Visually mining relational data

Guy Melançon, Yves Chiricota

Corresponding author:

Guy Melançon
CNRS UMR 5800 LaBRI
Université Bordeaux I
351 Cours de la Libération
33405 Talence Cedex
France

Email Guy.Melancon@labri.fr
Tél. +33 631 396 671 / +33 540 003 611
Fax +33 540 006 669

Second author :

Yves Chiricota
Département de mathématiques et d'informatique
Université du Québec à Chicoutimi
555, boul. de l'Université
Chicoutimi (Québec)
Canada G7H 2B1

Email Yves_Chiricota@uqac.ca

Tel. +1 418 545 5011 Ext. 5651
Fax +1 418 545 5017

Manuscript received May 2007, revised May 2007

Visually mining relational data

Yves Chiricota [1], Guy Melançon [2]

Abstract – *Mining relational data often boils down to computing clusters, that is finding sub-communities of data elements forming cohesive sub-units, while being well separated from one another. The clusters themselves are sometimes termed "communities" and the way clusters relate to one another is often referred to as a "community structure".*
Methods for identifying communities or subgroups in network data is the focus of intense research is different scientific communities and for different purposes. The present paper focuses on two novel algorithms producing multilevel community structures from raw network data. The two algorithms exploit an edge metric extending Watts's clustering coefficient to edges of a graph. The full benefit of the method comes from the multilevel nature of the community structure as it facilitates the visual interaction and navigation of the network by zooming in and out of components at any level. This multilevel navigation proves to be useful when visually exploring a network in search for structural patterns.

Keywords: *Network analysis, Visual Interaction, Graph Mining*

I. Introduction

Visual interaction is a powerful approach to assist the analysis and exploration of data. Exploration is actually a complex process since the user often needs to "see" the data before a strategy can be defined for tackling the problem to solve [1]. Visual assessment of cluster tendency [2] and interactive nearest neighbor search [3] are typical examples where visualization and interaction actively assist the data analyst. When exploring the data, a particular view might provide ideas as to what type of patterns can be searched and understood. Visualization does not only assist the analysis but actively contributes to its progress. As a particular case, the visual inspection of a network supports the analysis of its community structure and helps to answer questions concerning prominent actors or subgroups. Once a subgroup has been identified, and when the subgroup moreover appears as such within the visualization, it can be zoomed in to allow for a more detailed inspection of its own dynamics.

Network analysis and visualization has been the focus of fertile research for many years [4] [5] and many aspects contribute to make a graph visualization more effective at helping the exploration of its underlying data. Graph Drawing[1] is an active research area focusing on the design of layout procedures, where algorithms are mainly targeted at producing aesthetic drawings while being scalable [6] [7]. However, laying out a graph only forms one of a series of phases in the visualization process (see Fig. 1 below). Interaction also plays an essential role giving the user the ability to discover knowledge from the data representation. When it comes to interacting with very large graphs, the interaction is subsumed to the possibility of dealing with the entirety of the data while being able to build and maintain a coherent mental map. That is what scalability is mainly concerned with, although it is also linked to more algorithmic and technical questions. When dealing with relational data, scalability can be tackled through graph clustering, allowing to present the user with an abstract and more readable view of the data, helping her/him to navigate the whole information space while having a reduced number of graphical items to deal with at a time. Graph clustering is also sometimes termed "community identification", an expression originally coined by Newman, stemming from the analysis of social network [8] [9] and emphasizing the fact that an algorithm should be able to find clusters of elements sharing common properties. Graph clustering, or community identification based on the topology of the network, is a major question we will be looking at in this paper. Categories into which graph clustering techniques can be put have been surveyed by several authors [10] [11] [12] [13] [14] and from different points of view: VLSI design, data mining or graph theory, e.g.

Figure 1 gives a high-level view of the processes involved in visualization. The diagram describes visual knowledge discovery as the result of pipelined data transformation stages. In a sense, visualization embraces the whole pipeline although the "viewable" elements are only output during the later stages. Indeed, the usability of the visualization depends on its ability to "translate" the properties of the data, thus initially requiring strong data analysis. Clustering appears here as a filtering and enrichment step relying on data analysis and carrying onto the visualization mapping. It comes as a fruitful strategy to answer questions or tasks

[1] See the URL www.graphdrawing.org, for instance.

concerned with connectivity, as opposed to attribute-based tasks or lower-level topology-based task such as finding nodes with higher degrees. Indeed, clusters group elements that are "close" to one another - clustering primarily focuses on the notion of proximity. Once clusters have been defined, additional knowledge can be inferred by looking at how they connect with one another.

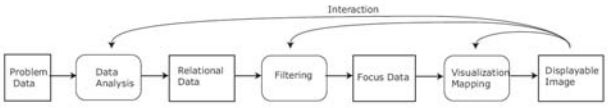

Fig. 1. Visualization as a high-level dataflow pipeline. The diagram is adapted from [15]; Chi's data state reference model [25] offers another point of view on the visualization process; see also van Wijk [26] for a "non-linear" version of the pipeline (but rather described as a dynamic system) targeted at usability taking user effort and acquired knowledge into account.

The different stages in Figure 1 must not be seen as isolated closed boxes, but rather as open processes acting on each other, particularly through user interaction (depicted by the return arrows). The visual inspection of the data might, for instance, help to evaluate the number q of "natural" clusters there might be, bringing the user back to the data analysis stage [15]. Metrics computed from different properties of the data (during the data analysis phase) can be used to map data elements to graphical cues, with the hope that visual patterns will emerge and ideally reveal structure. Typically, the user will want to filter out elements based on their metric value to look at the backbone structure of the network [16]. This filtering operation requires however some knowledge on the distribution of the metric within the data set under study and thus relies on the data analysis stage - in order to filter out the right proportion of data elements.

The approach we propose to study in this paper exploits metrics as a central ingredient to graph clustering. It aims at producing a clustering of a network in a reasonably short time in order to facilitate its visual and interactive inspection. It combines metric computation together with filtering, based on a canonical interpretation of a metric according to which the value of a graph element corroborates its relative importance in the network. This idea has revealed to be fertile in numerous situations [17] [16] [18] [19] [20]. Botafogo [21] had applied a similar approach to collections of hypertext documents. Henry [22] reported simple filtering techniques based on node attributes. The work by Newman and Girvan [23] [8] [9] also enters this perspective (see Sections II.2 and II.3).

We cluster graphs by filtering edges based on their strength metric, as originally introduced in [24] and [18]. In a sense, the strength of an edge can be seen as dual to its centrality - as defined by Newman [23] [8] (see Section II.2). After some edges have been deleted, the connected components appear as candidate clusters.

We shall look at two different filtering strategies for defining communities or clusters. The first strategy ("MinDisconnect", see Section III.1) performs a fine-grained filtering of edges in a manner similar to Girvan and Newman [8, Section II.B]. Weaker edges are filtered out first, almost one by one, leading to a detailed hierarchical decomposition of the whole network. The second method ("MQ", see Section IV) applies a threshold on edge strength, simultaneously filtering out a whole subset of weak edges. The threshold is selected in order to provide a "best possible" clustering with respect to a quality measure first introduced in [27, Section 3.3] in the context of reverse software engineering. This second method can be iterated on each cluster to obtain a hierarchical clustering of the original graph. As pointed out earlier, the full benefit of our approach comes not only from a simpler view of the graph, but also from the possibility to interactively navigate and explore a whole hierarchy of subgraphs.

II. Metric-based clustering

II.1. Clustering vs Drawing

Visualizing smaller graphs does not mandatorily require clustering. Even when dealing with average size graphs, drawing algorithms may sometimes be efficient at producing readable and informative views of graphs. Many algorithms however require graphs to share special properties, such as being trees or being planar (no edge crossing), having no directed cycles, etc. When a graph does not possess any special property, we are essentially left with one type of drawing algorithms, namely the force-directed layouts [6, Chap. 10] [7, Chap. 4].

Figure 2 (left) shows the layout of a graph using such a drawing algorithm, confirming our assertion on the readability of the layout for smaller graphs. Unfortunately, these techniques are somewhat inefficient for the visualization of larger graphs, either because their time complexity is too high or simply because they are unable to produce readable views from a large number of nodes and edges, as show the middle image in Figure 2. However, by clustering a graph we can offer the user an abstract view of the original graph consisting of a much smaller number of items, while being able to read patterns formed by sub-components. In turn, the set of components identified by the clustering process itself forms a graph that can most of the times be drawn using force-directed methods because of its smaller size, as confirmed by the image on the right of Figure 2 - high-level nodes actually contain subgraphs. This downsizing step sometimes requires the clustering technique to be re-applied on a component to further improve its readability and reduce its size. In doing so, we actually compute a hierarchy of subgraphs ultimately offering the user a multilevel view of the original data. In ideal cases, as we shall see with the air transportation network example (Section IV.3),

the various levels or scales present in the hierarchy can reinforce the interpretation of the original data.

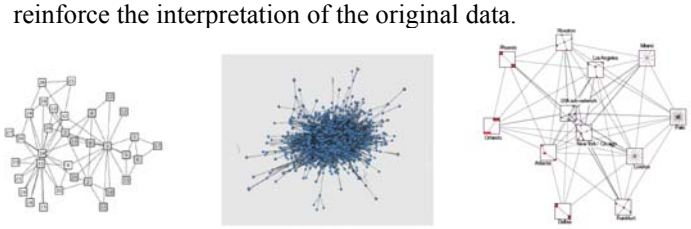

Fig. 2. Example layouts of force-directed algorithm on a small graph (left) and a larger graph (middle). The graph on the right has been clustered and then laid out using a force-directed method. (See also Fig. 4, Fig. 6 and Fig. 14.)

II.2. Metrics

Graph clustering is most of the time guided by topological properties. The methods we discuss in this paper rely on the computation of a metric for each edge or node of the graph. We shall now introduce notations that will be used throughout the paper. Let $G = (V,E)$ denote a graph with nodes V and edges E, and write $n = |V|$ and $m = |E|$. An *edge metric* is a map $E \rightarrow \mathbf{R}$ which computes a numerical value for each edge in the graph (here **R** stands for the set of real numbers). A node metric $V \rightarrow \mathbf{R}$ can be defined similarly. The value associated with a given edge e usually results from some algorithm.

One of the simplest example of a node metric is the one computing the *degree of a node* (number of neighbours). A more interesting example of a node metric is the *clustering coefficient* $c(v)$ of a node v introduced by Watts [28]. The clustering coefficient computes a ratio reflecting how much neighbours of the node v are likely to be connected. More precisely, let N_v denote the set of neighbours of v and let $r(N_v,N_v)$ denote the number of edges between neighbors of v. Thus, the clustering coefficient associated with v is computed as:

$$c(v) = \frac{r(N_v, N_v)}{|N_v|(|N_v|-1)/2} \qquad (1)$$

Observe that nodes belonging to a clique (a complete graph) have a clustering coefficient of 1 since any two neighbors of a node v are connected. Incidentally, the higher the clustering coefficient of a node, the higher the probability that it belongs to a clique. Watts had introduced the clustering coefficient of nodes in order to identify a key property shared by the so-called "small-world networks", a class of graphs that recently became the focus of intensive research. Small world networks indeed appear as more realistic models than random graphs, as defined by Erdös and Rényi [29] [30] (see also [31]). A graph qualifies as a small world graph if the average distance between any two nodes is low while the average clustering coefficient of nodes is high - when compared with Erdos-Renyi random graphs with the same number of nodes and edges. That is, a small world graph can be seen as formed of denser neighborhoods connected by pivot nodes so that routing a message in the network from end to end requires only a few steps through incident edges. Instances of small world networks appear in various application areas; the air traffic network we study in Section IV.3 is a typical example. For more examples and a survey on the subject see [32] [33] and [34].

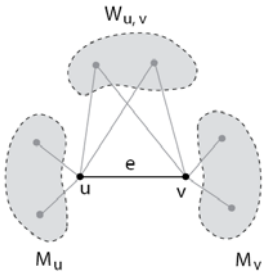

Fig. 3. Neighbors of nodes incident to an edge partition into three distinct subsets Wu;v, Mu and Mv.

We now recall the definition of the strength metric introduced in [24] [18] which can be seen as an extension of Watts' clustering coefficient to edges of a graph. This metric intuitively measures how much an edge is likely to connect two distinct communities in the network, or on the contrary contributes to the cohesiveness of its own community. In other words, weaker (low value) edges correspond to passageways between distinct communities. In order to define this metric, we need to introduce useful notations. Given a node $u \in V$, we denote by $N_G(u)$ or simply by N_u the set of neighbours of u in G. Consider an edge $e = \{u, v\}$ and define $W_{u,v} = N_u \cap N_v$, $M_u = N_u \setminus N_v$, and $M_v = N_v \setminus N_u$. That is, M_u is the set of neighbors of u that are not neighbors of v (similarly for M_v) and $W_{u,v}$ denotes the set of neighbours common to both u and v. Observe that $W_{u,v}$, M_u and M_v partition the set of neighbour nodes $N_u \cup N_v$ into three distinct subsets (see Figure 3).

First, we define the ratio:

$$\gamma_3(e) = \frac{|W_{u,v}|}{|M_u| + |M_v| + |W_{u,v}|}$$

which computes the proportion of 3-cycles (cycles of length 3) containing the edge e. The ratio γ_3 goes back to [35] and is known as the *Jaccard index*. It can also be defined as $(|N_u \cap N_v|) / (|N_u \cup N_v|)$. Note that we have a maximum number of 3-cycles containing e when $W_{u,v} = N_u = N_v$ (and thus $M_u = M_v = 0$), that is, when u and v have all their neighbors in common, in which case $\gamma_3(e) = 1$. A weighted version of the Jaccard index was given by Tanimoto [36] (see also [37]).

Next, given distinct subsets of nodes $A,B \subset V$, let $r(A,B)$ be the number of edges connecting nodes between A and B (but not connecting nodes of a same subset A or B). That is, $r(A,A) \leq |A| \cdot |B|$. We use the

same notation $r(A,A)$ to denote the number of edges between nodes in a subset A. In this case, we have $r(A,A) \le |A| \cdot (|A|-1)/2 = \binom{|A|}{2}$ and the maximum is reached if the subgraph induced on A is a *clique* (a complete graph). Define

$$\gamma_4(e) = \frac{r(M_u, W_{u,v}) + r(M_v, W_{u,v}) + r(M_u, M_v) + r(W_{u,v}, W_{u,v})}{|M_u| \cdot |W_{u,v}| + |M_v| \cdot |W_{u,v}| + |M_u| \cdot |M_v| + \binom{|W_{u,v}|}{2}}$$

so that $\gamma_4(e)$ computes the proportion of 4-cycles containing the edge e. The *strength metric* $\gamma(e)$ is defined as

$$\gamma(e) = \gamma_3(e) + \gamma_4(e).$$

Note that, strictly speaking, we should write the metric as $\gamma_G(e)$ and explicitly refer to G. This can be of importance when computing the strength of an edge in a subgraph induced from G. This full notation will be useful in Section IV.2.

The strength of an edge $\gamma(e)$ is thus a positive value with $0 \le \gamma(e) \le 2$. Intuitively, weak edges either have a small associated subset $W_{u,v}$ or are contained in only a few cycles of length 4 (or both). In other words, there are few edges connecting any two subsets chosen among $W_{u,v}$, M_u or M_v. As a consequence, there is a high probability that the edge e must be traversed when traveling between any two of these subsets. Having this in mind, we can interpret the strength metric as a *centrality* measure in a network, even though it relies only on close neighbourhoods. Centrality measures are usually defined for nodes of a network reflecting the overall importance of a node [38] (see also [39]). As a special case, the *betweenness* centrality of a node computes the number of shortest paths between pairs of other vertices which run through v. Girvan and Newman recently extended betweenness centrality to edges, where the betweenness of an edge e is simply the number of shortest paths between pairs of vertices that run along e [8]. Their use of the edge betweenness is much similar to our use of edge strength (see algorithm MinDisconnect presented in a forthcoming section).

II.3. Variations on strength

Goldberg et al. [40] independently studied variations on edge centrality pursuing a different goal than us in a bio-informatics context. They introduced what they call the geometric index for an edge e:

$$G(e) = \frac{|N_u \cap N_v|^2}{|N_u| \cdot |N_v|}.$$

Observe that we can go from the geometric index to Jaccard's as can be deduced from the fact that $|N_u \cup N_v| = |N_u| + |N_v| - |N_u \cap N_v|$ (Goldberg *et al.* actually also discuss the Jaccard index). The Jaccard metric however assigns a high value to an edge even if one of its ends is connected to few other nodes, which is not the case for the geometric index. We have experimentally noticed that the strength metric γ_G is not correlated with any of these two metrics. This actually reinforces the relevance of strength metric as a significant ingredient of our clustering algorithm. Golberg *et al.* also consider two other candidate metrics (see [40] for more details).

II.4. Complexity

The strength metric of an edge can be computed in a time proportional to $|A| \cdot |B|$ where A and B are any two of the neighborhoods $W_{u,v}$, M_u or M_v, leading to a worst time case of $O(|V|^2)$ since $\deg(v) \le |V|$ for all nodes $v \in V$. When G has bounded degree, that is when $\deg(v) = O(1)$ for all $v \in V$, $\gamma_G(e)$ can be computed in constant time, which is a reasonable hypothesis to make when G is sparse, although sparse graphs may contain a few nodes with a degree of order $O(|V|)$.

So assuming nodes in G have bounded degree $<< n$, the strength of all edges becomes computable in time $O(|V|)$. Note that theoretically speaking, the strength of all edges requires $O(|V|^2)$ even if it only contains a limited number of nodes with degree $O(|V|)$. However, in practice having $\deg(v) = O(|V|)$ for a small number of nodes remains under the constraints of interactive mining. Conversely, computing the betweenness centrality requires $O(|V| \cdot |E|)$ [Brandes, 2001] even when G has bounded degree. Observe that when G is sparse, $O(|V| \cdot |E|)$ is equivalent to $O(|V|^2)$. This actually makes our strength metric a real competitor to betweenness centrality.

III. Edge filtering: a basic navigation strategy

III.1. Mindisconnect

The strength metric admits an intuitive interpretation of weaker edges: since their neighbourhoods are not tightly connected, these edges mostly appear as passageways between denser neighbourhoods. Longer chains of weak edges may also exist. In this latter and extreme case, there might be edges of null strength. Conversely, strong edges sit in the middle of tightly connected neighbourhoods. This interpretation leads to a method for hierarchical clustering of connected graphs based on the removal of weak edges.

Observe that arbitrarily deleting the weakest edge might not straightforwardly disconnect the graph, depending on the topological surroundings of the edge. On the contrary, the deletion of too many edges may

disconnect the graph abruptly into several connected components. The idea behind the methodology we are about to describe is to attentively follow this "disconnection" process and halt as soon as the graph is disconnected in order to build a hierarchy. The process is conducted solely depending on the strength metric. Given a graph G, and having computed the strength metric γ_G for all edges in G, we then find the minimal value τ such that removing edges whose strength value is less than τ disconnects the graph. The process is then applied recursively on each connected component until a stopping condition is reached.

Here is a formal description of this algorithm. Write G_τ to denote the graph obtained from G by removing edges of strength lower than τ. Compute the connected components of G_τ and denote them as $C_1, \ldots, C_q$. Then $\Pi_\tau = (C_1, \ldots, C_q)$ forms a clustering of the original graph G. That is, the subsets C_i are pairwise distinct $C_i \cap C_j = 0$ for all $i \neq j$) and cover the whole set of nodes ($C_1 \cup \ldots \cup C_q = V$). More formally, what we have just defined is a map $[a, b] \rightarrow \wp(G)$ going from the metric range $[a, b]$ to the set $\wp(G)$ of all possible clusterings or partitions of G. Assume G is connected so it possesses a single connected component. Now, define τ_{disc} as the *minimum* strength value such that G_τ has more than one connected component. Obviously, τ_{disc} exists since G_b has $|V|$ connected components. Let $\Pi_{\tau disc}$ denote the clustering $(C_1, C_2, \ldots, C_q)$ obtained by filtering out edges of strength $\leq \tau_{disc}$. A hierarchical clustering is computed by iterating the procedure over each cluster Ci unless C_i is a clique in which case we set $q = 1$ and $C_1 = G$.

Algorithm 1 MinDisconnect($G(V,E)$, γ_G)

Compute γ_G for all edges $e \in E$
Calculate τ_{disc} = the minimum metric value τ such that G_τ has more than
one connected component
if τ_{disc} = maximum value reached by γ_G
    return
else
{
    Let $(C_1, C_2, \ldots, C_k) = G\tau_{disc}$
    For each C_i
    {
        Define the subgraph $G(C_i)$ induced from $G = (V,E)$
        Call MinDisconnect($G(C_i)$, γ_{Ci})
    }
}

This algorithm proceeds in a top-down (or divisive) manner and implicitly constructs a (general) tree in which leaves correspond to clusters of vertices of a graph G. Indeed, the process begins with a unique cluster (the initial graph), and "breaks" it into components, which will be in their turn broken down. Note that the strength metric is recomputed for each cluster occurring in the process, and as a metric associated with the graph G_{Ci}

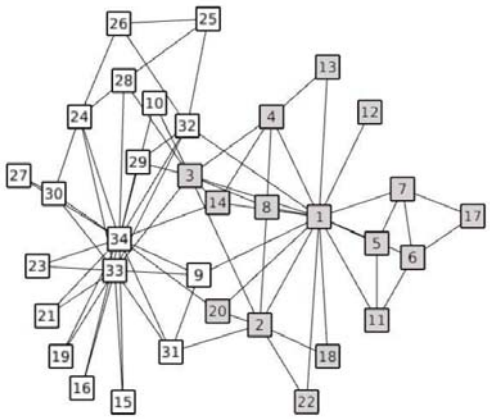

Fig. 4. A force-directed layout of Zachary's network showing connections between members of a karate club. White and greyed nodes indicate two groups that formed resulting from a disagreement between the club's officers (see [41]).

As a first (toy) example illustrating our method, we have applied MinDisconnect to the friendship graph borrowed from Zachary [41] (see Figure 4). Links in the graph correspond to friendship relations between members of a karate club. Figure 5 shows the cluster tree output by MinDisconnect. As the structure of the cluster tree indicates, node 12 has been isolated at the first iteration. This is no surprise since node 12 has degree one, and consequently the edge connecting it to the graph has null strength. Ignoring node 12, the algorithm comes up with two distinct groups of nodes that can be interpreted as sub-communities within the whole network. Zachary had studied the impact of a disagreement between two officers, resulting into two distinct subgroups within the club. Colors in Figure 4 indicate how the club was divided into these two sub-communities. A striking fact is how the friendship relations support the choice of individuals to go with one officer or the other (see [41] for more details).

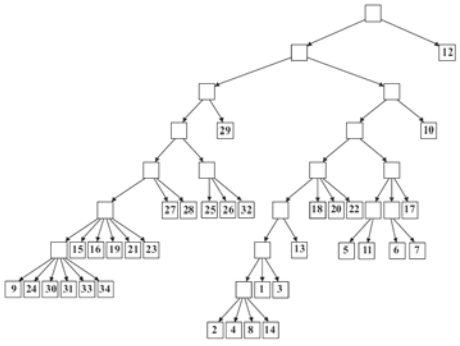

Fig. 5: the tree describes the cluster structure output by MinDisconnect. Apart from node 12 being isolated, the algorithm comes up with two distinct groups of nodes that can be interpreted as sub-communities within the whole network.

As a matter of fact, MinDisconnect was able to recover the two groups based on the topology of the friendship graph, but with one exception. Indeed, the only node that was misclassified is node 10 (and node 12). Observe however that node 10 sits at the top of the hierarchy indicating that its link to the community is weak, or put differently, that it sits on the border of the two communities. Conversely, clusters connect more intensively as we go down the tree.Please insert your figures with "inline wrapping" text style, as in this template (see Fig. 1).

We used this example to compare MinDisconnect with Girvan and Newman's approach (see [8, Sect. III.B]). Girvan and Newman however proceed by removing edges one by one, recomputing edge betweenness each time (although they only run over edges affected by the removal). Both methods identify two groups and slightly disagree with Zachary. They also disagree with one another, putting node 3 in different subgroups.

III.2. Complexity issues

Because MinDisconnect is recursive, and because it proceeds by filtering out edges, the number of times MinDisconnect is called is bounded above by $m = |E|$. Each call first computes the strength of edges and then finds a threshold value τ. Assuming the degree of nodes is bounded by a constant - which is a reasonable condition to require - the metric γ_G can be computed in time $O(n)$ (see Section II.4). Next, finding the "disconnecting value τ_{disc}" follows a divide and conquer strategy, recursively bisecting the range of the strength metric, involving at most a constant number of DFS searches on edges. Consequently, this step is also performed in time $O(n)$. Hence, the total time complexity is $O(mn)$, which reduces to $O(n^2)$ when G has bounded vertex degree. Without any particular condition on vertex degree, the worst case complexity is $O(n^3)$, making the technique applicable only to small graphs (with a few hundred nodes) to comply with interactivity. Note however that it is possible to further optimize the computation of strength on connected components by only recomputing strength values incrementally for edges at distance at most two from edges that are removed during the filtering process.

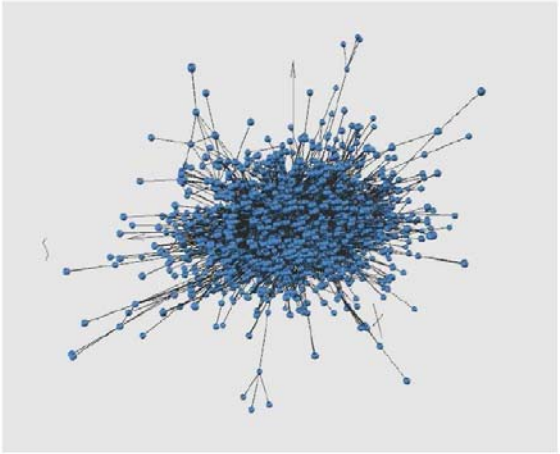

Fig. 6. Force directed layout of the word association network.

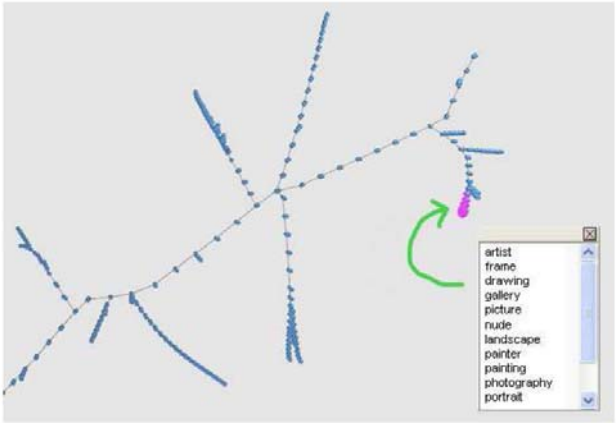

Fig. 7. The tree structure obtained after iterating MinDisconnect recursively.

III.3. Example : Word Association Network

We will now give a more elaborate example of the application of MinDisconnect to a network linking words according to spontaneous associations made by users. This network of associations between words is the result of an experiment conducted partly over the internet with interactive forms, and partly on-campus. People were asked to spontaneously reply a word when given a first one. Each person was asked to reply to 30 start words randomly selected from a larger subset. For the sake of coherence, the experiment was restricted to common nouns. The experiment was started using a small collection of common nouns. After a word had been given as a reply by a user, it became eligible as a start word in further experiments. This network thus collects different semantics between words - synonyms, homonyms, etc. It served as a common ground for a work gathering researchers from computer science, cognitive science and ergonomics. More details can be found in [42].

The original network we considered contained 10 000 nodes and about 100 000 edges. We performed our calculation on a single connected component

comprising 3 000 words and 7 000 edges. This network is scale-free and typically shows as a dense ball of wool , making any drawing algorithm useless when visually mining the graph, as confirmed by Figure 6.

As the example will show, the MinDisconnect algorithm is able to divide the network into semantically coherent pieces that can be further visually navigated and explored. Indeed, the strength metric acts just as we want by selecting stronger associations between subsets of words. Moreover, by calling MinDisconnect repeatedly on sub-clusters, we get a cluster tree as shown in Figure 7. The figure only shows part of the cluster tree laid out using a force directed algorithm. The thin structure of the tree results from the criteria we apply when defining sub-clusters $C_1, \ldots, C_q$. The algorithm is indeed sensitive to the smallest metric difference between two edges and will thus process them at different stages. In other words, a common computation step might filter out a single edge $e = \{u,v\}$ from cluster C, thus separating a single node u from it before reapplying MinDisconnect to sub-cluster $C - \{u\}$. Most often, MinDisconnect will discard a small subset of edges (less than 5). There are however regions that do form denser sub-trees of clusters. These are the regions of interest as they rely on stronger semantic associations.

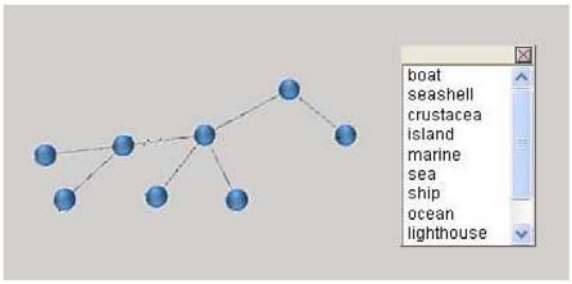

Fig. 8. A sub-cluster showing sea-related words.

The algorithm has been implemented within an interactive environment allowing for the selection of nodes from the layout. After selecting nodes, the user can recover the corresponding word list. Each leaf in the tree consists in a cluster typically containing two to four words. The user can thus examine larger subsets of words by selecting neighbor nodes or by simply selecting a small subtree as shown in Figure 7. The navigation offered by this clustering schema is used to pursue the study of this and other word association networks [43]. Figures 8 and 9 provide additional examples. Even non-expert users can assess the relevance of the clusters identified by MinDisconnect.

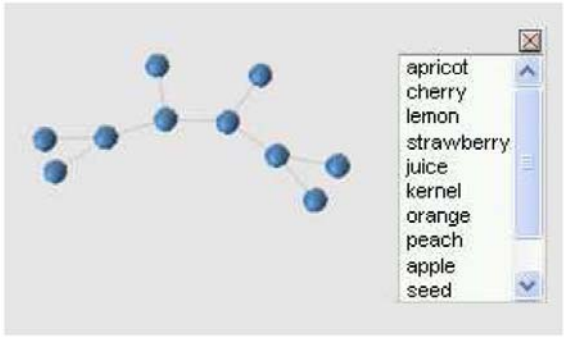

Fig. 9. A sub-cluster showing fruit-related words.

IV. Modularity quality

The algorithm presented in the previous section produces a fine-grained clustering by going through all possible values for edge strength, and by recursively selecting values that "minimally" disconnect subgraphs. The method we now describe differs from the previous one in that edges are filtered out in bulk. The filtering operation is operated with the hope that visual and structural simplification will help identify salient properties guiding further investigation. This filtering technique is even more useful if we allow the user to further examine each component and its subcomponents, and so forth. Applying the method recursively on each subcomponent leads to a hierarchical decomposition of the original graph, just as with "MinDisconnect" but leading to a coarser hierarchy.

IV.1. MQ: modularity quality of a clustering

Given a clustering $\Pi = (C_1, C_2, \ldots, C_q)$ of a graph G, a quality measure associated with the clustering Π (and the graph G) should provide a numerical value reflecting how *natural* the partition Π is with respect to the topology of the graph G. Naturality is used here to capture the idea that clusters should have high internal edge density, while having but a few external links (edges connecting nodes of different clusters).

Mancoridis *et al.* introduced such a function in the context of software reengineering [27]. This function, denoted as $MQ(\Pi,G)$ when applied to a graph G and a clustering Π, consists of two terms. The first term computes the mean value of edge density within clusters, and the second term appears as a penalty term computed from edge density ratios between clusters. More precisely, we have:

$$MQ(\Pi, G) = \frac{1}{q} \sum_{1 \le i \le q} \frac{r(C_i, C_i)}{\binom{|C_i|}{2}} - \frac{1}{\binom{q}{2}} \sum_{1 \le i < j \le q} \frac{r(C_i, C_j)}{|C_i| \cdot |C_j|}$$

Observe that $0 \le \frac{r(C_i,C_i)}{\binom{|C_i|}{2}} \le 1$ and $0 \le \frac{r(C_i,C_i)}{|C_i| \cdot |C_j|} \le 1$ so $MQ(\Pi;G)$ varies within [-1, 1]. The ratio computed by the first term measures how close clusters C_i are to cliques - the term $r(C_i,C_i)/\binom{|Ci|}{2}$ can actually be seen as an extension of Watts's clustering coefficient to clusters of the graph G. Similarly, the second term indicates how close edges between C_i and C_j are to a complete bipartite graph. In this case, the ratio $r(C_i,C_j)/|C_i| \cdot |C_j|$ could be interpreted as a dissimilarity between the sets C_i and C_j , by analogy to the link index introduced by Guha *et al.* [44] as part of the ROCK clustering algorithm.

Notations. Recall that Π_τ denotes the clustering induced from the threshold value τ (see section III.1). Note also that we can form a graph whose nodes correspond to the clusters $C_1, \ldots, C_q$, where edges connect two cluster nodes if there are edges in E connecting nodes in V between them. This graph is referred to as a *quotient* graph and is usually denoted as G / Π_τ . We save these notations for later use.

Clearly, the map $[a, b] \rightarrow \wp(G)$ defined in section III.1 does not cover the entire set $\wp(G)$. That is, filtering edges through all values τ will not exhaust all possible clusterings of the graph G. The hope here is that some values of τ may lead to clusterings of "good quality". To achieve this, we use MQ to survey the evolution of the clustering quality as τ varies. Put formally, we define a map $[a, b] \rightarrow \mathbf{R}$ giving the quality measure of the clustering induced from the threshold τ . More precisely, we compute $MQ(\Pi_\tau G)$ for all $\tau \in [a, b]$. Trusting MQ, we finally select a clustering of maximum quality. This can be achieved by properly sampling the range $[a, b]$, and for each threshold value τ compute the associated clustering τ and its quality measure. Thus the values $MQ(\Pi_\tau G)$ vary along a curve defined over the range $[a, b]$, and we seek to find its maximum and the corresponding threshold τ .

Figure 10 illustrates how MQ varies over the range of the strength metric applied to an example graph. As can easily be seen, the curve reaches its maximum value at $\tau \sim 1.7$. The graph used in this example is computed from a subset of the IMDB, the Internet Movie Database [2] . Nodes of the graph correspond to actors or actresses. Edges connect actors having played in a same movie or TV show. The graph is displayed on part (a) of Figure 11 and has been laid out using a force-directed algorithm.

This example exhibits specific properties due to its construction. Indeed, the graph was built by extracting actors having played with either Tom Cruise, Cameron Diaz or Jim Carrey, or with an actor having played with one of them, and so on up to a maximum distance of four links away. Consequently, the actors having played in a same movie all appear in a clique. Although this is not usual in a social network, it served as a good example and testbed for our methodology.

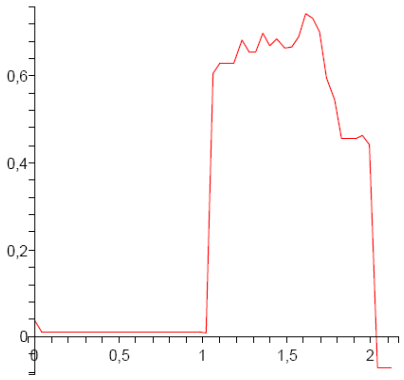

Fig. 10. The curve showing how MQ(¦τ ;G) varies with respect to τ .

Notice that the circled cluster in the left panel of Figure 11 is tightly connected, as confirmed on the middle image (b). Image (c) on the right provides an abstract view of the clustered graph with each cluster represented by a single node with size proportional to the number of nodes it contains, in a manner similar to [45]. Links between clusters are induced from links between nodes in the original graph and provide an overview of its topology. At first we might expect the cliques involved in that cluster to separate just like most of the other cliques do. This is not however the case, thus revealing intimate and closer links between actors of that subcommunity. Further analysis of the clusters shows they contain major actors such as Warren Beatty, Faye Dunaway, Dustin Hoffman and Jack Nicholson intuitively explaining why the links sitting in that neighbourhood are *stronger*.

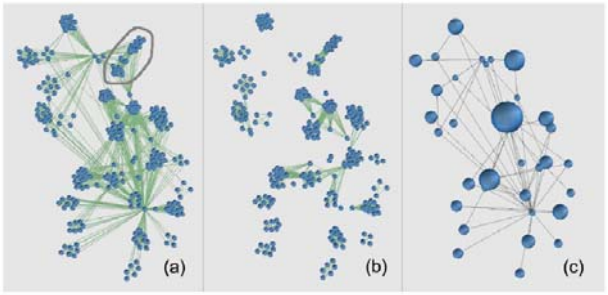

Fig. 11. Edge filtering applied to an IMDB subgraph.

More details on this introductory example can be found in [18]. We shall discuss another example in Section IV.3, requiring an extension of the strength metric to weighted graphs.

IV.2. Multilevel visualization and hierarchical clustering

As we pointed out earlier, iterating the method on each component isolated by the filtering process leads to a hierarchy of subgraphs. The process can be stopped using various halting conditions depending on a component size or variation on the strength of edges. For example, a component with edges all having a strength equal to 1 is obviously a clique so the process should not iterate further. Also, even though a component is not a clique, it might be the case that the maximum value reached by *MQ* is precisely when we take the component as a unique cluster.

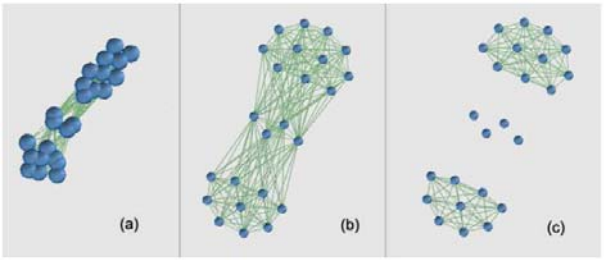

Fig. 12. Sub-clusters of the IMDB graph.

Part (a) of Figure 12 shows a close view of the circled subcomponent of the IMDB network in Figure 11. The middle image (b) is obtained by locally relaying out the cluster, providing a less constrained and more readable layout. The right image (c) shows s the effect of the filtering process when applied locally. Obviously the four nodes sitting in the middle of the image act as mediators between the top and bottom clusters. They have been identified as such by the strength metric, and the filtering process together with the *MQ* criteria selected a threshold extracting these mediator nodes while maintaining most of the edges in the two other clusters.

Let us describe the algorithm more formally. It takes as input a graph $G = (V,E)$, the strength metric and the quality measure *MQ* (although the algorithm is named after *MQ* here). We also introduce a variable p acting as a probability and used to properly sample the range $[a, b]$ over which γ_G varies. The variables τ_{max} and MQ_{max} are used to respectively store the metric value τ leading to a maximum *MQ* score. As mentioned before, the output can be coded as a hierarchy of subgraphs.

Algorithm 2 *MQ*-clustering($G(V,E)$, γ_G , *MQ*)

Compute γ_G for all edges $e \in E$
Set $\tau_{max} = 0$ and $MQ_{max} = 0$
Let p vary from 0 to 1 using a fixed increment ε
{
  Compute $\tau \in [a, b]$ such that the proportion of edges with strength $\leq \tau$ is p
  Filter the edges and define the induced subgraph G_τ
  Compute the connected components $(C_1, C_2, \ldots, C_q)$ of G_τ
  Define the clustering Π_τ
  Compute $MQ(\Pi_\tau, G)$ and compare with MQ_{max}
  Update τ_{max} and MQ_{max} accordingly
}
At this stage, we have selected the value τ_{max}. We also need to store
the corresponding clustering $\Pi_\tau = (C_1, C_2, \ldots, C_q)$ as well as the quotient
graph G / Π_τ

For each C_i
{
  Define the subgraph $G(C_i)$ induced from $G = (V,E)$
  Call *MQ*-clustering($G(C_i)$, $\gamma_{G(Ci)}$, *MQ*)
}

The description of the algorithm calls for a number of remarks. First, observe that the strength metric γ_G is recomputed at each stage in order to reflect the inner connectivity of a cluster after some of the edges have been filtered out. This is mandatory to avoid making decisions based on values inherited from the original graph. Indeed, the neighborhood of an edge and thus its strength $\gamma_{G(Ci)}$ in a cluster C_i might well be completely different from its neighborhood in the original graph G. This is even more true when considering a cluster deeply nested in the hierarchy. Second, the values $\tau \in [a, b]$ are sampled in a way such that the more compact regions are visited with more care. Running over the interval $[a, b]$ by linearly interpolating from a to b would lead to an inaccurate estimation of the *MQ* curve over $[a, b]$ (see [46] for more details). Finally, the complexity of the implied computations requires to fine tune the algorithmics and memory usage. We will not comment on these issues as they go beyond the scope of this paper and also intersect with future work (see section V).

One major benefit of this hierarchical clustering is the ability to navigate the network by zooming in and out of the subgraphs in the hierarchy, together with the possibility of navigating the network structure linking components with one another (at a given level of the hierarchy). The next section describes a use case where this feature enabled us to visually analyze the

worldwide air transportation network, allowing experts to gain insight on its dynamics [47].

IV.3. Worldwide air transportation network

Our second example application shows how the *MQ* clustering strategy helps to grasp the community structure of the worldwide air transportation network, and brings forward a visual representation of its nested sub-communities. The multiscale nature of this organization into sub-communities makes it necessary to be able to navigate through the hierarchy of subgroups. In this regard, our approach is complementary to the work by Guimerà *et al.* [48], who have studied the worldwide air transportation network by analyzing its centralities without the help of any visualization whatsoever. The analysis of the network supported by our method has been conducted with the help of experts who were able to assess its usefulness. The conclusions they were able to draw from the visualization and interactive exploration of the air transportation network can be accessed in [47] [3] .

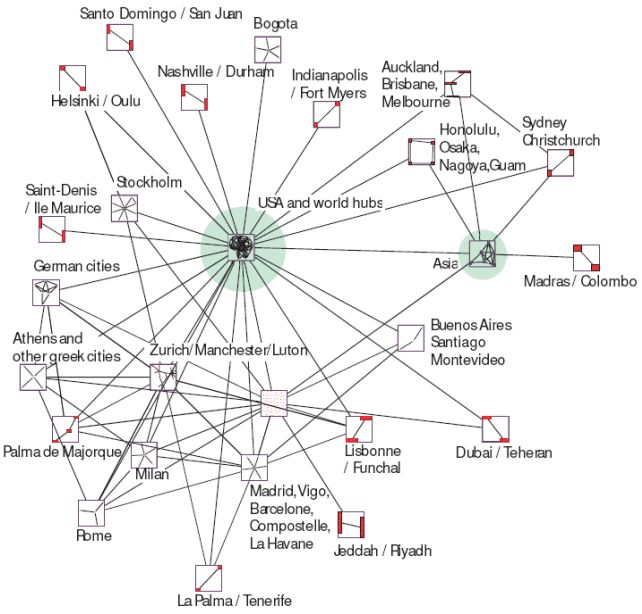

Fig. 13. Top level view of the hierarchically clustered worldwide air traffic network.

Obviously, a visualization of the airline connections drawn on a traditional 2D geographical representation of the earth is worthless, as the network contains a large amount of connections. Moreover, this type of representation could only reveal subgroups defined around regional strategies. The challenge here is to unveil sub-networks emerging from non territorial logics. Indeed, more and more companies today define their routes based on partnerships and logistic hubs installed in specific airports. The visual and multilevel analysis of the network allowed experts to identify sub-communities and suggested hypotheses explaining the overall organization of the worldwide network. The original dataset acquired from IATA comprises 1347 cities and 16526 connections. For the sake of readability, the images shown here only reveal a part of the total network consisting of connections involving at least 300 000 passengers in 2001, resulting in a network of 294 nodes and 1049 edges.

The multiscale representation shown in Figure 13 splits the network into several components and layers. Observe how *MQ* helped to go from a 294 nodes/1049 edges graph to a 25 nodes/90 edges graph providing a more readable representation of the data. The topology of the quotient graph globally is star-like with the exception of the bottom left region, which appears to be more densely connected. Clusters of cities in periphery strongly depend on the world hubs to connect to other parts of the world and do not show a strong participation in the overall exchanges. That however does not imply that they do not play such a role at another level. Investigation of deeper level components is required.

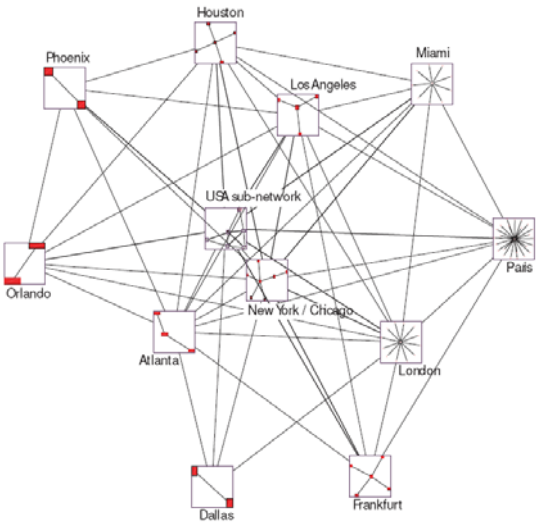

Fig. 14. World hubs form into a sub-community sitting at the center of the star-like topology of the worldwide air transportation network (Fig. 13).

A quick exploration of the center component reveals that it contains all principal air hubs: Atlanta, New York, Chicago, London and Paris are all there (Fig. 14). These capitals clearly drain most of the international traffic and impose routes to fly the world around because of airline partnerships (economical logic). Asia clearly stands apart from these core hubs because of strong territorial ties endorsed by national Asian airline companies (territorial logic). A large part of the European cities appear at the bottom left. This part of the network is more densely connected and confirms the organization of the European network on the fringe of the international hubs (the center component). Figure 15 shows a close-up of the sub-network organized around Zurich, Manchester (UK) and Luton. The other destinations in this sub-network, together with Luton (a

minor airport in the UK), clearly indicate tourism as a major aggregating force in this neighborhood. For a more detailed examination of the network and its subcomponents, the reader should consult [47].

Observe that the Zurich/Manchester/Luton sub-network does not admit any sub-community at a lower level. However, the center component containing the western world air hubs can be navigated four more levels down after entering the "USA sub-network" component (Fig. 14). Figure 16 shows a view of the USA sub-network. The central cluster of this sub-network can itself be explored two more levels down and reveal how Cleveland, Denver, Detroit, Minneapolis, Philadelphia, Tampa and other cities of similar "importance" organize.

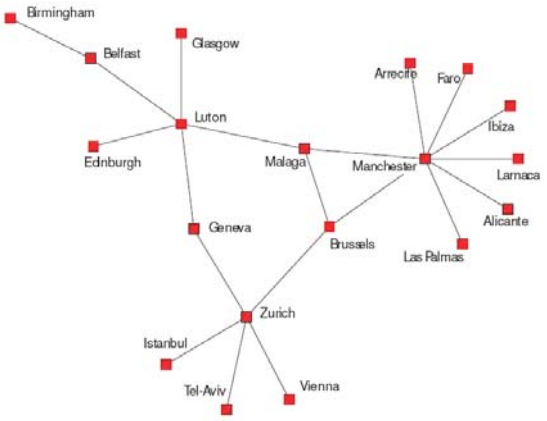

Fig. 15. Zurich/Manchester/Luton European sub-network.

IV.4. Taking the air traffic into account

In order to build the representation of the worldwide air transportation network, we had to extend the metric originally defined in [18]. Indeed, the network came equipped with weights counting the number of passengers that had traveled between each pair of cities [4]. The data did not sort passengers according to airline companies they had traveled with, but added passengers flying from different airports of a given city. For instance, the Paris traffic with any other city was obtained by summing departures/arrivals from/to the Charles de Gaulles or the Orly airports.

The frequency histogram in Figure 17 showing how passenger weights distribute over the network, reveals a scale-free phenomenon. This does not come as a surprise since only a few airports concentrate most of the world passenger traffic, or put differently, that a large amount of passengers go through only a few airline routes, as confirmed by the community structure revealed in Figure 13. The maximum (normalized) weight is however rather low with a value of 0.0073 for the Pusan-Seoul connection with a total of approximately 6 million passengers, compared to a total of 822 million (recall we only considered connections with more than 300 000 passengers per year).

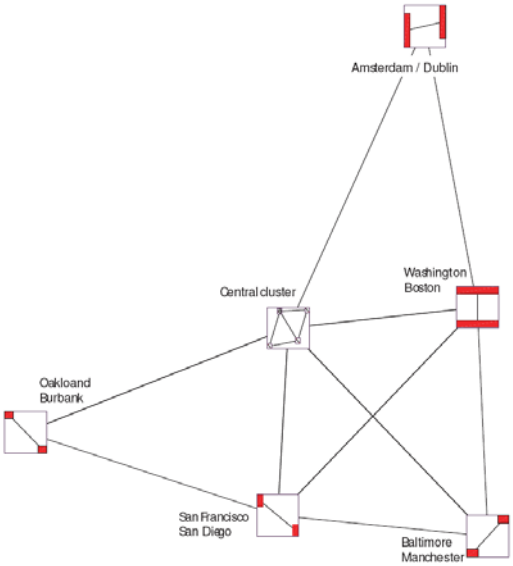

Fig. 16. Third level USA sub-network (sitting at the center of Fig. 14).

The clustering based on the strength metric defined in Section IV.2 does not take into account the weight of edges. Intuitively, we would like the algorithm to prefer edges with higher weight among those with equal strength metric. This can be achieved by computing for each edge its *weighted strength* $\gamma(e) = \gamma(e) \cdot \omega(e)$ obtained by multiplying its strength metric $\gamma(e)$ by its weight $\omega(e)$, as defined in the next paragraph.

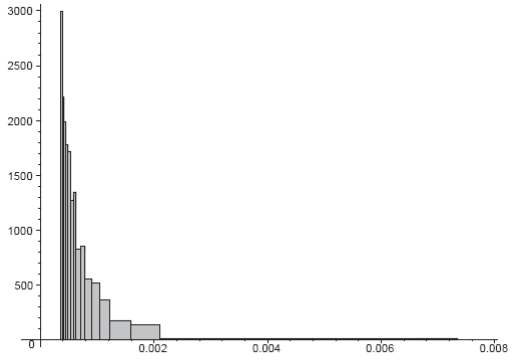

Fig. 17. Frequency distribution of weights for passenger connections.

This calculation requires to normalize weights to [0, 1]. This normalization must be done with care. Let e be an edge supporting a passenger traffic T. We assign edge e a weight $\omega(e)$ equal to the proportion of edges supporting a traffic at most equal to T. The proper weight $\omega(e)$ can be deduced using a cumulative histogram computed from that illustrated in Figure 17. This has the desired result as the strength of high traffic edges is not altered, while strength of low traffic edges is lowered according to their relative importance. For example, the Pusan-Seoul connection has a weight of 0.99 as expected, so its weighted strength almost equals

its "plain" strength. A connection supporting approximately 500 000 passengers representing 0.058% of all traffic has a normalized weight of $\omega(e) = 0.4$, because 40% of connections support at most 500 000 passengers. Its weighted strength $\gamma_\omega(e)$ thus equals 40% of its plain strength $\gamma(e)$.

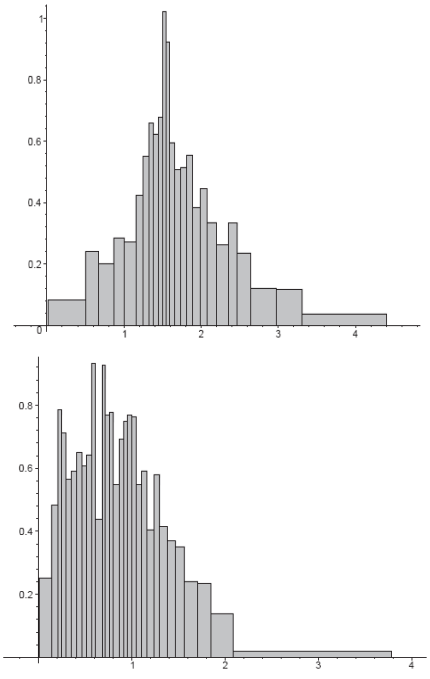

Fig. 18. Frequency distribution of strength for the worldwide air transportation network, before (left view) and after (right view) multiplication by normalized edge weights.

Figure 18 shows how edge weights affect the distribution of edge strength. As can be seen, the range of metric values moves towards the left due to the multiplication by normalized weights. More importantly, the distribution of the edge metric is affected with a much lower population of edges having a weighted strength above 1.0, and the peak frequency being translated from 0.75 to approximately 0.4.

IV.5. MQ clustering

The method developed in Section IV.2 basically requires two distinct ingredients, an edge metric together with a clustering quality measure.

The measure *MQ* we have been using to assess clustering quality possesses several interesting properties. In [24], we had already pointed out that *MQ* can be usefully approximated by a gaussian distribution with mean ~ -0.2 and standard deviation ~ 0.2, a fact that was unnoticed in [27] (see [49] for a proof). This allows us to assert the overall quality of our clustering technique since, for instance, there is only ½% probability that a clustering reaches an *MQ* value above 0.30. The probability goes down to 0.00125 for an *MQ* value of 0.40. We were moreover able to compare our clustering to those computed by the clustering framework Bunch [50] and observe that we produce clusterings of similar *MQ* quality [24, Table 1]. Bunch implements algorithms exploring the space of all possible clusterings following various strategies (hill climbing, genetic algorithms, etc.). Algorithm *MQ* (Section IV.2) however visits a considerably smaller portion of the solution space. Indeed, letting τ vary over $[a, b]$ defines a one-dimensional curve in the space $\wp(G)$ of all clustering of a graph G. Obviously the path prescribed by the strength metric and the filtering process passes through "fertile" regions.

V. Conclusion and future work

We have presented a strategy for the visualization and navigation of complex networks based on the identification of a multilevel community structure. The usefulness of our method has been assessed by experts and has proven to be useful for the visualization and analysis of a word association network and for the worldwide air transportation network. Examples from software re-engineering have also been studied. The method is actually part of Tulip's latest release [51][5].

Efficiency issues still remain to be studied to help the method reach full scalability. Although strength is defined as a local metric, its implementation can be helped by storing local information on N_u, N_v and $W_{u,v}$ (see Section II.2) to avoid recomputing $\gamma_G(C_i)$ from scratch (Section III.2). These issues are essential if we are to extend strength to cover cycles of length 5, and more as suggested by Raddichi *et al.* [52].

We also plan to carefully describe the behavior of *MQ* after local changes in the network in order to localize its computation and adapt to dynamic networks. This would allow for real-time community identification, and could help follow the evolution of the network and study the evolution of its community structure.

We plan to run a careful comparison between our work and that of Newman and Girvan. The combination of strength together with the modularity function *Q* used by Newman and Girvan could reveal useful. Conversely, we could mix *MQ* together with edge betweenness. The comparison should not only consider the community structure output by any of these combinations, but should also look at the multilevel community structure it produces. Indeed, we expect to get a better understanding of this methodology by introducing quality measures that take into account the whole cluster tree instead of only focusing on the clustering of a single component.

Finally, this work will soon be integrated into a fully interactive environment similar to that developed by van Ham and van Wijk [45], allowing a more fluid and focus+context navigation of the multilevel community structure.

Acknowledgements

This work is supported by CRSNG, Canada (Chiricota) and ANR SPANGEO, France (Melançon).

References

[1] Marcos, A. F., Muller, W., and Schumann, H., Visual knowledge discovery (special issue). *Computers & Graphics*, Vol. 28 n. 3 pp. 309-310, 2004.

[2] Huband, J. M., Bezdek, J. C., and Hathaway, R. J., bigvat: Visual assessment of cluster tendency for large data sets. *Pattern Recognition*, Vol. 38, n. 11, pp. 1875-1886, 2005.

[3] Aggarwal, C. C., On the use of human-computer interaction for projected nearest neighbor search. *Data Mining and Knowledge Discovery*, Vol. 13, n. 1, pp. 89-117, 2006.

[4] Card, S., Mackinlay, J., and Shneiderman, B. 1999. Readings in Information Visualization. Morgan Kaufmann Publishers, San Francisco.

[5] Herman, I., Marshall, M. S., and Melançon, G., Graph visualisation and navigation in information visualisation: A survey. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 6, n. 1, pp. 24-43, 2000.

[6] di Battista, G. d., Eades, P., Tamassia, R., and Tollis, I. G. 1999. Graph Drawing: Algorithms for the Visualisation of Graphs. Prentice Hall.

[7] Kaufmann, M. and Wagner, D. (Editors) 2001. Drawing Graphs, Methods and Models, volume 2025 of Lecture Notes in Computer Science. Springer.

[8] Girvan, M. and Newman, M. E. J. 2002. Community structure in social and biological networks. Proceedings of the National Academy Science USA, 99:7821-7826.

[9] Newman, M. E. J. and Girvan, M. 2004. Finding and evaluating community structure in networks. Physical Reviews, E 69(026113).

[10] Alpert, C. J. and Kahng, A. B. 1995. Recent developments in netlist partitioning: A survey. Integration: the VLSI Journal, 19(1-2):1-81.

[11] Michaud, P. 1997. Clustering techniques. Future Generation Computer System, 13:135-147.

[12] FjÄallstrom, P. 1998. Algorithms for graph partitioning: A survey. Linkoping Electronic Articles in Computer and Information Science, 3(10). www.ep.liu.se/ea/cis/1998/010/.

[13] Jain, A. K., Murty, M. N., and Flynn, P. J. 1999. Data clustering: a review. ACM Computing Surveys, 31(3):264-323.

[14] Berkhin, P. 2002. Survey of clustering data mining techniques. Technical report, Accrue Software.

[15] dos Santos, S. and Brodlie, K. 2004. Gaining understanding of multivariate and multidimensional data through visualization. Computers & Graphics, 28(3):311-325.

[16] Herman, I., Marshall, M. S., Melançon, G., Duke, D. J., Delest, M., and Domenger, J.-P. 1999. Skeletal images as visual cues in graph visualization. In: Gröller, E., Löffelmann, H., and Ribarsky, W. (Editors), Data Visualization '99, Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization, Wien. Springer-Verlag, pages 13-22.

[17] Herman, I., Melançon, G., and Delest, M. 1998. Tree visualisation and navigation clues for information visualisation. Computer Graphics Forum, 17(2):153-165.

[18] Auber, D., Chiricota, Y., Jourdan, F., and Melançon, G. 2003. Multiscale navigation of small world networks. In: IEEE Symposium on Information Visualisation, Seattle, GA, USA. IEEE Computer Science Press, pages 75-81.

[19] Auber, D., Delest, M., and Chiricota, Y. 2004. Strahler based graph clustering using convolution. In: Eighth International Conference on Information Visualisation (IV'04). IEEE Computer Society, pages 44-51.

[20] Huang, X., Eades, P., and Lai, W. 2005. A framework of filtering, clustering and dynamic layout graphs for visualization. In: Estivill-Castro, V. (Editor), 28th Australasian Computer Science Conference. Australian Computer Society, 38.

[21] Botafogo, R. A., Rivlin, E., and Schneiderman, B. 1992. Structural analysis of hypertexts: Identifying hierarchies and useful metrics. ACM Transactions on Information Systems, 10(2):142-180.

[22] Henry, T. R. 1992. Interactive Graph Layout: The Exploration of Large Graphs. Phd, University of Arizona.

[23] Newman, M. E. J. 2001. Scientific collaboration networks: II. Shortest paths, weighted networks, and centrality. Physical Review E, 64(016132).

[24] Chiricota, Y., Jourdan, F., and Melançon, G. 2003. Software components capture using graph clustering. In: 11th IEEE International Workshop on Program Comprehension, Portland, Oregon. IEEE / ACM, pages 217-226.

[25] Chi, E. H. 2000. A taxonomy of visualization techniques using the data state reference model. In: IEEE Symposium on Information Vizualization. IEEE Computer Society, page 69.

[26] Wijk, J. v. 2005. The value of visualization. In: Silva, C., Groeller, E., and Rushmeier, H. (Editors), IEEE Visualization. IEEE Computer Society, pages 79-86.

[27] Mancoridis, S., Mitchell, B. S., Rorres, C., Chen, Y., and Gansner, E. 1998. Using automatic clustering to produce high-level system organizations of source code. In: IEEE International Workshop on Program Understanding (IWPC'98), Ischia, Italy.

[28] [Watts and Strogatz, 1998] Watts, D. and Strogatz, S. H. 1998. Collective dynamics of "small-world" networks. Nature, 393:440-442.

[29] Erdös, P. and Rényi, A. 1959. On random graphs. Publ. Math. Debrecen, 6:290-297.

[30] Erdös, P. and Rényi, A. 1960. On the evolution of random graphs. Publ. Math. Inst. Hungar. Acad. Sci., 5:17-61.

[31] Bollobás, B. 1985. Random Graphs. Academic Press, London.

[32] Bornholdt, S. and Schuster, G. (Editors) 2003. Handbook of Graphs and Networks: From the Genome to the Internet. Wiley-VCH.

[33] Dorogovtsev, S. N. and Mendes, J. F. F. 2003. Evolution of Networks : From Biological Nets to the Internet and WWW. Oxford University Press.

[34] Chakrabarti, D. and Faloutsos, C. 2006. Graph mining: Laws, generators, and algorithms. ACM Computing Surveys, 38(1):2.

[35] Jaccard, P. 1912. New Phytol., 11:37-50.

[36] Tanimoto, T. 1958. An elementary mathematical theory of classification and prediction. Technical report, IBM Report.

[37] Willett, P., Barnard, J., and Downs, G. 1998. Chemical similarity searching. Journal of Chemical Information and Computer Sciences, 38:983-996.

[38] Freeman, L. C. 1977. A set of measures of centrality based upon betweeness. Sociometry, 40:35-41.

[39] Scott, J. P. 2000. Social Network Analysis: A Handbook. SAGE Publications, 2nd edition.

[40] Goldberg, D. S. and Roth, F. P. 2003. Assessing experimentally derived interactions in a small world. Proceedings of the National Academy of Sciences of the United States of America, 100(8):4372-4376.

[41] Zachary, W. 1977. An information flow model for conflict and fission in small groups. Journal of Anthropological Research, 33:452-473.

[42] Dion, D., Auber, D., Leblanc, B., and Melançon, G. 2003. Graphe d'associations verbales : élaboration et visualisation. In: Cognitique : vers une informatique plus cognitive et sociale, pages 223-232. Cépaduès-Editions.

[43] Dion, D. 2005. Graphes d'associations verbales et réseaux petit monde (Word association network and small world networks). Master thesis, cognitive science institute, Université Victor Segalen, Bordeaux, France.

[44] Guha, S., Rastogi, R., and Shim, K. 1999. Rock: a robust clustering algorithm for categorical attriutes. In: Proceedings of the 15th ICDE, Sydney, Australia. pages 512-521.

[45] van Wijk, J. and van Ham, F. 2004. Interactive visualization of small world graphs. In: Munzner, T. and Ward, M. (Editors), IEEE Symposium on Information Visualisation, Austin, TX, USA. IEEE Computer Science press.

[46] Herman, I., Marshall, M. S., and Melançon, G. 2000a. Density functions for visual attributes and effective partitioning in graph

visualization. In: Roth, S. F. and Keim, D. A. (Editors), IEEE Symposium on Information Visualization (InfoVis'2000), Salt Lake City, Utah, U.S. IEEE Computer Society, pages 49-56.
[47] Amiel, M., Melançon, G., and Rozenblat, C. 2005. Réseaux multi-niveaux : l'exemple des échanges aériens mondiaux. Mappemonde, 78.
[48] Guimera, R., Mossa, S., Turtschi, A., and Amaral, L. A. N. 2005. The worldwide air transportation network: anomalous centrality, community structure, and cities global roles. Proceedings of the National Academy of Sciences of the United States of America, 102(22):7794-7799.
[49] Delest, M., Fédou, J.-M., and Melançon, G. 2006. A quality measure for multi-level community structure. In: SYNASC 2006 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania.
[50] Mitchell, B. S., Mancoridis, S., Chen, Y.-F., and Gansner, E. 1999. Bunch: A clustering tool for the recovery and maintenance of software system structures. In: ICSM. pages 50-.
[51] Auber, D. 2003. Tulip - a huge graph visualization framework. In: Mutzel, P. and Jünger, M. (Editors), Graph Drawing Software, Mathematics and Visualization Series. Springer Verlag.
[52] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. 2004. Defining and identifying communities in networks. Proceedings of the National Acadademy of Science USA, 101:2658-2663.

Authors' information

[1] Université du Québec à Chicoutimi, Québec, Canada
[2] INRIA Futurs and CNRS UMR 5800 LaBRI, France

Yves Chiricota got a Ph. D. in combinatorial mathematic from Université du Québec à Montréal (Canada) in 1992. After working five years in the industry on developing 3D systems for the garment and clothing industries, he returned to the academics. He now holds a university professor position at Université du Québec à Chicoutimi (Canada).

His main scientific interests are Graph Visualization and Graph mining, with a special emphasis on graphics, systems development and graph combinatorics. His current interests lie in the improvement of algorithms for Graph Visualization, exploiting graphics hardware.

Guy Melançon also got a Ph. D. in combinatorial mathematic from Université du Québec à Montréal (Canada) in 1991. He worked as associate professor in Bordeaux until 1998, before joining CWI in Amsterdam (The Netherlands) as a full time researcher for two years. In 2000, he moved to Montpellier, France as a University professor in Computer Science; he has recently moved back to Université de Bordeaux I (France). He is now a permanent member of CNRS UMR 5800 LaBRI and head of the INRIA GRAVITÉ team.

His main scientific interests are Graph Visualization and Graph mining, with a special emphasis on graph combinatorics and human-computer interaction. He co-authored a survey on Graph Visualization [5] which ranks as a top reference in the field. His current interests lie in the Interactive Visualization and Mining of Dynamic Networks, with applications in Social Sciences and Strategic and Competitive Watch.

