the experiment are shown in Column 4 of Table 2. Our unsupervised sparsification approach with the virtual object $\overline{\mathbf{x}}$ (Row 3) outperforms sparsification with the optimum global threshold (Row 2).

Table 2. Averaged results of the experimental analysis. The first and the second row show the results with the minimum and the optimum global threshold respectively. The third and the fourth row report on our unsupervised approach, employing the virtual documents $\overline{\mathbf{x}}$ and $\widehat{\mathbf{x}}$. Column 4 reports on the F-measure in the first experiment (sparsification task), the rightmost column reports on the quality of the clusterings produced by MajorClust.

Approach	% of retained intra-class edges	% of discarded inter-class edges	F-measure (sparsification)	F-measure (clustering)
$\tau = min$	100.0%	6.0%	0.43	0.23
$\tau^* = 0.075$	59.9%	83.0%	0.59	0.61
$\overline{\varphi}$	66.8%	85.9%	**0.63**	0.68
$\widehat{\varphi}$	37.4%	97.4%	0.50	**0.76**

In the second experiment the thinned-out similarity graphs are given to MajorClust, a representative of the density-based cluster formation paradigm (cf. Figure 1). Here we use the classification-oriented F-measure, described, e.g., in [13], to determine the quality of the resulting clusterings. The average results are shown in the rightmost column of Table 2. The first row serves as a baseline: these values are achieved by applying the maximum global threshold that retains 100% of the intra-class edges. Note that sparsification in general raises the cluster quality. Comparing the different approaches to sparsification, our unsupervised approach with the virtual object $\overline{\mathbf{x}}$ again outperforms the global threshold sparsification. Interestingly, sparsification with the virtual object $\widehat{\mathbf{x}}$, which retains only 37.4% of the intra-class edges but discards 97.4% of the inter-class edges, attains the highest cluster qualities (Row 4).

4 Conclusion

The main contribution of this paper is a new, unsupervised approach to sparsification. We argue that existing cluster analysis technology is over-strained with the amount of noise that is typical for most categorization and classification tasks, e.g., in information retrieval. A preprocessing of the similarity graph in the form of a sparsification step considerably improves the cluster performance.

The outstanding property of the proposed rule is the consideration of the specific similarity distributions within the set of objects, while being parameterless at the same time. Our analysis shows that even sparsification with the optimum global threshold is outperformed. Recall in this context that a comparison to the optimum threshold is only of theoretical interest: in practical applications, cluster analysis happens unsupervised, and the optimum threshold is not at hand. This fact underlines the impact of the proposed strategy.

A still unanswered research question is the performance of our approach in comparison to a k-nearest neighbor approach. A preliminary evaluation of smaller document sets (up to 2000 documents) revealed, that our unsupervised approach to sparsification is as effective as the best performing mutual k-nearest neighbor graph in 86% of 126 different cases [4].

References

[1] P. E. Black. "Sparsification", in dictionary of algorithms and data structures (online). In U.S. National Institute of Standards and Technology, editors, *Algorithms and Theory of Computation Handbook*. CRC Press LLC, 2004. URL `http://www.itl.nist.gov/div897/sqg/dads/HTML/sparsificatn.html`.

[2] L. Ertöz, M. Steinbach, and V. Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *SDM*, 2003.

[3] B. S. Everitt. Cluster analysis. New York, Toronto, 1993.

[4] T. Gollub. Verfahren zur Modellbildung für das Dokumenten-Clustering. Diplomarbeit, Bauhaus-Universität Weimar, Fakultät Medien, Mediensysteme, April 2008. In German.

[5] S. Guha, R. Rastogi, and K. Shim. Rock: A robust clustering algorithm for categorical attributes. In *ICDE '99: Proceedings of the 15th International Conference on Data Engineering*, page 512, Washington, DC, USA, 1999. IEEE Computer Society.

[6] A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: a Review. *ACM Computing Surveys (CSUR)*, 31(3):264–323, 2000.

[7] G. Karypis, E.-H. Han, and V. Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling. Technical Report Paper No. 432, University of Minnesota, Minneapolis, 1999.

[8] L. Kaufman and P. J. Rousseuw. *Finding Groups in Data*. Wiley, 1990.

[9] V. Kumar. An introduction to cluster analysis for data mining. Technical report, CS Dept, University of Minnesota, USA, 2000.

[10] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4): 395–416, 2007.

[11] M. Minsky. Models, Minds, Machines. In *Proceedings of the IFIP Congress*, pages 45–49, 1965.

[12] T.G. Rose, M. Stevenson, and M. Whitehead. The Reuters Corpus Volume 1 - From Yesterday's News to Tomorrow's Language Resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, 2002.

[13] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, 2007.

[14] G. Salton, A. Wong, and C. S. Yang. A Vector Space Model for Automatic Indexing. *Commun. ACM*, 18(11):613–620, 1975.

[15] B. Stein and S. Meyer zu Eißen. Automatic Document Categorization: Interpreting the Perfomance of Clustering Algorithms. In Andreas Günter, Rudolf Kruse, and Bernd Neumann, editors, *KI 2003: Advances in Artificial Intelligence*, volume 2821 LNAI of *Lecture Notes in Artificial Intelligence*, pages 254–266. Springer, September 2003.