

## The graph clustering paradigm

In graph clustering one seeks to divide the node set of a graph into natural groups with respect to the edge relation. The first section of this chapter gives a brief account of three related ideas for accomplishing this task. They are formulated in terms of *path numbers*, *random walks*, and *shortest paths*. In Section 5.2 proposals towards graph clustering that have a combinatorial nature are discussed. A relaxation of one of them is the subject of Section 5.3. It is called *k-path clustering* and uses path numbers to detect cluster structure via single link clustering. This method links the combinatorial cluster notions with the *MCL* algorithm, as the starting point for the *MCL* algorithm is a localized version of *k-path clustering*. In Section 5.4 probabilistic cluster algorithms based on the ideas in the first section are briefly described. Random walks on graphs are introduced, corresponding with a localization of the context in which *k-path clustering* is applied. The standard way of describing a random walk on a graph associates a particular discrete Markov chain with the graph, and such is also the setup here. Section 5.5 begins with an example of (deterministically computed) random walks on an undirected graph possessing (weak) cluster structure. The initial characteristics of this stochastic process (c.q. Markov chain) are similar to phenomena observed in applying *k-path clustering* to the same graph (Section 5.3) but in the limit of the process all evidence of cluster structure has withered away. A new operator called *inflation* is inserted into the process, and an example run using the same input graph results in a limit which induces a cluster interpretation of the input graph in a generic way. The *MCL* algorithm and *MCL* process are formally described in the last section of this chapter. The relationship between the *MCL* process and cluster interpretation of graphs is the subject of Chapter 6, together with an analysis of convergence of the *MCL* process and stability of the limits with respect to the cluster interpretation. Chapter 7 gives conditions under which iterands of the process have real c.q. nonnegative spectrum, and which imply the presence of generalized cluster structure in the iterands.

### 5.1 Paths, walks, and cluster structure in graphs

What are natural groups? This is in general a difficult problem, but within the framework of graphs there is a single notion which governs many proposals. This notion can be worded in different ways. Let  $G$  be a graph possessing cluster structure, then alternative wordings are the following:

*a) The number of higher-length paths in  $G$  is large for pairs of vertices lying in the same dense cluster, and small for pairs of vertices belonging to different clusters.*

*b) A random walk in  $G$  that visits a dense cluster will likely not leave the cluster until many of its vertices have been visited.*

*c) Considering all shortest paths between all pairs of vertices of  $G$ , links between different dense clusters are likely to be in many shortest paths.*

These three notions are strongly related to each other. The situation can be compared to driving a car in an unfamiliar city in which different districts are connected by only a few roads, with many promising looking turns and roads unreachable due to traffic regulations. Viewing crossings and turns as vertices, and the accessible road segments between them as edges, the notions given above translate to a) There are many different ways of driving (not necessarily taking the shortest route) from  $A$  to  $B$  if  $A$  and  $B$  are in the same district, and only few if they are in different districts, under the condition that the number of roads segments visited is equal; b) Driving around randomly, but in line with traffic regulations, will keep you in the same district for a long time; c) If the transportation need of the locals is homogeneously distributed over all departure and destination points, then the roads connecting different districts will be congested.

The idea now is to measure or sample any of these — higher-length paths, random walks, shortest paths — and deduce the cluster structure from the behaviour of the sampled quantities. The cluster structure will manifest itself as a peaked distribution of the quantities, and conversely, a lack of cluster structure will result in a flat distribution. The distribution should be easy to compute, and a peaked distribution should have a straightforward interpretation as a clustering.

I propose to assemble the notions listed above under the denominator of the *graph clustering paradigm*, being well aware of the fact that the paradigm label is somewhat grandiloquent. However, the notions clearly share a common idea that is simple and elegant in that it gives an abstract and implicit description of cluster structure (rather than tying it to a particular optimization criterion); in that it is persistent, as it has surfaced at different times and places<sup>1</sup>; and in that it is powerful, as it can be tied to different graph-theoretical concepts, yielding different clustering methods.

The idea of using random walks to derive cluster structure is mainly found within the graph partitioning community. The various proposals utilizing it are discussed in Section 5.4. The following section describes proposals for graph clustering which have a strong combinatorial nature. One of these, the linkage-based  $k$ -path clustering method forms the connection between combinatorial and randomized methods. The single linkage paradigm can be seen as the connecting factor. This requires the dismissal of a notion which is seemingly central to single link clustering, namely the global interpretation of the (dis)similarity function. It is argued that this global interpretation hampers the combinatorial clustering methods introduced below; the introduction of random walks naturally requires a localized interpretation of graph connectivity properties.

---

<sup>1</sup>The number of occurrences is not large in itself, but it is significant considering the small number of publications dedicated to graph clustering.

## 5.2 Combinatorial cluster notions

In the clustering and pattern recognition communities, proposals have been made to define clusters in graphs which are more combinatorial in nature. An important contributor in this respect is David Matula, who wrote several articles on the subject. It is noteworthy that Matula's publication record (e.g. [120, 121, 122, 150]) indicates that his primary research interests are in graph theory and discrete mathematics. It seems that his publications in clustering in the setting of (simple) graphs came too early in the sense that at the time of writing there was little interest in the clustering community in simple graphs, except as a means of notation for the description of linkage-based algorithms such as single link and complete link clustering. In fact, Matula presents several graph cluster concepts in [121] as a series of refinements splitting the spectrum between single link and complete link clustering. The presentation of these findings in the setting of general similarity spaces and threshold graphs indicates that the time was not right for clustering in the setting of simple graphs per se. I see several reasons why the combinatorial notions have not caught on, among which the issue of justification in the setting of threshold graphs and the lack of genuine (simple) graph applications and problems. Equally important however are the relative intractability of the proposed notions, and their disposition to produce unbalanced clusterings. Let  $G = (V, E)$  be a graph. The following notions each define subgraphs of  $G$ .

- $k$ -bond            A maximal subgraph  $S$  such that each node in  $S$  has at least degree  $k$  in  $S$ .
- $k$ -component    A maximal subgraph  $S$  such that each pair of nodes in  $S$  is joined by  $k$  edge-disjoint paths in  $S$ .
- $k$ -block            A maximal subgraph  $S$  such that each pair of nodes in  $S$  is joined by  $k$  vertex-disjoint (except for endpoints) paths in  $S$ .

Each notion defines a corresponding hierarchical cluster method by letting  $k$  vary and at each level taking as cluster elements all  $k$ -objects and all singletons corresponding with nodes which are not in any  $k$ -object, where object may be any of *bond*, *component*, or *block*. These methods are hierarchical because every  $k + 1$ -object is contained within a  $k$ -object. For  $k = 1$  all three  $k$ -notions boil down to the connected components of  $G$ . Moreover, for fixed  $k$ , it is true that every  $k$ -block of  $G$  is a subgraph of some  $k$ -component, which is in turn a subgraph of some  $k$ -bond of  $G$ . This implies that the corresponding cluster methods are successive refinements, going from bond to component to block. In the clustering section of the graph partitioning survey article [8] of Alpert and Kahng one method is mentioned which is a refinement of the  $k$ -component method, namely the  $(K, L)$ -connectivity method proposed by Garbers et al in [61]. Nodes are  $(K, L)$ -connected if there exist  $K$  edge disjoint paths of length at most  $L$  between them.

Matula finds that  $k$ -components and  $k$ -blocks provide better resolution into cohesive groupings than  $k$ -bonds. The example given here in Figure 5 is taken from the article [121], and it shows a graph with its  $k$ -blocks, yielding the most refined clusterings. In this case, the overlapping clustering for  $k = 3$  looks reasonably good, although it is a pity that the fifth point in the leftmost 2-block ends up as a singleton in the 3-block clustering.

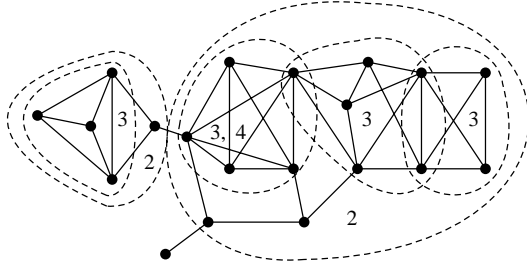
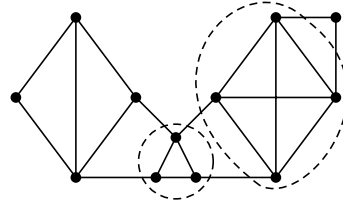
Figure 5. Graph with its  $k$ -blocks.

Figure 6. Graph with its 3-blocks.

The lack of balance is even stronger in the graph which is depicted in Figure 6, together with its 3-block clustering. For this graph, the 2-block clustering yields the whole vertex set as a single cluster and the 3-block clustering is very unsatisfactory. This evidence is neither incidental nor contrived. Rather, it is inherent to the  $k$ -object methods. They are very sensitive to local variations in node degree. Such sensitivity is unwanted in itself, and in this case leads to unbalanced clusterings. The  $k$ -object methods are much too restrictive in their definition of cohesive structure, especially taking into account the commonly accepted ‘loose’ objective of clustering. It is reasonable to demand that a clustering method for simple graphs can recognize disjoint unions of complete (simple) graphs of different sizes, or complete graphs which are sparsely connected. The  $k$ -object methods clearly fail to do so, and one reason for this is that local variations in connectivity have severe impact on the retrieved clusters.

Finally, the object methods are highly intractable. Matula [121] and Tomi [161] give time complexities  $\mathcal{O}(|V|^{3/2}|E|^2)$  for the retrieval of  $k$ -blocks and  $\mathcal{O}(\min(|V|^{8/3}|E|, |V||E|^2))$  for the retrieval of  $k$ -components. Among others, the algorithms require the solution of the minimum cut network flow problem. Since the number of edges  $|E|$  is surely at least  $|V|$  for interesting applications, the time complexities are at least cubic in the input size of the graph.

### 5.3 $k$ -Path clustering

Of the existing procedural algorithms, single link clustering has the most appealing mathematical properties. This is precisely because it allows *non-procedural* interpretations in terms of Minimal Spanning Trees and in terms of approximating metrics by ultrametrics (trees). See [80] for an extensive treatment of this subject. In this section I shall discuss a variant of single link clustering for graphs which I call  $k$ -path clustering. This variant is a further relaxation of the  $k$ -block and  $k$ -component methods, and its interpretation is related to the interpretation of the *MCL* algorithm. The basic observation underlying both methods is the fact that two nodes in some dense region will be connected by many more paths of length  $k, k > 1$ , than two nodes for which there is no such region. This section is mainly an exposition of ideas, and a few examples are

studied. The examples are intended to support the heuristic underlying the *MCL* algorithm, and they provide fruitful insights into the problems and benefits associated with refinements of graph similarities.  $k$ -Path clustering is conceptually much simpler than  $k$ -block and  $k$ -component clustering, but in terms of tractability it is only slightly more viable. It suffers less from a lack of balance in the clusters it produces, but it is still far from satisfactory in this respect.  $k$ -Block,  $k$ -component, and  $k$ -path clustering were also proposed by Tamura [160], who was apparently unaware of the work of Matula.

**5.3.1  $k$ -path clustering.** For  $k = 1$ , the  $k$ -path clustering method coincides with generic single link clustering. For  $k > 1$  the method is a straightforward generalization which refines the similarity coefficient associated with 1-path clustering. Let  $G = (V, w)$  be a graph, where  $V = \{v_1, \dots, v_t\}$ , let  $M = \mathcal{M}_G$  be the associated matrix of  $G$ . For each integer  $k > 0$ , a similarity coefficient  $Z_{k,G}$  associated with  $G$  on the set  $V$  is defined by setting  $Z_{k,G}(v_i, v_j) = \infty, i = j$ , and

$$(1) \quad Z_{k,G}(v_i, v_j) = (M^k)_{ij}, \quad i \neq j$$

Note that the values  $(M^i)_{pp}$  are disregarded. The quantity  $(M^k)_{pq}$  has a straightforward interpretation as the number of paths of length  $k$  between  $v_p$  and  $v_q$ ; this is the exact situation if  $G$  is a simple graph. If  $G$  has dense regions separated by sparse boundaries, it is reasonable to conjecture that there will be relatively many path connections of length  $k$  with both ends in the same region, compared with the number of path connections having both ends in different dense regions. For weighted graphs, the interpretation is in terms of path capacities rather than paths per se, and the formulation is now that the path capacities between different dense regions are small compared with the path capacities within a single dense region. The next example is one in which  $Z_{k,G}$  does not yet work as hoped for. It will be seen why and how that can be remedied. For sake of clear exposition, the examples studied are simple graphs.

**5.3.2 Odd and even.** The graph  $G_1$  in Figure 7 is a tetraeder with flattened tips. It clearly admits one good non-extreme clustering, namely the one in which each of the flattened tips, i.e. the four triangles, forms a cluster. The associated matrix  $M = \mathcal{M}_{G_1}$ , and the square  $M^2$  are shown in Figure 9.

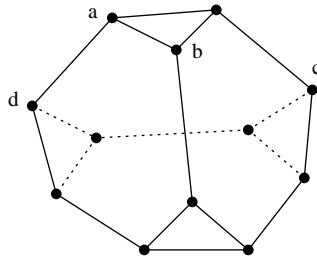


Figure 7. Topped tetraeder  $G_1$ .

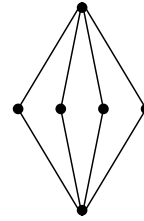


Figure 8. Bipartite graph  $G_2$ .

$$\begin{pmatrix}
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}
\quad
\begin{pmatrix}
3 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\
1 & 3 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 1 & 3 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 3 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 3 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 & 3 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 3 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 1 & 1 & 3 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 3 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 3 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 3 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 3
\end{pmatrix}$$

$M = \mathcal{M}_{G_1}$ 
 $M^2$

$$\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1
\end{pmatrix}
\quad
\begin{pmatrix}
4 & \mathbf{3} & \mathbf{3} & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 2 \\
\mathbf{3} & 4 & \mathbf{3} & 1 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 1 \\
\mathbf{3} & \mathbf{3} & 4 & 2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 2 & 4 & \mathbf{3} & \mathbf{3} & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & \mathbf{3} & 4 & \mathbf{3} & 1 & 0 & 0 & 1 & 2 & 1 \\
0 & 0 & 1 & \mathbf{3} & \mathbf{3} & 4 & 2 & 1 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 1 & 2 & 4 & \mathbf{3} & \mathbf{3} & 1 & 0 & 0 \\
1 & 2 & 1 & 0 & 0 & 1 & \mathbf{3} & 4 & \mathbf{3} & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & \mathbf{3} & \mathbf{3} & 4 & 2 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 2 & 4 & \mathbf{3} & \mathbf{3} \\
1 & 0 & 0 & 1 & 2 & 1 & 0 & 0 & 1 & \mathbf{3} & 4 & \mathbf{3} \\
2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & \mathbf{3} & \mathbf{3} & 4
\end{pmatrix}$$

$M + I$ 
 $(M + I)^2$

Figure 9. Several matrices associated with  $G_1$ .

For each of the coefficients  $Z_{k,G_1}$ , single link clustering immediately yields the whole vertex set of  $G_1$  as one cluster. How can this be? Somehow, the expectation that there would be relatively more  $k$ -length paths within the dense regions, in this case triangles, was unjustified. Now, on the one hand this is a peculiarity of this particular graph and especially of the subgraphs of the triangle type. For even  $k$ , spoilers are pairs like  $(a, c)$ , for odd  $k$ , these are pairs like  $(a, d)$ . This clearly has to do with the specific structure of  $G_1$ , where the set of paths of odd length leading e.g. from  $a$  to  $b$  does not profit from  $(a, b)$  being in a triangle, compared with the set of paths leading from  $a$  to  $d$ . On the other hand the behaviour of any similarity coefficient  $Z_{k,G}$  is in general very much influenced by the parity of  $k$ . There is a strong effect that odd powers of  $M$  obtain their mass from simple paths of odd length and that even powers of  $M$  obtain their mass from simple paths of even length. The only exceptions are those paths which include loops of odd length. Note that the only requirement for a loop of even length is the presence of an edge (inducing a loop of length 2).

**5.3.3 A countermeasure to parity dependence.** The observation in one of the previous paragraphs that paths containing circuits of odd length form an exception brings a solution to the problem of parity dependence. By adding loops to each node in  $G_1$ , the parity dependence is removed. Just as every edge induces the minimal loop of even length, every node now induces the minimal loop of odd length. On the algebra side, adding loops corresponds with adding the identity matrix to  $M$ . The numbers defining the new coefficients  $Z_{2,G_1+I}$  are found in Figure 9, where the largest off-diagonal matrix entries (diagonal entries are disregarded) are printed in boldface. Each coefficient now yields

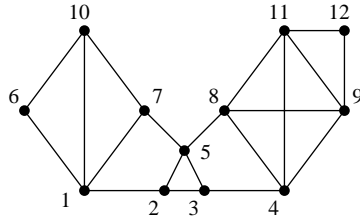


Figure 10. Graph  $G_3$ .

$$\begin{pmatrix} 5 & 2 & 1 & 0 & 2 & 3 & 3 & 0 & 0 & 4 & 0 & 0 \\ 2 & 4 & 3 & 1 & 3 & 1 & 2 & 1 & 0 & 1 & 0 & 0 \\ 1 & 3 & 4 & 2 & 3 & 0 & 1 & 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 2 & 5 & 2 & 0 & 0 & 4 & 4 & 0 & 4 & 2 \\ 2 & 3 & 3 & 2 & 5 & 0 & 2 & 2 & 1 & 1 & 1 & 0 \\ 3 & 1 & 0 & 0 & 0 & 3 & 2 & 0 & 0 & 3 & 0 & 0 \\ 3 & 2 & 1 & 0 & 2 & 2 & 4 & 1 & 0 & 3 & 0 & 0 \\ 0 & 1 & 2 & 4 & 2 & 0 & 1 & 5 & 4 & 0 & 4 & 2 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 & 4 & 5 & 0 & 5 & 3 \\ 4 & 1 & 0 & 0 & 1 & 3 & 3 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 & 4 & 5 & 0 & 5 & 3 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 3 & 0 & 3 & 3 \end{pmatrix}$$

Figure 11. The matrix  $(N+I)^2$ ,  $N = \mathcal{M}_{G_3}$ .

the best clustering, consisting of the set of four triangles. Adding loops helps in further differentiating the numbers  $Z_{k,G_1+I}(s, t)$  for fixed  $s$  and varying  $t$ .

For a less symmetrical example, consider the simple graph  $G_3$  depicted in Figure 10, also used on page 42. Its associated matrix after adding loops to each node is given next to it in Figure 11. Below are the results of single link clustering at all levels, using the similarity coefficient  $Z_{2,G_3+I}$ .

Level	Clustering
$\infty \dots 6$	$\{\text{singletons}(V)\} = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}, \{9\}, \{10\}, \{11\}, \{12\}\}$
5	$\{\{9, 11\}, \{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}, \{8\}, \{10\}, \{12\}\}$
4	$\{\{1, 10\}, \{4, 8, 9, 11\}, \{2\}, \{3\}, \{5\}, \{6\}, \{7\}, \{12\}\}$
3	$\{\{1, 6, 7, 10\}, \{2, 3, 5\}, \{4, 8, 9, 11, 12\}\}$
2, 1, 0	$\{V\} = \{\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}\}$

The clustering at level 3, which is the first in which no singletons remain, is rather pleasing. This clustering also results if the coefficient is taken to be  $Z_{3,G_3+I}$  (not given here). The coefficient  $Z_{4,G_3+I}$  starts out accordingly, however, before node 6 gets involved, the groups  $\{4, 8, 9, 11, 12\}$  and  $\{2, 3, 5\}$  are joined. This is caused by the fact that node 6 is located in the sparsest part of  $G_3$ . The weak spot of single link clustering, namely *chaining*, surfaces here in the specific case of  $k$ -path clustering.

The last example in this section is a graph  $G_2$  for which single link clustering with coefficient  $Z_{k,G_2}$ ,  $k > 1$ , initially groups points together which are not connected. The graph  $G_2$  in Figure 8 is a small bipartite graph. The upper and lower nodes have three simple paths of length 2 connecting them. Even in the presence of loops, the number of  $k$ -step paths,  $k > 1$ , will always be greater for the pair of top and bottom nodes than for any other pair. Bipartite graphs form a class of graphs for which it is natural to cluster each of the two node domains separately<sup>2</sup>. By adding multiple loops to each node of  $G_2$  it can be ensured that the resulting clustering corresponds with connected components only

<sup>2</sup>e.g. Document phrase databases naturally yield bipartite graphs. Clustering the two node domains then yields a document grouping and a phrase grouping.

(one in this case), but it is difficult to formulate a sufficient condition which guarantees this property for graphs in general. I conjecture that a sufficient condition is for a graph to have nonnegative spectrum. This is a non-trivial conjecture, since spectral properties have to be related to both the ordinal relationship among entries of a matrix power and the 0/1 structure of the matrix itself.

**5.3.4 A critical look at  $k$ -path clustering.** If  $k$ -path clustering were to be applied to large graphs, it would be desirable to work with varying  $k$  and the corresponding coefficients  $Z_{k,G}$ . However, for most application graphs in this research, the matrices  $M^k$  and  $(M + I)^k$  fill very rapidly due to high connectivity of the graphs. The potential number of nonzero elements equals  $10^{2N}$  for graphs of vertex-size  $|V| = 10^N$ . For  $N = 4$  this quantity is already huge and for  $N = 5$  it is clearly beyond current possibilities. More importantly, it is quadratically related to  $N$ . In large scale applications, this is known to be a bad thing. It is difficult to remedy this situation by a regime of removing smaller elements.

A second minus was mentioned in the discussion of the example graph  $G_3$  in Figure 10. I remarked that under the coefficient  $Z_{4,G_3}$  groups which had formed already started to unite before the last node left its singleton state. The coefficients  $Z_{k,G}$  do account for the local structure around a node. However, a region which is denser than another region with which it connected to a certain extent, will tend to swallow the latter up. This is the effect of chaining in  $k$ -path clustering. A third minus is related to the preceding and arises in the case of weighted graphs. Differentiation in the weight function will lead to the same phenomenon of heavy-weight regions swallowing up light-weight regions. It should be noted that this situation is problematic for every cluster method based on single link clustering.

On the credit side I find that at least in a number of examples the idea of considering higher length paths works well. The manoeuvre of adding loops to graphs is clearly beneficial, and the reason for this lies in the fact that parity dependence is removed, leading to a further differentiation of the associated similarity coefficient. The issue of parity dependence has been noted before: Alpert and Kahng criticize the  $(K, L)$ -connectivity method of Garbers et al — which is a variant of  $k$ -component clustering — for cutting a four-cycle (which is a bipartite graph) into disjoint paths.

## 5.4 Random walks and graphs

In this section I briefly discuss probabilistic cluster algorithms proposed in the graph partitioning community and the concept of random walks on graphs. An application of the latter is briefly described in Chapter 8, namely polynomial time approximation schemes based on Markov chains. These are interesting because a necessary condition is in general that the Markov chains be *rapidly mixing*, which essentially requires that the subdominant eigenvalue is well separated from the largest eigenvalue one. This relationship between the spectrum of a graph and its connectivity properties plays a role in many applications in graph theory (Chapter 8), and it does so too in the *MCL* process.

**5.4.1 Probabilistic cluster algorithms.** In the graph partitioning community, several randomized cluster algorithms have been proposed. I follow the survey article [8] by Alpert and Kahng which was written in 1995. Karger [99] proposed a heuristic where each vertex starts as a singleton cluster. Edges are iteratively chosen in random fashion, and each time the clusters incident to the currently chosen edge are contracted into a single cluster. A related approach was proposed by Bui et al in [29, 159]. A matching in a graph is a set of edges such that no pair of edges has a common vertex. They propose to find a random maximal matching and merge each pair of vertices into a cluster, resulting in a set of  $n/2$  clusters. Both proposals hinge on the fact that there are more edges within clusters than in between different clusters if cluster structure is present. Hagen and Kahng sample random walks for cycles in [75]; the basic setup is that if two nodes co-occur sufficiently often in a cycle, then they are joined within a cluster. Finally, Yeh et al [174] propose a method in which shortest paths between randomly chosen pairs of vertices are computed. Each edge has a cost associated with it, which is adjusted every time the edge is included in a shortest path. In dense clusters, alternative paths are easily found; this not being the case for vertices in different clusters, edges between them will inevitably acquire a higher check.

The basic idea underlying the *MCL* algorithm fits in the same paradigm, but two important distinctions are that random walks are computed *deterministically* and *simultaneously*. The crux of the algorithm is that it incorporates reinforcement of random walks.

**5.4.2 Random walks on graphs.** The standard way to define a random walk on a simple graph is to let a Young Walker take off on some arbitrary vertex. After that, he successively visits new vertices by selecting arbitrarily one of the outgoing edges.<sup>3</sup> This will be the starting point for the *MCL* algorithm. An excellent survey on random graphs is [114] by Lovász. An important observation quoted from this article is the following:

A random walk is a finite Markov chain that is time-reversible (see below). In fact, there is not much difference between the theory of random walks on graphs and the theory of finite Markov chains; every Markov chain can be viewed as a random walk on a directed graph, if we allow weighted edges.

The condition that (the chain generated by) a Markov matrix is time-reversible translates to the condition that the matrix is diagonally similar to a symmetric matrix (see below). In order to define random walks on weighted graphs in general, the weight function of a graph has to be changed such that the sum of the weight of all outgoing edges equals one. This is achieved by a generic rescaling step, which amounts to the localization of the weight function alluded to before.

**DEFINITION 2.** Let  $G$  be a graph on  $n$  nodes, let  $M = \mathcal{M}_G$  be its associated matrix. The **Markov matrix** associated with a graph  $G$  is denoted by  $\mathcal{T}_G$  and is formally defined by letting its  $q^{\text{th}}$  column be the  $q^{\text{th}}$  column of  $M$  normalized. To this end, let  $d$  denote the

---

<sup>3</sup>Basic notions investigated in the theory of random walks are the *access time*  $H_{i,j}$ , which is the expected number of steps before node  $i$  is visited starting from node  $j$ , the *cover time*, which is the expected number of steps to reach every node, and the *mixing rate*, which is a measure of how fast the random walk converges to its limiting distribution.

diagonal matrix that has diagonal entries the column weights of  $M$ , thus  $d_{kk} = \sum_i M_{ik}$ , and  $d_{ij} = 0, i \neq j$ . Then  $\mathcal{T}_G$  is defined as

$$(2) \quad \mathcal{T}_G = \mathcal{M}_G d^{-1}$$

The Markov matrix  $\mathcal{T}_G$  corresponds with a graph  $G'$ , which is called the **associated Markov graph** of  $G$ . The directed weight function of  $G'$ , which is encoded in the matrix  $\mathcal{T}_G$ , is called the **localized interpretation** of the weight function of  $G$ .  $\square$

This definition encodes exactly the transformation step used in the theory of random walks on graphs. Given an undirected graph  $G$ , the matrix  $N = \mathcal{T}_G$  is no longer symmetric, but is diagonally similar to a symmetric matrix. Something can be said about the spectrum of  $\mathcal{T}_G$  in terms of the spectrum of  $\mathcal{M}_G$  if  $G$  is undirected.

LEMMA 1. *Let  $G$  be undirected and void-free<sup>4</sup>, let  $M = \mathcal{M}_G$  be its associated matrix, let  $T = \mathcal{T}_G$  be its associated Markov matrix. Then the number of positive, negative, and zero eigenvalues are the same for  $T$  and  $M$ .*

Next denote by  $l$  and  $u$  the minimum respectively maximum column sum, that is,  $l = \min_k \sum_i M_{ik}$ , and  $u = \max_k \sum_i M_{ik}$ . Then

$$(3) \quad \frac{\lambda_k(M)}{u} \leq \lambda_k(T) \leq \frac{\lambda_k(M)}{l} \quad \lambda_k(T) > 0$$

$$(4) \quad \frac{\lambda_k(M)}{l} \leq \lambda_k(T) \leq \frac{\lambda_k(M)}{u} \quad \lambda_k(T) < 0$$

PROOF. Let  $d$  be the diagonal matrix of column lengths as defined in Definition 2. The matrix  $T = Md^{-1}$  is similar to the matrix  $d^{-1/2}Md^{-1/2}$ , which is congruent to the matrix  $M$ . Now the first statement of the lemma follows from Sylvester's law of inertia ([86], page 223). Because of congruence, the inertia of the matrices  $M$  and  $d^{-1/2}Md^{-1/2}$  are the same, and because of similarity, the spectra of the matrices  $d^{-1/2}Md^{-1/2}$  and  $T = Md^{-1}$  are the same, which is a stronger property than sharing the same inertia. The fact that the transition matrix  $T = d^{-1}$  is diagonally similar to the symmetric matrix  $d^{-1/2}Md^{-1/2}$  is in Markov theory phrased as that  $T$  is *time-reversible* or that  $T$  satisfies the *detailed balance condition*.

The second statement follows from Ostrowski's theorem ([86], page 224), which relates the eigenvalues of a hermitian matrix  $A$  to the eigenvalues of the matrix  $SAS^*$  in terms of bounding factors  $\lambda_1(SS^*)$  and  $\lambda_n(SS^*)$ . In the lemma, these factors are simply the largest and smallest eigenvalue of the matrix  $d^{-1}$ , equalling respectively  $1/l$  and  $1/u$ . It should be noted that this result can be refined by looking at principal submatrices of  $M$ . This is useful if there are a few columns of  $M$  of small weight compared with the other columns. This refinement is omitted here since it will not be needed.  $\square$

**5.4.3 A closer look at random walks.** Given a graph  $G$  and its associated Markov matrix  $T = \mathcal{T}_G$ , the value  $T_{pq}$  now indicates 'how much is the vertex  $q$  attracted to the vertex  $p$ ', and this is meaningful only in the context of the other values found in the

<sup>4</sup>All vertices are part of at least one edge.

$q^{\text{th}}$  column. It is still possible to move a node away from *all* its neighbours by increasing the weight of its loop. In Figure 12 the matrix  $M = \mathcal{T}_{G_3+I}$  (corresponding with the graph  $G_3$  in Figure 10) is given which results after the rescaling procedure, followed by three successive powers and a matrix labelled  $M^\infty$ . The matrix  $M$  is column stochastic. The fact that for each of its columns all nonzero values are homogeneously distributed can be interpreted as ‘each node is equally attracted to all of its neighbours’, or ‘at each node one moves to each of its neighbours with equal probability’.

All powers of  $M$  are column stochastic matrices too. For any Markov matrix  $N$ , the powers  $N^{(i)}$  have a limit, which is possibly cyclic (i.e. consisting of a sequence of matrices rather than a single matrix). A connected component  $C$  of a graph  $G$ , which has the property that the greatest common divisor of the set of lengths of all circuits in  $C$  is 1, is called *regular*. If for every vertex in  $C$  there is a path in  $C$  leading to any other vertex in  $C$  it is called *ergodic*. If the underlying graph of a Markov matrix  $N$  consists of ergodic regular components only, then the limit of the row  $N^{(i)}$  is non-cyclic. The graph  $G_3$  in Figure 10 clearly has this property, and the limit is found in Figure 12, denoted as  $M^\infty$ . The columns of  $M^\infty$  each equal the unique eigenvector of  $M$  associated with eigenvalue 1. This eigenvector  $e$  denotes the equilibrium state of the Markov process associated with  $M$ . A good review of Markov theory in the larger setting of nonnegative matrices can be found in [19]. Regrettably, the existing theory on Markov matrices is of little use in this thesis, because an essential ingredient of the *MCL* process is the operator  $\Gamma_r$  which acts on Markov matrices in a non-linear fashion.

### 5.5 An example MCL run

Consider Figure 12 again. As is to be expected, the equilibrium state  $e$  (each column of  $M^\infty$  equals  $e$ ) spreads its mass rather homogeneously among the states or vertices of  $G_3$ . However, the initial iterands  $M^k, k = 2, \dots$ , exhibit the same behaviour as did the matrices  $(N + I)^k$  in Figure 11, inducing the similarity coefficients  $Z_{k,G}$ . Transition values  $M^k_{pq}$  are relatively high if the vertices  $p$  and  $q$  are located in the same dense region. There is a correspondence between the numerical distribution of the column  $M^k_{p(q)}$ , and the distribution of the edges of  $G_3$  over dense regions and sparse boundaries.

**5.5.1 Boosting the multiplier effect.** The obvious interpretation of the new weight function is in terms of flow or random walks rather than in terms of path sets, but the observed behaviour of matrix multiplication is similar. The new interpretation of the weight function more or less suggests a speculative move. Flow is easier within dense regions than across sparse boundaries, however, in the long run this effect disappears. What if the initial effect is deliberately boosted by adjusting the transition probabilities? A logical model is to transform a Markov matrix  $T$  by transforming each of its columns. For each vertex, the distribution of its preferences (i.e. transition values) will be changed such that preferred neighbours are further favoured and less popular neighbours are demoted. A natural way to achieve this effect is to raise all the entries in a given column to a certain power greater than one (e.g. squaring), and rescaling the column to have sum 1 again. This has the advantage that vectors for which the nonzero entries are nearly homogeneously distributed are not so much changed, and that different column

$$\begin{pmatrix}
 0.200 & 0.250 & \text{---} & \text{---} & \text{---} & 0.333 & 0.250 & \text{---} & \text{---} & 0.250 & \text{---} & \text{---} \\
 0.200 & 0.250 & 0.250 & \text{---} & 0.200 & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\
 \text{---} & 0.250 & 0.250 & 0.200 & 0.200 & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\
 \text{---} & \text{---} & 0.250 & 0.200 & \text{---} & \text{---} & \text{---} & 0.200 & 0.200 & \text{---} & 0.200 & \text{---} \\
 \text{---} & 0.250 & 0.250 & \text{---} & 0.200 & \text{---} & 0.250 & 0.200 & \text{---} & \text{---} & \text{---} & \text{---} \\
 0.200 & \text{---} & \text{---} & \text{---} & \text{---} & 0.333 & \text{---} & \text{---} & \text{---} & 0.250 & \text{---} & \text{---} \\
 0.200 & \text{---} & \text{---} & \text{---} & 0.200 & \text{---} & 0.250 & \text{---} & \text{---} & 0.250 & \text{---} & \text{---} \\
 \text{---} & \text{---} & \text{---} & 0.200 & 0.200 & \text{---} & \text{---} & 0.200 & 0.200 & \text{---} & 0.200 & \text{---} \\
 \text{---} & \text{---} & \text{---} & 0.200 & \text{---} & \text{---} & \text{---} & 0.200 & 0.200 & \text{---} & 0.200 & 0.333 \\
 0.200 & \text{---} & \text{---} & \text{---} & \text{---} & 0.333 & 0.250 & \text{---} & \text{---} & 0.250 & \text{---} & \text{---} \\
 \text{---} & \text{---} & \text{---} & 0.200 & \text{---} & \text{---} & \text{---} & 0.200 & 0.200 & \text{---} & 0.200 & 0.333 \\
 \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & 0.200 & \text{---} & 0.200 & 0.333
 \end{pmatrix}$$

$$M = \mathcal{T}_{G_3+I}$$

$$\begin{pmatrix}
 0.257 & 0.113 & 0.063 & \text{---} & 0.100 & 0.261 & 0.175 & \text{---} & \text{---} & 0.258 & \text{---} & \text{---} \\
 0.090 & 0.225 & 0.175 & 0.050 & 0.140 & 0.067 & 0.100 & 0.040 & \text{---} & 0.050 & \text{---} & \text{---} \\
 0.050 & 0.175 & 0.225 & 0.090 & 0.140 & \text{---} & 0.050 & 0.080 & 0.040 & \text{---} & 0.040 & \text{---} \\
 \text{---} & 0.063 & 0.113 & 0.210 & 0.090 & \text{---} & \text{---} & 0.160 & 0.160 & \text{---} & 0.160 & 0.133 \\
 0.100 & 0.175 & 0.175 & 0.090 & 0.230 & \text{---} & 0.113 & 0.080 & 0.040 & 0.063 & 0.040 & \text{---} \\
 0.157 & 0.050 & \text{---} & \text{---} & \text{---} & 0.261 & 0.113 & \text{---} & \text{---} & 0.195 & \text{---} & \text{---} \\
 0.140 & 0.100 & 0.050 & \text{---} & 0.090 & 0.150 & 0.225 & 0.040 & \text{---} & 0.175 & \text{---} & \text{---} \\
 \text{---} & 0.050 & 0.100 & 0.160 & 0.080 & \text{---} & 0.050 & 0.200 & 0.160 & \text{---} & 0.160 & 0.133 \\
 \text{---} & \text{---} & 0.050 & 0.160 & 0.040 & \text{---} & \text{---} & 0.160 & 0.227 & \text{---} & 0.227 & 0.244 \\
 0.207 & 0.050 & \text{---} & \text{---} & 0.050 & 0.261 & 0.175 & \text{---} & \text{---} & 0.258 & \text{---} & \text{---} \\
 \text{---} & \text{---} & 0.050 & 0.160 & 0.040 & \text{---} & \text{---} & 0.160 & 0.227 & \text{---} & 0.227 & 0.244 \\
 \text{---} & \text{---} & \text{---} & 0.080 & \text{---} & \text{---} & \text{---} & 0.080 & 0.147 & \text{---} & 0.147 & 0.244
 \end{pmatrix}$$

$$M^2$$

$$\begin{pmatrix}
 0.213 & 0.133 & 0.069 & 0.013 & 0.090 & 0.259 & 0.198 & 0.020 & \text{---} & 0.238 & \text{---} & \text{---} \\
 0.106 & 0.158 & 0.148 & 0.053 & 0.136 & 0.069 & 0.095 & 0.046 & 0.018 & 0.077 & 0.018 & \text{---} \\
 0.055 & 0.148 & 0.158 & 0.095 & 0.134 & 0.017 & 0.060 & 0.078 & 0.050 & 0.025 & 0.050 & 0.027 \\
 0.013 & 0.066 & 0.119 & 0.161 & 0.085 & \text{---} & 0.023 & 0.156 & 0.165 & \text{---} & 0.165 & 0.151 \\
 0.090 & 0.170 & 0.168 & 0.085 & 0.155 & 0.054 & 0.126 & 0.096 & 0.050 & 0.069 & 0.050 & 0.027 \\
 0.155 & 0.052 & 0.013 & \text{---} & 0.033 & 0.205 & 0.116 & \text{---} & \text{---} & 0.182 & \text{---} & \text{---} \\
 0.158 & 0.095 & 0.060 & 0.018 & 0.101 & 0.155 & 0.158 & 0.026 & 0.008 & 0.173 & 0.008 & \text{---} \\
 0.020 & 0.058 & 0.098 & 0.156 & 0.096 & \text{---} & 0.033 & 0.152 & 0.163 & 0.013 & 0.163 & 0.151 \\
 \text{---} & 0.023 & 0.063 & 0.165 & 0.050 & \text{---} & 0.010 & 0.163 & 0.204 & \text{---} & 0.204 & 0.233 \\
 0.190 & 0.077 & 0.025 & \text{---} & 0.055 & 0.242 & 0.173 & 0.010 & \text{---} & 0.225 & \text{---} & \text{---} \\
 \text{---} & 0.023 & 0.063 & 0.165 & 0.050 & \text{---} & 0.010 & 0.163 & 0.204 & \text{---} & 0.204 & 0.233 \\
 \text{---} & \text{---} & 0.020 & 0.091 & 0.016 & \text{---} & \text{---} & 0.091 & 0.140 & \text{---} & 0.140 & 0.179
 \end{pmatrix}$$

$$M^3$$

$$\begin{pmatrix}
 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 \\
 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 \\
 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 \\
 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 \\
 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 \\
 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 \\
 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 \\
 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 \\
 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 \\
 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 & 0.077 \\
 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 & 0.096 \\
 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058 & 0.058
 \end{pmatrix}$$

$$M^\infty$$

Figure 12. Powers of  $M = \mathcal{T}_{G_3+I}$ , the Markov matrix associated with the graph  $G_3$  in Figure 10, loops added to  $G_3$

positions with nearly identical values will still be close to each other after rescaling. This is explained by observing that what effectively happens is that all *ratios*  $T_{p_1q}/T_{p_2q}$  are raised to the same power. Below four vectors and their image after rescaling with power coefficient 2 are listed. The notation  $\Gamma_r v$  is introduced right after these examples.

$$\begin{array}{l} \text{Vector } v: \\ \text{Image } \Gamma_2 v: \end{array} \begin{array}{ccccc} \begin{pmatrix} 0 \\ 3 \\ 0 \\ 1 \\ 2 \end{pmatrix} & \begin{pmatrix} 0 \\ 1/2 \\ 0 \\ 1/6 \\ 1/3 \end{pmatrix} & \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \\ 0 \end{pmatrix} & \begin{pmatrix} 0.151 \\ 0.159 \\ 0.218 \\ 0.225 \\ 0.247 \end{pmatrix} & \begin{pmatrix} 0.086 \\ 0.000 \\ 0.113 \\ 0.801 \\ 0.000 \end{pmatrix} \\ \begin{pmatrix} 0 \\ 9/14 \\ 0 \\ 1/14 \\ 4/14 \end{pmatrix} & \begin{pmatrix} 0 \\ 9/14 \\ 0 \\ 1/14 \\ 4/14 \end{pmatrix} & \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \\ 0 \end{pmatrix} & \begin{pmatrix} 0.110 \\ 0.122 \\ 0.229 \\ 0.245 \\ 0.295 \end{pmatrix} & \begin{pmatrix} 0.011 \\ 0.000 \\ 0.019 \\ 0.970 \\ 0.000 \end{pmatrix} \end{array}$$

DEFINITION 3. Given a matrix  $M \in \mathbb{R}^{k \times l}$ ,  $M \geq 0$ , and a real nonnegative number  $r$ , the matrix resulting from rescaling each of the columns of  $M$  with power coefficient  $r$  is called  $\Gamma_r M$ , and  $\Gamma_r$  is called the **inflation** operator with power coefficient  $r$ . Formally, the action of  $\Gamma_r : \mathbb{R}^{k \times l} \rightarrow \mathbb{R}^{k \times l}$  is defined by

$$(\Gamma_r M)_{pq} = (M_{pq})^r / \sum_{i=1}^k (M_{iq})^r$$

If the subscript is omitted, it is understood that the power coefficient equals 2. □

There are no restrictions on the matrix dimensions to fit a square matrix, because this allows  $\Gamma_r$  to act on both matrices and column vectors. There is no restriction that the input matrices be stochastic, since it is not strictly necessary, and the extended applicability is sometimes useful. The parameter  $r$  is assumed rather than required to be nonnegative. The reason is that in the setting of the MCL process nonnegative values  $r$  have a sensible interpretation attached to them. Values of  $r$  between 0 and 1 increase the homogeneity of the argument probability vector (matrix), whereas values of  $r$  between 1 and  $\infty$  increase the inhomogeneity. In both cases, the ordering of the probabilities is not disturbed. Negative values of  $r$  invert the ordering, which does not seem to be of apparent use.

DEFINITION 4. A nonnegative vector  $v$  is called **homogeneous** if all its nonzero entries are equal. A nonnegative matrix is called **column-homogeneous** if each of its columns is homogeneous. □

The set of homogeneous probability vectors is precisely the set of vectors which are invariant under  $\Gamma_r$ ,  $r \neq 1$ . When applied to vectors, the  $\Gamma_r$  operator has a nice mathematical property in terms of *majorization*. This is discussed in the following chapter, Section 6.2. Perhaps surprisingly, the  $\Gamma_r$  operator maps a rather large class of matrices with real spectrum onto itself, and if  $r \in \mathbb{N}$ , the subset of this class with nonnegative spectrum is preserved as well. These classes are introduced in Chapter 7.

**5.5.2 Iterating expansion and inflation.** Figure 13 gives the result of applying  $\Gamma_r$  to the Markov matrix  $M^2$  given in Figure 12. The vital step now is to iterate the process of alternately expanding information flow via normal matrix multiplication and contracting information flow via application of  $\Gamma_r$ . Thus, the matrix  $\Gamma_r M^2$  is squared, and the inflation operator is applied to the result. This process is repeated ad libitum. The invariant of the process is that flow in dense regions profits from both the expansion and the inflation step. A priori it is uncertain whether the process converges, or whether convergence will lead to a meaningful limit. However, the heuristic which leads to the formulation of the process suggests that something will happen for graphs possessing sparse boundaries. The transition values corresponding to edges crossing sparse boundaries are given a hard time by the process, and if anything, it is to be expected that they will tend to zero. This is exactly what happens for the example graph. The 5<sup>th</sup> iterand, the 9<sup>th</sup> iterand, and the invariant limit<sup>5</sup> of this process (provisionally denoted by  $M_{mcl}^\infty$ ) are given in Figure 13 as well.

The matrix  $M_{mcl}^\infty$  clearly is an idempotent under both matrix multiplication and the inflation operator. It has a straightforward interpretation as a clustering. Four nodes can be said to be an *attractor*, namely those nodes that have positive return probability. The nodes 9 and 11 are as much attracted to each other as they are to themselves. The rest of the vertex set of  $G_3$  can be completely partitioned according to the nodes to which they are attracted. Sweeping attractors and the elements they attract together, the partition  $\{4, 8, 9, 11, 12\} \{1, 6, 7, 10\} \{2, 3, 5\}$  results, also found earlier with  $k$ -path clustering.

In the next section the *MCL* process is formally described, and the relationship between equilibrium states of the *MCL* process and clusterings is formalized. A certain subset of the equilibrium states only admits an interpretation as a clustering with overlap. This is related to the presence of symmetry in the graphs and matrices used. Consider the matrix  $M$  depicted in Figure 14, corresponding with a line-graph on 7 nodes, loops added to each node. An *MCL* run with  $e_{(i)} \stackrel{\text{def}}{=} 2, r_{(i)} \stackrel{\text{def}}{=} 2$  results in the limit  $T_{mcl}^\infty$ . The nodes 2 and 6 are attractors, the node sets  $\{1, 3\}$ , and  $\{5, 7\}$ , are respectively attracted to them. The vertex 4 is equally attracted to 2 and 6. The formation of two clusters, or different regions of attraction, is explained by the fact that the nodes at the far ends, i.e. 1, 2, 6, 7 have higher return probability after the first iterations than the nodes in the middle. Given the symmetry of the graph, it is only natural that node 4 is equally attracted to both regions.

## 5.6 Formal description of the *MCL* algorithm

The basic design of the *MCL* algorithm is given in Figure 15; it is extremely simple and provides basically an interface to the *MCL* process, introduced below. The main skeleton is formed by the alternation of matrix multiplication and inflation in a for loop. In the  $k^{\text{th}}$  iteration of this loop two matrices labelled  $T_{2k}$  and  $T_{2k+1}$  are computed. The matrix  $T_{2k}$  is computed as the previous matrix  $T_{2k-1}$  taken to the power  $e_k$ . The matrix  $T_{2k+1}$  is computed as the image of  $T_{2k}$  under  $\Gamma_{r_k}$ . The row<sup>6</sup> of expansion powers  $e_{(i)}$  and the

<sup>5</sup>Idempotent under both  $\text{Exp}_2$  and  $\Gamma_2$ .

<sup>6</sup>The notation  $e_{(i)}$  is shorthand for  $\{e_i\}_{i \in N}$  and likewise  $r_{(i)}$  for  $\{r_i\}_{i \in N}$ .

$$\begin{pmatrix} 0.380 & 0.087 & 0.027 & -- & 0.077 & 0.295 & 0.201 & -- & -- & 0.320 & -- & -- \\ 0.047 & 0.347 & 0.210 & 0.017 & 0.150 & 0.019 & 0.066 & 0.012 & -- & 0.012 & -- & -- \\ 0.014 & 0.210 & 0.347 & 0.056 & 0.150 & -- & 0.016 & 0.046 & 0.009 & -- & 0.009 & -- \\ -- & 0.027 & 0.087 & 0.302 & 0.062 & -- & -- & 0.184 & 0.143 & -- & 0.143 & 0.083 \\ 0.058 & 0.210 & 0.210 & 0.056 & 0.406 & -- & 0.083 & 0.046 & 0.009 & 0.019 & 0.009 & -- \\ 0.142 & 0.017 & -- & -- & -- & 0.295 & 0.083 & -- & -- & 0.184 & -- & -- \\ 0.113 & 0.069 & 0.017 & -- & 0.062 & 0.097 & 0.333 & 0.012 & -- & 0.147 & -- & -- \\ -- & 0.017 & 0.069 & 0.175 & 0.049 & -- & 0.016 & 0.287 & 0.143 & -- & 0.143 & 0.083 \\ -- & -- & 0.017 & 0.175 & 0.012 & -- & -- & 0.184 & 0.288 & -- & 0.288 & 0.278 \\ 0.246 & 0.017 & -- & -- & 0.019 & 0.295 & 0.201 & -- & -- & 0.320 & -- & -- \\ -- & -- & 0.017 & 0.175 & 0.012 & -- & -- & 0.184 & 0.288 & -- & 0.288 & 0.278 \\ -- & -- & -- & 0.044 & -- & -- & -- & 0.046 & 0.120 & -- & 0.120 & 0.278 \end{pmatrix}$$

$\Gamma_2 M^2$ ,  $M$  defined in Figure 12

$$\begin{pmatrix} 0.448 & 0.080 & 0.023 & -- & 0.068 & 0.426 & 0.359 & -- & -- & 0.432 & -- & -- \\ 0.018 & 0.285 & 0.228 & 0.007 & 0.176 & 0.006 & 0.033 & 0.005 & -- & 0.007 & -- & -- \\ 0.005 & 0.223 & 0.290 & 0.022 & 0.173 & -- & 0.010 & 0.017 & 0.003 & 0.001 & 0.003 & 0.001 \\ -- & 0.018 & 0.059 & 0.222 & 0.040 & -- & 0.001 & 0.187 & 0.139 & -- & 0.139 & 0.099 \\ 0.027 & 0.312 & 0.314 & 0.028 & 0.439 & 0.005 & 0.054 & 0.022 & 0.003 & 0.010 & 0.003 & 0.001 \\ 0.116 & 0.007 & 0.001 & -- & 0.004 & 0.157 & 0.085 & -- & -- & 0.131 & -- & -- \\ 0.096 & 0.040 & 0.013 & -- & 0.037 & 0.083 & 0.197 & 0.001 & -- & 0.104 & -- & -- \\ -- & 0.012 & 0.042 & 0.172 & 0.029 & -- & 0.002 & 0.198 & 0.133 & -- & 0.133 & 0.096 \\ -- & 0.001 & 0.015 & 0.256 & 0.009 & -- & -- & 0.266 & 0.326 & -- & 0.326 & 0.346 \\ 0.290 & 0.021 & 0.002 & -- & 0.017 & 0.323 & 0.260 & -- & -- & 0.316 & -- & -- \\ -- & 0.001 & 0.015 & 0.256 & 0.009 & -- & -- & 0.266 & 0.326 & -- & 0.326 & 0.346 \\ -- & -- & 0.001 & 0.037 & 0.001 & -- & -- & 0.039 & 0.069 & -- & 0.069 & 0.112 \end{pmatrix}$$

$\Gamma_2(\Gamma_2 M^2 \cdot \Gamma_2 M^2)$

$$\begin{pmatrix} 0.807 & 0.040 & 0.015 & -- & 0.034 & 0.807 & 0.807 & -- & -- & 0.807 & -- & -- \\ -- & 0.090 & 0.092 & -- & 0.088 & -- & -- & -- & -- & -- & -- & -- \\ -- & 0.085 & 0.088 & -- & 0.084 & -- & -- & -- & -- & -- & -- & -- \\ -- & 0.001 & 0.001 & 0.032 & 0.001 & -- & -- & 0.032 & 0.031 & -- & 0.031 & 0.031 \\ -- & 0.777 & 0.798 & -- & 0.786 & -- & 0.001 & -- & -- & -- & -- & -- \\ 0.005 & -- & -- & -- & -- & 0.005 & 0.005 & -- & -- & 0.005 & -- & -- \\ 0.003 & 0.001 & -- & -- & 0.001 & 0.003 & 0.003 & -- & -- & 0.003 & -- & -- \\ -- & -- & 0.001 & 0.024 & -- & -- & -- & 0.024 & 0.024 & -- & 0.024 & 0.024 \\ -- & -- & 0.002 & 0.472 & 0.001 & -- & -- & 0.472 & 0.472 & -- & 0.472 & 0.472 \\ 0.185 & 0.005 & 0.001 & -- & 0.004 & 0.185 & 0.184 & -- & -- & 0.185 & -- & -- \\ -- & -- & 0.002 & 0.472 & 0.001 & -- & -- & 0.472 & 0.472 & -- & 0.472 & 0.472 \\ -- & -- & -- & 0.001 & -- & -- & -- & 0.001 & 0.001 & -- & 0.001 & -- \end{pmatrix}$$

$(\Gamma_2 \circ \text{Squaring})$  iterated four times on  $M$

$$\begin{pmatrix} 1.000 & -- & -- & -- & -- & 1.000 & 1.000 & -- & -- & 1.000 & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & 1.000 & 1.000 & -- & 1.000 & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\ -- & -- & -- & 0.500 & -- & -- & -- & 0.500 & 0.500 & -- & 0.500 & 0.500 \\ -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- & -- \end{pmatrix}$$

$M_{mcl}^\infty$

Figure 13. Iteration of  $(\Gamma_2 \circ \text{Squaring})$  with initial iterand  $M$  defined in Figure 12.

Entries marked ‘--’ are either zero because that is the exact value they assume (this is true for the first two matrices) or because the computed value fell below the machine precision.

$$\begin{pmatrix} 0.5000 & 0.3333 & -- & -- & -- & -- & -- \\ 0.5000 & 0.3333 & 0.3333 & -- & -- & -- & -- \\ -- & 0.3333 & 0.3333 & 0.3333 & -- & -- & -- \\ -- & -- & 0.3333 & 0.3333 & 0.3333 & -- & -- \\ -- & -- & -- & 0.3333 & 0.3333 & 0.3333 & -- \\ -- & -- & -- & -- & 0.3333 & 0.3333 & 0.5000 \\ -- & -- & -- & -- & -- & 0.3333 & 0.5000 \end{pmatrix}$$

Initial iterand  $T_1 = M$ 

$$\begin{pmatrix} 0.3221 & 0.2393 & 0.0493 & 0.0028 & 0.0000 & -- & -- \\ 0.6138 & 0.6120 & 0.2664 & 0.0420 & 0.0021 & 0.0000 & -- \\ 0.0606 & 0.1275 & 0.4259 & 0.2165 & 0.0383 & 0.0010 & 0.0000 \\ 0.0035 & 0.0200 & 0.2159 & 0.4662 & 0.2143 & 0.0200 & 0.0034 \\ 0.0000 & 0.0011 & 0.0403 & 0.2259 & 0.4311 & 0.1282 & 0.0607 \\ -- & 0.0000 & 0.0022 & 0.0436 & 0.2652 & 0.6116 & 0.6137 \\ -- & -- & 0.0000 & 0.0029 & 0.0490 & 0.2392 & 0.3220 \end{pmatrix}$$

Intermediate iterand  $T_5$  ( $k$  equals 2)

$$\begin{pmatrix} 0.0284 & 0.0280 & 0.0191 & 0.0015 & 0.0000 & 0.0000 & 0.0000 \\ 0.9647 & 0.9631 & 0.8226 & 0.1205 & 0.0016 & 0.0000 & 0.0000 \\ 0.0066 & 0.0082 & 0.0768 & 0.1362 & 0.0087 & 0.0000 & 0.0000 \\ 0.0003 & 0.0006 & 0.0686 & 0.4309 & 0.0673 & 0.0006 & 0.0003 \\ 0.0000 & 0.0000 & 0.0109 & 0.1677 & 0.0863 & 0.0088 & 0.0069 \\ 0.0000 & 0.0000 & 0.0020 & 0.1414 & 0.8173 & 0.9627 & 0.9644 \\ 0.0000 & 0.0000 & 0.0000 & 0.0018 & 0.0187 & 0.0280 & 0.0284 \end{pmatrix}$$

Intermediate iterand  $T_9$  ( $k$  equals 4)

$$\begin{pmatrix} -- & -- & -- & -- & -- & -- & -- \\ 1.0000 & 1.0000 & 1.0000 & 0.5000 & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- & -- & -- \\ -- & -- & -- & 0.5000 & 1.0000 & 1.0000 & 1.0000 \\ -- & -- & -- & -- & -- & -- & -- \end{pmatrix}$$

Limit  $T_{mcl}^\infty$  (idempotent under  $\text{Exp}_2$  and  $\Gamma_2$ ).Figure 14. *MCL* run on a line-graph on 7 nodes

row of inflation powers  $r_{(i)}$  influence the granularity of the resulting partition. The matrices in Figure 13 correspond with an *MCL* session in which  $e_{(i)} \leq 2$  and  $r_{(i)} \leq 2$ . If the current iterand is sufficiently close to an idempotent matrix the process stops and the last resultant is interpreted according to Definition 8 and Theorem 1 in the next chapter. The theorem provides a mapping from the set of nonnegative column allowable idempotent matrices to the set of overlapping clusterings. There are exceptional cases in which the iterands cycle around a periodic limit. These cases, and the issues of convergence and equilibrium states at large, are discussed in the following chapter. It is useful to

```

MCL (G, Δ, e(i), r(i)) {
    # G is a voidfree graph.
    # ei ∈ ℕ, ei > 1, i = 1, ...
    # ri ∈ ℝ, ri > 0, i = 1, ...

    G = G + Δ;
    T1 = TG;

    # Possibly add (weighted) loops.
    # Create associated Markov graph
    # according to Definition 2.

    for k = 1, ..., ∞ {
        T2k = Expek(T2k-1);
        T2k+1 = Γrk(T2k);
        if (T2k+1 is (near-) idempotent) break;
    }
    Interpret T2k+1) as clustering according to Definition 8;
}

```

Figure 15. The basic MCL algorithm. Convergence is discussed in Chapter 6.

speak about the algebraic process which is computed by the MCL algorithm in its own right. To this end, the notion of an MCL process is defined.

DEFINITION 5. A nonnegative column-homogeneous matrix  $M$  which is idempotent under matrix multiplication is called **doubly idempotent**.  $\square$

DEFINITION 6. A general MCL **process** is determined by two rows of exponents  $e_{(i)}$ ,  $r_{(i)}$ , where  $e_i \in \mathbb{N}$ ,  $e_i > 1$ , and  $r_i \in \mathbb{R}$ ,  $r_i > 0$ , and is written

$$(5) \quad (\cdot, e_{(i)}, r_{(i)})$$

An MCL process for stochastic matrices of fixed dimension  $d \times d$  is written

$$(6) \quad (\cdot^{d \times d}, e_{(i)}, r_{(i)})$$

An MCL process with input matrix  $M$ , where  $M$  is a stochastic matrix, is determined by two rows  $e_{(i)}$ ,  $r_{(i)}$  as above, and by  $M$ . It is written

$$(7) \quad (M, e_{(i)}, r_{(i)})$$

Associated with an MCL process  $(M, e_{(i)}, r_{(i)})$  is an infinite row of matrices  $T_{(i)}$  where  $T_1 = M$ ,  $T_{2i} = \text{Exp}_{e_i}(T_{2i-1})$ , and  $T_{2i+1} = \Gamma_{r_i}(T_{2i})$ ,  $i = 1, \dots, \infty$ .  $\square$

In practice, the algorithm iterands converge nearly always to a doubly idempotent matrix. In the next section it is shown that the MCL process converges quadratically in the neighbourhood of doubly idempotent matrices. A sufficient property for associating a (possibly overlapping) clustering with a nonnegative column allowable matrix is that the matrix is idempotent under matrix multiplication. In Chapter 7 it is shown that the mapping of idempotent matrices onto overlapping clusterings according to Definition 8 can

be generalized towards a mapping of time-reversible Markov matrices with nonnegative spectrum onto directed acyclic graphs. This is not a generalization in the strict sense, because stochastic idempotent matrices are in general not time-reversible. However, in Chapter 7 it is shown that the *MCL* process offers a perspective in which idempotent matrices are the extreme points of the set of time-reversible Markov matrices with nonnegative spectrum. The figure below shows the clustering resulting from applying the *MCL* algorithm with standard parameters  $e_{(i)} \stackrel{\text{def}}{=} 2$  and  $r_{(i)} \stackrel{\text{def}}{=} 2$  to the example graph in Figure 5 taken from [121], loops added to the graph.

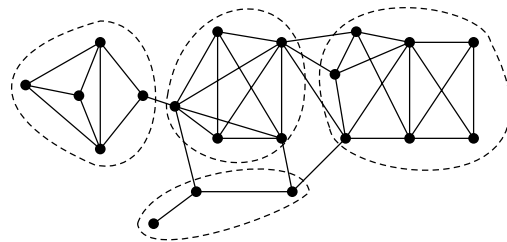


Figure 16. *MCL* Clustering of the graph in Figure 5.