Centrum voor Wiskunde en Informatica

A new cluster algorithm for graphs

S. van Dongen

Information Systems (INS)

INS-R9814 December 31, 1998

Report INS-R9814
ISSN 1386-3681

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

CWI is the National Research Institute for Mathematics and Computer Science. CWI is part of the Stichting Mathematisch Centrum (SMC), the Dutch foundation for promotion of mathematics and computer science and their applications.
SMC is sponsored by the Netherlands Organization for Scientific Research (NWO). CWI is a member of ERCIM, the European Research Consortium for Informatics and Mathematics.

Copyright © Stichting Mathematisch Centrum
P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

A New Cluster Algorithm for Graphs

Stijn van Dongen
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

ABSTRACT

A new cluster algorithm for graphs called the *Markov Cluster algorithm* (*MCL* algorithm) is introduced. The graphs may be both weighted (with nonnegative weight) and directed. Let G be such a graph. The *MCL* algorithm simulates flow in G by first identifying G in a canonical way with a Markov graph G_1. Flow is then alternatingly expanded and contracted, leading to a row of Markov Graphs G_i. The expansion step is done by computing higher step transition probabilities (TP's), the contraction step creates a new Markov graph by favouring high TP's and demoting low TP's in a specific way. The heuristic underlying this approach is the expectation that flow between dense regions which are sparsely connected will evaporate. The stable limits of the process are easily derived and in practice the algorithm converges very fast to such a limit, the structure of which has a generic interpretation as an overlapping clustering of the graph G. Overlap is limited to cases where the input graph has a symmetric structure inducing it. The contraction and expansion parameters of the algorithm influence the granularity of the output. The algorithm is space and time efficient with a space quality/time trade–off, works very well for a wide range of test cases, and lends itself to drastic scaling. Experiments with a scaled C–implementation have been conducted on graphs having several tens of thousands of nodes. This report describes the algorithm, its complexity, and experimental results. The algorithm is introduced by first considering a generalization of generic single link clustering for graphs called k–path clustering.

1991 Mathematics Subject Classification: 05B20, 05B25, 60J15, 62H30, 68T10, 90C35.
Keywords and Phrases: Clustering, graph clustering, random walk, Markov matrix, flow simulation.
Note: Work carried out under project INS 3.2, Digital Libraries.

1. Introduction

In this report I describe the Markov Cluster (*MCL*) algorithm, a new cluster algorithm for graphs which is based on simulation of flow expansion and flow contraction in graphs. This algorithm is specifically suited to sparse graphs, i.e. graphs for which the average node degree is an order of magnitude smaller than the number of nodes in the graph. The algorithm is in fact motivated by considering how the concept of 'cluster' in the setting of sparse graphs can be formalized to some extent.

The idea that clustering in the setting of sparse graphs may very well merit from a separate approach, as opposed to viewing this problem as a minor variant of clustering in a more general setting, does not seem to be widespread. The proposed distinction is between clustering in the setting of sparse graphs and clustering in the setting of attribute spaces or vector spaces. In the first case, the relationship between elements is typically of the kind 'share a property or not'. A famous example springs from the Erdös number defined for a mathematician M, which is the length of the shortest chain of mathematicians $M_1, \ldots, M_l$, such that M_1 is Erdös and M_l is M, and such that M_i and M_{i-1} have co-authored an article. In this case, mathematicians are viewed as the nodes in a graph, and there is a link between two mathematicians (perhaps of strength s) if they have co-authored an article (s articles). In the second case, the relationship between elements is typically defined in terms of a measure on the difference between one or more attributes possessed by the elements, such as weight, length, age, et cetera. The two settings can be viewed as different ends of the same spectrum, and this issue is discussed extensively in [2]. This report deals exclusively with clustering in the setting of sparse graphs.

The basic idea is that dense regions in sparse graphs correspond with regions in which the number of k-length paths is relatively large, for small $k \in \mathbb{N}$. Random walks of length k thus have higher probability for paths with beginning and ending in the same dense region than for other paths. This is especially true if one looks at the subset of all random walks departing from a specific node. If this node is situated in a dense region, random walks departing from it will in general have a tendency to end in the same region. The crucial element in the *MCL* algorithm is that this effect is deliberately boosted by an iterative procedure. First, an input graph G is mapped in a generic way onto a Markov matrix M_1. Then the set of transition probabilities is iteratively recomputed via expansion and inflation. The expansion step corresponds with normal matrix multiplication (on stochastic matrices), the contraction step corresponds with a parametrized operator Γ_r, called the ***inflation operator***, which acts column-wise on (column) stochastic matrices. Henceforth, the term inflation will be used rather than contraction. Via expansion, nodes are able to see new neighbours; via inflation, favoured neighbours are further promoted, and less favoured neighbours are further demoted. For nearly all undirected graphs G, this process triggered by G converges very fast. The structural characteristics of the matrix limit of the process may be very different from the initial Markov matrix M_1. The associated graph of the matrix limit can have a larger number of connected components than the original input graph. This is in fact what makes the algorithm work, since the strongly connected components of the limit, joined with the respective node-sets that reach them, are interpreted as an overlapping clustering of the original graph. As in the usual Markov process, the 'nice' limits are idempotent matrices.

An infinite sequence consisting of repeated alternation of expansion and inflation constitutes a new algebraic process called the Markov Cluster (*MCL*) process. If the inflation operator Γ_r is parametrized such that all inflation steps correspond with the identity operator, a normal Markov process results. Interpretation of the limit then yields a clustering which corresponds with the set of connected components of the original graph. In general, the parameters of the *MCL* process influence the granularity of the clustering. For all testcases, the correlation between input graphs, algorithm parameters, and output clusterings is in line with heuristic considerations. Some examples show surprising strengths of the algorithm. Most notable among these are separating power and the absence of chaining. The algorithm performs well in recognizing clusters which are spherical in nature, also if chains are present. All clusterings that are found have the property that the clusters in them correspond with regions in which there are relatively many k-length paths within. In this sense, it is impossible to find bad clusterings. The number of clusters is influenced by the flow characteristics of the *MCL* process. The cluster granularity can be affected by varying the flow parameters, but the number of clusters need not (and can not) be specified explicitly. The parametrizations of the *MCL* process which are useful for clustering purposes, generally lead to intermediate matrix iterands and equilibrium states which are very sparse in a 'weighted' sense. That is, a column may have many nonzero entries, but most of them are very small compared to the largest entries within the column. This gives the means to scale the algorithm drastically, by applying columnwise pruning.

The heuristic underlying the *MCL* algorithm is first motivated by considering k-path clustering, a generalization of generic single link clustering for graphs. Several phenomena of concern to the *MCL* algorithm also play a role in the conceptually simpler setting of k-path clustering. An example of this is the effect of adding loops to a graph in order to improve, for the purpose of clustering, its topological properties. I have not found a description of k-path clustering in the literature. In this report, it serves as a tool supporting the main contribution, the *MCL* algorithm.

The *MCL* process is a complex process which, to the best of knowledge, is described here for the first time. The inflation operator , introduced in Section 4, does not distribute over the normal matrix product, as acts on matrices column-wise. For a normal Markov process, the columns of any iterand lie within the convex hull of the columns of any previous iterand. This is not true for the *MCL* process, which is due to again. However, the *MCL* process has remarkable convergence properties. The 'nice' equilibrium states of the process are easily derived, and in practice the algorithm converges nearly always to such a limit. Exceptions to this rule are quite rare. The only ones found so far were made by construction, and are in line with heuristic considerations. They are discussed in Section 7. This section also gives a classification of the theoretically possible equilibrium states of the process, with examples from each of the introduced classes. The class of nice equilibrium states is further subdivided into two categories in Section 8. It is shown that in the neighbourhood of the equilibrium states in the first subclass the *MCL* process converges quadratically towards equilibrium. Then it is shown that the equilibrium states S in the second subclass are instable, but that the *MCL* process converges quadratically at least on a macroscopic scale, once close enough to such an equilibrium state S. That is, it is proven that the structural form of the elements of *MCL* process converges towards the same block structure as present in S. Roughly speaking, the conclusion is that the phenomenon of cluster overlap is instable in nature, and that otherwise the instability of an equilibrium state, c.q. perturbation followed by convergence towards another equilibrium state, does not change the associated clustering.

In Section 2 the setting, notation, and terminology are formally introduced. Section 3 introduces k-path clustering. The section after that shows an example run of the *MCL* algorithm. Section 5 gives a formal description of the *MCL* process and the *MCL* algorithm. This includes a theorem which gives the means to formally associate an overlapping clustering with the limits generically resulting from an *MCL* process. Section 6 gives mathematical properties of the inflation operator . In Section 7 the equilibrium states of the *MCL* process are categorized. These include instable states for which expansion and inflation act as each other inverse. In Section 8 local convergence properties of the *MCL* process are studied, followed by a section in which cluster properties of the algorithm are discussed. Section 10 is concerned with complexity and scalability of the algorithm. It is shown that the algorithm can be scaled drastically for large graphs allowing clusterings in which the natural cluster size is relatively small. This section ends with a benchmark proposal. Conclusions and further research make up the last section.

2. Basic terminology

This section introduces the needed terminology for graphs, matrices, (dis)similarity spaces, and clusterings.

Definition 1 Let V be a finite collection of elements, enumerated $v_0, \ldots, v_{t-1}$.

i) A **weighted graph** G on V is a pair V, w, where w is a function mapping pairs of elements of V to the nonnegative reals: $w : V \quad V \,!\, I\!R_{0}$.

   a) G is called **undirected** if w is symmetric, it is called **directed** otherwise.

   b) G is said to be **irreflexive** if there are no *loops* in G, that is, $w\ v, v \quad 0; 8\ v\ 2\ V$.

ii) A **dissimilarity space** $D \quad V, d$ is a pair V, d, where s is a symmetric function mapping $V \quad V$ to $I\!R_{0}$, satisfying $s\ u, v \quad 0\ (\,)\quad u \quad v$. The function d is called a **dissimilarity measure** or **dissimilarity coefficient**.

iii) A **similarity space** is a pair (V, s), where s is a symmetric function mapping $V \times V$ to $\mathbb{R}_{>0} \cup \{\infty\}$, satisfying $s(u,v) = \infty \iff u = v$. The function s is called a **similarity measure** or **similarity coefficient**.

The definition of a dissimilarity coefficient is identical to that of an irreflexive undirected weighted complete graph (see below for the definition of weighted complete). However, this is not a very natural relationship if the set of all irreflexive undirected weighted graphs is considered. The problem is that the weight function of undirected weighted graphs which are not complete is best viewed as being of a similarity nature, where longer distance dependencies give extra information about elements which are not immediately related. More sophisticated approaches such as the ones suggested in [4] construct a dissimilarity measure from the weight function by considering such longer distance dependencies. The dissimilarity measure thus constructed looks very different from the original weight function. It follows that a dissimilarity space is better viewed as a relaxation of a metric space than as a special instance of the set of irreflexive undirected weighted graphs. In this report, I shall be using similarity coefficients rather than dissimilarity coefficients (in Section 3). The two types are interchangeable if it is only the relative ordering of the values assumed by them that matters (e.g. one can be mapped onto the other by the map $x \mapsto 1/x$). The latter condition holds in the setting of single link clustering, and this method (see Definition 3) will be used in Section 3.

Definition 2 Let $G = (V, w)$ be a finite weighted directed graph with $|V| = t$. The **associated matrix** of G lying in $\mathbb{R}_{\geq 0}^{t \times t}$, denoted $\mathcal{M}_G$, is defined by setting the entry $(\mathcal{M}_G)_{pq}$ equal to $w(v_p, v_q)$. Given a matrix $M \in \mathbb{R}_{\geq 0}^{N \times N}$, the **associated graph** of M is written Γ_M, which is the graph (V, w) with $|V| = N$ and $w(v_p, v_q) = M_{pq}$.

Matrices and graphs of dimension N are indexed using indices running from 0 to $N-1$. Other enumeration types are usually indexed starting with 1. If u, v are nodes for which $w(u, v) > 0$, I say that there is an arc going from v to u with weight $w(u, v)$. Then v is called the **tail node**, and u is called the **head node**. The reason for this ordering lies in the fact that graphs will be transformed later on into stochastic matrices, and that I find it slightly more convenient to work with column stochastic matrices rather than row stochastic matrices. A **column stochastic** matrix is a nonnegative matrix in which the entries of each column sum to one. The **degree** of a node is the number of arcs originating from it. A graph is called **void–free** in this report if there is at least one arc originating from each node. A matrix is called void-free if its associated graph is void-free, that is, it has no zero columns. Note that a column stochastic matrix is by definition void-free. A **path** of length p in G is a sequence of nodes $v_{i_0}, \ldots, v_{i_p}$ such that $w(v_{i_k}, v_{i_{k-1}}) > 0$, $k = 0, \ldots, p$. The path is called a **circuit** if $i_0 = i_p$, it is called a **simple path** if all indices i_k are distinct, i.e. no circuit is contained in it. A circuit is called a **loop** if it has length 1. If the weight function w is symmetric then the arcs (v_k, v_l) and (v_l, v_k) are not distinguished, and G is said to have an **edge** (v_l, v_k) with weight $w(v_l, v_k)$. The two points v_l, v_k are then said to be connected. A **simple graph** is an undirected graph in which every nonzero weight equals 1. The simple graph on t points in which all node pairs $u, v, u \neq v$, are connected via an edge (yielding $t(t-1)$ edges in all) is denoted by K_t, and is called the **complete** graph on t points. A weighted directed graph for which $w(u, v) > 0, \forall u \neq v$, is called a **weighted complete** graph.

Let $G = (V, w)$ be a finite directed weighted graph $G = (V, w)$. In this report the interpretation of the weight function w is that the value $w(u, v)$ gives the *capacity* of the arc (path of length 1) going from v to u, or (equivalently) the *number of paths of length* 1 from v to u. The latter interpretation takes the concept 'the number of' to a higher level of abstraction, since w may assume all values in the nonnegative reals. Let G be a simple graph, let $M = \mathcal{M}_G$ be its associated matrix. The capacity interpretation of the weight function w is very natural in view of the fact that the pq entry of the

k^{th} power M^k gives exactly the number of paths of length k between v_p and v_q. This can be verified by a straightforward computation. There is in fact nothing special about the weight function of a simple graph with respect to this property (except that it allows easy manual verification), and the given interpretation of the entries of M^k extends to the class of finite weighted directed graphs.

Notation The graph which is formed by adding loops to G is denoted by $G + I$. In general, if Δ is a nonnegative diagonal matrix, then $G + \Delta$ denotes the graph which results from adding to each node v_i in G a loop with weight Δ_{ii}.

Definition 3 A **partition** or **clustering** of V is a collection of pairwise disjoint sets $\{V_1, \ldots, V_d\}$ such that each set V_i is a nonempty subset of V and the union $\cup_{i=1,\ldots,d} V_i$ is V. A partition Π is called (**top** respectively **bottom**[1]) **extreme** if respectively $\Pi = \{V\}$ and $\Pi = \{\text{singletons } V\} = \{\{v_0\}, \ldots, \{v_{t-1}\}\}$.

A **hierarchical clustering** of V is a finite ordered list of partitions $\Pi_i, i = 1, \ldots, n$ of V, such that for all $1 \leq i < j \leq n$ the partition Π_j can be formed from Π_i by conjoining elements of Π_i, where $\Pi_1 = \{\text{singletons } V\} = \{\{v_0\}, \ldots, \{v_{t-1}\}\}$ and $\Pi_n = \{V\}$.

An **overlapping clustering** of V is a collection of sets $\{V_1, \ldots, V_d\}$, $d \in \mathbb{N}$, such that each set V_i is a nonempty subset of V, the union $\cup_{i=1,\ldots,d} V_i$ is V, and each subset V_i is not contained in the union of the other subsets $V_j, j \neq i$. The latter implies that each subset V_i contains at least one element not contained in any of the other subsets, and this in turn implies the inequality $d \leq t$.

Let S be a similarity coefficient defined on $V = \{v_0, \ldots, v_{t-1}\}$. Let $s_0, \ldots, s_n$ be the row of different values that S assumes on $V \times V$, in strictly descending order and with the value 0 added. Remember that $S(u, u) = 1, u \in V$. Thus, $1 = s_0 > s_1 > \ldots > s_n \geq 0$.

The **single link clustering** of the pair (V, S) is the nested collection of partitions $\Pi_i, i = 1, \ldots, n$, where each Π_i is the partition induced by the transitive closure of the relation in which two elements $u, v,$ are related iff $S(u, v) \geq s_i$. According to this definition, subsequent partitions may be equal, $\Pi_1 = \{\text{singletons } V\}$, and $\Pi_n = \{V\}$.

Miscellaneous notation

Expressions in which single indices or subscripted or superscripted simple expressions are enclosed in parentheses denote the object which results from letting the index run over its natural boundaries. E.g. (e_i) denotes a vector or a row (the context should leave no doubt which of the two), $(T_{k\,i})$ denotes the k^{th} row of the matrix T, and $((T^i)_{kl})$ denotes the set of kl entries of the powers of T. The fact that each of the entries in a row (e_i) equals the same constant c is concisely written as $(e_i) \stackrel{c}{=} c$.

Let M be a matrix in $\mathbb{R}^{n \times n}$, let α and ω both be sequences of distinct indices in the range $0..n-1$. The submatrix of M corresponding with row indices from α and column indices from ω is written $M[\alpha \mid \omega]$. The principal submatrix with both row and column indices from α is written $M[\alpha]$.

Finally, in this report account will be given of experiments done with the *MCL* algorithm. This means that the realm of finite precision arithmetic is entered. Numerical expressions denote floating point numbers if and only if a dot is part of the expression.

[1] The set of all partitions forms a lattice of which these are the top and bottom elements.

3. k–path clustering

Of the existing procedural algorithms, single link clustering has the most appealing mathematical properties, see e.g. [4]. In this section I shall discuss a variant of single link clustering for graphs which I call k–path clustering. This variant is closely related to 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 quite 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. An example of this is the fact that the cardinality of the set of paths of length k between two nodes depends very much on the parity of k. Following the exposition of k–path clustering its pros and cons are reconsidered.

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 $V = \{v_0, \ldots, v_{t-1}\}$, let $M = \mathcal{M}_G$ be the associated matrix of G. For each integer $k > 0$, I define a similarity coefficient $Z_{k,G}$ associated with G on the set V, by setting $Z_{k,G}(v_i, v_j) = \infty, i = j$, and

$$Z_{k,G}(v_i, v_j) = (M^k)_{ij}, \qquad i \neq j \tag{3.1}$$

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. 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.

Note that the values $(M^i)_{pp}$ are disregarded, which is slightly inelegant and thus unsatisfactory. The problem is not unfamiliar, since it occurs in the generic formulation of single link clustering for simple graphs (yielding the connected components as groups). In this formulation, under the flag of dissimilarity, the values M_{pq} which are zero ($p \neq q$) are set to ∞.

Odd and even

The graph G_1 in Figure 1 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 3. In the same figure the first rows of the matrices $M^i, i = 1, \ldots, 5$ are listed. The other rows of M^i can be found by a suitable permutation of the first row of M^i, due to the symmetry of M.

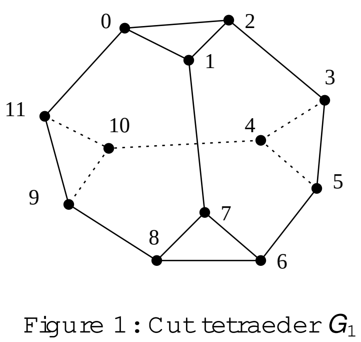

Figure 1: Cut tetraeder G_1

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 $(0, 3)$, for odd k, these are pairs like $(0, 11)$. This clearly has to do with the specific structure of G_1, where the set of paths of odd length leading e.g. from 0 to 1 does not profit from $(0, 1)$ being in a triangle, compared with the set of paths leading from 0 to 11. On the other hand the behaviour of

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

$M = M_{G_1}$ $\qquad\qquad$ M^2

k ; j	0	1	2	3	4	5	6	7	8	9	10	11
1	0	1	1	0	0	0	0	0	0	0	0	1
2	3	1	1	1	0	0	0	1	0	1	1	0
3	2	5	5	1	2	1	1	1	2	1	1	5
4	15	8	8	8	3	4	4	8	3	8	8	4
5	20	31	31	15	20	15	15	15	20	15	15	31

Figure 2: The values M^k_{0j}, $k = 1,\dots,5$, $j = 0,\dots,11$.

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). Loops of odd length only arise in case triangles are present, and start to have influence for paths of length at least 3.

Positioning oneself in a particular node, the similarity distribution around this node is seen to depend heavily on the parity of the simple path connections with the node. It is desirable that while moving away from a node along a simple path, the similarities at least have a tendency to decline radially. In general, if this is not the case, single link clustering may produce clusters which are not connected in the graph. Definition 4 simplifies the formulation of a sufficient criterion for pairs of graphs and similarity coefficients so that clusters at all levels correspond with connected components in the graph. Let G be a graph and let Z be a similarity coefficient defined for G.

Definition 4 The **radial declination number** of the pair (G, Z) is the number of ordered node pairs (s, t), $s \neq t$, such that there is at least one path $s = v_1, v_2, \dots, v_l = t$ between s and t with the property that $Z(s, v_i)$ is decreasing in i, $2 \leq i \leq l$.

The pair (G, Z) is said to have **full radial declination** if the radial declination number is maximal (which is $N(N-1)$ if N is the cardinality of the node set of G).

Lemma 1 A pair (G, Z) which has full radial declination is guaranteed to yield groups which are connected under the neighbour relation inherited from G, at each stage of single link clustering.

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

$M \;\; I;\; M \;\; \mathrm{M}_{G_1}$ $\qquad\qquad$ $M \;\; I^{2}$

k ; j	0	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	0	0	0	0	0	0	0	0	1
2	4	3	3	1	0	0	0	1	0	1	1	2
3	12	11	11	4	2	1	1	4	2	4	4	8
4	42	38	38	18	11	8	8	18	11	18	18	28
5	146	136	136	75	55	45	45	75	55	75	75	106

Figure 3: The values $M \;\; I^{k}_{0j}; k \;\; 1;\ldots;5; j \;\; 0;\ldots;11$.

The proof of this lemma is trivial. Note that each of the pairs $G_1; Z_{k;G_1} \; ; k \;\; 1;\ldots;5$, has full radial declination, with a decrease that unfortunately equals zero along vital edges. It shall be seen that full radial declination cannot be guaranteed unconditionally for the coefficients $Z_{k;G}$, and that in fact this property is not the right property to ask for under all circumstances. First I introduce a countermeasure to the parity dependence of the coefficients $Z_{k;G}$. This countermeasure is not sufficient to assure full radial declination, but it does add power to the coefficients $Z_{k;G}$ and k-path clustering.

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 $Z_{k;G_1 \; I}$ are found in Figure 3, where the largest off-diagonal matrix entries (diagonal entries are disregarded) are printed in boldface. Each coefficient now yields 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_2 depicted in Figure 4. Its associated matrix after adding loops to each node is given next to it in Figure 5. Below are the results of single link clustering at all levels, using the similarity coefficient $Z_{2;G_2 \; I}$.

Level	Clustering
$1;\ldots;6$	{singletons V} {{0};{1};{2};{3};{4};{5};{6};{7};{8};{9};{10};{11}}
5	{{8;10};{0};{1};{2};{3};{4};{5};{6};{7};{9};{11}}
4	{{0;9};{3;7;8;10};{1};{2};{4};{5};{6};{11}}
3	{{0;5;6;9}{1;2;4};{3;7;8;10;11}}
2;1;0	{V} {{0;1;2;3;4;5;6;7;8;9;10;11}}

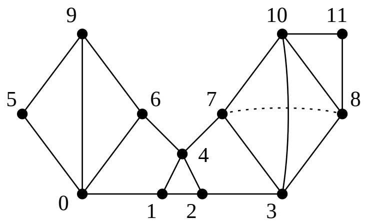

Figure 4: Graph G_4.

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

Figure 5: The matrix $N \; I^2; \; N \; \mathrm{M}_{G_2}$.

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_2+I} (not given here). The coefficient Z_{4,G_2+I} starts out accordingly, however, before node 5 gets involved, the groups $\{3,7,8,10,11\}$ and $\{1,2,4\}$ are joined. This is caused by the fact that node 5 is located in the sparsest part of G_2. The weak spot of single link clustering, namely *chaining*, surfaces here in the specific case of k–path clustering.

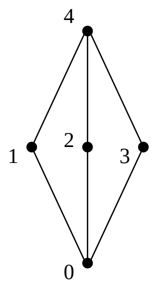

Figure 6: Bipartite graph G_3.

The last example in this section is a graph G_3 for which single link clustering with coefficient $Z_{k,G_3}, k > 1$, initially groups points together which are not connected. The graph G_3 in Figure 6 is a small bipartite graph. The two nodes 0 and 4 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 $(0,4)$ than for any other pair. Generally, bipartite graphs have low radial declination number. However, the bipartite graphs form a class of graphs for which it is natural to cluster each of the two node domains separately[2]. More troublesome are non–bipartite graphs which locally have bipartite characteristics. For these graphs it has to be accepted that intermediate results of single link clustering may contain clusters which are not connected in the graph. The procedure of adding loops does not help in this case.

The following conjecture, if true, substantiates the usefulness of adding loops. It says that for all finite undirected graphs G there exists $N \in \mathbb{N}$ such that for all $k \in \mathbb{N}$ the pair $(G, Z_{k,G+NI})$ has full radial declination. The counterpart of Conjecture 1, stated in Lemma 2, is easy to prove. The lemma states that for all $k, N \in \mathbb{N}$, graphs G exist such that the pair $(G, Z_{k,G+NI})$ does not have full radial declination. The lemma is proven by explicit construction of a class of such graphs G. Intuitively, these are also the 'difficult' graphs with regard to the conjecture. It is easy to see however that the conjecture holds for this class of difficult graphs G, which is heuristic support for it. All the same the conjecture is not easily proven, and its status still unsettled.

Conjecture 1 For all finite undirected graphs G with positive weight function, there exists $N \in \mathbb{N}$, such that for all $k \in \mathbb{N}, k > 1$,

$$(G, Z_{k,G+NI}) \text{ has full radial declination.}$$

[2] e.g. Document phrase–databases naturally yield bipartite graphs. Clustering the two node domains then yields a document grouping and a phrase grouping.

Lemma 2 For all $k \in N, k > 1$ and all $N \in \mathbb{N}, N > 1$, there exists a finite undirected graph G with positive weight function such that

$$(G, Z_{k,G+NI}) \quad \text{does not have full radial declination.}$$

Proof Let k and N be numbers in $\mathbb{N}$, both greater than one. Construct a graph G as follows. Let c be a number in $\mathbb{N}$. Begin with c subgraphs, each of which is a line-graph on $k-1$ nodes. For each subgraph, label one of the ends α, label the other end ω. Now create G by identifying all nodes α as one and identifying all nodes ω as one. Thus, the nodes α and ω have degree c in G, and all other nodes of G have degree 2. There are exactly c paths of length k between α and ω, independent of the presence of loops. The graph G_3 in Figure 6 has this form with parameters $k = 2$ and $c = 3$. Let β be a neighbour of ω. Because $k > 1$ it is true that $\alpha \neq \beta$, and the shortest path between α and β has length $k-1$. In the graph $G' = G + NI$ all paths of length k between α and β must visit a loop exactly once. Such a path has multiplicity N (the weight of the loop), and there are exactly k such paths. It follows that if c is chosen such that $c > Nk$, then $Z_{k,G'}(\alpha, \omega) = c > Nk = Z_{k,G'}(\alpha, \beta)$. Because of symmetry, it then follows that the pair $(G + NI, Z_{k,G'})$ does not have full radial declination.

A critical eye on 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 seems rather difficult to remedy this situation by a regime of disposing of "smaller" elements.

A second minus was mentioned in the discussion of the example graph G_2 in Figure 4. I remarked that under the coefficient Z_{4,G_2} 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 topological 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. With the plusses and minuses of k–path clustering in mind, the *MCL* algorithm is now ready to come to the fore.

4. Introducing the *MCL* algorithm

Given the considerations in the last sections, it seems interesting to investigate models in which the weight of an edge is not interpreted globally, contrary to single link clustering. The *MCL* algorithm achieves this by, for each vertex, rescaling the sum of all weights of all outgoing arcs to one, which in effect boils down to a transformation of the graph G into a Markov graph.

Definition 5 Let G be a graph on d nodes, let $M \quad M_G$ be its associated matrix. The **Markov matrix** associated with a graph G is denoted by T_G and is formally defined by letting its q^{th} column be the q^{th} column of M normalized, as follows.

$$T_{G\,pq} \qquad M_{pq} = \sum_{i\;0}^{d\;1} M_{iq}$$

The Markov matrix T_G corresponds with a graph G^0, which is called the **associated Markov graph** of G. The directed weight function of G^0, which is encoded in the matrix T_G, is called the **localized interpretation** of the weight function of G.

Given a graph G, the *MCL* algorithm first creates the associated Markov matrix $M \quad T_G$, which has the effect that variations in the weight function of G, for each tail node involved, affect the whole corresponding column in M. The matrix M is no longer symmetric. The value M_{pq} now indicates "how much the vertex q is attracted to the vertex p", and this is meaningful only in the context of the other values found in the q^{th} column. It is still possible to move a node away from *all* its neighbours by increasing the weight of its loop. Before the *MCL* algorithm is fully introduced by means of an example session, I consider the behaviour of powers of M. In Figure 7 the matrix $M \quad T_{G_2}$ (corresponding with the graph G_2 in Figure 4) is given which results after the rescaling procedure, followed by three successive powers and a matrix labeled M^1. 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".

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 consists of ergodic regular components only, then the limit of the row N^i is non-cyclic. The graph G_2 in Figure 4 clearly has this property, and the limit is found in Figure 7, denoted as M^1. The columns of M^1 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 setting of nonnegative matrices can be found in [1].

As is to be expected, the equilibrium state e spreads its mass rather homogeneously among the states or vertices of G_2. However, the initial iterands $M^k; k \quad 2; \ldots$, exhibit the same behaviour as did the matrices $N \quad I^k$ in Figure 5, 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_2 over dense regions and sparse boundaries.

Boosting the multiplier effect

The obvious interpretation of the new weight function is in terms of flow rather than in terms of path sets, but the observed behaviour of matrix multiplication is similar. The new interpretation of the

$$\begin{pmatrix}
0.2000 & 0.2500 & & & & 0.3333 & 0.2500 & & & 0.2500 & & \\
0.2000 & 0.2500 & 0.2500 & & 0.2000 & & & & & & & \\
 & 0.2500 & 0.2500 & 0.2000 & 0.2000 & & & & & & & \\
 & & 0.2500 & 0.2000 & & & & 0.2000 & 0.2000 & & 0.2000 & \\
 & 0.2500 & 0.2500 & & 0.2000 & & 0.2500 & 0.2000 & & & & \\
0.2000 & & & & & 0.3333 & & & & 0.2500 & & \\
0.2000 & & & & 0.2000 & & 0.2500 & & & 0.2500 & & \\
 & & & 0.2000 & 0.2000 & & & 0.2000 & 0.2000 & & 0.2000 & \\
 & & & 0.2000 & & & & 0.2000 & 0.2000 & & 0.2000 & 0.3333 \\
0.2000 & & & & & 0.3333 & 0.2500 & & & 0.2500 & & \\
 & & & 0.2000 & & & & 0.2000 & 0.2000 & & 0.2000 & 0.3333 \\
 & & & & & & & & 0.2000 & & 0.2000 & 0.3333
\end{pmatrix}$$

$M = T_{G_2+I}$

$$\begin{pmatrix}
0.2567 & 0.1125 & 0.0625 & & 0.1000 & 0.2611 & 0.1750 & & & 0.2583 & & \\
0.0900 & 0.2250 & 0.1750 & 0.0500 & 0.1400 & 0.0667 & 0.1000 & 0.0400 & & 0.0500 & & \\
0.0500 & 0.1750 & 0.2250 & 0.0900 & 0.1400 & & 0.0500 & 0.0800 & 0.0400 & & 0.0400 & \\
 & 0.0625 & 0.1125 & 0.2100 & 0.0900 & & & 0.1600 & 0.1600 & & 0.1600 & 0.1333 \\
0.1000 & 0.1750 & 0.1750 & 0.0900 & 0.2300 & & 0.1125 & 0.0800 & 0.0400 & 0.0625 & 0.0400 & \\
0.1567 & 0.0500 & & & & 0.2611 & 0.1125 & & & 0.1958 & & \\
0.1400 & 0.1000 & 0.0500 & & 0.0900 & 0.1500 & 0.2250 & 0.0400 & & 0.1750 & & \\
 & 0.0500 & 0.1000 & 0.1600 & 0.0800 & & 0.0500 & 0.2000 & 0.1600 & & 0.1600 & 0.1333 \\
 & & 0.0500 & 0.1600 & 0.0400 & & & 0.1600 & 0.2267 & & 0.2267 & 0.2444 \\
0.2067 & 0.0500 & & & 0.0500 & 0.2611 & 0.1750 & & & 0.2583 & & \\
 & & 0.0500 & 0.1600 & 0.0400 & & & 0.1600 & 0.2267 & & 0.2267 & 0.2444 \\
 & & & 0.0800 & & & & 0.0800 & 0.1467 & & 0.1467 & 0.2444
\end{pmatrix}$$

M^2

$$\begin{pmatrix}
0.2127 & 0.1329 & 0.0687 & 0.0125 & 0.0900 & 0.2587 & 0.1975 & 0.0200 & & 0.2378 & & \\
0.1063 & 0.1575 & 0.1475 & 0.0530 & 0.1360 & 0.0689 & 0.0950 & 0.0460 & 0.0180 & 0.0767 & 0.0180 & \\
0.0550 & 0.1475 & 0.1575 & 0.0950 & 0.1340 & 0.0167 & 0.0600 & 0.0780 & 0.0500 & 0.0250 & 0.0500 & 0.0267 \\
0.0125 & 0.0663 & 0.1188 & 0.1605 & 0.0850 & & 0.0225 & 0.1560 & 0.1647 & & 0.1647 & 0.1511 \\
0.0900 & 0.1700 & 0.1675 & 0.0850 & 0.1545 & 0.0542 & 0.1263 & 0.0960 & 0.0500 & 0.0688 & 0.0500 & 0.0267 \\
0.1552 & 0.0517 & 0.0125 & & 0.0325 & 0.2045 & 0.1163 & & & 0.1815 & & \\
0.1580 & 0.0950 & 0.0600 & 0.0180 & 0.1010 & 0.1550 & 0.1575 & 0.0260 & 0.0080 & 0.1725 & 0.0080 & \\
0.0200 & 0.0575 & 0.0975 & 0.1560 & 0.0960 & & 0.0325 & 0.1520 & 0.1627 & 0.0125 & 0.1627 & 0.1511 \\
 & 0.0225 & 0.0625 & 0.1647 & 0.0500 & & 0.0100 & 0.1627 & 0.2036 & & 0.2036 & 0.2326 \\
0.1902 & 0.0767 & 0.0250 & & 0.0550 & 0.2420 & 0.1725 & 0.0100 & & 0.2253 & & \\
 & 0.0225 & 0.0625 & 0.1647 & 0.0500 & & 0.0100 & 0.1627 & 0.2036 & & 0.2036 & 0.2326 \\
 & & 0.0200 & 0.0907 & 0.0160 & & & 0.0907 & 0.1396 & & 0.1396 & 0.1793
\end{pmatrix}$$

M^3

$$\begin{pmatrix}
0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 \\
0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 \\
0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 \\
0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 \\
0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 \\
0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 \\
0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 \\
0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 \\
0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 \\
0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 & 0.0769 \\
0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 & 0.0962 \\
0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577 & 0.0577
\end{pmatrix}$$

M^{∞}

Figure 7: Powers of $M = T_{G_2+I}$, the Markov matrix associated with the graph G_2 in Figure 4, loops added to G_2

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 logical candidate for achieving this effect is by raising 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 positions with nearly identical values will still be close to each other after rescaling. This is explained by observing that what effectively happens is the exponentiation of the *ratios* of all the pairs $(T_{p_1 q}, T_{p_2 q})$. Below four vectors and their image after rescaling with power coefficient 2 are listed. The notation $\Gamma_r v$ is introduced in Definition 6.

$$\text{Vector } v: \quad \begin{pmatrix} 0 \\ 3 \\ 0 \\ 1 \\ 2 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1/2 \\ 0 \\ 1/6 \\ 1/3 \end{pmatrix} \quad \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0.1513 \\ 0.1587 \\ 0.2177 \\ 0.2251 \\ 0.2472 \end{pmatrix} \quad \begin{pmatrix} 0.0861 \\ 0.0000 \\ 0.1126 \\ 0.8013 \\ 0.0000 \end{pmatrix}$$

$$\text{Image } \Gamma_2 v: \quad \begin{pmatrix} 0 \\ 9/14 \\ 0 \\ 1/14 \\ 4/14 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 9/14 \\ 0 \\ 1/14 \\ 4/14 \end{pmatrix} \quad \begin{pmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0.1104 \\ 0.1215 \\ 0.2287 \\ 0.2445 \\ 0.2949 \end{pmatrix} \quad \begin{pmatrix} 0.0112 \\ 0.0000 \\ 0.0191 \\ 0.9697 \\ 0.0000 \end{pmatrix}$$

Definition 6 Given a matrix $M \in I\!R^{k \times l}$, $M \geq 0$, and a real number $r > 0$, the matrix resulting from rescaling each of the columns of M with power coefficient r is called $\Gamma_r M$, and Γ_r is called the **inflation** operator with power coefficient r. Formally, the action of $\Gamma_r : I\!R^{k \times l} \to I\!R^{k \times l}$ is defined by

$$(\Gamma_r M)_{pq} = (M_{pq})^r \Big/ \sum_{i=0}^{k-1} (M_{iq})^r$$

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

Definition 7 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 $\Gamma_r, r \neq 1$ (See Section 6). When applied to vectors, the Γ_r operator has a nice mathematical property in terms of *majorization*. This is discussed in Section 6. Figure 8 gives the result of applying Γ_r to the Markov matrix M^2 given in Figure 7. The vital step now is to iterate the process of alternatingly expanding information flow via normal matrix multiplication and contracting information flow via application of Γ_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 3^{rd} iterand, the 6^{th} iterand, and the stable limit of this process (provisorily denoted by M^1_{mcl}) are given in Figure 8 as well.

The matrix M^1_{mcl} 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 8 and 10 are as much attracted to each other as they are to themselves. The rest of the vertex set of G_2 can be completely partitioned according to the nodes to which they are attracted. Sweeping attractors and the elements they attract together, the partitioning φ3;7;8;10;11γ φ0;5;6;9γ φ1;2;4γ results, also found earlier with k-path clustering.

In the next section the *MCL* process is introduced, 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 4, corresponding with a line-graph on 7 nodes, loops added to each node. An *MCL* run with $e_i \; ^c \; 2; r_i \; ^c \; 2$ results in the stable limit T^1_{mcl}. The nodes 1 and 5 are attractors, the node sets φ0;2γ, and φ4;6γ, are respectively attracted to them. The vertex 3 is equally attracted to 1 and 5. The formation of two clusters, or different regions of attraction, is explained by the fact that the nodes at the far ends, i.e. 0;1;5;6 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 3 is equally attracted to both regions.

5. Formal description of the *MCL* algorithm

The general design of the *MCL* algorithm is given in Figure 10. The main skeleton is formed by the alternation of matrix multiplication and inflation in a for loop. The k^{th} iteration of this loop yields two matrix resultants labeled T_{2k} and $T_{2k \; 1}$. The resultants with odd index $2k \quad 1$ are fed to the routine *Expand* which takes the $e_k{}^{th}$ power of this matrix, yielding the $2k^{th}$ resultant. The inflation operator $_{r_k}$ is applied to resultants with even index $2k$, yielding the next odd-labeled matrix $T_{2k \; 1}$. The expansion row e_i and the inflation row r_i influence the granularity of the resulting partition. The matrices in Figure 8 correspond with an *MCL* session in which $e_i \; ^c \; 2$ and $r_i \; ^c \; 2$. The stopcriterion tests whether the current iterand is sufficiently close to an idempotent matrix. If this is the case, the process stops and the last resultant is fed to an interpretation routine. Definition 12 in this section, using Theorem 1, provides a mapping from the set of nonnegative void-free 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 section. It is useful to 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 8 Denote the operator which raises a square matrix to the k^{th} power, $k \; 2 \; N; k \quad 1$, by Exp_k.

Definition 9 A nonnegative column-homogeneous matrix M which is idempotent under matrix multiplication is called **doubly idempotent**.

Definition 10 A general *MCL* **process** is determined by two rows e_i, r_i, where $e_i \; 2 \; I\!N; e_i > 1$, and $r_i \; 2 \; I\!R; r_i > 0$, and is written

$$;e_i \; ; r_i \tag{5.1}$$

$$\left(\begin{array}{cccccccccccc}
0.3801 & 0.0867 & 0.0268 & & 0.0767 & 0.2945 & 0.2012 & & & 0.3195 & & \\
0.0467 & 0.3469 & 0.2099 & 0.0171 & 0.1503 & 0.0192 & 0.0657 & 0.0115 & & 0.0120 & & \\
0.0144 & 0.2099 & 0.3469 & 0.0555 & 0.1503 & & 0.0164 & 0.0460 & 0.0090 & & 0.0090 & \\
 & 0.0268 & 0.0867 & 0.3021 & 0.0621 & & & 0.1839 & 0.1433 & & 0.1433 & 0.0828 \\
0.0577 & 0.2099 & 0.2099 & 0.0555 & 0.4057 & & 0.0832 & 0.0460 & 0.0090 & 0.0187 & 0.0090 & \\
0.1416 & 0.0171 & & & & 0.2945 & 0.0832 & & & 0.1836 & & \\
0.1131 & 0.0685 & 0.0171 & & 0.0621 & 0.0972 & 0.3326 & 0.0115 & & 0.1466 & & \\
 & 0.0171 & 0.0685 & 0.1753 & 0.0491 & & 0.0164 & 0.2874 & 0.1433 & & 0.1433 & 0.0828 \\
 & & 0.0171 & 0.1753 & 0.0123 & & & 0.1839 & 0.2876 & & 0.2876 & 0.2782 \\
0.2464 & 0.0171 & & & 0.0192 & 0.2945 & 0.2012 & & & 0.3195 & & \\
 & & 0.0171 & 0.1753 & 0.0123 & & & 0.1839 & 0.2876 & & 0.2876 & 0.2782 \\
 & & & 0.0438 & & & & 0.0460 & 0.1204 & & 0.1204 & 0.2782
\end{array}\right)$$

$\Gamma_2 M^2$; M defined in Figure 7

$$\left(\begin{array}{cccccccccccc}
0.4478 & 0.0801 & 0.0226 & 0.0003 & 0.0681 & 0.4257 & 0.3593 & 0.0004 & 0.0000 & 0.4319 & 0.0000 & \\
0.0176 & 0.2849 & 0.2280 & 0.0070 & 0.1759 & 0.0056 & 0.0330 & 0.0049 & 0.0003 & 0.0068 & 0.0003 & 0.0000 \\
0.0048 & 0.2226 & 0.2895 & 0.0224 & 0.1726 & 0.0004 & 0.0101 & 0.0172 & 0.0030 & 0.0007 & 0.0030 & 0.0009 \\
0.0002 & 0.0180 & 0.0590 & 0.2217 & 0.0400 & 0.0000 & 0.0008 & 0.1870 & 0.1386 & 0.0000 & 0.1386 & 0.0990 \\
0.0265 & 0.3121 & 0.3136 & 0.0276 & 0.4389 & 0.0052 & 0.0539 & 0.0215 & 0.0033 & 0.0098 & 0.0033 & 0.0009 \\
0.1161 & 0.0069 & 0.0005 & 0.0000 & 0.0036 & 0.1574 & 0.0846 & 0.0000 & & 0.1308 & & \\
0.0963 & 0.0403 & 0.0127 & 0.0004 & 0.0371 & 0.0831 & 0.1968 & 0.0008 & 0.0000 & 0.1035 & 0.0000 & 0.0000 \\
0.0002 & 0.0115 & 0.0417 & 0.1723 & 0.0287 & 0.0000 & 0.0016 & 0.1982 & 0.1326 & 0.0001 & 0.1326 & 0.0964 \\
0.0000 & 0.0013 & 0.0147 & 0.2558 & 0.0088 & & 0.0001 & 0.2655 & 0.3264 & 0.0000 & 0.3264 & 0.3456 \\
0.2904 & 0.0209 & 0.0022 & 0.0000 & 0.0170 & 0.3225 & 0.2596 & 0.0001 & 0.0000 & 0.3164 & 0.0000 & \\
0.0000 & 0.0013 & 0.0147 & 0.2558 & 0.0088 & & 0.0001 & 0.2655 & 0.3264 & 0.0000 & 0.3264 & 0.3456 \\
 & 0.0000 & 0.0008 & 0.0367 & 0.0005 & & 0.0000 & 0.0388 & 0.0694 & & 0.0694 & 0.1116
\end{array}\right)$$

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

$$\left(\begin{array}{cccccccccccc}
0.9973 & 0.0002 & 0.0001 & 0.0000 & 0.0002 & 0.9973 & 0.9973 & 0.0000 & & 0.9973 & & \\
0.0000 & 0.0002 & 0.0002 & 0.0000 & 0.0002 & 0.0000 & 0.0000 & 0.0000 & & 0.0000 & & \\
0.0000 & 0.0001 & 0.0001 & 0.0000 & 0.0001 & 0.0000 & 0.0000 & 0.0000 & & 0.0000 & & \\
 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & & & 0.0000 & 0.0000 & & 0.0000 & 0.0000 \\
0.0000 & 0.9995 & 0.9996 & 0.0000 & 0.9995 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & & 0.0000 & 0.0000 & 0.0000 & & & 0.0000 & & \\
0.0000 & 0.0000 & 0.0000 & & 0.0000 & 0.0000 & 0.0000 & & & 0.0000 & & \\
 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & & & 0.0000 & 0.0000 & & 0.0000 & 0.0000 \\
 & 0.0000 & 0.0000 & 0.5000 & 0.0000 & & & 0.5000 & 0.5000 & & 0.5000 & 0.5000 \\
0.0027 & 0.0000 & 0.0000 & & 0.0000 & 0.0027 & 0.0027 & & & 0.0027 & & \\
 & 0.0000 & 0.0000 & 0.5000 & 0.0000 & & & 0.5000 & 0.5000 & & 0.5000 & 0.5000 \\
 & & 0.0000 & 0.0000 & & & & 0.0000 & 0.0000 & & 0.0000 & 0.0000
\end{array}\right)$$

$\Gamma_2 \cdot \mathit{Squaring}$ iterated six times on M

$$\left(\begin{array}{cccccccccccc}
1.0000 & & & & & 1.0000 & 1.0000 & & & 1.0000 & & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & & \\
 & 1.0000 & 1.0000 & & 1.0000 & & & & & & & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & & \\
 & & & 0.5000 & & & & 0.5000 & 0.5000 & & 0.5000 & 0.5000 \\
 & & & & & & & & & & & \\
 & & & 0.5000 & & & & 0.5000 & 0.5000 & & 0.5000 & 0.5000 \\
 & & & & & & & & & & &
\end{array}\right)$$

M_{mcl}^{∞}

Figure 8: Iteration of $\Gamma_2 \cdot \mathit{Squaring}$ with initial iterand M defined in Figure 7.

These matrices were computed by an implementation of the *MCL* algorithm. 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 (which is true for the last two matrices). Entries equal to ‘0.0000’ are greater than zero but smaller than the precision with which these matrices are printed, i.e. 10^{-4}.

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

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)

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

Stable limit T_{mcl}^{∞}

Figure 9: *MCL* run on a line-graph on 7 nodes

```
                                                # G is a weighted directed void-free graph.
                                                # e_i ∈ IN, e_i > 1, i = 1,....
MCL(G, ⟨e_i⟩, ⟨r_i⟩) {                          # r_i ∈ IR, r_i > 0, i = 1,....

    T_1 := T_G;                                 # Create associated Markov graph.
                                                # according to Definition 5
    for k = 1,...,∞ {
        T_{2k} = expand(T_{2k-1}, e_k);
        T_{2k+1} = inflate(T_{2k}, r_k);

        status = stopcriterion(Information(MCL, k));

        if (status == IDEMPOTENT)       break;  # See Theorem 1 and Definition 12.
        elif (status == PERIODIC)       exit;   # Exceptional cases, see Sections 7, 9.
        elif (status == WAITING)        ;
    }
    clustering = interpret(T_{2k+1});
}

proc expand(T, n) {                             # Matrix T is square column stochastic.
                                                # n ∈ IN, n > 1.
    R := Exp_n(T);
    return R;
}

proc inflate(T, r) {                            # Matrix T is square column stochastic.
                                                # r ∈ IR, r > 0.
    R := Γ_r(T);
    return R;
}

proc stopcriterion(information) {
    # Recognizes convergence towards equilibrium states.
}

proc information(MCL, k) {
    # Has access to a subset of all information generated up until the k^th step.
    # Returns a subset of this subset.
}

proc interpret(T) {
    # T is a nonnegative void-free idempotent matrix.
    # Return the associated clustering according to Definition 12.
}
```

Figure 10: A general setup for the *MCL* algorithm

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

$$^{d\ d}; e_{i}; r_{i} \tag{5.2}$$

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

$$M; e_{i}; r_{i} \tag{5.3}$$

Associated with an *MCL* process $M; e_{i}; r_{i}$ is an infinite row of matrices T_{i} where $T_1 \quad M$, T_{2i} $\mathrm{Exp}_{e_i}\ T_{2i\ 1}$, and $T_{2i\ 1} \quad {}_{r_i}\ T_{2i}$, $i \quad 1; \ldots; 1$.

In practice, the algorithm iterands converge nearly always to a doubly idempotent matrix. A sufficient property for associating a (possibly overlapping) clustering with a nonnegative void-free matrix is that the matrix is idempotent under matrix multiplication. The following theorem characterizes the structural properties of nonnegative void-free idempotent matrices. Using this theorem, Definition 12 establishes a mapping from the class of nonnegative void-free idempotent matrices to the set of overlapping clusterings. Nonnegative doubly idempotent matrices do not have stronger structural properties than matrices which are idempotent under matrix multiplication only. A further characterization of doubly idempotent column stochastic matrices is given in Theorem 2. The following theorem can easily be derived from the decomposition of nonnegative idempotent (not necessarily void-free) matrices given in [1]. However, I choose to give a proof along different lines here, which is inspired more by graph-theoretical considerations.

Theorem 1 Let M be a nonnegative void-free idempotent matrix of dimension N, let Γ_M be its associated graph. For $s; t$, nodes in Γ_M, write $s \to t$ if there is an arc in Γ_M from s to t. By definition, $s \to t \iff M_{ts} \quad 0$. Let $\ ;\ ;\ $ be nodes in Γ_M. The following implications hold.

$$\to \quad \perp \quad \to \quad \Rightarrow \quad \to \tag{5.4}$$

$$\to \quad \perp \quad \to \quad \Rightarrow \quad \to \tag{5.5}$$

$$\to \quad \Rightarrow \quad \to \tag{5.6}$$

Proof The first statement follows from the fact that $M \quad M^2 \quad M \quad M \quad > 0$. Suppose the second statement does not hold. Denote by V the set of nodes which reach , denote by V the set of nodes reachable from . Then $V \quad ;$ because $\to$, and $V \quad ;$ because M is void-free. It is furthermore true that $V \therefore V \quad ;$ and that there is no arc going from V to V, for this would imply $\to$ and $\to$ by 5.4. For $u; w \in V \ ; v \in V$, the property $u \to v \to w$ implies $v \in V$. For $u; w \in V; v \in V$, the property $u \to v \to w$ implies $v \in V$. It follows that for all 2-step paths between node pairs respectively lying in V and V only indices lying in the same node set V, respectively V, need be considered. Reorder M and partition the matrix such that its upper left block has the form

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

where the indices of the diagonal block A_{11} correspond with all the elements in V, and the indices of the diagonal block A_{22} correspond with all the elements in V. It follows from the construction of V and V that all entries of A_{21} are positive, since for all $u \in V; v \in V$, it is true that $u \to \ \to \ \to v$. Similarly, $A_{12} \quad 0$. The observation made on 2-step paths with beginning and ending in V, resp. V, implies that $A_{11} \quad {A_{11}}^2$ and $A_{22} \quad {A_{22}}^2$. Furthermore, the inequality $A_{21} \quad A_{21}A_{11} \quad A_{22}A_{21}$ holds. Multiplying both sides on the left with A_{22} and on the right with A_{11}, the inequality $A_{22}A_{21}A_{11}$

$2A_{22}A_{21}A_{11}$ results. The fact that A_{21} is positive, and the fact that A_{11} contains one positive row, i.e. the row corresponding with , imply that $A_{21}A_{11}$ is positive too. Since A_{22} is nonzero, this implies that the product $A_{22}A_{21}A_{11}$ is nonnegative, leading to a contradiction. The third statement follows by observing that there must be a path of infinite length going from to in Γ_M, that is, a path containing a circuit. If this were not the case, there would exist a k 2 $I\!N$ such that M^k 0, whereas M 0. The existence of such a circuit implies by 5.5 and 5.6 that ! .

Definition 11 Let Γ_M $V; w$ be the associated graph of a nonnegative void-free idempotent matrix of dimension N, where V $\phi 0; \ldots ; N$ 1γ. The node 2 V is called an **attractor** if M 0. If is an attractor then the set of its neighbours is called an **attractor system**.

By Theorem 1, each attractor system in Γ_M induces a weighted subgraph in Γ_M which is complete. Theorem 1 furthermore provides the means to formally associate an overlapping clustering with each nonnegative void-free idempotent matrix. Let M be an arbitrary nonnegative idempotent matrix, let Γ_M $V; w$ be its associated graph. Denote by V_x the set of attractors of Γ_M. Denote the 'arc from to ' relationship in G by ! . The first two statements in Theorem 1 imply that ! is transitive and symmetric on V_x, and ! is reflexive on V_x by definition of V_x. Accordingly, ! induces equivalence classes on V_x. Denote the set of equivalence classes by $\phi E_1; \ldots ; E_d\gamma$. The definition below requires the input of a void-free matrix, in order to be able to distribute the elements of $V \nu V_x$ over the classes E_i.

Definition 12 Let M be a nonnegative void-free idempotent matrix. Let Γ_M $V; w$ be its associated graph, let ! be the arc relation associated with Γ_M. Let V_x be the set of attractors in Γ_M, let E $\phi E_1; \ldots ; E_d\gamma$ be the set of equivalence classes of ! on V_x. Define a relation on E V by setting $E;$ 1 if 9 2 E with ! , and $E;$ 0 otherwise. The overlapping clustering $\mathrm{X}\Lambda_M$ $\phi C_1; \ldots ; C_d\gamma$ associated with M, defined on V, has d elements. The i^{th} cluster $C_i; i$ $1; \ldots ; d$ is defined by Equation 5.7.

$$C_i \quad v\ 2\ V \varphi \quad E_i; v \quad 1 \tag{5.7}$$

Note that the number of clusters is equal to the number of strongly connected components in the directed graph Γ_M. The inclusion E_i C_i implies that each cluster has at least one element which is unique for this cluster. All this is in line with the procedures followed while studying the example in the previous section. It should be noted that there is in general a very large number of nonnegative void-free idempotent matrices which yield the same overlapping clustering according to Definition 12. This is caused by the fact that the number of attractors and the distribution of the attractors over the clusters may both vary without resulting in different clusterings. E.g., printing attractors in boldface, the clustering $\phi\phi 0; 1\gamma, \phi 2; 3; 4\gamma\gamma$ results from all 21 possible combinations of the distributions $\phi\mathbf{0}; 1\gamma$, $\phi 0; \mathbf{1}\gamma$, and $\phi\mathbf{0}; \mathbf{1}\gamma$ for the first cluster, and the distributions $\phi\mathbf{0}; 1; 2\gamma$, $\phi 0; \mathbf{1}; 2\gamma$, $\phi 0; 1; \mathbf{2}\gamma$, $\phi\mathbf{0}; \mathbf{1}; 2\gamma$, $\phi\mathbf{0}; 1; \mathbf{2}\gamma$, $\phi 0; \mathbf{1}; \mathbf{2}\gamma$, and $\phi\mathbf{0}; \mathbf{1}; \mathbf{2}\gamma$ for the second cluster. Another example may show the extent to which complicated structure can be present in nonnegative idempotent matrices. The matrix

$$\begin{pmatrix}
1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/6 & 0 & 0 & 1/5\\
1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/6 & 0 & 0 & 1/5\\
1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/6 & 0 & 0 & 1/5\\
0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 1/7 & 0 & 0\\
0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 1/7 & 0 & 0\\
0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 1/7 & 0 & 0\\
0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 1/7 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 1/6 & 1/7 & 1/2 & 1/5\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1/2 & 0 & 0 & 1/6 & 1/7 & 1/2 & 1/5\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1/6 & 1/7 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{5.8}$$

is nonnegative idempotent and gives rise to the set $V_x = \{0,1,2,3,4,5,6,7,8,9\}$, to the equivalence classes $\{0,1,2\}$, $\{3,4,5,6\}$, $\{7,8\}$, $\{9\}$, and to the overlapping clustering $\{0,1,2,10,11,14\}$, $\{3,4,5,6,12\}$, $\{7,8,11,12,13,14\}$, $\{9,11,12\}$. This matrix is also doubly idempotent and column stochastic. The *MCL* algorithm converges for nearly all input graphs to a doubly idempotent column stochastic limit. For fixed dimension t, the class of doubly idempotent column stochastic matrices is finite, but extremely large. The fact that it is finite is easy to see: There is only a finite number of values that each matrix entry can assume, namely the set of rationals $\{0,1,1/2,\ldots,1/t\}$. The set of doubly idempotent column stochastic matrices of dimension t can equivalently be characterized in the following terms.

Theorem 2 Fix $t \in \mathbb{N}$, $t > 0$. Let $V = \{0,\ldots,t-1\}$ be a set of indices of cardinality t. Let V_x be a non-empty subset of V, let $\sim$ be an equivalence relation on V_x, inducing equivalence classes $\mathcal{E} = \{E_1,\ldots,E_d\}$. Let θ be a relation defined on $\mathcal{E} \times V$ such that $\theta(E,\alpha) = 1$ for all $E \in \mathcal{E}$, $\alpha \in E$, and $\theta(E,\alpha) = 0$ for all $E \in \mathcal{E}$, $\alpha \in V_x \backslash E$, and such that for all $\alpha \in V \backslash V_x$ there exists at least one $E \in \mathcal{E}$ with $\theta(E,\alpha) = 1$.

For fixed $t \in \mathbb{N}$, the set of 4-tuples $(V, V_x, \sim, \theta)$ is isomorphic to the set of doubly idempotent column stochastic matrices of dimension t.

Proof By Theorem 1 one can construct V_x, $\sim$, and θ, given a nonnegative void-free doubly idempotent matrix M, analogous to Definition 12. The fact that M is void-free gives the desired property that $\theta(E,\alpha)$ is not identical to zero for fixed $\alpha \in V \backslash V_x$ and varying E. Vice versa, given a tuple $(V, V_x, \sim, \theta)$, a doubly idempotent column stochastic matrix M can be constructed as follows. Define $N_\alpha = \sum_{E \in \mathcal{E}} \theta(E,\alpha)|E|$, that is '$N_\alpha$ equals the sum of the size of the attractor spaces to which α is attracted'. Define M by setting

$$M_{\beta\alpha} = \begin{cases} 1/N_\alpha & \text{if } \exists E \in \mathcal{E} \text{ with } \beta \in E \wedge \theta(E,\alpha) = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{5.9}$$

A straightforward calculation verifies that M is doubly idempotent and column stochastic.

The results in this section, especially Definition 12, which uses Theorem 1, establish a clear relationship between nonnegative void-free idempotent matrices and overlapping clusterings. In practice as observed so far, the equivalence classes $E_1,\ldots,E_d$ (see Definition 12) tend to be singleton sets, and overlap in the setting of undirected graphs has been observed only for graphs having certain symmetries. This is discussed in Section 9.

6. Mathematical properties of the Γ operator

The Γ operator preserves the majorization relationship between a probability vector and its image. This is stated in Lemma 3. Concerning just Γ, this is a nice property. However, it does not give enough foothold by itself for describing the intricate interaction of the Γ operator with the Exp operator. This subject will be discussed in greater detail in a forthcoming paper. The Γ operator furthermore distributes over the Kronecker product of matrices, which is stated in Theorem 3. Combined with the distributivity of normal matrix exponentiation over the Kronecker product, this yields the result that for each *MCL* process the Kronecker product of the respective iterands corresponding with two input matrices A and B, is equal to the iterands corresponding with the input matrix which is the Kronecker product of A and B. This property is used in Section 7 to show the existence of certain periodic limits of the *MCL* process.

Following [10], if z denotes a real vector of length n, then $z_{[0]} \geq z_{[1]} \geq \cdots \geq z_{[n-1]}$ denote the components of z in decreasing order. The vector $z_\# \equiv (z_{[0]}, z_{[1]}, \ldots, z_{[n-1]})$ is called the **decreasing rearrangement** of z.

Definition 13 Let x, y be real nonnegative vectors of length n. The vector x is said to **majorize** the tuple y if 6.1 and 6.2 hold. This is denoted by $x \succ y$.

$$x_{[0]} + x_{[1]} + \cdots + x_{[k]} \geq y_{[0]} + y_{[1]} + \cdots + y_{[k]} \qquad k \in 0, \ldots, n-2 \tag{6.1}$$

$$x_{[0]} + x_{[1]} + \cdots + x_{[n-1]} = y_{[0]} + y_{[1]} + \cdots + y_{[n-1]} \tag{6.2}$$

The relationship $\succ$ entails a rather precise mathematical notion of one vector being more "spread out" than another vector. It turns out that the inflation operator Γ_r widens probability vectors for values $r > 1$, and contracts probability vectors for values $r < 1$, which is stated in Lemma 3. This lemma follows from the fact that the vectors ω and $\Gamma_r \omega$ satisfy the conditions of Lemma 4. The latter lemma, which can be found in [10], is proven.

Lemma 3 Let ω be a probability vector, let r be a real number, $r > 0$. The two inequalities 6.3, 6.4, are implied by the fact that ω and $\Gamma_r \omega$ satisfy the conditions of Lemma 4. The two equalities 6.5, 6.6 are obvious.

$$\Gamma_r \omega \succ \omega \qquad r > 1 \tag{6.3}$$

$$\Gamma_r \omega \prec \omega \qquad r < 1 \tag{6.4}$$

$$\Gamma_r \omega = \omega \qquad r = 1 \tag{6.5}$$

$$\Gamma_r \omega = \omega \qquad \omega \text{ is homogeneous} \tag{6.6}$$

Lemma 4 ([10], page 179) If $a_i > 0, i \in 0, \ldots, n-1$, $b_0 \geq b_1 \geq \cdots \geq b_{n-1} > 0$, and $b_0/a_0 \geq b_1/a_1 \geq \cdots \geq b_{n-1}/a_{n-1}$, then $a/\sum a \prec b/\sum b$.

Proof The hypothesis implies $a_0 \geq \cdots \geq a_{n-1} > 0$. We need to prove for $k \in 0, \ldots, n-2$,

$$\sum_{j=0}^{k} a_j \Big/ \sum_{l=0}^{n-1} a_l \leq \sum_{j=0}^{k} b_j \Big/ \sum_{l=0}^{n-1} b_l$$

This follows from

$$\sum_{j=0}^{k} a_j \sum_{l=0}^{n-1} b_l \quad \sum_{j=0}^{k} b_j \sum_{l=0}^{n-1} a_l \qquad \sum_{j=0}^{k} a_j \sum_{l=k+1}^{n-1} b_l \quad \sum_{j=0}^{k} b_j \sum_{l=k+1}^{n-1} a_l$$

$$\sum_{j=0}^{k} \sum_{l=k+1}^{n-1} a_j a_l \; \frac{b_l}{a_l} \quad \frac{b_j}{a_j} \qquad 0$$

Theorem 3 Let $A;B$ be nonnegative matrices of respective dimensions $s_1 \quad t_1$ and $s_2 \quad t_2$, let r 2 $I\!R$ be greater than zero. Denote the Kronecker product by . Equation 6.7 holds.

$$_r\; A \quad B \qquad {}_r A \quad {}_r B \tag{6.7}$$

Proof Use the following notation for the Kronecker product of matrices. Let $A \quad B_{\,i;j;k;l}$ denote the entry $A \quad B_{\,is_2 \;\; k;jt_2 \;\; l}$, which is by definition equal to $A_{ij}B_{kl}$. Here $i \quad 0;\ldots;s_1 \quad 1$, $k \quad 0;\ldots;s_2 \quad 1$, $j \quad 0;\ldots;t_1 \quad 1$, and $l \quad 0;\ldots;t_2 \quad 1$. I prove identity 6.7 by proving that the ratios between two entries in the same column is the same on both sides of Equation 6.7. Let $i;j;k;l$ be as above and let $i^0;k^0$ be additional indices within the same bounds as respectively i and k. The indices $j;l$, identify the $jt_1 \quad l^{\,th}$ column on both sides of 6.7. The two index pairs $i;k$ and $i^0;k^0$ identify two row entries in this column.

$$\frac{{}_r\; A \quad B_{\;i;j;k;l}}{{}_r\; A \quad B_{\;i^0;j;k^0;l}} \qquad \frac{A \quad B_{\;i;j;k;l}}{A \quad B_{\;i^0;j;k^0;l}}^{!\;r} \qquad \frac{A_{ij}B_{kl}}{A_{i^0j}B_{k^0l}}^{!\;r}$$

$$\frac{A_{ij}}{A_{i^0j}}^{!\;r} \quad \frac{B_{kl}}{B_{k^0l}}^{\;r} \qquad \frac{{}_rA_{\;ij}}{{}_rA_{\;i^0j}} \; \frac{{}_rB_{\;kl}}{{}_rB_{\;k^0l}}$$

$$\frac{{}_rA \quad {}_rB_{\;i;j;k;l}}{{}_rA \quad {}_rB_{\;i^0;j;k^0;l}}$$

Theorem 4 Let $A;B$, be square column stochastic matrices with no further restrictions imposed on their respective dimensions. Let $K \quad A \quad B$ be their Kronecker product. Suppose all three are input to the same MCL process $\;;e_i\;;r_i\;$. Denote the respective iterand pairs by $A_{2i};A_{2i\;\;1}$, $B_{2i};B_{2i\;\;1}$, $K_{2i};K_{2i\;\;1}$, $i \quad 1;\ldots;1$. Identity 6.8 holds.

$$K_j \quad A_j \quad B_j \qquad\qquad j \quad 2;\ldots;1 \tag{6.8}$$

Proof The theorem follows from the observation that both matrix exponentiation and distribute over the Kronecker product.

7. Equilibrium states of the *MCL* process

In order to characterize the equilibrium states of the *MCL* process, I make two extra assumptions on the input rows $r_{(i)}$ and $e_{(i)}$. These are

i) $r_i \geq c$ eventually, $c \in I\!R; c > 1$.

ii) $e_i \geq 2$ eventually.

The main purpose of these requirements is to study for specific parameters whether matrices exist corresponding with periodic limits. This question will be answered affirmatively below. The first requirement implies that the process differs genuinely from the usual Markov process. It is necessary to require $r_i > 1$ eventually in order to ensure that the limit of the corresponding *MCL* process can in principle have structural properties which are different from the original input graph in terms of the number and distribution of the (strongly) connected components. Consider a regular ergodic input graph (all example graphs in the figures except G_3 in Figure 6 are regular and ergodic). The structural properties of all intermediate iterands are identical, and positive entries can thus only *tend* to zero eventually, they can not become equal to zero eventually. It is true only for the limit of the process that it may differ structurally from the input graph.

The implementation with which experiments were carried out so far uses rows $r_{(i)}$ and $e_{(i)}$ which have even much simpler structure. The rows are allowed to have a prefix of the same length N, in which each row is constant. Both rows may assume another constant on the postfix of infinite length starting at position $N+1$. The examples in Section 9 use such input rows.

An equilibrium state corresponds with an *MCL* process $(M; e_{(i)}; r_{(i)})$ with $e_{(i)} \stackrel{c}{=} 2$, and $r_{(i)} \stackrel{c}{=} c > 1$, for which the associated row of matrix pairs $(T_{2i}; T_{2i+1})$ is periodic. A periodic row of objects is a row consisting of a finite list of objects repeated infinitely many times. The period of a periodic row is the minimum cardinality of such a finite list, the period of a constant row is 1. An equilibrium state can be associated with the input matrix M, with the infinite row $(T_{2i}; T_{2i+1})$ generated by M, and with a finite list of matrices constituting a cycle of period p in $(T_{2i}; T_{2i+1})$. A priori, I distinguish three different types L_i ($i = 1; \ldots; 3$) of equilibrium states for the *MCL* process with column stochastic input matrix M, input row $r_{(i)} \stackrel{c}{=} c > 1$, and input row $e_{(i)} \stackrel{c}{=} 2$. A matrix M is said to be of type L_i if its associated output row is of type L_i. In order of decreasing strength of properties, the types L_i are:

- L_1 M is doubly idempotent, implying that all matrices T_{2i} and T_{2i+1} are equal.
- L_2 The row of pairs $(T_{2i}; T_{2i+1})$ has period 1. Even iterands are $\mathrm{Exp}_2 \circ \Gamma_c$-$id$, odd iterands are $\Gamma_c \circ \mathrm{Exp}_2$-$id$, and $T_{2i} \neq T_{2i+1}$.
- L_3 The row of pairs $(T_{2i}; T_{2i+1})$ has period $p > 1$, that is, $T_{2i} = T_{2(i+p)}$ and $T_{2i+1} = T_{2(i+p)+1}$. The even iterands T_{2i} are idempotents under p iterations of the operator $(\mathrm{Exp}_2 \circ \Gamma_c)$, the odd iterands T_{2i+1} are idempotents under p iterations of the operator $(\Gamma_c \circ \mathrm{Exp}_2)$.
- L_{3a} As above, where the matrix T_1 is the Kronecker product of a column homogeneous column stochastic cyclic matrix P with odd period and a matrix A which is of type L_2 or L_1. An example of such P is a permutation matrix containing cycles of odd period only.

Each of the classes L_1, L_2, and L_3 is non-empty. The most important class of equilibrium states is the large class L_1 of doubly idempotent matrices. These matrices are invariant under arbitrary *MCL* processes. For dimensions $2; 3; 4; 5$ a few matrices of L_2 type for $c = 2$ can be found quickly

$$\begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix} \quad \begin{pmatrix} 1-2p & p & p \\ p & 1-2p & p \\ p & p & 1-2p \end{pmatrix} \quad \begin{pmatrix} 1-2p-q & p & q & p \\ p & 1-2p-q & p & q \\ q & p & 1-2p-q & p \\ p & q & p & 1-2p-q \end{pmatrix}$$

$$\begin{pmatrix} 1-2p-2q & p & q & q & p \\ p & 1-2p-2q & p & q & q \\ q & p & 1-2p-2q & p & q \\ q & q & p & 1-2p-2q & p \\ p & q & q & p & 1-2p-2q \end{pmatrix}$$

General form for $\Gamma_2 \circ \mathrm{Exp}_2$ $-id$ matrices in dimensions 2, 3, 4, and 5. Explicit solutions for the resulting equations are given below.

$$R_{2a} \quad \begin{pmatrix} .77184 & .22816 \\ .22816 & .77184 \end{pmatrix} \qquad p = \tfrac{2}{3} - \sqrt[3]{v} - \frac{1}{18\sqrt[3]{v}}; \quad v = \tfrac{17}{216} - \tfrac{1}{72}\sqrt{33}$$

$$R_{3a} \quad \begin{pmatrix} 2/3 & 1/6 & 1/6 \\ 1/6 & 2/3 & 1/6 \\ 1/6 & 1/6 & 2/3 \end{pmatrix} \qquad p = \tfrac{1}{6}$$

$$R_{4a} \quad \begin{pmatrix} 0.60205 & 0.13265 & 0.13265 & 0.13265 \\ 0.13265 & 0.60205 & 0.13265 & 0.13265 \\ 0.13265 & 0.13265 & 0.60205 & 0.13265 \\ 0.13265 & 0.13265 & 0.13265 & 0.60205 \end{pmatrix} \qquad q = p; \ p = \tfrac{5}{18} - \sqrt[3]{v} - \frac{1}{162\sqrt[3]{v}}; \quad v = \tfrac{67}{23328} - \tfrac{1}{2592}\sqrt{57}$$

$$R_{4b} \quad \begin{pmatrix} 0.38592 & 0.11408 & 0.38592 & 0.11408 \\ 0.11408 & 0.38592 & 0.11408 & 0.38592 \\ 0.38592 & 0.11408 & 0.38592 & 0.11408 \\ 0.11408 & 0.38592 & 0.11408 & 0.38592 \end{pmatrix} \qquad q = \tfrac{1}{2} - p; \ p = \tfrac{1}{3} - \sqrt[3]{v} - \frac{1}{72\sqrt[3]{v}}; \quad v = \tfrac{17}{1728} - \tfrac{1}{576}\sqrt{33}$$

$$R_{4c} \quad \begin{pmatrix} 0.59594 & 0.17610 & 0.05205 & 0.17610 \\ 0.17610 & 0.59594 & 0.17610 & 0.05205 \\ 0.05205 & 0.17610 & 0.59594 & 0.17610 \\ 0.17610 & 0.05205 & 0.17610 & 0.59594 \end{pmatrix} \qquad q = p - 4p^2; \ p = \tfrac{1}{3} - \sqrt[3]{v} - \frac{1}{8\sqrt[3]{v}}; \quad v = \tfrac{13}{864} - \tfrac{1}{288}\sqrt{57}$$

$$R_{5a} \quad \begin{pmatrix} 0.5568 & 0.1108 & 0.1108 & 0.1108 & 0.1108 \\ 0.1108 & 0.5568 & 0.1108 & 0.1108 & 0.1108 \\ 0.1108 & 0.1108 & 0.5568 & 0.1108 & 0.1108 \\ 0.1108 & 0.1108 & 0.1108 & 0.5568 & 0.1108 \\ 0.1108 & 0.1108 & 0.1108 & 0.1108 & 0.5568 \end{pmatrix} \qquad q = p; \ p = \tfrac{13}{60} - \sqrt[3]{v} - \frac{11}{3600\sqrt[3]{v}}; \quad v = \tfrac{233}{216000} - \tfrac{1}{36000}\sqrt{1545}$$

$$R_{5b} \quad \begin{pmatrix} 0.5346 & 0.2087 & 0.0239 & 0.0239 & 0.2087 \\ 0.2087 & 0.5346 & 0.2087 & 0.0239 & 0.0239 \\ 0.0239 & 0.2087 & 0.5346 & 0.2087 & 0.0239 \\ 0.0239 & 0.0239 & 0.2087 & 0.5346 & 0.2087 \\ 0.2087 & 0.0239 & 0.0239 & 0.2087 & 0.5346 \end{pmatrix}$$

Values are numerically found roots of a polynomial of degree 8 which is irreducible over the rationals.

Figure 11: $\Gamma_2 \circ \mathrm{Exp}_2$ $-id$ matrices, computed in Maple.

by algebraic computation. They are depicted in Figure 7. The general graph forms on n nodes, $n = 2,\ldots,5$, which were used to derive these examples, are invariant under the automorphism group of the ring-graph[3] of order n. Note that the matrix R_{4b} is the Kronecker product of the matrices $1/2 J_2$ and R_{2a}, where J_2 is the all-one matrix of dimension 2. It seems likely that higher dimensional versions of the forms in Figure 7 have solutions as well. The only clusterings suiting ring graphs are the two extreme clusterings. Slight perturbations of either the *MCL* process parameters or the input graphs lead the *MCL* algorithm to converge towards a limit of the L_1 type, corresponding with one of the two extreme clusterings. For example, setting $p = 101/601$ in the 3-dimensional matrix form in Figure 7 leads the algorithm to convergence to the identity matrix, setting $p = 99/601$ leads the algorithm to converge to $1/3\, J$, where J is the all-one matrix. The same behaviour results after respectively setting $c = 201/100$ and $c = 199/100$. For the latter settings, it is in line with heuristic considerations that a slight increase in inflation leads the algorithm to converge towards a matrix corresponding with the top extreme partition (i.e. $\phi V \gamma$), and that a slight decrease in inflation leads the algorithm to converge to a matrix corresponding with the bottom extreme partition (i.e. $\phi V \gamma$).

These observations suggest that the class L_2 consists of equilibrium states which are very instable by nature. The image of the column vectors under either Γ_2 or Exp_2 is very different from the original vector. For this class, expansion and inflation act as each others inverse. A slight perturbation of the *MCL* process parameters or the equilibrium state leads to one of the two getting the upper hand.

So far, all limits resulting from inputting undirected graphs were of the L_1 type. If the condition $e_i \stackrel{c}{=} 2$ is relaxed to $e_i \stackrel{c}{=} k$, where $k \in \mathbb{N}$ is a constant, examples of the L_{3a} type can be found as well by selecting bipartite graphs, setting $e_i = 3$, and refraining from adding loops. This is not surprising, since in bipartite graphs paths of odd length always go from one of the two node sets to the other. As was the case with ring-graphs, the relationship between parameter choice, expected behaviour, and observed behaviour fully agree, so this is an agreeable situation.

The class L_3 is nonempty for rows $e_i \stackrel{c}{=} 2$ as well. It is easy to construct matrices of the L_{3a} type, by taking the Kronecker product of L_1- or L_2-type matrices and permutation matrices containing odd permutations only, illustrating the use of Theorem 3. Denote by $L_x \setminus L_y$ the class of matrices satisfying the L_x constraints but not satisfying the L_y constraints. It is an open question whether matrices of the type $L_3 \setminus L_{3a}$ exist. If they exist, I expect them by all means to be as sensitive to perturbations of parameter settings and matrix values as are the matrices of the L_2 type. While the L_3 and L_2 classes are of interest for studying the *MCL* process, they do not form a weak spot of the *MCL* algorithm. If a graph constructed from some application such as a thesaurus or a database leads to an *MCL* process which at any stage approaches an L_2 or L_3 type matrix, the input graph must be considered unsuited for clustering purposes.

8. Stability of equilibrium states

In this section the stability of the equilibrium states in L_1 is considered. The setting is as follows. Let M be the associated matrix of an equilibrium state in L_1, let Δ be a perturbation matrix such that $M + \Delta$ is stochastic. For various types of perturbation Δ the limit or set of possible limits of the perturbed *MCL* process $(M + \Delta; e_i \stackrel{c}{=} 2; r_i \stackrel{c}{=} 2)$ is investigated. The states in L_1 which are stable in every respect correspond with doubly idempotent matrices which have precisely one nonzero entry (equal to 1) in each column. This is stated in Theorem 5. A doubly idempotent matrix M corresponds with an instable equilibrium state if it has columns with more than one nonzero entry. Two cases can be distinguished: the case where all columns with multiple entries correspond with nodes which are attracted to or are part of a single attractor system having more than one attractor (Lemma 6), and the case where p is not an attractor and is attracted to two different attractor systems (Lemma 7). For

[3] See Definition 15 for the precise definition of a ring graph.

both cases, it is of interest in which respects the associated clustering of a limit resulting from the perturbed *MCL* process may differ from the associated clustering of M.

In the first case, the equilibrium state is shown to be stable on a macroscopic scale which corresponds with the cluster structure derived from M (Theorem 7). A perturbation of M may thus lead the *MCL* process M $; e_i; r_i$ to converge towards a different equilibrium state. Theorem 7 guarantees that this new equilibrium state yields a cluster interpretation which is identical to or a refinement of the associated clustering of M. For a restricted class of perturbations , Theorem 8 guarantees that the new equilibrium state yields a cluster interpretation which is identical to the associated clustering of M. These are perturbations only affecting the principal submatrices M , where is any index set describing an attractor system in M. In words, Theorem 8 states that for such a perturbation an attractor system cannot split into a number of smaller attractor systems. The proof of the theorem is elegant and gives confidence that the result extends to arbitrary perturbations (a proof is not given nor known however).

In the second case, if a perturbation of column p is unevenly spread over the attractor systems towards which p is attracted, then the process $M; e_i; r_i$ will converge towards a state in which p is attracted to just one of those systems. This means that the phenomenon of cluster overlap is instable in nature (Lemma 7).

In a forthcoming paper it will be shown that intermediate iterands of a *MCL* process have structure, comparable to but more general than the structure of doubly idempotent matrices as described by Theorem 1. This structure enables one to associate several overlapping clusterings with each iterand in the process. Besides giving extra insight in the *MCL* process and in the interaction of Exp_k and $_r$, this gives the means to detect cluster overlap (and consequently graph symmetries) at earlier stages, which is partial consolation for the instability of the phenomenon of overlap.

The following theorem identifies the equilibrium states in L_1 for which the associated matrix M is attractor for all input matrices M with regard to the *MCL* process M $; e_i$ ^{c} $2; r_i$ ^{c} $2)$, for small enough.

Theorem 5 *The MCL process with standard parameters* $; e_i$ ^{c} $2; r_i$ ^{c} $2)$*, converges quadratically in the neighbourhood of each nonnegative idempotent column stochastic matrix for which every column has one entry equal to* 1 *and all the other entries equal to* 0*.*

The formulation of this theorem is rather non-technical. What I shall prove is Lemma 5.

Lemma 5 *Let* M 2 $I\!R_0$ $^{n\ n}$ *be a nonnegative idempotent column stochastic matrix for which every column has one entry equal to* 1 *and all other entries equal to* 0*. Let* x_i *be the row index such that* $M_{x_i i}$ 1*. Let* $f > 0$ *be a real number and let be a matrix in* $I\!R^{n\ n}$*, the columns of which add to zero, such that* M *is column stochastic and nonnegative, and such that* M $_{x_i i}$ 1 f*. Define the matrix by* M ^{2} M *.*

For f $1{=}4$ *the inequality* $\max_{i;j}$ φ $_{ij}$ φ $8f^2$ *holds.*

Proof The structure of nonnegative idempotent matrices as described in Theorem 1 implies the equality x_{x_i} x_i, by the implication i ! x_i ⇒ x_i ! x_i. It furthermore follows from the definition of that $\max_{i;j}$ φ $_{ij}$ φ f. Consider the entry M $^2{}_{x_i i}$. The inequalities M $^2{}_{x_i i}$ M $^2{}_{x_i x_i}$ M $^2{}_{x_i i}$ 1 f ^{2} 1 $2f$ hold. Now consider the entry P M $_{x_i i}$. It is true that P_k M $_{ki}{}^2$ 1 f ^{2}. Furthermore, $\mathrm{P}_{k\ x_i}$ M $_{ki}$ f and thus $\mathrm{P}_{k\ x_i}$ M $_{ki}{}^2$ f^2. It follows that $\sum_{k \notin x_i} (\Gamma M)_{ki} \leq f^2/(1-f)^2$, and consequently $(\Gamma M)_{ki} \geq 1 - f^2/(1-f)^2$. For $f < 1/4$ the inequality $1 - f^2/(1-f)^2 \geq 1 - 2f^2$ holds. Combining this inequality and the previous one yields the desired result.

Theorem 6 *The equilibrium states of the MCL process in L_1 for which the associated doubly idempotent matrices have one or more columns with more than one nonzero entry are instable.*

Two cases are distinguished in proving this theorem, namely the case in which a column with more than one nonzero entry corresponds with an attractor, and the case in which it corresponds with a non-attractor. Both cases are illustrated with simple examples which generalize in a straightforward manner to higher dimensional and more complex cases.

Lemma 6 *Let M, Δ_f and L be the matrices*

$$M = \begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix} \quad \Delta_f = \begin{pmatrix} f & f \\ -f & -f \end{pmatrix} \quad L = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$$

For each $f > 0$ the MCL process $(M + \Delta_f, e_i \equiv 2, r_i \equiv 2)$ converges towards L.

Proof The matrix $M + \Delta_f$ is idempotent under matrix multiplication for arbitrary f, as it is a rank-1 stochastic matrix. Direct computation shows that $(\Gamma_2(M + \Delta_f))_{11}$ equals $(1/4 + f^2 + f)/(1/2 + 2f^2) = 1/2 + 2f/(1 + 4f^2)$. Thus $\Gamma_2(M + \Delta_f)$ can be written as $M + \Delta_{2f/(1+4f^2)}$. For small f, the deviation of $\Gamma_2(M + \Delta_f)$ from M is nearly twice as large as the deviation of $M + \Delta_f$ from M. The lemma follows.

The proof of the following lemma is nearly identical.

Lemma 7 *Let M, Δ_f and L be the matrices*

$$M = \begin{pmatrix} 1 & 0 & 1/2 \\ 0 & 1 & 1/2 \\ 0 & 0 & 0 \end{pmatrix} \quad \Delta_f = \begin{pmatrix} 0 & 0 & f \\ 0 & 0 & -f \\ 0 & 0 & 0 \end{pmatrix} \quad L = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

For each $f > 0$ the MCL process $(M + \Delta_f, e_i \equiv 2, r_i \equiv 2)$ converges towards L.

The previous results do not imply that the *MCL* algorithm is built on quicksand. The instability of the phenomenon of cluster overlap cannot be helped, if only the limit of the *MCL* process is taken into account. As mentioned before, there is a cure for this by looking at the specific structure which is present in all iterands of the process. This issue will be dealt with in a forthcoming paper.

The instability of attractor systems consisting of more than one element is not a serious issue if only regarding clustering purposes. If one is interested in graph symmetries which may be exhibited by such larger attractor systems then the remarks in the previous paragraph hold here as well. Below it is shown that perturbation of doubly idempotent matrices M by a matrix Δ for which the associated clustering $\mathcal{X}$ does not have overlap, lead the iterands of the *MCL* process $(M + \Delta, e_i \equiv 2, r_i \equiv 2)$ to stay within a class of matrices the block structure of which only admits a clustering which is a refinement of $\mathcal{X}$. These statements are assembled in Theorem 7, which is preceded by two more technical lemmas. This result is extended by Theorem 8, which demonstrates that for a specific class of perturbations the notion 'a refinement of' in Theorem 7 can be strengthened to 'identical to'.

If a diagonal block structure can be mapped onto part of a column stochastic matrix M such that the mass of the columns in this part is highly concentrated in the blocks, then the entries outside the diagonal blocks tend to zero quadratically in the MCL process $(M, e_{(i)} = c \geq 2, r_{(i)} = c \geq 2)$. If it is moreover assumed that the mass of the columns in the remaining part is (for each column separately) concentrated in a set of rows corresponding to at most one diagonal block, then the entries not belonging to these rows tend to zero as well. Conceptually, the proof is very similar to that of Lemma 5. The more complicated setting requires substantial elaboration.

Let M be a column stochastic matrix of dimension n, let $f > 0$ be a real number. Assume that there is a strictly increasing row of indices $k_0, \ldots, k_l$ with $k_0 = 0$ and $k_l \leq n$ such that the mass of the columns in each principal submatrix $M[k_i \ldots k_{i+1} - 1]$, $i = 0, \ldots, l-1$ is greater than or equal to $1 - f$. (Remember that M is indexed from 0 to $n-1$). It is convenient to denote the set of indices $\{k_x, \ldots, k_{x+1} - 1\}$ by Γ_x, indicating the x^{th} diagonal block.

Lemmas 8 and 9 hold, and are preparatory to Theorem 7. The corresponding statements for matrices which are permutation-similar to a matrix with the required block structure are omitted and easily seen to be true, since both matrix multiplication and inflation distribute over simultaneous permutation of rows and columns.

Lemma 8 *Let f, M and $k_0, \ldots, k_l$ be as above. Let T_{2i} and T_{2i+1} be the iterands of the MCL process $(M, e_{(i)} = c \geq 2, r_{(i)} = c \geq 2)$, where $T_0 = M$. Let Γ_x be the range of indices $\{k_x, \ldots, k_{x+1} - 1\}$ and let q be an index in Γ_x. For f small enough, the entries $(T_i)_{jq}$ tend to zero for all j with $j \notin \Gamma_x$ as i goes to infinity.*

Proof Suppose that $k_l < n$. Thus, the block diagonal structure (the blocks of which have large mass) does not fully cover M, as the last block is indexed by the range $k_{l-1} \ldots k_l - 1$. This is the more general case where nothing is assumed about the remaining columns $k_l \ldots n - 1$. The proof where $k_l = n$ simplifies and will be omitted. Let Γ_x and q be as in the lemma, so $q \in \Gamma_x$. Let p be any index, $0 \leq p < n$.

Consider the p^{th} entry of the q^{th} column of M^2. Consider first the case where $k_l \leq p < n$. The identity $(M^2)_{pq} = \sum_{i=0}^{n-1} M_{pi} M_{iq}$ holds. Split the latter sum into the parts $\sum_{i \in \Gamma_x} M_{pi} M_{iq}$ and $\sum_{i \notin \Gamma_x} M_{pi} M_{iq}$. For $i \in \Gamma_x$ the inequality $M_{pi} \leq f$ holds. Since $\sum_{i \in \Gamma_x} M_{iq} \leq 1$, the first sum is smaller then or equal to f. By similar reasoning it is found that the second sum is smaller than or equal to f^2.

Now consider the case where $p \in \Gamma_y$, $y \neq x$. Write the entry $(M^2)_{pq}$ in three parts: $\sum_{i \in \Gamma_x} M_{pi} M_{iq}$, $\sum_{i \in \Gamma_y} M_{pi} M_{iq}$, and $\sum_{i \notin \Gamma_x \cup \Gamma_y} M_{pi} M_{iq}$. For the first part, $M_{pi} \leq f$ and the entries M_{iq} sum to less than one. For the second part, the entries M_{pi} sum to less than $|\Gamma_y|$ and $M_{iq} \leq f$. For the third part, $M_{pi} \leq f$ and the entries M_{iq} sum to less than f. Combining these results yields that the full sum is smaller than or equal to $f + |\Gamma_y| f + f^2$. So after multiplication, the combined mass of all entries in column q which are not in Γ_x is bounded from above by $n(n + l)f + f^2$, which is of order f.

Estimate the entry $(\Gamma M)_{pq}$ as follows. The sum of squares $\sum_{i=0}^{n-1} {M_{iq}}^2$ is bounded from above by $1/n$. For $p \notin \Gamma_x$ the inequality ${M_{pq}}^2 \leq f^2$ holds and thus $(\Gamma M)_{pq} \leq n f^2$. The combined mass of all entries in column q which are not in Γ_x is thus bounded from above by the (crude) estimate $n^2 f$, which is of order f^2. Combination of this with the result on multiplication yields the following. If f is viewed as the error with which M deviates from the block structure imposed by the index sets Γ_x (in the index range $0 \ldots k_l - 1$), then application of $\Gamma \cdot \mathrm{Exp}_2$ to M yields a matrix for which the new error is of order f^2. This proves the lemma.

Lemma 9 *Let f, M and $k_0; \ldots; k_l$ be as in Lemma 8. Assume moreover that $k_l < n$ and that for each $q \quad k_l$ there exists a block indexed by $_x \quad \phi k_x; \ldots; k_{x\ 1} \quad 1\gamma$ such that the mass in the submatrix $M \quad _x \varphi q$ (which is part of column q) is bounded from below by $1 \quad f$. Let T_i be the iterands of the MCL process $M; e_i \ ^c\ 2; r_i \ ^c\ 2$. Then, for f small enough, all entries $T_{i\ pq}$ tend to zero for p ₿ $_x$ as i goes to infinity.*

Proof The proof is very similar to that of Lemma 9. Consider the p^{th} entry of the q^{th} column of M^2. First consider the case where $k_l \ _{\mathrm{P}} p < n$. The identity $M^2{}_{pq} \quad \mathrm{P}^{n\ 1}_{i\ 0} M_{pi} M_{iq}$ holds. Split the latter sum into the parts $\mathrm{P}_{i2\ x} M_{pi} M_{iq}$ and $\mathrm{P}_{i₿\ x} M_{pi} M_{iq}$. As in the proof of Lemma 9 it is found that the two parts are respectively bounded from above by f and f^2.

Now consider the case where $p\ 2 \quad _y; y \quad x$. Writing the entry $M^2{}_{pq}$ in three parts: $\mathrm{P}_{i2\ x} M_{pi} M_{iq}$, $\mathrm{P}_{i2\ y} M_{pi} M_{iq}$, and $\mathrm{P}_{i₿\ x[\ y} M_{pi} M_{iq}$, it is found that these parts are respectively bounded by f, $\varphi\ _y \varphi f$, and f^2. After multiplication, the combined mass of all entries in column q which are not in $_x$ is bounded from above by $n\ n \quad l\ f \quad f^2$, which is of order f.

The entry $M\ _{pq}$ is estimated as before, yielding $M\ _{pq} \quad nf^2$, and bounding the combined mass of the entries $M\ _{pq}, q$ ₿ $_x$ by $n^2 f$. Viewing f as the error with which column q deviates from the structure imposed by $_x$ gives that applying Exp_2 to M yields a matrix for which the new error is of order f^2. This proves the lemma.

Theorem 7 is a general result on perturbation of equilibrium states for which the associated matrix M may have columns with more than one nonzero entry. It states that the associated clustering of any idempotent limit resulting from the perturbed process must be a refinement of the clustering associated with M. The proof of the theorem is a direct consequence of Lemmas 8 and 9.

Theorem 7 *Let M be a doubly idempotent matrix in $\mathbb{R}_0^{n\ n}$ for which the associated clustering X is free of overlap. Let $f > 0$ and let be a matrix in $\mathbb{R}^{n\ n}$, the columns of which sum to zero and for which $\max_{i;j} \varphi_{ij}\varphi \quad f$. The iterands T_i of the MCL process $M \quad ; e_i \ ^c\ 2; r_i \ ^c\ 2$, for f small enough, have the property that $T_{i\ pq}$ tends to zero as i goes to infinity, if q 6 p in the associated graph of M. Consequently, an idempotent limit resulting from the process $M \quad ; e_i \ ^c\ 2; r_i \ ^c\ 2$ corresponds with a clustering which is identical to or a refinement of X.*

The following theorem extends this result for a restricted class of perturbations, namely those that only affect the principal submatrices of the doubly idempotent matrix M which correspond to an attractor system in the associated clustering of M. Theorem 1 implies that such a submatrix has the form $\frac{1}{k} J_k$, where J_k is the all one matrix of dimensions $k \quad k$. Theorem 8 is concerned with limits which may possibly result from the *MCL* process $\frac{1}{k} J_k \quad ; e_i \ ^c\ 2; r_i \ ^c\ 2$, where is as before. It appears that for small perturbations it is guaranteed that the iterands of the process approach arbitrarily close towards the set of rank 1 stochastic matrices, without actually pinpointing a particular limit point. This implies that an idempotent limit of the perturbed process $M \quad ; e_i \ ^c\ 2; r_i \ ^c\ 2$, where M is doubly idempotent and only affects the attractor systems of M, is guaranteed to yield an associated clustering which is the same as that of M, except for the cases where overlap occurs.

Theorem 8 *Let M be a doubly idempotent matrix in $\mathbb{R}_0^{n\ n}$ for which the associated clustering X is free of overlap. Let $f > 0$ and let be a matrix in $\mathbb{R}^{n\ n}$, the columns of which sum to zero, for which $\max_{i;j} \varphi\varphi_{ij} \quad f$, and for which $_{kl} \quad 0 \ni \quad k$ and l are attractors in the same attractor system in M. That is, only affects the diagonal blocks of M corresponding with its attractor systems.*

An idempotent limit resulting from the process $M \; ; e_i \;\; c \; 2; r_i \;\; c \; 2$, has an associated clustering which is identical to X.

This theorem is a consequence of the following lemma. Note that the diagonal blocks of M corresponding with its attractor systems are of the form $\frac{1}{k}J_k$.

Lemma 10 *Let $f > 0$ be a real number, let J be an arbitrary rank 1 column stochastic matrix in $IR_0{}^{n \;\; n}$, let $\;\; 2 \; IR^{n \;\; n}$ be a matrix the columns of which sum to zero and for which $\max_{i;j} |\;|_{ij} \;\; f$. For f small enough, the matrix $_2 \;\; J \;\;\; ^2$ can be written as J^0 $\;$, where J^0 is rank 1 column stochastic, the columns of $\;$ sum to zero and $\max_{i;j} |\;|_{ij} \;\; cf^2$, where $c > 0$ is a constant independent from J, $\;$, and f.*

Proof Consider $J \;\;\; ^2$. This product can be written as $J^2 \;\; J \;\;\; J \;\;\; ^2$. The identities $J^2 \;\; J$ and $J \;\;\; 0$ hold. Furthermore, the sum $J \;\;\; J$ is a rank 1 column stochastic matrix. Thus the product $J \;\;\; ^2$ can be written as the sum of a rank 1 column stochastic matrix and ^{2}. It is easy to show that $\max_{i;j} |\,^2|_j \;\; nf^2$, which is of order f^2.

Now consider the result of applying $_2$ to $J \;\;$, and compare this with $_2 J$. First compute the renormalization weight for the l^{th} column of $_2 J \;\;$. This equals $\sum_i J_{il} \;\;\; _{il}{}^2$. Split this sum into the parts $\sum_i J_{il}{}^2$, $2 \sum_i \;_{il} J_{il}$, and $\sum_i \;_{il}{}^2$. Then $2|\sum_i \;_{il} J_{il}| \;\; 2f$, and $\sum_i \;_{il}{}^2 \;\; nf^2$. It follows that $\sum_i J_{il} \;\;\; _{il}{}^2$ can be written as $\sum_i J_{il}{}^2 \;\;\; _d$, where $|\;_d| \;\; 2f \;\;\; nf^2$ (and the d stands for denominator).

Observe that $J_{kl} \;\;\; _{kl}{}^2 \;\;\; J_{kl}{}^2 \;\; 2J_{kl} \; _{kl} \;\;\; _{kl}{}^2$ can be written as $J_{kl}{}^2 \;\;\; _e$, where $|\;_e| \;\; 2f \;\;\; f^2$. It follows that $_2 \; J \;\;\;\; _{kl}$ can be estimated as below.

$$\frac{J_{kl} \;\;\; e}{\sum_i J_{il}{}^2 \;\;\; d} \qquad \frac{J_{kl} \;\;\; _{kl}{}^2}{\sum_i J_{il} \;\;\; _{il}{}^2} \qquad \frac{J_{kl} \;\;\; e}{\sum_i J_{il}{}^2 \;\;\; d}$$

Now let $a=b$ be a positive fraction less than or equal to one, let x and y be real numbers. Observe that

$$\frac{a \;\; x}{b \;\; y} \qquad \frac{a}{b} \quad \frac{x \;\; ay=b}{b \;\; y} \qquad \frac{a}{b} \quad \frac{|x| \;\; |y|}{b \;\; y}$$

$$\frac{a \;\; x}{b \;\; y} \qquad \frac{a}{b} \quad \frac{x \;\; ay=b}{b \;\; y} \qquad \frac{a}{b} \quad \frac{|x| \;\; |y|}{b \;\; y}$$

Finally,

$$_2 J \;\; _{kl} \quad \frac{|\;_e| \;\; |\;_d|}{\sum_i J_{il}{}^2 \;\;\; |\;_d|} \qquad _2 \; J \qquad _{kl} \qquad _2 J \;\; _{kl} \quad \frac{|\;_e| \;\; |\;_d|}{\sum_i J_{il}{}^2 \;\;\; |\;_d|}$$

Since $\sum_i J_{il}{}^2 \;\;\; 1=n$ it follows that the difference $| \; _2 \; J \;\;\;\; _{kl} \;\;\; _2 J \; _{kl}|$ can be bounded by cf, where $c > 0$ is a constant depending on n only. This, combined with the result on $J \;\;\; ^2$ proves the lemma.

Remark For the proof of Theorem 8 one needs also consider the behaviour of columns in M, the associated nodes of which are not attractors. It is an easy exercise to show that such columns exhibit the same behaviour as the columns of the attractor systems to which they are attracted. This concludes a first series of results on the stability and instability of the equilibrium states in L_1 in both the usual sense and a 'macroscopic' sense.

Remark In the next section experimental results are discussed. In many cases the phenomena of overlap and attractor systems of cardinality greater than one occur. Current evidence suggests that

Parameter	Meaning	Default setting
-a	Loop weight	0
-r	Initial inflation constant	2
-e	Initial expansion constant	2
-l	Initial prefix length	0
-R	Main inflation constant	2
-E	Main expansion constant	2

Figure 12: *MCL* implementation legend.

the *MCL* process converges so fast that idempotence can be recognized long before instability of overlap and attractor systems begin to play a role. This is certainly related to the fact that the examples given here concern small graphs. However, the crucial property is that the natural cluster size is not too large. Thus, large graphs G for which the natural cluster size is relatively small may also lead the *MCL* process $(T_{G}, e_{(i)}, r_{(i)})$ to converge towards idempotence before instability starts to play a role. For further discussion of this subject see the considerations in Section 10 regarding scalability of the *MCL* algorithm.

9. Cluster properties of the *MCL* algorithm

The current implementation in use at the *CWI* (henceforth called (*CWI*) implementation) with which experiments are carried out was mentioned briefly in Section 7. Its legend will be used here in order to describe the results of several *MCL*-runs on various test-graphs for various parametrizations. The implementation allows input rows $e_{(i)}$, $r_{(i)}$ which have very simple structure. The default setting is that both rows are constant. Optionally, both rows may assume another constant on a prefix of length l, where l can be specified as well. This amounts to five parameters related to the input rows specifying *MCL* process parameters. The sixth parameter indicates whether loops are added to an input graph G. If this parameter assumes a value $c \in \mathbb{R}_{\geq 0}$, the *MCL* implementation takes the graph $G + cI$ as actual input. The parameters of the *MCL* implementation must somehow be specified. Here I will use the customary UNIX notation in which parameters are indicated by strings starting with a hyphen, followed by the value which the parameter must assume. The six parameters, their meaning, and default setting are found in Figure 9. The length l of the initial prefix is indicated by '-l', the constant values assumed by $e_{(i)}$ resp. $r_{(i)}$ on the initial prefix by '-e' and '-r', the constant values on the infinite postfix by '-E' and '-R', and the loop weight by '-a' (stemming from auto-nearness).

The parameter setting -a 2 -r 1.5 -e 2 -l 10 -R 2.5 thus corresponds with an *MCL* process for which $e_i = 2$ and $r_i = 1.5$, $i = 1, \ldots, 10$, and $e_i = 2$ and $r_i = 2.5$, $i = 11, \ldots, \infty$. An input graph G will furthermore have loops of weight 2 added to each node. The parameter setting -a 1 -R 1.7 corresponds with an *MCL* process for which $e_{(i)} = {}^{c}2$ and $r_{(i)} = {}^{c}1.7$ with loops of weight 1 added to the input graph. The default setting corresponds with an *MCL* process for which $e_{(i)} = {}^{c}2$ and $r_{(i)} = {}^{c}2$ and no loops added.

Attractors

In practice, for input graphs constructed from real applications for the purpose of clustering (and thus with no strong symmetries present), the equivalence classes $E_1, \ldots, E_d$ (see Definition 12) tend to be singleton sets. In all situations observed so far where this is not the case, see e.g. the limit matrix in Figure 8, the elements in an equivalence class of cardinality greater than one are the orbit of a graph automorphism. For the graph in Figure 4, the nodes 8 and 10 share exactly the same set of neighbours, and are for that reason indistinguishable with regard to structural properties of the graph. The mapping on the node set of G_2 which only interchanges the nodes 8 and 10 is an automorphism of the graph G_2. As a second example, consider the graph G_1 in Figure 1. An *MCL* run with parametrization -a 1 results in 4 clusters, each of which is a triangle in the graph, and where each node is an attractor. Printing attractors in boldface, this is the clustering $\{\{\mathbf{0}, \mathbf{1}, \mathbf{2}\}, \{\mathbf{3}, \mathbf{4}, \mathbf{5}\}, \{\mathbf{6}, \mathbf{7}, \mathbf{8}\},$

$\{\mathbf{9},\mathbf{10},\mathbf{11}\}\}$. The cluster $\{\mathbf{0},\mathbf{1},\mathbf{2}\}$, and likewise the other clusters, is the orbit of either of its elements under rotation of G_1 around the symmetry axis orthogonal to the plane spanned by $0, 1,$ and 2.

Generally, attractors are located in local centra of density, which is best illustrated by large graphs with clear islands of cohesion. If a graph fits the picture, so to speak, of a 'gradient of density', the attractors are found in the thickest parts of the graph. This effect is to some extent illustrated by the graph in Figure 14. For this graph, it interferes however with another phenomenon occurring when a graph possesses borders. In that case, the return probabilities of nodes which lie just before those borders, profit immediately and maximally after one exponentiation step from the 'dead end' characteristic of the border. The border of the graph in Figure 14 is the outline of the grid. This explains why nearly all of the attractors in Figure 17 are nodes lying at distance one from the outline of the grid.

Overlap

The phenomenon of overlap in the setting of undirected graphs has only been observed so far for input graphs with specific symmetry properties. For these cases, if the *MCL* algorithm produces two clusters C_1, C_2 with nonempty intersection, there exists an automorphism which transforms C_1 into C_2 while leaving the intersection invariant. An example is the line graph on 7 nodes the associated matrix of which is found in Figure 4. The automorphism maps i onto $6 - i$, $i = 0, \ldots, 6$, which leaves the intersection $\{3\}$ invariant. Existence of such an automorphism means that the overlapping part forms a subset of the graph from which the graph looks the same in different directions. If those different directions correspond also with different islands of cohesion, it is rather nice if the overlapping part is not arbitrarily divided among the resulting clusters. Another example of this phenomenon can be found in Figure 17. Overlap occurs at several levels of granularity, and it always corresponds with a symmetry of the graph. For undirected graphs, the amount of possible overlap tends to be proportional to the amount of symmetry present. Naturally, any amount of overlap can be constructed by taking directed input graphs which are variants of the graph associated with the matrix 5.8.

For a fixed *MCL* process, small perturbations in the input graph generally do not affect the output clustering. An exception to this is the case where overlap occurs, as discussed in Section 7. If the symmetry corresponding with the overlap is perturbed, the overlap disappears. Consider the line graph on 7 nodes again. If the edge $\{3, 2\}$ is split into two arcs of different weights, namely the arc $2 \to 3$ with weight 1 and the arc $3 \to 2$ with weight $100/99$, an *MCL* run with parametrization -a 1 and otherwise default settings results in the partition (non-overlapping clustering) $\{\{0,1,2\},\{3,4,5,6\}\}$. This is explained by the fact that node 3 is now slightly more attracted to node 4 than it is to node 2.

The effect of adding loops

For small graphs and graphs with bipartite characteristics such as rectangular grids, adding loops is a beneficial manoeuvre. The reason for this is the same as it was for k-path clustering. The possible dependence of the transition probabilities on the parity of the simple path lengths in the graph is removed. More generally, adding loops of weight c to a graph has the effect of adding c to all the eigenvalues in its spectrum. Since negative eigenvalues correspond with periodic behaviour of the associated matrix, this has the effect of opposing periodicity. The effect of adding loops on the output clusterings of the *MCL* algorithm is that connectedness of the clusters in the graph is promoted. For an *MCL* process with default settings, specifically -E 2, and an input graph G, the value (T_{2k}) will be nonzero iff there is a path of length 2^k going from to . Clustering the graph G_2 with default parameters (i.e. -a 0 -R 2 -E 2) yields the clustering $\{\{0,4,9\},\{1,5,6\},\{2,3,7,8,10,11\}\}$. The fact that e.g. nodes 4 and 0 are now joined together, whereas nodes 4 and 1 are not, is explained by observing that there are two paths of length 2 connecting the former pair, whereas the paths of even length connecting 4 and 1 all have length at least 4. As a result, the transition probabilities $(T_{2k})_{14}$ and $(T_{2k})_{41}$ soon diminish in comparison with the transition probabilities $(T_{2k})_{04}$ and $(T_{2k})_{40}$.

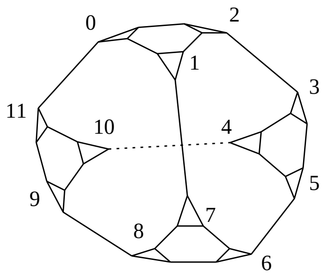

Figure 13: Graph G_4.

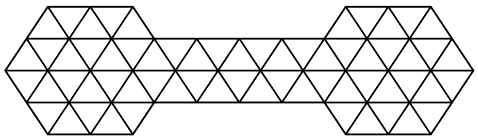

Figure 14: Graph G_5.

All example graphs in this section had loops added to them when they were input to the *MCL* algorithm. For nearly all of them this was absolutely necessary in order to get output which makes sense as a clustering. An exception is the halter shaped grid in Figure 17, which is explained by the presence of many circuits of length 3, which reduce the parity dependence as discussed in Section 3. The convergence of the *MCL* process is not notably affected by the presence or absence of loops.

The effect of inflation on cluster granularity

There is a clear correlation between the inflation parameter and the granularity of the resulting output. The higher the parameter r, the more the inflation operator Γ_r demotes flow along long path distances in the input graph. This is illustrated for the graph G_5 in Figure 14. Figure 17 gives the result of six *MCL* runs for G_5 in which the inflation parameter is varied from 1.4 to 2.5, while all other parameters are kept the same (i.e. -a 1 -e 2). Note that the corresponding overlapping clusterings are strongly related to each other. The set of all clusterings excluding the one corresponding with inflation parameter -r 1.4 is a set of nested overlapping clusterings. This is very satisfactory, as one expects clusters at different levels of granularity to be related to each other. The clusterings at the first three levels -r x, $x \in \{1.4, 1.5, 1.7\}$, have good visual appeal. It holds for all clusterings that the sizes of the respective clusters are evenly distributed, except perhaps for the clustering with parameter -r 2.0.

The second example in which the inflation parameter is varied while other parameters are kept the same concerns the graph G_4 in Figure 13. It is derived from the graph G_1 in Figure 1 by replacing each of the 12 nodes in G_1 by a triangle. Note that G_4 is a simple graph: The length of the edges in the picture do not correspond with edge weights. Now G_4 clearly allows two extreme clusterings $\Pi_1 = \{singletons(V)\}$ and $\Pi_4 = \{V\}$, a clustering Π_2 in which each of the newly formed triangles forms a cluster by itself, and a clustering Π_3 with 4 clusters in which each cluster consists of the 9 nodes corresponding with 3 newly formed triangles. Clustering with parameters -a 1 -e 2 -r x, where x varies, yields the following. Choosing $x \in [1.0, 1.2]$ results in the top extreme clustering Π_4, choosing $x \in [1.3, 1.4]$ in the clustering Π_3, choosing $x \in [1.4, 3.0]$ in the clustering Π_2, and choosing $x \in [3.1, \infty)$ results in the bottom extreme clustering Π_1. The range of x for which the clustering Π_3 results is small. This has to do with the fact that the clustering Π_3 is rather coarse. The dependencies associated with Π_k correspond with longer distances in the graph G_4 than the dependencies associated with Π_2. If the inflation parameter increases, the latter dependencies (in the form of random walks) soon profit much more from the inflation step than the former dependencies. By letting expansion continue a while before starting inflation, this can be remedied. Figure 13 shows several parameter settings and the resulting clusterings.

Parametrization				
-l	-r	-R		Clustering
0	–	1:0	1:2	Π_4
0	–	1:3	1:4	Π_3
0	–	1:5	3:0	Π_2
0	–	3:0	1	Π_1
1	1	1:0	1:3	Π_4
1	1	1:4	1:7	Π_3
1	1	1:8	5:3	Π_2
1	1	5:4	1	Π_1
2	1	1:0	1:4	Π_4
2	1	1:5	2:4	Π_3
2	1	2:5	6:8	Π_2
2	1	6:9	1	Π_1
-a 1 set everywhere				

Figure 15: *MCL* runs for the graph G_4 in Figure 13. The clusterings $\Pi_1;:::;\Pi_4$ are defined in the accompanying text.

x	-l	-r	-R	
5	2	1:2	2:1	3:2
	2	1:0	2:1	4:0
	2	0:8	2:1	5:3
6	2	1:2	2:1	2:4
	2	1:0	2:1	2:8
	2	0:8	2:1	3:3
7	3	1:2	2:1	2:7
	3	1:0	2:3	3:9
	3	0:8	2:9	6:4
8	3	1:2	2:1	2:3
	3	1:0	2:3	2:9
	3	0:8	2:9	4:3
9	3	1:2	2:1	
	3	1:0	2:3	2:5
	3	0:8	2:9	3:3
-a 1 set everywhere				

Figure 16: Parametrizations for which the *MCL* algorithm finds 10 clusters of size x each for the input graph *TORUS* $10;x$, x 5;:::;9.

The third example concerns the input graph G_2 from Figure 4 and parameters -a 1 -R x . Choosing x 2 1:5;1:8 results in the clustering ϕϕ3;7;8;10;11γ;ϕ0;1;2;4;5;6;9γγ, whereas choosing x 2 1:9;4:3 results in the clustering ϕ3;7;8;10;11γ,ϕ0;5;6;9γ,ϕ1;2;4γ). This corresponds with the fact that lower inflation parameters imply that longer path distance dependencies can influence the cluster distribution. In Section 3 it was indicated that the k-path clustering coefficient $Z_{4;G_2}$ results at some level in the clustering ϕϕ1;2;3;4;7;8;10;11γ,ϕ0;5;6;9γγ. Note that the similarity coefficient $Z_{4;G_2}$ is by definition affected by path distances of length up to 4. For the k-path clustering, it seems as if the relatively large and dense cluster ϕ3;7;8;10;11γ (which contains a complete graph as subgraph on the four nodes 3;7;8;10) has gobbled up the neighbouring lighter-weight cluster ϕ1;2;4γ. It is remarkable that the *MCL* clustering leads the latter cluster instead to be joined with the other lighter-weight cluster ϕ0;5;6;9γ. This may indicate that heavy-weight islands of cohesion do not have an absorbing quality with regard to the *MCL* algorithm. I think this is a desirable property, which promotes clusterings being wellbalanced.

The last examples are rectangular torus-graphs. A k-dimensional rectangular torus graph generalizes a ring graph in k dimensions. It is most conveniently defined as a sum of ring graphs, defined on the Cartesian product of the respective node sets.

Definition 14 Let G_i $V_i;w_i$;i 1;:::;n be an n-tuple of simple graphs. The **sum graph** S of $G_1;:::;G_n$ is defined on the Cartesian product V_1 ::: V_n. Two vertices $x_1;:::;x_n$ and $y_1;:::;y_n$ are connected in S if exactly one of the pairs $x_i;y_i$ is connected in G_i, and x_i y_i for the remaining n 1 pairs.

Definition 15 The 1-dimensional **torus graph** or **ring graph** of cardinality t is the simple graph defined on the integers modulo t: 0;:::;t 1 , where there is an edge between i and j iff i j 1 mod t or j i 1 mod t .

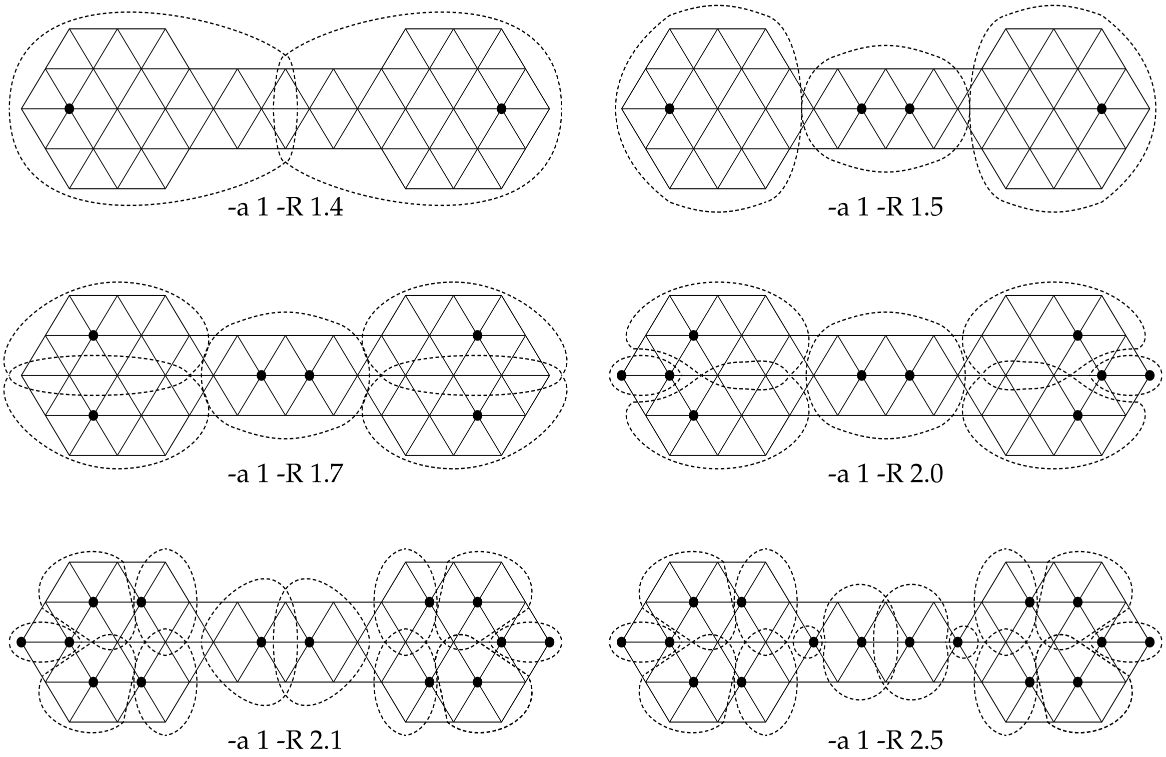

Figure 17: Clusterings resulting from the *MCL* algorithm for the graph in Figure 14. Dotted nodes are attractors.

A graph is called a k–dimensional torus graph if it is the sum graph of k ring graphs. It can be identified with a k–tuple $t_1;\ldots;t_k$, where t_i is the cardinality of the node set of the i^{th} ring graph. The torus graph corresponding with this k–tuple is denoted $TORUS\ t_1;\ldots;t_k$.

Here I will use only 2– and 3–dimensional simple torus graphs. A 2–dimensional torus graph $TORUS\ k;l$ can be thought of as a rectangular grid of width k and depth l, where nodes lying opposite on parallel borders are connected. In Section 7 it appeared that periodic MCL limits exist which have the same automorphism group as ring graphs. A two dimensional torus graph G $TORUS\ k;l$ where k l has the same homogeneity properties as ring graphs. It is interesting to see what happens if $k > l$. Consider a node pair $u_1;u_2$ lying on a ring of length l in G at a (shortest path) distance t l from each other, and a node pair $v_1;v_2$ in G, also lying at distance t from each other, but not lying on such a ring. The transition probability associated with going in l steps from u_1 to u_2 is larger than the transition probability associated with going in l steps from v_1 to v_2, because u_1 can reach u_2 in two ways along the ring on which they both lie, while this is not true for v_1 and v_2. Is it possible to find an MCL process in which this effect is boosted such that a clustering of G in k clusters of size l each results? This is indeed the case, and it requires the usage of input rows r_i which are not constant everywhere. If l is very close to k, it is furthermore beneficial to use an initial inflation parameter which is close to or smaller than 1. Without this, the return probability of each node grows too large before paths of length l start to have influence, which is after $\lceil \log_2 l \rceil$ exponentiation steps (assuming e_i ^{c} 2). Figure 9 shows parameter settings for which the MCL algorithm output divides the graphs $TORUS\ 10;x$ in 10 clusters of cardinality x each, x $5;\ldots;9$. These are of course not the only parametrizations achieving this, but among the parametrizations found they lead to fast convergence of the MCL process.

The last example concerns the 3–dimensional torus graph $TORUS\ 3;4;5$. A priori it is to be expected that the non–extreme clusterings which the MCL algorithm can possibly produce are the clustering Π_2 corresponding with 20 subgraphs isomorphic to $TORUS\ 3$ and the clustering Π_3 corresponding with 5 subgraphs isomorphic to $TORUS\ 3;4$. Denote the top and bottom extreme clusterings by Π_1 $\{$singletons $V\}$ and Π_4 $\{V\}$ respectively. The table below gives four parameter ranges yielding the four clusterings Π_i.

Parametrization				
-l	-r	-R		Clustering
2	1.2	1:0	2:3	Π_4
2	1.2	2:4	3:3	Π_3
2	1.2	3:4	6:4	Π_2
2	1.2	6:5	1	Π_1

The torus examples illustrate the strong separating power of the MCL process. The clusterings shown for the torus graphs, the tetraeder-shaped graphs in Figures 1 and 13, and the grid in Figure 14 illustrate that the MCL algorithm 'recognizes' structure even if the node degrees in the input graph are homogeneously distributed and the connectivity of the graph is high. The inflation parameter clearly is the main factor influencing the granularity of the output clusterings. By all evidence, the output clustering changes for specific values of the inflation parameter constant (either the prefix or the postfix value), and stays the same for in-between intervals. By prolonging expansion, coarser clusterings can be found.

10. Complexity and scalability

The complexity of the *MCL* algorithm, if nothing special is done, is $O\ N^3$ where N is the number of nodes of the input graph. The factor N^3 corresponds to the cost of one matrix multiplication on two matrices of dimension n. The inflation step can be done in $O\ N^2$ time. I will leave the issue aside here of how many steps are required before the algorithm converges to a doubly idempotent matrix. In practice, this number lies typically somewhere inbetween 5 and 20. The real bottleneck of the algorithm is the exponentiation step. The *CWI* implementation has two different modes, corresponding with different approaches towards matrix multiplication. In the first or *standard* mode, product matrices are computed by the standard matrix multiplication procedure, taking $O\ N^3$ time. This mode is useful for testing the algorithm on graphs with up to several thousands of nodes. The second or *prune* mode works a lot faster and somewhat more inaccurate. It allows handling of graphs with up to a hundred thousand nodes, and this bound may well be further stretched. There are some restrictions on the type of graphs fit for application of the prune mode. This is discussed at the end of this section. Pruning reduces the complexity to $O\ Nn$, where n is small compared to N and 3.

Column pruning and row pruning

In prune mode, it is not the exact product of two matrices which is computed, but a matrix very close to it. The basic idea stems from the observation that *MCL* runs on sparse graphs yield limit matrices which are extremely sparse, and intermediate matrices which are rather sparse. The columns of limit matrices most often have just one positive element, perhaps only a few, whereas rows of a matrix limit may have as many entries as the largest resulting cluster. The columns of intermediate iterands generally have a few larger entries, several small entries, and many entries which are close to zero.

The prune mode uses the preceding in the following way. For every matrix iterand, the number of positive entries any column can have is uniformly bounded by a constant n, where n can be specified in advance. This is referred to as **column pruning**. A matrix product is computed by successively computing each column of the product. All positive entries of a column are computed, yielding possibly more than n entries. The n largest of these are kept, the rest is set to zero, and the remaining (n largest) entries are rescaled to have sum 1 again. The larger n is, the closer this approximate product will be to the actual product. With the right architecture, the complexity of computing the approximate product is $O\ N^2 n$ (see below). A second relaxation can reduce this (depending on the value of n) further to $O\ Nn\ ;$ 3. When computing the c^{th} column of the product AB, under the condition that the columns of both A and B have at most n positive entries, only those rows of A need be considered which have nonzero inner product with the c^{th} column of B. It is straightforward to see that there are at most n^2 such rows. This yields a complexity of $O\ Nn^3$.

Now consider the c^{th} column of AB. It is computed by multiplying a given set of rows of A with the c^{th} column of B. An entry AB_{rc} is the sum of the scalar products $A_{ri}B_{ic}; i\ \ 0;\ldots;N\ \ 1$. In the context of the *MCL* algorithm, the values A_{ri} and B_{ic} both have the interpretation 'how much node i (resp. c) is attracted to node r (resp. i). It is an option in the implementation to consider only rows r for which at least one index i exists such that node c is 'sufficiently' attracted to node i, and node i is 'sufficiently' attracted to node r. Otherwise the entry AB_{rc} is simply not computed and set to zero. This selection scheme (further specified below) is referred to as **row pruning**. The motivation is that a node can reach a new neighbour only if there is at least one 2-step path of sufficient weight to it. This procedure is further justified by the fact that it is very probable that all rows thus disregarded, would otherwise have been dismissed in the column pruning phase. The notion of sufficiency is defined in terms of a cut value defined for probability vectors, separating larger transition probabilities from smaller transition probabilities. The cut value is parametrized relative to the notion of *mass center* defined for probability vectors. This notion is related to the behaviour of . A slight digression in the form of a definition and a lemma formalizing this follows below.

Definition 16 Let be a probability vector of dimension d, let r be a real number greater than one. The **mass center** of order r of is defined as

$$_r \qquad \left(\sum_{i=0}^{d-1} {}_i^{\,r} \right)^{\frac{1}{r-1}} \qquad (10.1)$$

Lemma 11 Let be a probability vector of dimension d. The indices i for which ${}_r\ {}_i\ \ {}_i$ are those indices for which ${}_i\ \ {}_r$.

Proof If ${}_i\ \ {}_r$ holds then

$$\frac{{}_r\ {}_i}{{}_i} \qquad \frac{{}_i^{\,r-1}}{\sum {}_i^{\,r}} \qquad \frac{{}^{\,r-1}}{\sum {}_i^{\,r}} \qquad 1$$

Given a probability vector , the number ${}_r$ separates the transition probabilities which grow under ${}_r$ from those which shrink under ${}_r$. A logical candidate for the cut value mentioned above, defined for probability vectors, is to define it as a function of their mass center, e.g. the mass center of order 2. The *CWI* implementation essentially uses a slight refinement of this scheme. The combination of column pruning and row pruning yields a complexity $O(Nn)$ for matrix multiplication, where varies from 3 during the early stages of the *MCL* algorithm to 2 and subsequently 1 during the middle stages. When the iterands are very close to a sparse doubly idempotent matrix the complexity is effectively $O(N)$. This is achieved by an architecture for matrices and matrix multiplication in which only non-zero elements are stored.

Applicability requirements on pruning

For large graphs, the range in which n can be chosen depends on the computing power available. In general there will be some bound on n. This then affects the type of graphs which can be clustered. The combination of column pruning and row pruning has been tested extensively for *MCL* processes with constant expansion row $e_i^{\,c}$ 2, and graphs with up to 60,000 nodes having an additional density characteristic. The additional characteristic is that the natural size of the clusters occurring in the graph is in the same order of magnitude as the pruning constant n. Detection of substantially larger clusters requires prolonged expansion. The matrix iterands then become less sparse, the iterand columns become more spread out, and convergence towards an equilibrium state will take up much more time and memory. The fact that the columns become more spread out implies that the degree in which pruning perturbs the *MCL* process is inversely proportional to the degree of expansion prolongation and the size of clusters sought for. It follows that in the setting of large graphs the main strength of the *MCL* algorithm lies in retrieving small to medium sized clusters. Column pruning and row pruning can then be applied to speed up the algorithm.

Convergence in the setting of pruning

The convergence properties in the setting sketched above do not change noticeably, and the resulting clusterings are still very satisfactory. Clusterings of graphs with up to a thousand nodes resulting from both the standard mode and the prune mode with otherwise identical parametrizations were compared. The respective clusterings sometimes differed slightly (e.g. a node moving from one cluster to another) and were often identical. The parameter n can be chosen surprisingly low without causing deterioration of the output. Setting $n = 50$ for graphs with 60,000 nodes is very well possible. An example of pruning is given in Figure 18. The equilibrium state and several matrix iterands are given for the *MCL* process with input graph G_2, and pruning constant $n = 4$. The iterand indices correspond with the iterand indices in Figure 8. The clustering resulting from this adapted process is the same as the clustering resulting from the standard process.

Benchmarking proposal

Many experiments were carried out for the *MCL* algorithm on several graph types with specific characteristics, such as the graphs occurring throughout this report. In all these cases, the correlation between input graphs, heuristic expectations and output clusterings is very satisfactory. A substantial number of experiments was also done on graphs constructed from document-phrase databases and thesaurus structures such as Roget's thesaurus. These experiments are promising and indicate that the *MCL* algorithm is suited for facilitating the tasks of finding semantic structure in such objects. This constitutes an interesting line of research. Meanwhile, support for the soundness of the *MCL* process can come from two directions. Firstly, it is desirable to have a theoretical framework answering some of the many questions accompanying the process. This is discussed in the next section. The second possibility is the creation of a benchmarking package for graph clustering. Here I present a proposal for such a package. A test graph for the purpose of sparse graph clustering must be a graph with certain a priori known characteristics which may vary within certain bounds, which can also be specified a priori. A candidate scheme for such test graphs is the following.

Definition 17 Let N, E be integers, let $\Pi = \{P_1, \ldots, P_t\}$ be a partition of the integers $1, \ldots, N$, let p and q be fractions lying between 0 and 1, and let f and g be functions mapping the positive reals to the positive reals. An undirected (N, E, Π, p, q, f, g) testgraph $G = (V, w)$ is an undirected graph having the following properties,

- G has N nodes.
- G has E edges.
- The average within degree of each subset P_i is at least $f(|P_i|)\, p\, 2E/N$, where $0 < p < 1$.
- The minimum within degree of each subset P_i is at least $g(|P_i|)\, q\, 2E/N$, where $0 < q < p < 1$.

Here the average within degree of a subset $S \subset V$ is the average degree of the nodes in S in the subgraph on S inherited from G. The notion of minimum within degree is defined analogously.

The functions f and g have been introduced so that the density properties of the subsets P_i may vary depending on the sizes $|P_i|$. If f and g are chosen constant 1, then the fractions p and q give an indication of the number of edges that each node can choose outside the subset P_i to which it belongs. Note that the number $2E/N$ is the average node degree in the graph G. Examples:

- The graph G_1 is a $(N = 12, E = 18, \Pi = \Theta, p = 2/3, q = 2/3, f \equiv g \equiv 1)$ testgraph, where Θ is the clustering of G_1 into four triangles.
- The graph G_2 is a $(N = 12, E = 20, \Pi = \Theta, p = 3/5, q = 3/5, f \equiv g \equiv 1)$ testgraph, where Θ is the clustering $\{\{0,5,6,9\}, \{1,2,4\}, \{3,7,8,10,11\}\}$.

$$
\begin{pmatrix}
\begin{array}{cccccccccccc}
0.4048 & 0.0941 & & & 0.0907 & 0.2945 & 0.2232 & & & 0.3234 & & \\
 & 0.3763 & 0.2276 & & 0.1779 & 0.0192 & & & & & & \\
 & 0.2276 & 0.3763 & 0.0628 & 0.1779 & & & & & & & \\
 & & 0.0941 & 0.3419 & & & & 0.2078 & 0.1459 & & 0.1459 & 0.0828 \\
0.0615 & 0.2276 & 0.2276 & & 0.4800 & & 0.0923 & & & 0.0189 & & \\
0.1508 & & & & & 0.2945 & 0.0923 & & & 0.1858 & & \\
0.1204 & 0.0743 & & & 0.0735 & 0.0972 & 0.3690 & & & 0.1484 & & \\
 & & 0.0743 & 0.1984 & & & & 0.3247 & 0.1459 & & 0.1459 & 0.0828 \\
 & & & 0.1984 & & & & 0.2078 & 0.2928 & & 0.2928 & 0.2782 \\
0.2625 & & & & & 0.2945 & 0.2232 & & & 0.3234 & & \\
 & & & 0.1984 & & & & 0.2078 & 0.2928 & & 0.2928 & 0.2782 \\
 & & & & & & & 0.0519 & 0.1226 & & 0.1226 & 0.2782
\end{array}
\end{pmatrix}
$$

$_2 M^2$ for *MCL* process with pruning.

$$
\begin{pmatrix}
\begin{array}{cccccccccccc}
0.4608 & 0.0659 & & & 0.0645 & 0.4329 & 0.3760 & & & 0.4360 & & \\
 & 0.2940 & 0.2561 & & 0.1863 & 0.0007 & & & & & & \\
 & 0.2411 & 0.3283 & & 0.1863 & & & & & & & \\
 & & 0.0393 & 0.2562 & & & & 0.1945 & 0.1446 & & 0.1446 & 0.1034 \\
0.0216 & 0.3616 & 0.3476 & & 0.5254 & & 0.0443 & & & 0.0098 & & \\
0.1188 & & & & & 0.1559 & 0.0889 & & & 0.1297 & & \\
0.0995 & 0.0374 & & & 0.0376 & 0.0864 & 0.2147 & & & 0.1072 & & \\
 & & 0.0286 & 0.1975 & & & & 0.2104 & 0.1383 & & 0.1383 & 0.1004 \\
 & & & 0.2641 & & & & 0.2817 & 0.3284 & & 0.3284 & 0.3463 \\
0.2993 & & & & & 0.3241 & 0.2761 & & & 0.3173 & & \\
 & & & 0.2641 & & & & 0.2817 & 0.3284 & & 0.3284 & 0.3463 \\
 & & & 0.0181 & & & & 0.0318 & 0.0603 & & 0.0603 & 0.1037
\end{array}
\end{pmatrix}
$$

$_2$ $_2 M^2$ $_2 M^2$, for *MCL* process with pruning.

$$
\begin{pmatrix}
\begin{array}{cccccccccccc}
0.9973 & 0.0000 & & & 0.0000 & 0.9973 & 0.9973 & & & 0.9973 & & \\
 & 0.0000 & 0.0000 & & 0.0000 & & & & & & & \\
 & 0.0000 & 0.0000 & & 0.0000 & & & & & & & \\
 & & & 0.0000 & & & & 0.0000 & 0.0000 & & 0.0000 & 0.0000 \\
0.0000 & 0.9999 & 0.9999 & & 0.9999 & & 0.0000 & & & 0.0000 & & \\
0.0000 & & & & & 0.0000 & 0.0000 & & & 0.0000 & & \\
0.0000 & 0.0000 & & & 0.0000 & 0.0000 & 0.0000 & & & 0.0000 & & \\
 & & & 0.0000 & & & & 0.0000 & 0.0000 & & 0.0000 & 0.0000 \\
 & & & 0.5000 & & & & 0.5000 & 0.5000 & & 0.5000 & 0.5000 \\
0.0027 & & & & & 0.0027 & 0.0027 & & & 0.0027 & & \\
 & & & 0.5000 & & & & 0.5000 & 0.5000 & & 0.5000 & 0.5000 \\
 & & & 0.0000 & & & & 0.0000 & 0.0000 & & 0.0000 & 0.0000
\end{array}
\end{pmatrix}
$$

$_2$ *Squaring* iterated six times on M for *MCL* process with pruning

$$
\begin{pmatrix}
\begin{array}{cccccccccccc}
1.0000 & & & & & 1.0000 & 1.0000 & & & 1.0000 & & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & & \\
 & 1.0000 & 1.0000 & & 1.0000 & & & & & & & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & & \\
 & & & & & & & & & & & \\
 & & & 0.5000 & & & & 0.5000 & 0.5000 & & 0.5000 & 0.5000 \\
 & & & & & & & & & & & \\
 & & & 0.5000 & & & & 0.5000 & 0.5000 & & 0.5000 & 0.5000 \\
 & & & & & & & & & & &
\end{array}
\end{pmatrix}
$$

M^1_{mcl}

Figure 18: Iteration of $_2$ *Squaring* with initial iterand M defined in Figure 7. Pruning with pruning constant n 5 is applied throughout the process.

- The graph G_2 is also a N 12; E 20; Π Θ; p 9=10; q 3=4; f; g testgraph, where Θ is as before and where f : x , 1 1=x, and g f .

The parameters p and q, in conjunction with the functions f and g, determine the density characteristics of the graph G. High values of p and q imply that the partition Π is a good partition clustering of the graph G, since in that case there can be only few edges inbetween different partition elements. Lowering p and q corresponds with gradually clouding this clear picture. I intend to create a benchmark package with which test graphs as defined here can be randomly generated. Testing the *MCL* algorithm on such graphs will be a good testcase for its robustness and performance.

11. Conclusions and further research

The *MCL* algorithm described here seems very promising. Its design is simple, and is basically nothing more than alternation of matrix expansion and matrix inflation. Alternation of these two operators appears to be intrinsically related to cluster structure present in the input graph. A sensible interpretation of the algorithm, which was in fact the heuristic leading to its formulation, is that flow is simulated by iterated expansion and contraction. The algorithm is notable for its strong separating power, as evidenced by the various experiments here described. Convergence of the algorithm is fast. It was shown that the algebraic process employed by the algorithm converges quadratically in the neighbourhood of its equilibrium states which correspond with doubly idempotent matrices. In a forthcoming article the inflation operator will be shown to have a number of properties which give the *MCL* algorithm a sound mathematical foundation. Below some further questions are listed which seem both interesting and difficult.

Questions

- Does the input of an arbitrary nonnegative matrix always lead to a limit of one of the types L_3, L_2, or L_1?
- Do there exist matrices of the L_3 type which are not of the L_{3a} type?
- Do there exist matrices M which are not of L_2 or L_1 type that lead to a limit of one of these types? That is, do such matrices M exist for which the basin of attraction is greater than just the set $\phi M \gamma$?
- For a fixed *MCL* process ; e_i ; r_i , what can be said about the basins of attraction of the *MCL* process. Are they connected?

  What can be said about the union of all basins of attraction for all limits which correspond with the same overlapping clustering (i.e., differing only in the distribution of attractors)?
- Can the set of limits reachable from a fixed nonnegative matrix M for all *MCL* processes M; e_i; r_i be characterized? Can it be related to a structural property of M?
- Under what conditions do the clusters in the cluster interpretation of the limit of a convergent *MCL* process M; e_i; r_i correspond with connected subgraphs in the associated graph of M?

References

1. Abraham Berman and Robert J. Plemmons. *Nonnegative Matrices In The Mathematical Sciences*. Computer Science and Applied Mathematics. Academic Press, 1979.
2. Stijn van Dongen. Graph clustering and information structure. In Kraak and Wassermann [8], pages 13–31.
3. Brian S. Everitt. *Cluster Analysis*. Hodder & Stoughton, third edition, 1993.
4. Michiel Hazewinkel. Classification in mathematics, discrete metric spaces, and approximation by trees. Technical Report AM-R9505, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam, April 1995.
5. Michiel Hazewinkel. Tree-tree matrices and other combinatorial problems from taxonomy. Technical Report AM-R9507, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam, April 1995.
6. Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.
7. Nicholas Jardine and Robin Sibson. *Mathematical Taxonomy*. Wiley Series In Probabilistic And Mathematical Statistics. John Wiley & Sons, 1971.
8. Esther Kraak and Renata Wassermann, editors. *Proceedings Accolade 97*. University of Amsterdam, 1998.
9. Albert W. Marshall and Ingram Olkin. *Inequalities: Theory of Majorization and Its Applications*, volume 143 of *Mathematics In Science And Engineering*. Academic Press, 1979.
10. Albert W. Marshall, Ingram Olkin, and Frank Proschan. Monotonicity of ratios of means and other applications of majorization. In Shisha [14], pages 177–197.
11. Henryk Minc. *Nonnegative Matrices*. Wiley Interscience Series In Discrete Mathematics And Optimization. John Wiley & Sons, 1988.
12. Boris Mirkin. *Mathematical Classification and Clustering*. KAP, 1996.
13. E. Seneta. *Non–negative matrices and Markov chains*. Springer, second edition, 1981.
14. Oved Shisha, editor. *Inequalities*. Academic Press, 1967.