Energy-Based Clustering of Graphs with Nonuniform Degrees

Andreas Noack

Institute of Computer Science,
Brandenburg University of Technology at Cottbus,
PO Box 10 13 44, 03013 Cottbus, Germany
an@informatik.tu-cottbus.de

Abstract. Widely varying node degrees occur in software dependency graphs, hyperlink structures, social networks, and many other real-world graphs. Finding dense subgraphs in such graphs is of great practical interest, as these clusters may correspond to cohesive software modules, semantically related documents, and groups of friends or collaborators. Many existing clustering criteria and energy models are biased towards clustering together nodes with high degrees. In this paper, we introduce a clustering criterion based on normalizing cuts with edge numbers (instead of node numbers), and a corresponding energy model based on edge repulsion (instead of node repulsion) that reveal clusters without this bias.

1 Introduction

It is increasingly recognized that the degrees of the nodes in many graph models of real-world systems vary widely [1], with examples including dependencies between software artifacts, citations of scientific articles, hyperlink structures (like the World Wide Web, dictionaries, and thesauri), social networks, and neural networks. Dense subgraphs of these graphs are of great scientific and practical interest, because these clusters are candidates for cohesive software modules, research areas, semantically related terms or documents, groups of closely interacting people, and functional units of the nervous system.

The first challenge in the identification of such clusters is to formalize the notion of a cluster. Section 2 shows that several existing cut-based clustering criteria are biased towards certain cluster sizes, and derives two unbiased clustering criteria by appropriately normalizing the cut. There are two unbiased clustering criteria because the two natural measures of cluster size, namely the number of nodes and the number of edges, are equivalent (up to a constant factor) only for graphs with uniform degrees.

The second challenge is the computation and the presentation of the clusters. Section 3 introduces two energy models that reveal the clusters corresponding to the two clustering criteria. This enables the computation of clusters with existing energy minimization algorithms (like the algorithm of Barnes and Hut [3, 19]) that scale to graphs with thousands of nodes. The presentation as graph drawing facilitates the comprehension of the cluster structure, because viewers naturally interpret closely positioned nodes as strongly related [4, 6]. Section 4 presents example drawings of various real-world graphs.

P. Healy and N.S. Nikolov (Eds.): GD 2005, LNCS 3843, pp. 309–320, 2005.

1.1 Basic Definitions

For a set M, let $|M|$ be the number of elements of M, and let $M^{(2)}$ be the set of all subsets of M which have exactly two elements. A *bipartition* of a set M is a pair (M_1, M_2) of sets with $M_1 \cup M_2 = M$, $M_1 \cap M_2 = \emptyset$, $M_1 \neq \emptyset$, and $M_2 \neq \emptyset$.

A *graph* $G = (V, E)$ consists of a finite set V of *nodes* and a finite set E of *edges* with $E \subseteq V^{(2)}$. Because drawings can be computed separately for different components of a graph, we restrict ourselves to connected graphs, i.e. graphs where every pair of nodes is connected by a path.

For a node v, the *degree* $\deg(v)$ is the number $|\{u \mid \{u, v\} \in E\}|$ of nodes adjacent to v. The total degree $\sum_{v \in V_1} \deg(v)$ of all nodes in a set V_1 is denoted by $\deg(V_1)$. For two sets of nodes V_1 and V_2, the number of edges $\left|\{\{v_1, v_2\} \in E \mid v_1 \in V_1, v_2 \in V_2\}\right|$ between V_1 and V_2 is called the *cut* between V_1 and V_2 and denoted by $\mathrm{cut}(V_1, V_2)$. We often identify a set of nodes V_1 with the subgraph $(V_1, \{e \in E \mid e \subseteq V_1\})$ it induces.

A *d-dimensional drawing* of the graph G is a vector $p = (p_v)_{v \in V}$ of node positions $p_v \in \mathbb{R}^d$. For a drawing p and two nodes $u, v \in V$, the length of the difference vector $p_v - p_u$ is called the *distance* of u and v in p and denoted by $||p_v - p_u||$.

2 Graph Clustering Criteria

Informally, we denote by a graph cluster a subgraph with many internal edges and few edges to the remaining graph. This can be formalized by defining a measure for the coupling between subgraphs, such that a smaller coupling corresponds to a better clustering. This section discusses such measures, starting with the cut. The main result is that the cut is biased, and has to be normalized with the sizes of the subgraphs. For graphs with uniform degrees, normalizing the cut with the number of nodes of the subgraphs is equivalent to normalizing the cut with the number of edges, but for graphs with nonuniform degrees, these two alternatives lead to considerably different notions of a cluster. For clarity, the discussion is restricted to the coupling between two subgraphs, the generalization to more subgraphs is straightforward.

2.1 The Cut

A simple measure of the coupling between two disjoint sets of nodes V_1 and V_2 of a graph (V, E) is their cut $\mathrm{cut}(V_1, V_2)$. There exist efficient algorithms for finding a bipartition of a given graph with the minimum cut [22].

However, the cut prefers bipartitions that consist of a very small and a very large subgraph, as the following calculation shows. Among the $\frac{1}{2}(|V|^2 - |V|)$ unordered pairs of nodes from V, there are $|V_1| \cdot |V_2|$ pairs of one node from V_1 and one node from V_2. So the expected cut between V_1 and V_2 is $\frac{2|V_1| \cdot |V_2|}{|V|^2 - |V|}|E|$, which is much smaller for bipartitions with $|V_1| \ll |V_2|$ than for bipartitions with $|V_1| = |V_2|$.

2.2 The Node-Normalized Cut

An unbiased measure of the coupling between two disjoint sets of nodes V_1 and V_2 called *node-normalized cut* is obtained by normalizing the cut with the expected cut (and ignoring constant factors for simplicity):

$$\text{nodenormcut}(V_1, V_2) = \frac{\text{cut}(V_1, V_2)}{|V_1| \cdot |V_2|}$$

For a fixed graph (V, E) and all clusters sizes $|V_1|$ and $|V_2|$, the node-normalized cut has the same expected value $\frac{2|E|}{|V|^2-|V|}$.

This measure is also known as ratio of the cut, and has been used in VLSI design [2] and software engineering [16]. Computing a bipartition with minimum node-normalized cut is NP-complete, but approximable in polynomial time within factor $O(\log(|V|))$ [15].

The node-normalized cut is still biased towards bipartitions with a very small and a very large subgraph if the number of edges is used as measure of subgraph size. Consider two bipartitions of the set of nodes V into two sets V_1 and V_2 of equal cardinality, where $\deg(V_1) = \deg(V_2)$ in the first bipartition, and $\deg(V_1) \ll \deg(V_2)$ in the second bipartition. (Note that such bipartitions only exist in graphs with nonuniform degrees.) Then the expected cut, and therefore the node-normalized cut, is much larger for the first bipartition than for the second.

The following calculation makes this more precise. The $|E|$ edges of a graph (V, E) have $\deg(V) = 2|E|$ end nodes. So there are $\frac{1}{2}\left(\deg(V)^2 - \sum_{v \in V} \deg(v)^2\right)$ unordered pairs of end nodes. (The subtrahend accounts for "pairs" of two equal end nodes.) Among these pairs, there are $\deg(V_1)\deg(V_2)$ pairs of one node from V_1 and one node from V_2. So the expected cut between $|V_1|$ and $|V_2|$ is $\frac{2\deg(V_1)\deg(V_2)}{\deg(V)^2 - \sum_{v \in V} \deg(v)^2}|E|$, which is much smaller for bipartitions with $\deg(V_1) \ll \deg(V_2)$ than for bipartitions with $\deg(V_1) = \deg(V_2)$.

2.3 The Edge-Normalized Cut

Normalizing the cut with the expected cut (without constant factors) results in another measure of coupling called *edge-normalized cut*:

$$\text{edgenormcut}(V_1, V_2) = \frac{\text{cut}(V_1, V_2)}{\deg(V_1)\deg(V_2)}$$

For a fixed graph (V, E) and all clusters sizes $\deg(V_1)$ and $\deg(V_2)$, the edge-normalized cut has the same expected value $\frac{2|E|}{\deg(V)^2 - \sum_{v \in V} \deg(v)^2}$.

A similar measure has been introduced (without a systematic derivation) by Shi and Malik [20] as normalized cut:

$$\text{ncut}(V_1, V_2) = \frac{\text{cut}(V_1, V_2)}{\deg(V_1)} + \frac{\text{cut}(V_1, V_2)}{\deg(V_2)}.$$

Because $(\deg(V_1) + \deg(V_2))\,\text{edgenormcut}(V_1, V_2) = \text{ncut}(V_1, V_2)$, the values of the two measures differ only by the constant factor $\deg(V)$ if $V_1 \cup V_2 = V$. The problem of deciding whether a given graph has a bipartition with an edge-normalized cut smaller than a given constant is NP-complete [20].

2.4 Related Work: Other Measures of Coupling

Other measures of the coupling between two disjoint sets of nodes V_1 and V_2 of a graph (V, E) include the expansion [14]

$$\text{expansion}(V_1, V_2) = \frac{\text{cut}(V_1, V_2)}{\min(|V_1|, |V_2|)}$$

and the conductance [14]

$$\text{conductance}(V_1, V_2) = \frac{\text{cut}(V_1, V_2)}{\min(\deg(V_1), \deg(V_2))}.$$

Computing a bipartition with minimum expansion is NP-complete, but approximable in polynomial time within factor $O(\log(|V|))$ [15].

The expansion is biased towards similarly-sized clusters: For $|V_1| = |V| - 1$ and $|V_2| = 1$, the expected expansion is $\frac{2|E|}{|V|}$, while for $|V_1| = |V_2| = \frac{1}{2}|V|$, the expected expansion is only $\frac{|E|}{|V|-1}$. The conductance has a similar bias when the total degree is used as measure of cluster size.

3 Energy Models for Graph Clustering

One particular way to compute and present the cluster structure of graphs is energy-based graph drawing. That the results are drawings and not partitions of the set of nodes has several benefits: Drawings facilitate the comprehension of the cluster structure, because viewers naturally interpret closely positioned nodes as strongly related [4, 6], and enable the navigation from one cluster to closely related clusters. Drawings show how clearly clusters are separated, and how closely nodes are associated with their cluster.

In an earlier paper [17], we introduced the LinLog energy model for visualizing clusters with respect to the node-normalized cut. The main result of this section is that replacing repulsion between nodes with repulsion between edges adapts the LinLog model to the edge-normalized cut (and thus to graphs with nonuniform degrees).

3.1 The Edge-Repulsion LinLog Energy Model

The *node-repulsion LinLog energy* of a drawing p is defined in [17] as

$$U_{NodeLinLog}(p) = \sum\nolimits_{\{u,v\} \in E} ||p_u - p_v|| - \sum\nolimits_{\{u,v\} \in V^{(2)}} \ln ||p_u - p_v||$$

To avoid infinite energies we assume that different nodes have different positions, which is no serious restriction because we are interested in drawings with low energy. The first term of the difference can be interpreted as attraction between adjacent nodes, the second term as repulsion between different nodes.

In the *edge-repulsion LinLog energy model* the repulsion between nodes is replaced by repulsion between edges. In our formalization, the repulsion does not act between entire edges, but only between their end nodes. So the repulsion between two nodes is weighted by the number of edges of which they are an end node, i.e. by their degrees:

$$U_{EdgeLinLog}(p) = \sum\nolimits_{\{u,v\} \in E} ||p_u - p_v|| - \sum\nolimits_{\{u,v\} \in V^{(2)}} \deg(u) \deg(v) \ln ||p_u - p_v||$$

The beauty of edge repulsion lies in its symmetry: Edges cause both attraction and repulsion. In other words, nodes that attract strongly also repulse strongly. More precisely, each node has consistently – in terms of attraction and repulsion – an influence on the

drawing proportional to its degree. (This can be visualized by setting the size of a node to its degree, as in the figures in Sect. 4.) As a beneficial side effect, this symmetry can also facilitate the introduction and weighting of additional forces [18].

In a node-repulsion LinLog drawing of a graph with very nonuniform degrees, the positions of the nodes mainly reflect their degrees: The (strongly attracting) high-degree nodes are mostly placed at the center, and the (weakly attracting, but equally repulsing) low-degree nodes at the borders. This bias is removed in the edge-repulsion LinLog model. For graphs with uniform node degrees, both models have equivalent minima up to scaling.

3.2 Interpretation of Edge-Repulsion LinLog Drawings

The theorems and proofs about the interpretation of node-repulsion LinLog drawings in [17] can be adapted to edge-repulsion LinLog. This subsection only presents a simplified version to illustrate the difference between node repulsion and edge repulsion.

Let $G = (V, E)$ be a graph, and let (V_1, V_2) be a bipartition of the set of nodes V into two cohesive (dense), loosely coupled subgraphs. Let p be a drawing of G with minimum edge-repulsion LinLog energy. How is the distance of V_1 and V_2 in p related to their coupling?

Due to the high cohesion and low coupling, the distances *within* V_1 and *within* V_2 should be much smaller than the distance *between* V_1 and V_2 in p. For our discussion, this situation can be reasonably closely approximated by assuming that all nodes in V_1 have the same position and all nodes in V_2 have the same position in p. Let d be the Euclidean distance of these two positions.

Ignoring the energy between nodes of the same subgraph (which is irrelevant for the distance d between the subgraphs), we obtain the following edge-repulsion LinLog energy of the drawing p:

$$U(d) = \operatorname{cut}(V_1, V_2)\, d - \deg(V_1) \deg(V_2) \ln d$$

Because p is a drawing with minimum energy, this function has a global minimum at d, so $U'(d) = 0$.

$$0 = U'(d) = \operatorname{cut}(V_1, V_2) - \deg(V_1) \deg(V_2)/d$$

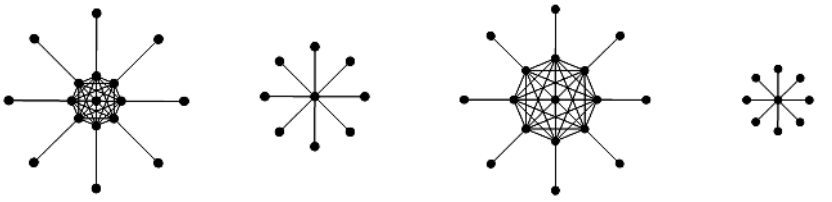
$$d = \frac{\deg(V_1) \deg(V_2)}{\operatorname{cut}(V_1, V_2)} = \frac{1}{\operatorname{edgenormcut}(V_1, V_2)}$$

So the distance d between V_1 and V_2 in the drawing with minimum edge-repulsion LinLog energy is the inverse of their edge-normalized cut. For the node-repulsion LinLog energy model, we only need to replace $\deg(V_1) \deg(V_2)$ with $|V_1| \cdot |V_2|$ in all terms, so the distance is the inverse node-normalized cut.

This simple analysis method is not meant to replace a more detailed examination (as done in [17] for node-repulsion LinLog), but it allows a quick approximate assessment of the clustering properties for many energy models.

3.3 Related Work

Energy Models for Clustering. The force and energy models of Eades [7], Fruchterman and Reingold [8], Davidson and Harel [5], and Kamada and Kawai [13] tend to



(a) (Node-repulsion) Fruchterman-Reingold (b) Edge-repulsion Fruchterman-Reingold

Fig. 1. Two small graphs

enforce uniform (or other given) edge lengths, to support tasks like following paths and identifying neighbors. The LinLog energy models reveal clusters, which generally requires some long (between-cluster) and short (within-cluster) edges. So the two goals of clustering and uniform edge lengths are contradictory and cannot be achieved with a single energy model. But classes of energy models like r-PolyLog [17] allow the user to choose any compromise.

Edge Repulsion. In many force and energy models, including those of Eades [7] and Fruchterman and Reingold [8], adjacent nodes attract and all pairs of nodes repulse. Like node-repulsion LinLog, these models tend to draw dense subgraphs too small (because attraction dominates repulsion) and sparse subgraphs too large.

Figure 1a shows examples for the Fruchterman-Reingold model: The complete subgraph of the left graph contains most edges, but uses only a small part of the drawing area. Much area is wasted by the unnecessarily long edges to the eight peripheral nodes. The (sparse) right graph is drawn much larger than the (dense) complete subgraph, although it contains much fewer edges. Further examples are given in Sect. 4.

Like for LinLog, replacing node repulsion with edge repulsion improves the balance between attraction and repulsion, because both are caused by the edges. Figure 1b shows that this leads to a more uniform information density and thus better readability.

A related concept is the repulsive force between edges and nodes proposed by Davidson and Harel [5]. This force was introduced exclusively for improving readability, and not for enabling interpretations with respect to the cluster structure.

Algorithms for Energy Minimization. As usual in force- and energy-based graph drawing (with the exception of Hall's energy model [11]), we have no practical algorithm that finds global minima of the LinLog energy models. In our experiments we use the hierarchical energy minimization algorithm of Barnes and Hut [3], which was introduced to graph drawing by Quigley and Eades [19]. Its runtime is in $O(|E| + |V| \log |V|)$ per iteration. The overall runtime grows somewhat faster because the number of iterations needed for convergence tends to grow with $|V|$. Some other efficient minimization algorithms are not expected to find good energy minima for clustering energy models like LinLog and for graphs with small diameter [9, 12, 21, 10].

4 Examples

This section shows example drawings of the edge-repulsion LinLog energy model, and, for comparison, of the node-repulsion LinLog energy model and the well-known

Fruchterman-Reingold force model [8]. The first subsection illustrates the differences between the models with drawings of a pseudo-random graph. The second subsection shows that drawings of the edge-repulsion LinLog model can provide non-trivial and useful insights into the structure of real-world graphs.

In all figures, the area of each circle that represents a node is proportional to the degree of the node, with the exception that there is a minimum area to ensure visibility. Some drawings were rotated manually. (Rotation does not change the energy.) In most drawings, the edges are omitted to avoid clutter.

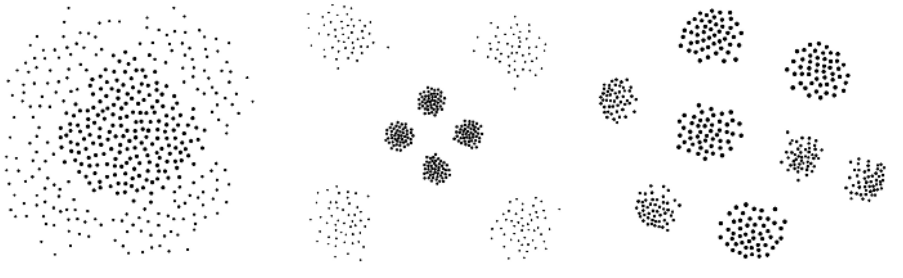
An effective visualization of large graphs requires panning and zooming, and interactive showing and hiding of node labels and edges. Therefore we provide VRML files (offering the first three features) of the drawings on a supplementary web page[1].

4.1 Pseudo-Random Graph

Figure 2 shows a pseudo-random graph with eight cluster of 50 nodes. The probability of an edge $\{u, v\}$ is

- 1 if u and v belong to the same of the first four clusters,
- 0.5 if u and v belong to the same of the second four clusters,
- 0.2 if u and v belong to different of the first four clusters,
- 0.05 if u and v belong to different of the second four clusters, and
- 0.1 if u belongs to one of the first and v belongs to one of the second four clusters.

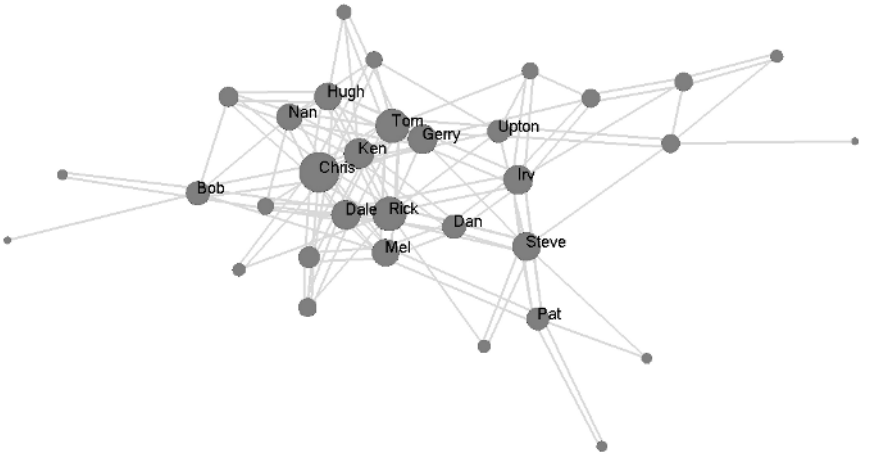
Both LinLog models reveal the clusters, but their drawings differ because the degrees of the nodes are nonuniform. The node-repulsion LinLog drawing places the first four clusters more closely than the second four clusters, which reflects that node-normalized cuts between the first four clusters are higher than between the second four clusters. In the edge-repulsion LinLog drawing the distances between all clusters are similar, which reflects that the edge-normalized cuts between all pairs of clusters are similar.



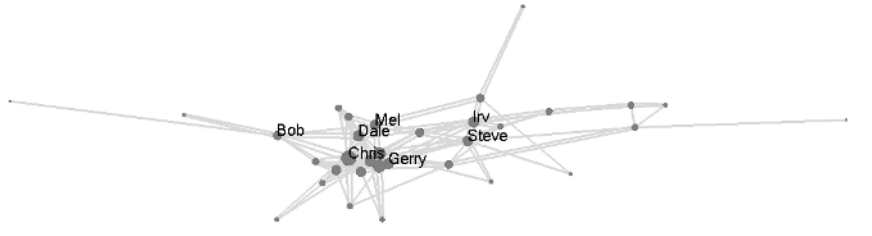
(a) Fruchterman-Reingold (b) Node-repulsion LinLog (c) Edge-repulsion LinLog

Fig. 2. Pseudo-random graph

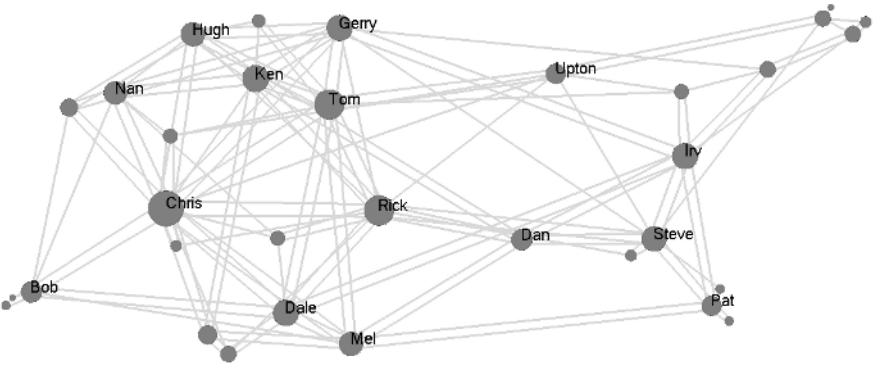
[1] http://www-sst.informatik.tu-cottbus.de/GD/erlinlog.html



(a) Fruchterman-Reingold model

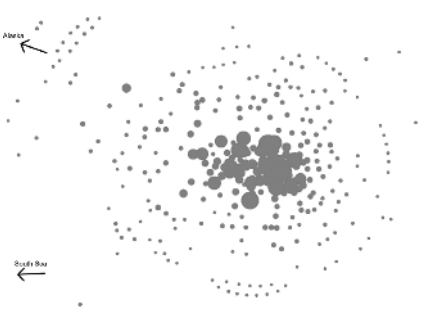


(b) Node-repulsion LinLog model

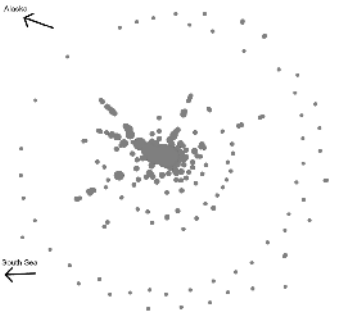


(c) Edge-repulsion LinLog model

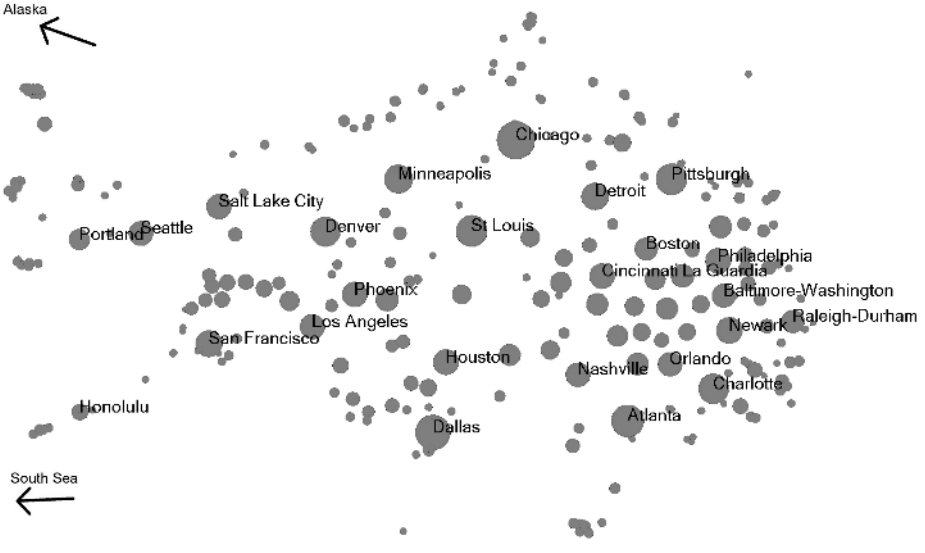
Fig. 3. Friendship network (33 nodes, 147 edges). Double edges correspond to reciprocated relationships, single edges to non-reciprocated relationships.



(a) Fruchterman-Reingold model



(b) Node-repulsion LinLog model



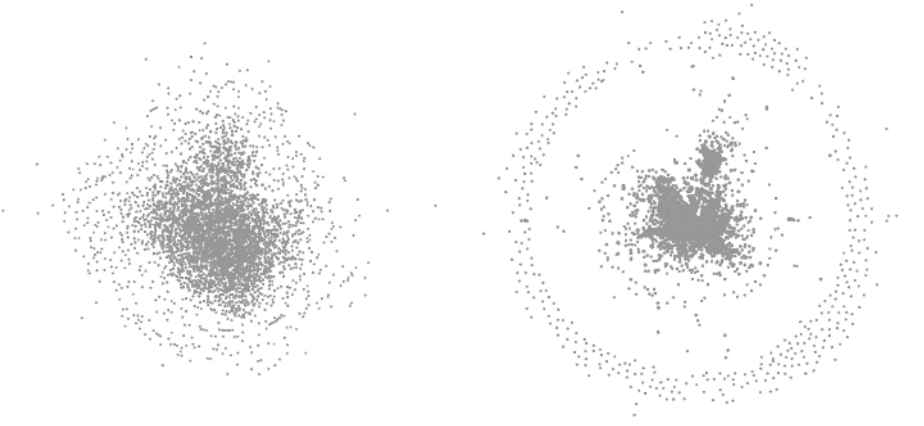
(c) Edge-repulsion LinLog model

Fig. 4. Direct flights between US airports (332 nodes, 2126 edges). The airports in Alaska and the South Sea (e.g. Guam) are omitted to improve readability.

4.2 Real-World Graphs

The graphs in Fig. 3 to 5 were obtained from the Pajek project[2]. In the drawings of the Fruchterman-Reingold model (Fig. 3a to 5a) and the node-repulsion LinLog model (Fig. 3b to 5b), nodes with high degree are placed in the center, and nodes with low degree near the borders. So the positions of the nodes mainly reflect their degree.

[2] http://vlado.fmf.uni-lj.si/pub/networks/data/



(a) Fruchterman-Reingold model

(b) Node-repulsion LinLog model

(c) Edge-repulsion LinLog model

Fig. 5. Hyperlinks between terms in the Online Dictionary for Library and Information Science ODLIS (2896 nodes, 18238 edges)

In Fig. 3, only the edge-repulsion LinLog drawing clearly reflects that there are two groups of friends – the left group around Chris and Rick and the right group around Steve and Irv – which are mainly connected by Upton and Dan.

Figure 4c shows that the edge-repulsion LinLog model discovers (roughly) the relative geographical locations of the US airports from the airline routing graph. Besides providing insights into the structure of the airline routes, this example impressively shows that the LinLog model can discover non-obvious knowledge in graphs.

The edge-repulsion LinLog drawing of the Online Dictionary for Library and Information Science (ODLIS) is shown in Fig. 5c, but the VRML file on the supplementary web page[3] gives a better impression how well semantically related terms are grouped on all scales. Such grouping is useful e.g. for discovering the global topic areas (like publishing, printing, computer science, etc.), identifying entry points for the exploration of topics, or finding semantically related terms even if they are not explicitly linked.

Drawings of three additional graphs are provided on the supplementary web page[3]. The grouping of papers from the Graph Drawing symposium in a drawing of the citation graph reflects research areas. However, there is some noise in the drawing because many papers have too few citations to be clearly assigned to a group. Again, such drawings have many applications, from getting an overview of the field with its subfields and landmark papers to identifying related papers even if they have no direct citation relationship. A drawing of Roget's thesaurus provides a nice map of (parts of) the English language by grouping semantically related categories, with benefits similar to the ODLIS visualization. The third drawing reflects how often files of a software system changed together in the development process. Because changes should be localized in subsystems, groups of files in this graph help to decompose the system into subsystems or to improve an existing subsystem hierarchy.

5 Conclusion

Cut-based measures for the coupling of subgraphs should be normalized with the size of the subgraphs to avoid biases. For graphs with nonuniform degrees, the number of edges is often a more appropriate measure of the size of subgraphs than the number of nodes. (For uniform degrees, both are equivalent.) Accordingly, energy models should use edge repulsion instead of (or in addition to) node repulsion to avoid dense accumulations of nodes with high degrees. In drawings of one such energy model, called edge-repulsion LinLog, the distance of groups of nodes is approximately inversely proportional to their coupling. Drawings of this energy model can provide deep and useful insights into the structure of real-world graphs from various domains, which are not possible with previous energy models.

References

1. Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.
2. Charles J. Alpert and Andrew B. Kahng. Recent directions in netlist partitioning: A survey. *Integration, the VLSI Journal*, 19(1-2):1–81, 1995.

[3] http://www-sst.informatik.tu-cottbus.de/GD/erlinlog.html

3. Josh Barnes and Piet Hut. A hierarchical O(N log N) force-calculation algorithm. *Nature*, 324:446–449, 1986.
4. Jim Blythe, Cathleen McGrath, and David Krackhardt. The effect of graph layout on inference from social network data. In *Proc. GD 1995*, pages 40–51. Springer-Verlag, 1996.
5. Ron Davidson and David Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15(4):301–331, 1996.
6. Edmund Dengler and William Cowan. Human perception of laid-out graphs. In *Proc. GD 1998*, pages 441–443. Springer-Verlag, 1998.
7. Peter Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.
8. Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software – Practice and Experience*, 21(11):1129–1164, 1991.
9. Pawel Gajer, Michael T. Goodrich, and Stephen G. Kobourov. A multi-dimensional approach to force-directed layouts of large graphs. In *Proc. GD 2000*, pages 211–221. Springer-Verlag, 2001.
10. Stefan Hachul and Michael Jünger. Drawing large graphs with a potential-field-based multilevel algorithm. In *Proc. GD 2004*, pages 285–295. Springer-Verlag, 2004.
11. Kenneth M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17(3):219–229, 1970.
12. David Harel and Yehuda Koren. A fast multi-scale method for drawing large graphs. In *Proc. GD 2000*, pages 183–196. Springer-Verlag, 2001.
13. Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31(1):7–15, 1989.
14. Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, 2004.
15. Tom Leighton and Satish Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Proc. 29th Annual Symposium on Foundations of Computer Science (FOCS 1988)*, pages 422–431. IEEE, 1988.
16. S. Mancoridis, B. S. Mitchell, C. Rorres, Y. Chen, and E. R. Gansner. Using automatic clustering to produce high-level system organizations of source code. In *Proc. 6th IEEE International Workshop on Program Comprehension (IWPC 1998)*, pages 45–52. IEEE, 1998.
17. Andreas Noack. An energy model for visual graph clustering. In *Proc. GD 2003*, pages 425–436. Springer-Verlag, 2004.
18. Andreas Noack and Claus Lewerentz. A space of layout styles for hierarchical graph models of software systems. In *Proc. 2nd ACM Symposium on Software Visualization (SoftVis 2005)*, pages 155–164. ACM, 2005.
19. Aaron J. Quigley and Peter Eades. FADE: Graph drawing, clustering, and visual abstraction. In *Proc. GD 2000*, pages 197–210. Springer-Verlag, 2001.
20. Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
21. Chris Walshaw. A multilevel algorithm for force-directed graph drawing. In *Proc. GD 2000*, pages 171–182. Springer-Verlag, 2001.
22. Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.