A Hierarchical Network Simplification Via Non-Negative Matrix Factorization

Markus Diego Dias*,
Moussa R. Mansour*,
Fabio Dias*‡
*ICMC - Universidade de São Paulo
São Paulo, Brasil

Fabiano Petronetto†
†Departamento de Matemática
Centro de Ciências Exatas
Universidade Federal do Espírito Santo
Espírito Santo, Brasil

Cláudio T. Silva‡,
L. Gustavo Nonato*‡
‡Tandon School of Engineering
New York University
New York, USA

***Abstract*—Visualization tools play an important part in assisting analysts in the understanding of networks and underlying phenomena. However these tasks can be hindered by visual clutter. Simplification/decimation schemes have been a main alternative in this context. Nevertheless, network simplification methods have not been properly evaluated w.r.t. their effectiveness in reducing complexity while preserving relevant structures and content. Moreover, most simplification techniques only consider information extracted from the topology of the network, altogether disregarding additional content.**

In this work we propose a novel methodology to network simplification that leverages topological information and additional content associated with network elements. The proposed methodology relies on non-negative matrix factorization (NMF) and graph matching, combined to generate a hierarchical representation of the network, grouping the most similar elements in each level of the hierarchy. Moreover, the matrix factorization is only performed at the beginning of the process, reducing the computational cost without compromising the quality of the simplification. The effectiveness of the proposed methodology is assessed through a comprehensive set of quantitative evaluations and comparisons, which shows that our approach outperforms existing simplification methods.

I. Introduction

Networks are important structures for modeling systems whose elements bear a pairwise relationship. Typical examples are social networks, where people interact according to their friendship relation; and scientific networks, where researchers are connected according to their level of collaboration. In this context, visualization tools are crucial, revealing gist information w.r.t. group formation and patterns as well as directing the analysts' attention to specific parts of a network.

However, visualization tasks can be impaired on massive networks due to visual clutter. Simplification/decimation has been a main alternative to make large networks manageable in terms of visualization. Indeed, 43 of the 81 methods analyzed for network group structure visualization by Vehlow et al. [1] rely on hierarchical groups to assist visualization tasks.

Despite its importance, network simplification methods have not been properly evaluated for their effectiveness in reducing network complexity while preserving relevant structures and content. Most techniques consider only the topology, disregarding information associated with the elements. Neglecting this information can lead to configurations where groups, and their content, barely reflect the original ones.

We propose a novel methodology for network simplification based on topological information and content associated with network elements, relying on non-negative matrix factorization (NMF) [2] and graph matching [3]. It can be recurrently applied to generate a hierarchical representation of the network, grouping the most similar elements in each level of the hierarchy. The matrix factorization is only performed at the start of the process, reducing computational cost without compromising the quality of the result.

The effectiveness of our methodology is assessed through a set of quantitative measures, where we adapt community formation metrics to gauge the quality of the network simplification, compared against well known techniques, showing that the proposed method outperforms those methods.

In summary, the main contributions of this work are:

- A network simplification mechanism that combines Non-Negative Matrix Factorization and graph matching.
- A hierarchical representation where each level is a meta graph whose similarity between meta nodes is derived directly from the original network, avoiding computationally costly processing at each level of the hierarchy.
- A comprehensive set of comparisons showing the effectiveness of our approach when compared against existing network simplification techniques.

II. Related Work

In order to better contextualize our contribution, we group existing network simplification techniques into two main categories, edge/node removal and edge/node collapse.

Edge/Node removal. These methods remove elements while preserving some information of interest. Betweenness centrality was used by Girvan and Newman [4], [5] to set edge weights and to perform the simplification considering the resulting minimum spanning tree. Conversely, Jia et al. [6] remove edges with smaller centrality values in order to preserve paths between groups. Sensitivity can be defined as the derivative of the centrality, mostly used to indicate variations such as the addition or removal of elements [7], or to create the minimum spanning tree for simplification [8].

Edges/nodes removal methods are not appropriate for certain applications, mainly when the node connectivity is relevant for the analysis. Moreover, they consider only the topology, disregarding additional content associated with network edges and nodes, constraining their usefulness.

Edge/Node collapse. These methods group nodes based on a similarity or optimality, where simplified representations have meta nodes connected by edges. Phrase Nets [9] groups nodes with identical neighborhood structures, keeping an edge between groups whose nodes are connected in the original network. Dinkla et al. [10] proposed a similar approach, in the context of gene network analysis, grouping nodes with identical neighbor set. As most edge/node removal techniques, these methods rely only on the topological structure of the network, neglecting possible attributes associated with nodes and edges. Power graph analysis [11], [12], [13] also perform graph simplification based on the nodes neighborhood.

However, there are methods that collapse elements based on optimality criteria or on the content of the elements in the network. The method proposed by Newman [14] is a typical example, where edges are collapsed to minimize a modularity function. Spectral clustering [15] is widely used to partition a network while preserving a balance between the cost of cutting edges and the size of the partitions. The K-SNAP technique [16] partitions a network to maximize an inter-cluster cost function, which takes into account attributes associated with the nodes. Based on the K-SNAP technique, Zhang et al. [17] proposed a scheme to sort nodes according to their attributes, performing the partition based on the resulting order. Minimal Description Length was used by Navlakha et al. [18] to generate a simplified representation from which the original network can be recovered with minimal cost.

Graph matching was also used, providing a natural hierarchy by its recurrent application, as exploited by Karpis and Kumar [19] to build a hierarchy from binary partitions. It has been an important mechanism to create hierarchical representations, including graph drawing [20], [21]. However, most matching-based methods rely only on the topology of the network.

Matrix factorization has been used to consider attributes when performing the simplification. Wang et al [22] use NMF to define similarity between nodes. Vegas [23] uses SymNMF [24] to summarize citation networks. Multivis [25] uses e-mail information as attributes to generate a tensor, that is decomposed to simplify the network. While NMF is effective to define groups of similar entities, it was not properly explored to generate hierarchical representations where similarity among elements is enforced in each level of a hierarchy.

Edge/node removal and collapsing for visualization. Although visual resources are alternatives, such as edge bundling [26], edge/node removal and collapse approaches remain essential to visualization tasks, as shown by Auber et al. [27], enabling the visual exploration of a hierarchical network, constructed by removing the weakest edges. Edge strength is obtained from the topology and the hierarchy is set by thresholding the edge strength at different levels. Topological information was also used in Ask-Graph View [28], relying on clustering to build a hierarchy. Topological Fisheye [29] combines topological and geometrical information, the latter obtained by embedding the network into a Cartesian visual space, to hierarchically simplify a network through edge collapses computed from a graph matching mechanism.

General mechanisms to assist network visualization have been proposed by OnionGraph [30] and Pivot Graph [31], considering the topology and node attributes to define similarities among nodes, allowing semantic aggregation during simplification. Node attributes are also considered by Elmqvist et al. [32], which build hierarchies composed of meta nodes representing similar nodes. Nevertheless, none of those attribute-aware techniques consider NMF methods to guide the construction of the hierarchy. Most hierarchical visualization techniques assume the hierarchy is given, performing only the visualization [33], [34]. Further, graph matching techniques perform the collapses based only on topological information.

Our approach is a step towards filling this gap between topology and content, by combining NMF and graph matching to produce hierarchical representations where similarities are preserved. It differs from the existing techniques, as it builds the hierarchy from a combination of NMF and graph matching. This combination leverages the solid mechanism to use the attribute information to define similarities between nodes in each level of the hierarchy from the NMF, with optimal collapses/grouping from the graph matching, enabling a reliable representation for visualization purposes.

III. Matrix Factorization and Graph Hierarchies

Let $G = (V, E, X)$ be a network, where V is the set of nodes, E is the set of edges, and X is a matrix where each column corresponds to attributes associated with a node. A hierarchical representation of G is built, where meta nodes at each level represent (meta)nodes from the previous level. The hierarchy is built by collapsing edges connecting similar (meta)nodes. Each collapse leads to a new meta node representing a pair of collapsed (meta)nodes. Three main issues are addressed when building each level of the hierarchy:

1) How to measure the similarity between adjacent nodes? We use the NMF to define this similarity, considering the additional information provided by the matrix X.
2) How to decide which edges must be collapsed? We use Graph Matching to find a set of edges to be collapsed, allowing for multiple collapses simultaneously.
3) How to merge the content of the collapsed (meta)nodes? The content of the resulting meta node is derived from the NMF, rather than from the original data.

As illustrated in Figure 1, the three main steps of our methodology are recurrently employed to define a hierarchical representation for the network.

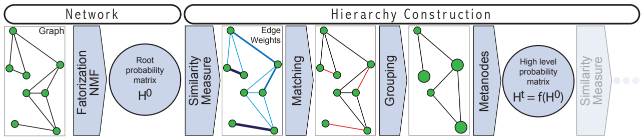


Fig. 1. Overview of our method.

A. Matrix Factorization

NMF. [2] Assuming that all entries of a matrix X are greater or equal to zero, a non-negative factorization of X is a decomposition of the form $X \sim W \cdot H$, where the product $W \cdot H$ approximates X. All entries of both W and H are also positive scalars or zero. The factorization is computed by an alternating constrained optimization scheme that finds the minimum of the expression:

$$\min_{W,H} \|X - WH\|_F^2 \tag{1}$$

subject to the non-negativity constraint for W and H. If X has dimensions $m \times n$, W and H will have dimensions $m \times k$ and $k \times n$, where k is a user defined parameter often set $k << min(m, n)$ (as we want to reduce the size of the dimension), and $\|\cdot\|_F$ is the Frobenius norm.

Since the NMF only uses additive operations, the columns of W are prone to be formed by pieces of information that approximate X when combined by H. The pieces in W lead to an easier interpretation of X [2]. A typical example is topic modeling, where each entry x_{ij} in X corresponds to the number of times a word i appears in the document j. Since the decomposition $X \sim W \cdot H$ expresses the columns of X in the basis given by the columns of W, each column w_s can be interpreted as a topic associated with the documents represented in the columns of X. The larger the value w_{is} the more important the word i is for the topic.

From the matrix H one can identify the important basis vectors (topics) to represent the content of each node. Since the j-th column of H corresponds to the j-th column of X (the j-th node in the network G) and that the s-th entry h_{sj} multiplies the s-th column of W; the larger the value of h_{sj} the more relevant w_s is to representing the content of node j. In terms of topic analysis, h_{sj} indicates the relevant topics for each document. Therefore, similar nodes can be found by identifying similar columns in H.

Dividing each column of H by the sum of its entries makes possible to interpret h_{sj} as the probability of the node j be represented by column w_s, that is, the probability of x_j belongs to topic w_s. We call h_j the *probability vector*, resorting to this notion of probabilities to define the similarity between nodes (see section III-B).

GNMF. Although effective in several contexts, the classical NMF formulation as described in Equation 1 has several pitfalls. For instance, it does not take into account neighborhood relations, so distinct basis vectors can represent the content of neighbor nodes, what is not expected, since neighbor nodes should share similar properties. Neighborhood relation can be incorporated in the decomposition through regularization terms. The Graph Regularized Non-negative Matrix Factorization (GNMF) [35] proposes the following cost function:

$$\min_{W,H} \left(\|X - WH\|^2 - \lambda \, \mathbf{Tr}(WLW^t) \right) \tag{2}$$

subject to the non-negativity of W and H. In the regularization term on the right, $\mathbf{Tr}$ is the trace of a matrix and L is the Laplacian matrix of G. The Laplacian matrix is given by $L = D - A$, where A is the (weighted) adjacency matrix of G and D is a diagonal matrix with entries d_{ii} equal to the sum of the elements in the i-th row of A. The parameter $\lambda \geq 0$ (smoothness parameter) controls the importance of the regularization in the optimization process.

SymNMF. It can be more convenient to deal with a weighted $n \times n$ adjacency matrix A than the data matrix X. Entries in the adjacency matrix A correspond to the similarity between nodes; each non-zero entry a_{ij} in A corresponds to the similarity between nodes i and j from G. Therefore, the weighted adjacency matrix A can be defined based on application dependent similarity measures, what enables great flexibility to the analysis. A popular construction is $A = X^\top \cdot X$, which corresponds to measuring similarity based on the dot product between the attribute vectors of each node.

The Symmetric Non-negative Matrix Factorization (SymNMF) [24] has as cost function the following expression:

$$\min_{H} \left\| A - H^t H \right\|_F^2 \tag{3}$$

SymNMF can be more easily interpreted in the context of clustering, where the largest entry h_{ij} of column h_j indicates that node j belongs to cluster i.

B. Node Similarity

Each column h_j of the matrix H is a k dimensional vector, since H is $k \times n$, with entries h_{ij} corresponding to the probability of the attribute vector associated with node j being well represented by the basis vector w_i. Given two nodes j and s, and their corresponding vectors h_j and h_s in H, we say those two nodes are similar if their joint probability $p(h_j, h_s) = \sum_{i=1}^{k} h_{ij} h_{is}$ is close to 1. Since the joint probability can be written as

$$\sum_{i=1}^{k} h_{ij} h_{is} = < h_j, h_s > = \|h_j\| \|h_s\| \cos(h_j, h_s) \tag{4}$$

the similarity between two nodes is given by the cosine of the angle between the corresponding columns in H, multiplied by their norms. Therefore, we assign the cosine of the angle between probability vectors h_j and h_s as weight to the edge connecting the nodes j and s. Collapsing edges with large weights corresponds to merging highly similar nodes.

Edges could be collapsed one at a time, simplifying the network gently in each step. However, this procedure does not correspond to what is usually expected of a hierarchical mechanism, which should significantly simplify or expand the representation between adjacent levels. To this end, we collapse edges based on a graph matching scheme.

C. Matching and grouping

There are advantages in edge collapsing based on graph matching. Matching typically finds a large number of edges to be collapsed simultaneously, enabling a true decimation procedure. Moreover, the collapse of an edge does not conflict with the others, so the procedure is computationally simpler.

A subset of edges $M \subset E$ is a matching in G if no two edges in M are adjacent, that is, edges in M do not share a common node. This property guarantees that edges in M can be collapsed without conflicts. A matching M is called maximal if there is no other matching M' such that $M \subset M'$. Given a weighted set of edges E, where $w(e)$ is the weight associated with the edge $e \in E$, let $C(M) = \sum_{e \in M} w(e)$ be the total cost of a matching M, and $\mathcal{M}$ be the collection of all matchings on G. A matching $M \in \mathcal{M}$ is a maximum weighted matching (MWM) if $C(M') \leq C(M)$ for every $M' \in \mathcal{M}$.

Ideally, the collapses should be based on a maximum weighted matching (MWM). However, MWM is costly, only suitable for small networks. For larger networks, we propose a greedy approximation to the MWM, the *sorted maximal matching* (SMM), computed by sorting the edges in E in descending order of weights; then a matching set M is built by adding edges to M in the sorted order. If an edge to be added is incident to an edge already in M then it is discarded and the next edge in the sorted list is considered. The process follows until all edges are considered.

The SMM is not guaranteed to be maximal nor of maximum weight. However, it always includes the edge with the largest weight in the matching list, thus ensuring that the two most similar nodes will always be collapsed in each step of the hierarchy construction.

Filtering out low weight edges. Ideally the matching set M would have the largest number of edges. However, there are cases where non-conflicting edges could be added to M, but their small weight indicates low similarity between (meta)nodes. To avoid that issue, we filter out edges according to a threshold δ before creating the matching set.

D. Hierarchy

We adopt superscript indices to represent levels of the hierarchy, j^t corresponds to a (meta)node in the t-th level of the hierarchy, $t = 0$ is the original network. We denote by $|j^t|$ the number of nodes from the original graph merged into j^t. Since the column h_j of H corresponds to the column x_j in the attribute matrix X (1), we define a new probability matrix H^t with columns given by:

$$h_j^t = \frac{1}{|j^t|} \sum_{s \in j^t} h_s \qquad (5)$$

In other words, h_j^t is the average of the columns in H corresponding to nodes in j^t. Entries in h_j^t can also be interpreted as probability vectors. Our merging mechanism avoids repeated computations of the NMF for each level, which would be computationally demanding and more difficult to interpret, since the basis vectors (topics) in W would change in each level. Therefore, the proposed methodology renders the hierarchical construction computationally viable and mathematically sound.

IV. Datasets and Metrics

The effectiveness of our methodology is assessed using six networks with multivariate data associated with the nodes. Those datasets, jointly with four quality metrics, are used to evaluate and compare our approach against other methods.

A. Datasets

Artificial. This network has 30 nodes, evenly distributed in 6 classes, designed to have more edges within classes than between classes. Each node corresponds to a document represented by a word-frequency vector. Nodes in the same classes share the same non-empty entries in the word-frequency vector, with different values. Moreover, word-frequency vectors from different classes are orthogonal, nodes from different classes have no similarity. Figure 2 depicts a node-link visualization of this dataset and three levels of the hierarchy.

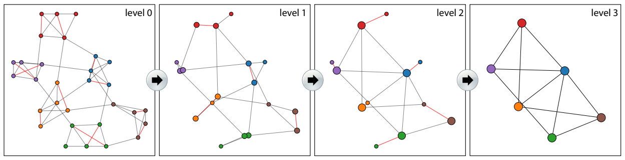


Fig. 2. A node-link visualization of the Artificial Network hierarchy. The meta node size represents the number of the nodes in the finer levels grouped in this. Edges that were matched and collapsed in each step was colored red. The nodes are colored on a bottom-up way.

College Football. Proposed by Givan and Newman [5], it provides information about the game table of a College Football Division in 2000. It has 115 nodes, each representing a team, divided into 12 conferences. Teams from the same conference tend to face each other more often than teams from distinct conferences, with the exception of a conference. The network is constructed by creating edges between nodes (teams) that face each other in the season.

We associate multivariate data to each node i by creating a feature vector x_i with dimension 115. Each entry x_{ij} stores the number of times the team i played against the team j. Since the teams are divided in 12 conferences, we aim to build a hierarchical representation where the coarser level contains exactly 12 nodes (figure 3).

VIS Conference. The VIS Conference dataset [36] contains information of papers published at the IEEE VAST, InfoVis, and SciVis conferences. Each author is represented by a node,

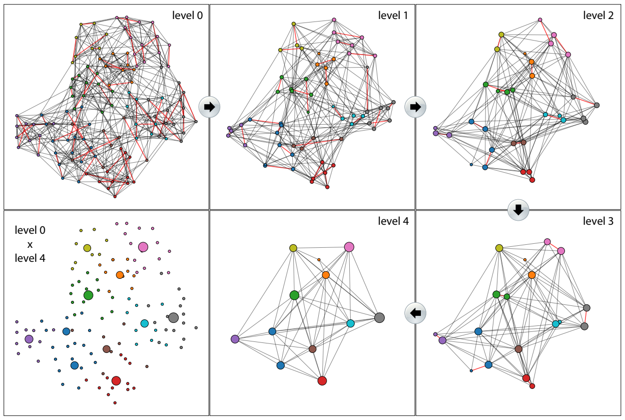


Fig. 3. College Football Network and the five levels of the hierarchical representation from our approach (clockwise from top left). The nodes are colored on a bottom to top way.

connected if the authors collaborated in at least once. The node information is derived from the titles of the papers authored, as a term-frequency matrix. We consider only the largest connected component, with 966 nodes and 4,323 edges.

Movies. The Movies Dataset contains information about movies from the Movie Database (TMDb). Each movie is represented as a node, and two nodes are connected if the movies have at least one actor in common. We considered two different sets of attributes, one is a genre × movies matrix, and the other keywords × movies.

MovieLens. MovieLens is a research site of the University of Minnesota. It helps users to find movies they like, rate movies, and build profiles. Each user is represented as a node and users are linked if they rated the same movie with maximum score (5), resulting in a network with 478 nodes and 39,991 edges. The multivariate information associated with the nodes is generated from the ratings. Specifically, X is a movies×users matrix, where each column corresponds to the rate a particular user gave to the movies.

B. Metrics of Validation

The effectiveness of our methodology is assessed by three different metrics from other works, modularity, Δ–Measure, and K-Way Ratio Cut Cost Metric. These three metrics quantify the quality of the clusters on a graph. In our tests, each meta node in the coarser level is considered as a cluster comprising nodes from the original network, allowing the use of these metrics.

Modularity. It was used by Newman [14] and Wang et al. [22] to validate their simplification methods. Networks with high modularity have dense connections within defined groups and sparse connections among different groups. Assuming that the nodes are labeled according their group, let e_{ij} be the fraction of edges connecting nodes from group i to group j and $a_i = \sum_j e_{ij}$. The modularity Q is defined as:

$$Q = \sum_i (e_{ii} - a_i^2) \tag{6}$$

where $Q = 0$ indicates random groupings and $Q = 1$ indicates the maximum modularity, created by well structured groups.

Δ–Measure. The Δ-measure [16], [17] assesses the quality of group formation by measuring pairwise relationships between the groups. Let $\Phi = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_k\}$ be a partition of the nodes from G such that $\mathcal{G}_i \cap \mathcal{G}_j = \emptyset$, for all $i \neq j$, and

$$P_{\mathcal{G}_j}(\mathcal{G}_i) = \{u |\ u \in \mathcal{G}_i \text{ and } \exists\ v \in \mathcal{G}_j \text{ s.t. } (u,v) \in E\}. \tag{7}$$

Making $p_{i,j} = (|P_{\mathcal{G}_j}(\mathcal{G}_i)| + |P_{\mathcal{G}_i}(\mathcal{G}_j)|)/(|\mathcal{G}_i| + |\mathcal{G}_j|)$ we define the Δ-measure as:

$$\Delta(\Phi) = \sum_{\mathcal{G}_i, \mathcal{G}_j \in \Phi} = (\delta_{\mathcal{G}_j}(\mathcal{G}_i) + \delta_{\mathcal{G}_i}(\mathcal{G}_j)) \tag{8}$$

where,

$$\delta_{\mathcal{G}_j}(\mathcal{G}_i) = \begin{cases} |P_{\mathcal{G}_j}(\mathcal{G}_i)| & \text{if } p_{i,j} \leq 0.5 \\ |\mathcal{G}_i| - |P_{\mathcal{G}_j}(\mathcal{G}_i)| & \text{otherwise} \end{cases} \tag{9}$$

To obtain the average contribution of the groups, we divide it by k. The smaller the result, the better the group formation.

This definition does not account for edge weights, so we apply Equation (8) only on edges with weight greater than zero. We call this version the modified Δ–measure.

K-Way Ratio Cut Cost Metric. The K-Way Ratio Cut Cost Metric [15] measures the cost of a graph cut generating a k-way partition $\Phi = \{\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_k\}$. Let E_h be the sum of the weights of the edges with exactly one end in $\mathcal{G}_h$. The cut cost can then be defined as:

$$cost(\mathcal{G}) = \sum_{h=1}^{k} \frac{E_h}{|\mathcal{G}_h|} \tag{10}$$

Similarly to the Δ–measure, the smaller the K-Way Ratio Cut Cost the better the partition.

V. Results and Comparisons

In this section we present a comprehensive set of experiments and comparisons to validate and show the effectiveness of the proposed methodology. We divide the experiments in three main groups:

1) Assessment of the proposed graph matching process,
2) Evaluation of the mechanism employed to measure the similarity between (meta)nodes, which define weights to the edges, and
3) Comparison against existing simplification techniques.

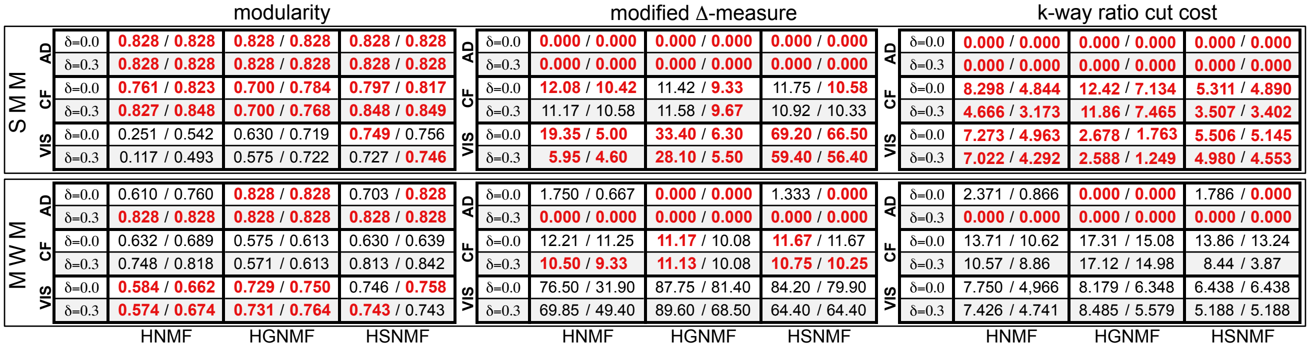
Factorization methods denoted with capital H (HNMF, HGNMF, HSNMF) indicate that they are employed as described in our methodology. Factorization methods without the capital H correspond to the traditional methods (NMF, GNMF, SNMF). In these methods, a node j will belong to the cluster i if h_{ij} is the largest entry in h_j, as is usual with NMF-based data clusterization [22], [24]. Therefore, in traditional methods, no hierarchical structure is obtained, and the simplification is accomplished in a single step. As previously stated, the Artificial and College Football datasets were simplified until 6 and 12 meta nodes were obtained in the coarser level. The other datasets were simplified to 10 meta nodes in the coarser level.

A. Graph Matching and Filtering

To evaluate the effectiveness of the our matching, we compare it with MWM. We also assess the impact of ignoring edges with small weights from process (SMM$_f$ and MWM$_f$).

The tables in Figure 4 show the quality of the simplified networks for the Artificial (AD), College Football (CF) and VIS (VIS CCN) datasets, with and without filtering the edges. The tables in Figure 4 also show the quality of the simplification, using three different factorizations, HNMF, HGNMF and HSNMF. Different factorization schemes generate different H matrices, thus affecting the edge weights and matching.

According to the modularity measure, the proposed SMM mechanism resulted in better quality simplifications for the AC and CF datasets, regardless of the factorization method. In the VIS dataset, the MWM was superior, although SSM+SMNF



			modularity			modified Δ-measure			k-way ratio cut cost		
			HNMF	HGNMF	HSNMF	HNMF	HGNMF	HSNMF	HNMF	HGNMF	HSNMF
SMM	AD	δ=0.0	**0.828 / 0.828**	**0.828 / 0.828**	**0.828 / 0.828**	**0.000 / 0.000**	**0.000 / 0.000**	**0.000 / 0.000**	**0.000 / 0.000**	**0.000 / 0.000**	**0.000 / 0.000**
		δ=0.3	**0.828 / 0.828**	**0.828 / 0.828**	**0.828 / 0.828**	**0.000 / 0.000**	**0.000 / 0.000**	**0.000 / 0.000**	**0.000 / 0.000**	**0.000 / 0.000**	**0.000 / 0.000**
	CF	δ=0.0	**0.761 / 0.823**	**0.700 / 0.784**	**0.797 / 0.817**	**12.08 / 10.42**	11.42 / **9.33**	11.75 / **10.58**	**8.298 / 4.844**	**12.42 / 7.134**	**5.311 / 4.890**
		δ=0.3	**0.827 / 0.848**	**0.700 / 0.768**	**0.848 / 0.849**	11.17 / 10.58	11.58 / **9.67**	10.92 / 10.33	**4.666 / 3.173**	**11.86 / 7.465**	**3.507 / 3.402**
	VIS	δ=0.0	0.251 / 0.542	0.630 / 0.719	**0.749** / 0.756	**19.35 / 5.00**	**33.40 / 6.30**	**69.20 / 66.50**	**7.273 / 4.963**	**2.678 / 1.763**	**5.506 / 5.145**
		δ=0.3	0.117 / 0.493	0.575 / 0.722	0.727 / **0.746**	**5.95 / 4.60**	**28.10 / 5.50**	**59.40 / 56.40**	**7.022 / 4.292**	**2.588 / 1.249**	**4.980 / 4.553**
MWM	AD	δ=0.0	0.610 / 0.760	**0.828 / 0.828**	0.703 / **0.828**	1.750 / 0.667	**0.000 / 0.000**	1.333 / **0.000**	2.371 / 0.866	**0.000 / 0.000**	1.786 / **0.000**
		δ=0.3	**0.828 / 0.828**	**0.828 / 0.828**	**0.828 / 0.828**	**0.000 / 0.000**	**0.000 / 0.000**	**0.000 / 0.000**	**0.000 / 0.000**	**0.000 / 0.000**	**0.000 / 0.000**
	CF	δ=0.0	0.632 / 0.689	0.575 / 0.613	0.630 / 0.639	12.21 / 11.25	**11.17** / 10.08	**11.67** / 11.67	13.71 / 10.62	17.31 / 15.08	13.86 / 13.24
		δ=0.3	0.748 / 0.818	0.571 / 0.613	0.813 / 0.842	**10.50 / 9.33**	**11.13** / 10.08	**10.75 / 10.25**	10.57 / 8.86	17.12 / 14.98	8.44 / 3.87
	VIS	δ=0.0	**0.584 / 0.662**	**0.729 / 0.750**	0.746 / **0.758**	76.50 / 31.90	87.75 / 81.40	84.20 / 79.90	7.750 / 4,966	8.179 / 6.348	6.438 / 6.438
		δ=0.3	**0.574 / 0.674**	**0.731 / 0.764**	**0.743** / 0.743	69.85 / 49.40	89.60 / 68.50	64.40 / 64.40	7.426 / 4.741	8.485 / 5.579	5.188 / 5.188

Fig. 4. From top to bottom, quality measurements (median/best values) from the modularity function, modified Δ–measure, k–way Ratio Cut Cost Metric applied to the Artificial, College Football, and VIS datasets.

obtained similar results. Regarding the Δ–measure and K-way Ratio Cut, where smaller values are better, the proposed SMM clearly created simplifications of better quality in most of the cases, regardless of the factorization method. Notice that the filtering mechanism tends to further improve the simplification, with a few exceptions. Although not shown, similar results were found in the other datasets.

B. Node Similarity and Edge Weights

The similarity measure between nodes is essential for a good simplification. Our approach relies on the NMF and on the cosine of the angle between the probability vectors, but there are other measures that could be used to gauge the similarity between nodes, some of which present lower computational cost. In order to show that this adopted combination is a viable alternative, we compare it against other four mechanisms to measure the similarity between nodes. The node similarity is used only to assign edge weights, running the simplification process according to the proposed pipeline (Figure 1).

Neighborhood Similarity. Let N_i be the set of nodes in the neighborhood of a node i, the neighborhood similarity between nodes i and j is given by:

$$\mathcal{N}(i,j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|} \tag{11}$$

This topology-based measure [29] disregards node attributes.

Clustersize Similarity. Clusters tend to have groups with a uniform number of elements. The *clustersize* similarity between two meta nodes j_1^t and j_2^t is given by:

$$\mathcal{C}(j_1^t, j_2^t) = 1 - \frac{|j_1^t| + |j_2^t|}{\max\limits_{j_i^t, j_k^t \in V^t} \{|j_i^t| + |j_k^t|\}} \tag{12}$$

where $|j^t|$ is the number of nodes contained in the meta node j^t and V^t is the set of (meta)nodes in the t-th level. This is topology-based measure [29] that ignores node attributes.

Degree Similarity. Let d_i be the degree of node i. The degree similarity is given by:

$$\mathcal{D}(i,j) = 1 - \frac{d_i \cdot d_j}{\max\limits_{r,s \in V} \{d_r \cdot d_s\}} \tag{13}$$

This is also a topology-based metric.

Cosine without NMF. To investigate if the NMF-based probability vector indeed brings benefits of the similarity between nodes, we considered it directly to the attribute vectors:

$$\text{cosine}(i,j) = \cos(x_i, x_j) \tag{14}$$

where x_i and x_j are the attribute vectors of i and j.

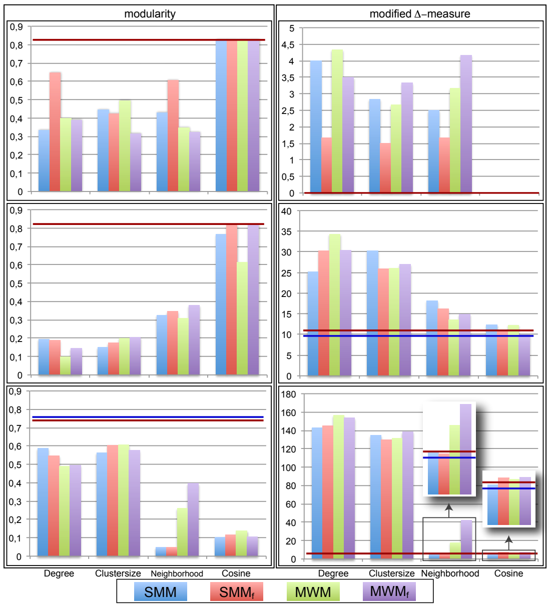


Fig. 5. Comparison against other node similarity measures. The horizontal lines are the median (red line) and the best result (blue line) of our method with the SMM approach (Figure 4). The best result is the bigger on the modularity graphics and the smaller on the Δ–measure graphics. For the Artificial Dataset the median and best are the same in each metric of validation.

Figure 5 shows the result of using these measures to compare nodes and assign weight to edges. The horizontal bars correspond to the proposed approach (matrix factorization + cosine of probability vectors) best results (red) and best median (blue) in the SMM. We also evaluate all the measures with MWM to show that our method with SMM also performs better than other measures with MWM. Our measure outperformed or was similar to the best results presented by other metrics, in all cases and datasets.

C. Comparison Against other Simplification Techniques

We conclude the tests with a set of comparisons against existing network simplification methods. More specifically, we compare our approach against the simplification techniques proposed by Newman [14] and Chan et al. [15].

The comparisons use the quality metrics described in Section IV, also employed by Newman [14] and Chan et al. [15], where the proposed methods were optimized for their respective metrics. We are using them to assess the three methods, leading to fair comparison.

Results obtained from the comparisons are summarized in Figure 6, where each row corresponds to a quality metric and each column to a dataset. Since our approach uses NMF with a random initial condition, we ran each configuration 20 times, assembling quality measure results in the box plots depicted in the figure. Each column in Figure 6 is divided into four color strips, one for each factorization method. Each color bar contains two box plots, which corresponds to the proposed SMM scheme without (left in each color strip) and with (right in each color strip) the edge weight filtering mechanism. The brownish color strips (rightmost color strip in each column) correspond to the results of the traditional NMF clustering; clusters of nodes are computed directly from the factorized matrices in a single step. Since the methods by Newman [14] and Chan et al. [15] are deterministic, their quality measures are shown as horizontal lines in Figure 6.

For the Artificial dataset, composed of nodes distributed in six classes, the simplification was accomplished until six meta nodes remain in the coarser level. Similarly, the other methods were configured to generate six clusters in the final representation. Our approach was superior to Newman's method in all metrics, with the best results reached by HGNMF. Chan's method presented a tendency to obtain better results in this dataset, although, considering the median behavior of our approach, both perform quite similarly.

Regarding the other comparisons, varying the metrics and datasets, the HGNMF version of our approach outperformed Newman's method in 19 out of 20 comparisons. Considering the performance of our approach regardless of the factorization and filtering mechanism, we obtained better results in all datasets when compared to Newman's method. The HGMNF variant of our approach outperformed Chan's method in 11 out of 20 comparison. Considering all possible factorizations, our approach was superior in 16 out of 20 comparisons.

Generally speaking, the HSNMF variation has better performance when the modularity metric is used, while the HGNMF tends to be superior according to the other metrics. Comparing the HNMF, HGNMF, and HSNMF against their traditional counterpart, i.e., NMF, GNMF, and SNMF, we clearly show that the hierarchical variants presented better results in all the cases, except for the College Football dataset, where the traditional non-negative factorization performed slightly better.

VI. Discussion and Limitations

The proposed methodology obtained good results in most of the experiments and evaluations considered, showing our choices as to factorization procedure, edge weight assignment, graph matching, and merging of attributes worked in unison.

Our methodology outperformed the methods of Newman [14] and Chan et al. [15] in most cases, even though those two methods have been conceived to optimize some of the metrics we used in the comparisons.

A limitation of our method occurs when the network has the topology of a star graph. In this configuration, our approach can only collapse one edge at a time.

VII. Conclusion

We presented a new method to create a hierarchical clustering on a network with multivariate information on the nodes. With the NMF resulting matrices, we can group the network and classify the nodes on each pass of the hierarchy using the topics determined by the NMF. To choose the nodes that we group together, we created a similarity measure based on the results of the NMF that measures how much two nodes should be on the same cluster. To validate our similarity measure based on NMF, we compared our results with others similarity measures based on topology and based on the multivariate information. To validate our method, we compared our hierarchical clustering with the traditional NMF clustering and with other classical network clustering methods.

Acknowledgment

Markus acknowledges CAPES for the financial support. FAPESP Grants 2011/22749-8, 2014/12815-1, 2013/07375-0, 2016/04190-7, 2016/04391-2. The views expressed do not reflect the official policy or position of FAPESP.

References

[1] C. Vehlow, F. Beck, and D. Weiskopf, "The state of the art in visualizing group structures in graphs," in *EuroVis - STARs*, 2015.

[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, 1999.

[3] J.-A. Bondy and U. S. R. Murty, *Graph theory*, ser. Graduate texts in mathematics. New York, London: Springer, 2007.

[4] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review*, vol. E 69, no. 026113, 2004.

[5] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *National Academy of Sciences*, 2002.

[6] Y. Jia, J. Hoberock, M. Garland, and J. Hart, "On the visualization of social and other scale-free networks," *IEEE TVCG*, vol. 14, no. 6, 2008.

[7] C. D. Correa, T. Crnovrsanin, and K.-L. Ma, "Visual reasoning about social networks using centrality sensitivity." *IEEE TVCG*, 2012.

[8] C. D. Correa, T. Crnovrsanin, K. liu Ma, and K. Keeton, "The derivatives of centrality and their applications in visualizing social networks," University of California at Davis, Tech. Rep. CSE 2009-5, 2009.

[9] F. van Ham, M. Wattenberg, and F. B. Vigas, "Mapping text with phrase nets." *IEEE TVCG*, vol. 15, no. 6, 2009.

[10] K. Dinkla, M. A. Westenberg, and J. van Wijk, "Compressed adjacency matrices: Untangling gene regulatory networks." *IEEE TVCG*, 2012.

[11] V. Yoghourdjian, T. Dwyer, G. Gange, S. Kieffer, K. Klein, and K. Marriott, "High-quality ultra-compact grid layout of grouped networks," *IEEE TVCG*, vol. 22, no. 1, pp. 339–348, 2016.

[12] T. Dwyer, N. H. Riche, K. Marriott, and C. Mears, "Edge compression techniques for visualization of dense directed graphs," *IEEE TVCG*, vol. 19, no. 12, pp. 2596–2605, 2013.

[13] T. Dwyer, C. Mears, K. Morgan, T. Niven, K. Marriott, and M. Wallace, "Improved optimal and approximate power graph compression for clearer visualisation of dense graphs," *2014 IEEE PacificVIS*, 2014.

[14] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, September 2003.

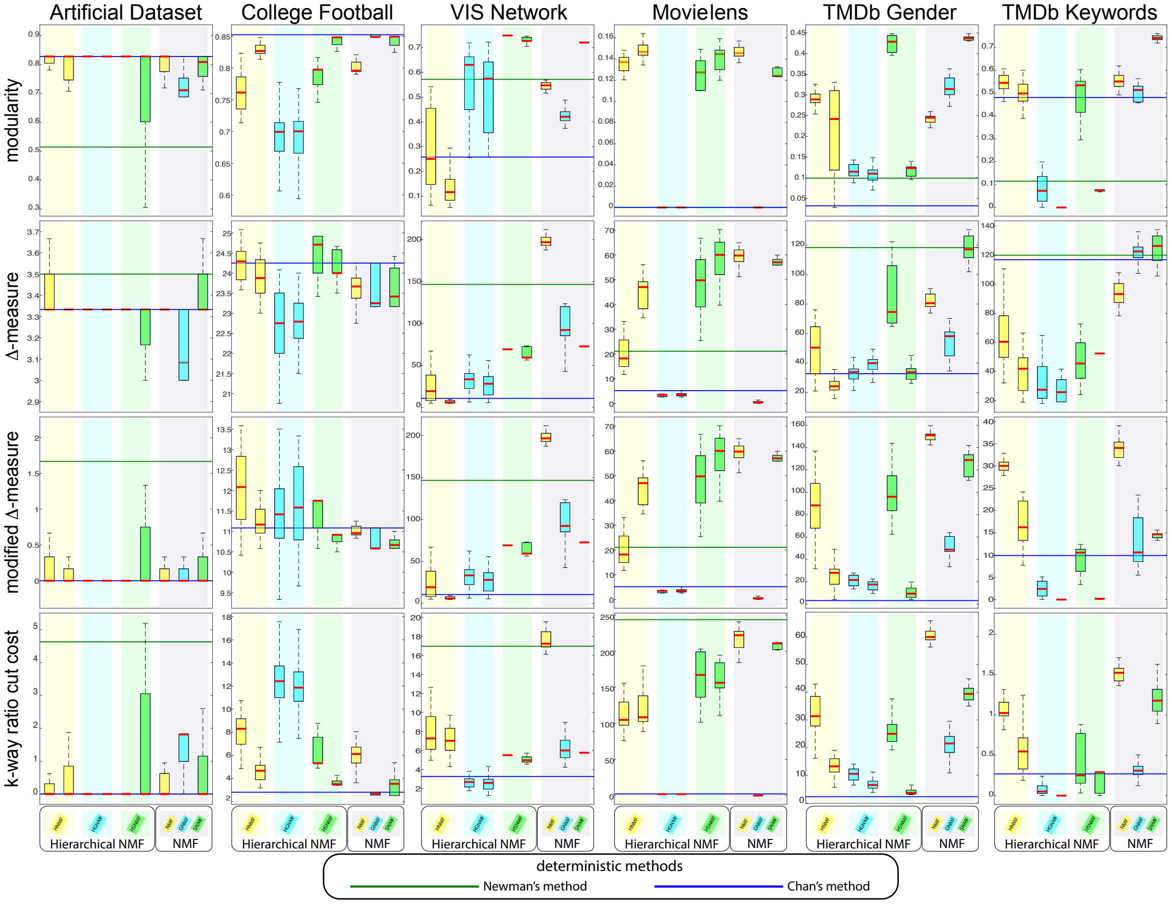


Fig. 6. Comparison with other simplification techniques. Each column (dataset) is divided into four color strips. The first three are from our method, one for each factorization method. Each color bar contains two box plots, which corresponds to the proposed SMM scheme without (left in each color strip) and with (right in each color strip) the edge weight filtering mechanism. The brownish color strips (rightmost color strip in each column) correspond to the results of the traditional NMF clustering. The methods by Newman [14] and Chan et al. [15] are deterministic and their quality measures are shown as horizontal lines.

[15] P. K. Chan, M. D. F. Schlag, and J. Y. Zien, "Spectral k-way ratio-cut partitioning and clustering." *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 13, no. 9, pp. 1088–1096, 1994.

[16] Y. Tian, R. A. Hankins, and J. M. Patel, "Efficient aggregation for graph summarization." in *SIGMOD*. ACM, 2008, pp. 567–580.

[17] N. Zhang, Y. Tian, and J. M. Patel, "Discovery-driven graph summarization." in *ICDE*. IEEE Computer Society, 2010.

[18] S. Navlakha, R. Rastogi, and N. Shrivastava, "Graph summarization with bounded error." in *SIGMOD*, J. T.-L. Wang, Ed. ACM, 2008.

[19] G. Karpis and V. Kumar, "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

[20] C. Walshaw, "A multilevel algorithm for force-directed graph drawing," in *Proceedings of the 8th International Symposium on Graph Drawing*, ser. GD '00. London, UK, UK: Springer-Verlag, 2001, pp. 171–182.

[21] R. Hadany and D. Harel, *A Multi-Scale Algorithm for Drawing Graphs Nicely*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.

[22] R.-S. Wang, S. Zhang, Y. Wang, X.-S. Zhang, and L. Chen, "Clustering complex networks and biological networks by nonnegative matrix factorization with various similarity measures," *Neurocomputing*, 2008.

[23] L. Shi, H. Tong, J. Tang, and C. Lin, "Vegas: Visual influence graph summarization on citation networks." *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 12, pp. 3417–3431, 2015.

[24] D. Kuang, H. Park, and C. H. Q. Ding, "Symmetric nonnegative matrix factorization for graph clustering." in *SDM*. SIAM / Omnipress, 2012.

[25] J. Sun, S. Papadimitriou, C.-Y. Lin, N. Cao, S. Liu, and W. Qian, "Multivis: Content-based social network exploration through multi-way visual analysis." in *SDM*. SIAM, 2009, pp. 1064–1075.

[26] H. Zhou, P. Xu, X. Yuan, and H. Qu, "Edge bundling in information visualization," *Tsinghua Science and Technology*, 2013.

[27] D. Auber, Y. Chiricota, F. Jourdan, and G. Melanon, "Multiscale visualization of small world networks." in *INFOVIS*, 2003.

[28] J. Abello, F. van Ham, and N. Krishnan, "Ask-graphview: A large scale graph visualization system." *IEEE TVCG*, 2006.

[29] E. R. Gansner, Y. Koren, and S. C. North, "Topological fisheye views for visualizing large graphs." *IEEE TVCG*, 2005.

[30] Q. Liao, L. Shi, H. Tong, Y. Hu, Y. Zhao, and C. Lin, "Hierarchical focus+context heterogeneous network visualization," *2014 IEEE PacificVIS*, vol. 0, pp. 89–96, 2014.

[31] M. Wattenberg, "Visual exploration of multivariate graphs," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 811–819.

[32] N. Elmqvist, T.-N. Do, H. Goodell, N. Henry, and J.-D. Fekete, "Zame: Interactive large-scale graph visualization." in *PacificVis*, 2008.

[33] C. Vehlow, F. Beck, and D. Weiskopf, "Visualizing dynamic hierarchies in graph sequences," *IEEE TVCG*, 2016.

[34] R. Blanch, R. Dautriche, and G. Bisson, "Dendrogramix: A hybrid tree-matrix visualization technique to support interactive exploration of dendrograms," in *2015 IEEE PacificVIS*, April 2015, pp. 31–38.

[35] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE PAMI*, 2011.

[36] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko, "Visualization publication dataset," 2015.