Untangling Hairballs*
From 3 to 14 Degrees of Separation

Arlind Nocaj, Mark Ortmann, and Ulrik Brandes

Computer & Information Science, University of Konstanz, Germany

Abstract. Small-world graphs have characteristically low average distance and thus cause force-directed methods to generate drawings that look like hairballs. This is by design as the inherent objective of these methods is a globally uniform edge length or, more generally, accurate distance representation. The problem arises in graphs of high density or high conductance, and in the presence of high-degree vertices, all of which tend to pull vertices together and thus clutter variation in local density.

We here propose a method to draw online social networks, a special class of hairball graphs. The method is based on a spanning subgraph that is sparse but connected and consists of strong ties holding together communities. To identify these ties we propose a novel measure of embeddedness. It is based on a weighted accumulation of triangles in quadrangles and can be determined efficiently. An evaluation on empirical and generated networks indicates that our approach improves upon previous methods using other edge indices. Although primarily designed to achieve more informative drawings, our spanning subgraph may also serve as a sparsifier that trims a hairball graph before the application of a clustering algorithm.

1 Introduction

Online social networks such as Facebook friendship graphs are an amalgamation of a variety of social relations. The existence of a friendship tie might be due to shared interests, spatial proximity, kinship, or professional relations to name but a few. When such a multitude of relations is conflated in the same network, any two nodes are likely to be connected via at most a few links – thus leading to a *small world* effect [21]. Visualizations of these graphs using standard layout methods such as force-directed placement produce drawings in which variation in local structure is hidden in a densely-looking, overlap-ridden *hairball*. An example is given in Fig. 1(a).

Various approaches to reduce the clutter in drawings of small worlds and other hairball graphs have been proposed [12], most notably *edge bundling* [10], *edge lensing* [11], modified *layout algorithms* or *representations* [1,7,29], and *graph simplification* [2,17,18,20,23,30]. The idea of graph simplification is to identify

* We gratefully acknowledge financial support from DFG under grants GRK 1042 and Br 2158/6-1. The proposed method is available in visone.

C. Duncan and A. Symvonis (Eds.): GD 2014, LNCS 8871, pp. 101–112, 2014.

a subset of edges such that only the resulting graph, the so-called backbone, needs to be laid out. We adopt this approach and propose a new method to trim hairballs.

Problem formulations in graph simplification include the preservation of properties such as cuts [2], spectra [23,24], connectivity [30], collapsing substructures into supernodes [18], and emphasizing deeply embedded connections [17,20]. As graph invariants such as cuts are more easily affected by noise in empirical networks, we opt for locally defined graph simplification criteria.

Substantiated by the sociological work of Simmel [22], Nick et al. [17] define the strength of an edge by the number of triangles it is contained in, and then determine the degree of structural embeddedness for an edge by comparing the ranked neighborhoods of its two vertices. The purpose is to identify edges that are more likely to be inside of cohesive groups than between such groups. If the initial edge strengths are uniform, the Simmelian backbone reduces to a backbone proposed by Satuliri et al. [20]. In both methods, the backbone is obtained by finally removing all edges with weights below a specified nodal or network wide threshold.

These filtering techniques are related to, but not the same as graph partitioning. Since we want to use them for graph drawing, the difference is even greater because maintaining connectedness becomes a crucial constraint. Otherwise, the layout algorithm is oblivious to edges of the original graph connecting vertices in different components of the backbone as can be inferred from Fig. 1(b). When connected components happen to be placed far apart, these edges will run across the drawing and produce even worse clutter.

We present an efficient preprocessing technique that allows to draw a certain class of small-world social networks with standard layout algorithms that would produce hairball layouts otherwise. Our main contributions are:

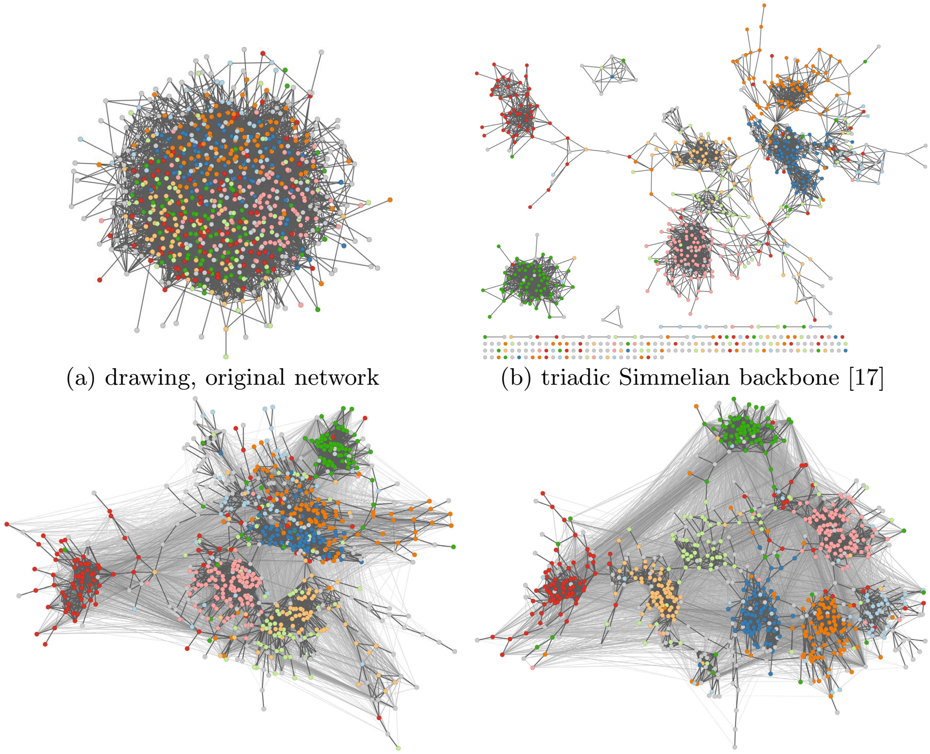
- a novel method to identify strong ties,
- the use of the union of all maximum spanning trees as a sparsifier that maintains connectedness and avoids subtree-ordering ambivalence, and
- an evaluation on observed and generated networks.

We outline our overall method for drawing hairball graphs in the next section and describe our edge embeddedness metric in Sect. 3. Different metrics are evaluated in Sect. 4 and we conclude in Sect. 5.

2 Drawing Algorithm

The main challenges in drawing hairball graphs are their high density, low diameter and noisy group structure. Therefore, our goal is to find a backbone of the graph that retains deeply embedded edges and thus can be used to draw the original graph, e.g., by a force-directed method [13] to reveal the actual variation in cohesiveness.

Since most drawing methods cannot put vertices of different graph components into a meaningful spatial relation, cf. Fig. 1(b), we need to maintain the graph connectivity to retain the global context.



(a) drawing, original network

(b) triadic Simmelian backbone [17]

(c) triadic Simmelian backbone with UMST

(d) quadrilateral Simmelian backbone with UMST (our approach)

Fig. 1. Facebook friendships at California Institute of Technology (Caltech36). Vertex color corresponds to dormitory (gray for missing values), but has not been utilized in the layout algorithm. The layout in (a) is based on the entire hairball graph, whereas (b)-(d) use edge embeddedness, which spreads the graph while keeping cohesive groups together. Embeddedness mapped to edge color; backbone edges dark gray.

This leads to the following requirements on our backbone:

(i) Edges should be favored based on their structural embeddedness only.
(ii) Connectedness has to be maintained.

Two common approaches to simplify a graph $G = (V, E, w)$ with vertex set V, edge set E, and edge weight $w : E \to \mathbb{R}_{\geq 0}$, are *sampling* [2,23] and *thresholding* [1,17,20]. Note that we assume that w reflects the embeddedness of an edge and a higher value corresponds to stronger embeddedness. Although sampling can be used for sparsification purposes the random selection of edges violates both of our requirements. In contrast thresholding guarantees that edges are favored by their weights and consequently their structural properties, as it retains only the top k percent of edges with respect to w. Nevertheless, neither nodal nor network wide thresholding can ensure that the simplified graph stays connected.

Algorithm 1: Hairball Drawing Algorithm

Input: Undirected Graph $G = (V, E)$ and sparsification ratio $s \in [0, 1]$.
Output: Vertex positions $P \in \mathbb{R}^{|V| \times 2}$

$w \leftarrow$ embeddedness weights of edges
sort edges by non-increasing weight
$E_{\text{union}} \leftarrow$ UMST with respect to w
$E_{\text{threshold}} \leftarrow \{e \in E \,:\, w(e) \geq w(e_{\lceil (1-s)|E| \rceil})\}$
$P \leftarrow$ layout determined from spanning subgraph $(V, E_{\text{union}} \cup E_{\text{threshold}})$

Algorithm 2: UMST: Union of all Maximum Spanning Trees

Input: Undirected Graph $G = (V, E)$ and edge weights $w : E \to \mathbb{R}_{\geq 0}$.
Data: Union-Find datastructure
Output: Edges belonging to any MST

$E_{\text{union}} \leftarrow \emptyset$
partition edges by weight into buckets $B_1, \ldots, B_k$
sort buckets by decreasing weight
for $i \leftarrow 1$ ***to*** k **do**
  $M \leftarrow \emptyset$
  foreach $e = (u, v) \in B_i$ **do**
    if *find(u)* $\neq$ *find(v)* **then** $M \leftarrow M \cup \{e\}$
  foreach $e = (u, v) \in M$ **do** `union`(u, v)
  $E_{\text{union}} \leftarrow E_{\text{union}} \cup M$

Sparse connected subgraphs of edges not likely to be between cohesive groups have been proposed, e.g., by van Ham and Wattenberg [28] (planar graphs) and Tumminello et al. [27] (graph of bounded genus). A minimally connected subgraph of edges with high weights is a *maximum spanning tree* (MST), and Mantegna [14] proposed these as a backbone. Trees, however, have severe drawbacks: firstly, they do not maintain any local variation in density and, secondly, they introduce a subtree ordering ambiguity. While the first also means that arbitrary choices must be made when edges have equal embeddedness, the second creates a degree of freedom that is almost as bad as disconnected components.

We combine thresholding (to maintain local variation) with the union of all maximum spanning trees (UMST; to maintain connectedness). The UMST does not only solve the problem of tie breaks but also reduces the ordering problem by resulting in higher connectivity (Fig. 1(b)-(d)).

The complete algorithm to compute the layout of a hairball graph is presented in Alg. 1. Note that the UMST only contributes the (strongest) edges necessary to connect the components that result from the thresholding process.

Kruskal's algorithm for minimum spanning trees is easily adapted to determine the union of all maximum spanning trees. Since every edge of maximum weight that has not been processed yet could be chosen next, we batch-process them before components are merged; cf. Algorithm 2.

The final layout emphasizes variation in local density by considering only deeply embedded edges as expressed by the weights introduced in the next section.

3 Edge Embeddedness by Accumulating Triadic Effects

Real world networks are often aggregates of different relations, which can hamper the detection of subgroups or clusters. Our goal is to determine strong embedded edges, which are likely to be in dense groups, so that we can use them to emphasize the inherent structure. The assumption here is that vertices in the same subgroup of a network are connected stronger with each other than to members outside of the group.

Satuliri et al. [20] propose to capture the embeddedness of an edge $e = (u, v)$ by the Jaccard coefficient over u's and v's neighborhood. Nick et al. [17] suggest a more general framework, consisting of the following main steps:

1. For each edge, determine its strength
2. For each vertex, rank all its neighbors according to the edge strength
3. For each edge, determine its redundancy

The approach of Satuliri et al. can be seen as using a uniform edge strength for step one and the Jaccard coefficient for the redundancy in step three. Contrary to this, Nick et al. use the number of triangles an edge is embedded in (Simmelian strength) for step one and the best prefix Jaccard coefficient for step three. The latter chooses k such that the Jaccard coefficient of the first top k ranked neighbors of u and v is maximized. The effect of this ranking measure is that the highly ranked neighbors have more importance attached, since fewer common vertices are needed to get a high coefficient.

A more intuitive interpretation of this framework is that for an edge $e = (u, v)$ the edge strength allows us to determine the most important neighbors of u and v. If these most important neighbors are *the same*, e is strongly embedded; otherwise e is connecting two vertices, which are likely to be in different groups.

We follow the main idea, but propose a different edge strength than the number of triangles.

Consider the setting in Fig. 2. Clearly, edge e is strongly embedded. Compared to all other edges it closes many triangles resulting in an increase of the *group performance* [5] by introducing mediator effects. Similar to this, an edge (s, t) connecting two triangles at e introduces additional mediator effects on the triangles, which in turn increases the importance of e. We call these edges *mediator edges* on e.

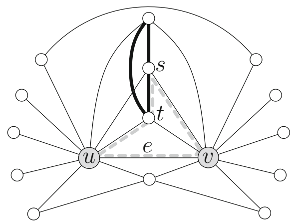


Fig. 2. Triangles at edge e [17,20] do not capture *mediator edges* (bold), while quadrangles do.

Counting the number of triangles at e does not capture the importance of mediator edges. But since each mediator edge creates two quadrangles at e, cf. dashed-contour in Fig. 2, we can use the number of quadrangles containing e

to capture this mediator effect. While there can be additional quadrangles at e, they will be counted only once from e's perspective, which makes their influence rather low. Furthermore, counting the two different types of quadrangles at e would be too time consuming and therefore we will not distinguish between them.

Using the absolute number of quadrangles poses difficulties, when the network contains subgroups of different densities. Hence, we normalize this absolute value by putting it into relation to all edges at vertex u and v. Let $q(u, v)$ be the number of quadrangles containing edge $(u, v) \in E$. We define the *quadrilateral* edge embeddedness as

$$Q(u, v) = \frac{q(u, v)}{\sqrt{q(u) \cdot q(v)}},$$

where $q(v) = \sum_{w \in N(v)} q(v, w)$, for $v \in V$, and $N(v)$ the neighborhood of v. We use the geometric mean over the arithmetic mean, since it takes the dependency of two variables into stronger consideration. Note that edge-metrics using quadrangles have already been proposed by Auber et al. [1] and Radicchi et al. [19], but are different from our method as they focus on density. For a comparison of different edge metrics we refer the reader to Melançon and Sallaberry [16].

Computation and Time Complexity

The quadrangles of a graph G can be listed in $\mathcal{O}(m\alpha(G))$ [6], where m is the number of edges and $\alpha(G)$, the *arboricity* of G, is the minimum number of edge-disjoint forests necessary to cover all edges of G. While the arboricity can be as large as $\sqrt{m}$, it is bounded from above by the h-index of a graph which in turn is found to be very small in social networks [8]. Together with the normalization, the computation of the edge strengths takes $\mathcal{O}(m\alpha(G))$ time.

Neighbors can be ranked in $\mathcal{O}(m \log \triangle(G))$ time and redundancy can be computed in $\mathcal{O}(m\triangle(G))$, where $\triangle(G)$ is the maximum vertex degree. For example, the overall backbone computation took 0.2s on a network with 762 vertices and 16k edges (Caltech65) and 2.3s on a network with 2970 vertices and 100k edges (Smith60) with our Java 7 implementation and an Intel Core i7-2600K CPU@3.40GHz. The approach thus scales to large networks and we turn to the evaluation of its effectiveness in the next section.

4 Evaluating Methods for Edge Embeddedness

In this section we introduce the dataset and a graph model, from which we generate artificial hairball graphs. Then we explain our output quality indicators and the different edge embeddedness methods. For each graph and edge embeddedness method, we iteratively increase the sparsification ratio by 10% and compute the corresponding backbone. Layouts are computed using stress majorization [9] initialized by PivotMDS [3] as suggested in [4].

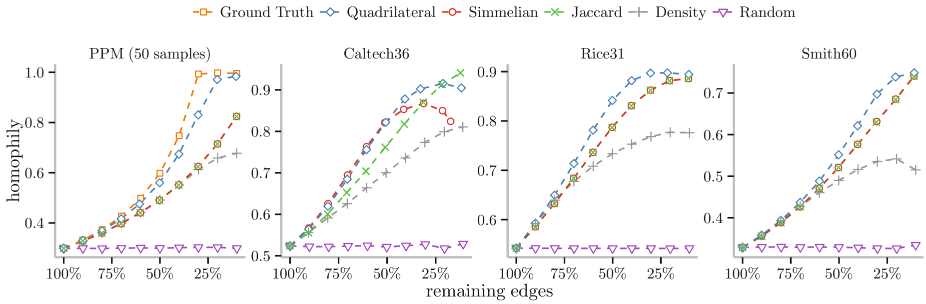


Fig. 3. Homophily (y-axis) is plotted against the number of remaining edges (x-axis) for the synthetic model (PPM) and three of the Facebook networks. Overall Quadrilateral performs better than the others. For the synthetic networks it comes very close to the ground truth.

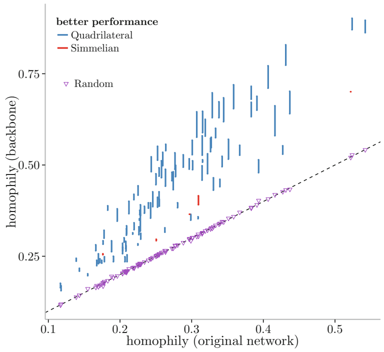
4.1 Dataset and Model

As real world samples, we use the *Facebook100* dataset [26], which contains social relations of 100 higher educational institutes in the US. The network size varies from 762 to 41K vertices and from 16K to 1.6M edges. The dataset is directly from Facebook, not sampled, and thus very complete in terms of capturing the social relations according to a widely used service at that time. Additional attributes obtained from the Facebook profiles are gender, year of graduation, dormitory, etc. Due to incomplete profiles, a number of attribute values are missing. We will use the dormitory attribute for our evaluation, because it has been argued to be important for the creation of social relations in many of the networks [26].

Note that, in spite of a strong empirical association with homophilous attribute values, no ground-truth group structure is available for Facebook networks. Therefore, we also generated artificial networks from a model that represents the idealized version of the networks we are considering in this application.

A simple model generating random graphs with cohesive groups that are connected into a small world is the *planted partition model* (PPM) [15]. Let $\mathcal{C} = \{C_1, \ldots, C_k\}$ be a partition of V for a graph $G = (V, E)$. Then $\mathcal{C}$ is called a clustering of G with class $c(v) \in \mathcal{C}$ for a vertex $v \in V$. The probability of an edge (u, v) is p_{in} if $c(u) = c(v)$ and p_{out} if $c(u) \neq c(v)$. We generated 50 graphs from a PPM with 500 vertices, $k = 9$, $p_{in} = 0.3$, and $p_{out} = 0.01$. On top of that, we ran a random noise model with $p_{in} = p_{out} = 0.1$ to obfuscate the underlying group structure. The resulting graphs are very dense, have a low diameter, and are real hairballs without any visible structure when laid out using force-directed methods. The presented results of our model are averaged over 50 samples.

Fig. 4. Dormitory-homophily of different backbones, with sparsification ratio 70%, (y-axis) compared to the homophily in the original network (x-axis) for all Facebook100 networks. Points above/below the dashed line indicate homophily increase/decrease respective the original network. Simmelian and Quadrilateral homophily values for corresponding networks have been connected by colored segments comparing their performance.



4.2 Edge Embeddedness Methods

We compare different methods which assign a weight $w : E \to \mathbb{R}_{\geq 0}$ to each edge $e = (u, v) \in E$ depicting its embeddedness. All these methods are then extended using our UMST approach to guarantee the connectivity, such that a layout can be computed from the resulting graph. We use the following approaches to assign a weight to the edges.

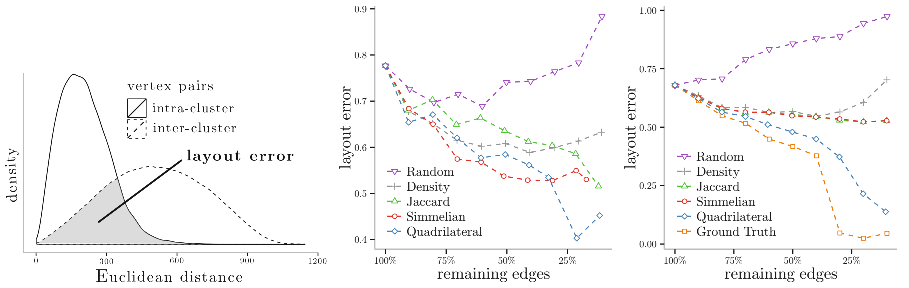
Random: Assigns uniform random weights, as base line.
Jaccard: Jaccard coefficient, $\frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$, as proposed by Satuliri et al. [20].
Simmelian: Triadic Simmelian backbone, as proposed by Nick et al. [17].
Quadrilateral: Quadrilateral Simmelian backbone, based on our embeddedness method, which accumulates triadic effects at an edge with quadrangles (Sect. 3).
Density: Metric by Auber et al. [1] accumulating densities of different subgroups in the local neighborhood.
Ground Truth: Knowledge of class membership in the synthetic network is used to assign directly a low value to inter-cluster edges and a big value to intra-cluster edges.

4.3 Quality Metrics

In contrast to the synthetic networks there is no ground truth available for the Facebook networks. This makes it hard to evaluate outcomes of the different methods. Nevertheless, it was found that for many of the Facebook networks, the housing structure (dormitory attribute) is very relevant for the underlying formation of social relations [17,26]. We, therefore, use the dormitory attribute as a reference for evaluation.

Assume that we know the ground truth, meaning the class membership $c(v)$ of each vertex. A *perfect* algorithm, for example, would first remove all inter-cluster edges before starting to remove intra-cluster edges while obeying the required sparsification ratio. Since inter-cluster edges are removed priorly, this increases the ratio between intra-cluster or homophily edges and the total number of edges.

If the edge embeddedness methods perform similar to this, the ratio of homophily edges



(a) layout error (b) Caltech36 (c) synthetic networks

Fig. 5. Layout error of different edge embedding methods combined with our UMST for (b) a real world network and (c) synthetic networks. (a) shows the layout error for a single point of the line chart in (b).

$$\text{homophily}(G) = \frac{\#\text{homophily edges}}{\#\text{homophily edges} + \#\text{heterophily edges}}$$

should monotonically increase, while gradually removing edges from the network according to their weight. Edges for which the class membership (attribute) of at least one vertex is missing are neglected.

Additionally, we would like to see how well this class membership is reflected in the layouts. Vertex pairs of the same class should have a small Euclidean distance, while pairs of different classes should have a large Euclidean distance. Looking at the curve of the Euclidean distance distribution of the intra-cluster and inter-cluster vertex pairs in Fig. 5(a), we define the layout error as the intersection area of these two curves. The layout error can also be interpreted as the percentage of vertex pairs, where the distinction whether they are in the same cluster or not cannot clearly be made based on the Euclidean distance. Since the computation of this quality metric is very time intensive, it was not feasible to analyze all 100 Facebook networks with it.

4.4 Results and Discussion

An interesting observation from Fig. 3 is that Jaccard and Simmelian perform very similar for most Facebook networks. Our method (Quadrilateral) clearly manages to distinguish between the different types of edges better than the other methods, especially in earlier phases of the sparsification.

For all 100 Facebook networks, the difference in homophily between Simmelian and Quadrilateral is shown by the length of a vertical segment in Fig. 4. While both approaches increase the percentage of homophily edges (all segments above the diagonal dashed line), Quadrilateral clearly performs better, especially for networks with higher percentage of homophily edges.

Although the homophily of Jaccard and Quadrilateral is nearly the same for the last but one step of the Caltech network (Fig. 3) the Quadrilateral embedding

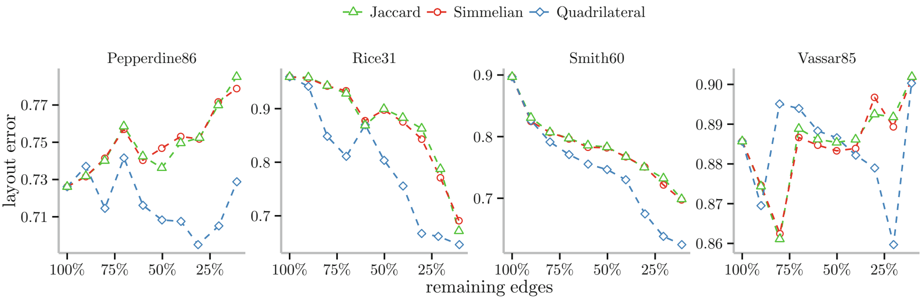


Fig. 6. Layout error of Facebook networks w.r.t. the dormitory attribute. While improvement is not clear for Pepperdine86 and Vassar85, the layout is improved a lot for the networks with high homophily (Rice31 and Smith60).

creates the superior layout (Fig. 5). Furthermore, for the synthetic networks, Quadrilateral comes very close to the ground truth (Fig. 5).

Figure 6 shows the layout error for four Facebook networks and the three best performing edge metrics (according to homophily). The layout clearly improves for the Rice and Smith network, but not much for the other two. One possible explanation for this could be that the dormitory attribute is not the explanatory variable for the formation of social relations in these two networks.

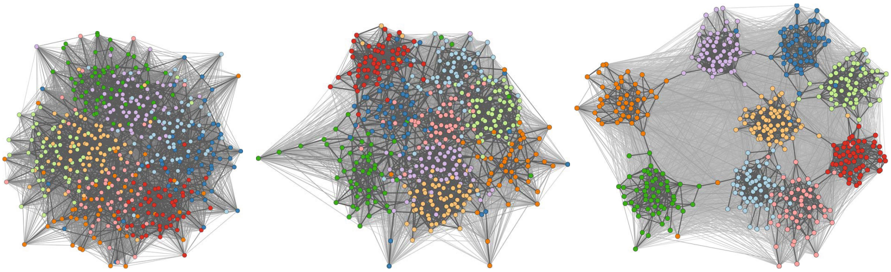
The effectiveness of our layout quality metric can also be verified, by looking at the final drawings in Fig. 1(c) and 1(d). In the latter many clusters, as light green and light blue, are more clearly visible. For the synthetic networks our method comes also very close to the ground truth, in terms of layout error (Fig. 5(c)). Again, the drawings of a synthetic network (Fig. 7) support this conclusion.

5 Conclusion

We proposed a sparsification approach to draw hairball graphs as encountered in online social networks. It is based on the idea that pairwise distances (the "degrees of separation") need to be increased without disrupting tightly-knit groups. The deeply embedded edges such groups are made of are identified using a suitably modified Simmelian backbone [17], and overall layout organization is stabilized by maintaining connectedness via the union of all maximum spanning trees.

An evaluation with empirical and generated networks showed that our novel metric manages to reveal relations deeply embedded in latent primary groups. In the resulting drawings such groups are separated from each other but still positioned in their global context. On the Facebook100 dataset, average distances increased from about 3 in the original friendship networks to about 14 in the backbone, thus easing the layout task for force-directed algorithms.

Our proposed edge embeddedness metric proved to be more effective than previous approaches with respect to improving layout quality by way of amplifying homophily. It is thus likely to be useful as a preprocessing step for graph clustering algorithms as well.



(a) Jaccard [20] (b) triadic Simmelian backbone [17] (c) our quadrilateral Simmelian backbone

Fig. 7. Layouts of the same synthetic network determined by different edge embeddedness methods combined with our UMST (sparsification ratio of 80%).
Colors encode groups – ground truth.

By design, our technique appears to be best suited for small-world networks with multiple centers. While these are common, especially in social media, it will be interesting to identify variants for hierarchically clustered graphs and single-centered core-periphery structures such as the network of world trade [25].

References

1. Auber, D., Chiricota, Y., Jourdan, F., Melançon, G.: Multiscale visualization of small world networks. In: INFOVIS. IEEE Computer Society (2003)
2. Benczúr, A.A., Karger, D.R.: Approximating s-t minimum cuts in $\tilde{o}(n^2)$ time. In: Miller, G.L. (ed.) STOC, pp. 47–55. ACM (1996)
3. Brandes, U., Pich, C.: Eigensolver methods for progressive multidimensional scaling of large data. In: Kaufmann, M., Wagner, D. (eds.) GD 2006. LNCS, vol. 4372, pp. 42–53. Springer, Heidelberg (2007)
4. Brandes, U., Pich, C.: An experimental study on distance-based graph drawing. In: Tollis, I.G., Patrignani, M. (eds.) GD 2008. LNCS, vol. 5417, pp. 218–229. Springer, Heidelberg (2009)
5. Burt, R.S.: Structural holes versus network closure as social capital. Social Capital: Theory and Research, pp. 31–56 (2001)
6. Chiba, N., Nishizeki, T.: Arboricity and subgraph listing algorithms. SIAM J. Comput. 14(1), 210–223 (1985)
7. Cohen, J.D.: Drawing graphs to convey proximity: An incremental arrangement method. ACM Trans. Comput.-Hum. Interact. 4(3), 197–229 (1997)
8. Eppstein, D., Spiro, E.S.: The h-index of a graph and its application to dynamic subgraph statistics. J. Graph Algorithms Appl. 16(2), 543–567 (2012)
9. Gansner, E.R., Koren, Y., North, S.C.: Graph drawing by stress majorization. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 239–250. Springer, Heidelberg (2005)
10. Holten, D., van Wijk, J.J.: Force-directed edge bundling for graph visualization. Comput. Graph. Forum 28(3), 983–990 (2009)
11. Hurter, C., Telea, A., Ersoy, O.: Moleview: An attribute and structure-based semantic lens for large element-based plots. IEEE TVCG 17(12), 2600–2609 (2011)

12. Jankun-Kelly, T.J., Dwyer, T., Holten, D., Hurter, C., Nöllenburg, M., Weaver, C., Xu, K.: Scalability considerations for multivariate graph visualization. In: Kerren, A., Purchase, H.C., Ward, M.O. (eds.) Multivariate Network Visualization 2013. LNCS, vol. 8380, pp. 208–236. Springer, Heidelberg (2013)
13. Kobourov, S.G.: Force-directed drawing algorithms. In: Tamassia, R. (ed.) Handbk. of Graph Drawing and Visualization, pp. 383–408. Chapman & Hall/CRC (2013)
14. Mantegna, R.N.: Hierarchical structure in financial markets. The European Physical Journal B-Condensed Matter and Complex Systems 11(1), 193–197 (1999)
15. McSherry, F.: Spectral partitioning of random graphs. In: FOCS, pp. 529–537. IEEE Computer Society (2001)
16. Melançon, G., Sallaberry, A.: Edge metrics for visual graph analytics: A comparative study. In: IV, pp. 610–615. IEEE Computer Society (2008)
17. Nick, B., Lee, C., Cunningham, P., Brandes, U.: Simmelian backbones: amplifying hidden homophily in facebook networks. In: Rokne, J.G., Faloutsos, C. (eds.) ASONAM, pp. 525–532. ACM (2013)
18. Pfaltz, J.L.: The irreducible spine(s) of undirected networks. In: Lin, X., Manolopoulos, Y., Srivastava, D., Huang, G. (eds.) WISE 2013, Part II. LNCS, vol. 8181, pp. 104–117. Springer, Heidelberg (2013)
19. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proc. Natl. Acad. Sci. 101(9), 2658–2663 (2004)
20. Satuluri, V., Parthasarathy, S., Ruan, Y.: Local graph sparsification for scalable clustering. In: Sellis, T.K., Miller, R.J., Kementsietsidis, A., Velegrakis, Y. (eds.) SIGMOD Conference, pp. 721–732. ACM (2011)
21. Schnettler, S.: A structured overview of 50 years of small-world research. Social Networks 31(3), 165–178 (2009)
22. Simmel, G.: The sociology of Georg Simmel, vol. 92892. Simon and Schuster (1950)
23. Spielman, D.A., Srivastava, N.: Graph sparsification by effective resistances. SIAM J. Comput. 40(6), 1913–1926 (2011)
24. Spielman, D.A., Teng, S.H.: Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In: Babai, L. (ed.) STOC, pp. 81–90. ACM (2004)
25. Subramanian, A., Wei, S.J.: The WTO promotes trade, strongly but unevenly. Journal of International Economics 72(1), 151–175 (2007)
26. Traud, A.L., Kelsic, E.D., Mucha, P.J., Porter, M.A.: Comparing community structure to characteristics in online collegiate social networks. SIAM Review 53(3), 526–543 (2011)
27. Tumminello, M., Aste, T., Di Matteo, T., Mantegna, R.: A tool for filtering information in complex systems. Proc. Natl. Acad. Sci. 102(30), 10421–10426 (2005)
28. Van Ham, F., Wattenberg, M.: Centrality based visualization of small world graphs. Computer Graphics Forum 27(3), 975–982 (2008)
29. Zaidi, F., Sallaberry, A., Melançon, G.: Revealing hidden community structures and identifying bridges in complex networks: An application to analyzing contents of web pages for browsing. In: Web Intelligence, pp. 198–205. IEEE (2009)
30. Zhou, F., Mahler, S., Toivonen, H.: Network simplification with minimal loss of connectivity. In: Webb, G.I., Liu, B., Zhang, C., Gunopulos, D., Wu, X. (eds.) ICDM, pp. 659–668. IEEE Computer Society (2010)