Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Analysis of Markov Influence Graphs

Joost Berkhout, Bernd F. Heidergott

http://pubsonline.informs.org/journal/opre/

OPERATIONS RESEARCH
Vol. 67, No. 3, May–June 2019, pp. 892–904
ISSN 0030-364X (print), ISSN 1526-5463 (online)

METHODS

Analysis of Markov Influence Graphs

Joost Berkhout,[a,b] Bernd F. Heidergott[b]

[a] Centrum Wiskunde and Informatica, 1098 XG Amsterdam, Netherlands; [b] Department of Econometrics and Operations Research, Vrije Universiteit Amsterdam, 1081 HV Amsterdam, Netherlands

Contact: joost.berkhout@cwi.nl, https://orcid.org/0000-0001-5883-9683 (JB); b.f.heidergott@vu.nl, https://orcid.org/0000-0002-3389-2311 (BFH)

Received: May 17, 2016
Revised: June 21, 2017; May 23, 2018
Accepted: July 27, 2018
Published Online in Articles in Advance: April 3, 2019

Subject Classifications: probability: stochastic model applications; networks/graphs: stochastic
Area of Review: Stochastic Models

https://doi.org/10.1287/opre.2018.1813

Abstract. The research presented in this paper is motivated by the growing interest in the analysis of networks found in the World Wide Web and of social networks. In this paper, we elaborate on the Kemeny constant as a measure of connectivity of the weighted graph associated with a Markov chain. For finite Markov chains, the Kemeny constant can be computed by means of simple algebra via the deviation matrix and the ergodic projector of the chain. Using this fact, we introduce a new decomposition algorithm for Markov chains that splits the graph the Markov chain is defined on into subgraphs, such that the connectivity of the chain measured by the Kemeny constant is maximally decreased. We discuss applications of our decomposition algorithm to influence ranking in social networks, decomposition of a social network into subnetworks, identification of nearly decomposable chains, and cluster analysis.

Supplemental Material: The online appendix is available at https://doi.org/10.1287/opre.2018.1813.

Keywords: Markov influence graphs • social networks • deviation matrix • Markov multichains • Kemeny decomposition algorithm • nearly decomposable Markov chains

1. Introduction

Consider a directed graph with finite node set $\mathbb{S} = \{1, \ldots, n\}$ and set of edges $\mathscr{E} \subset \mathbb{S} \times \mathbb{S}$. Let a Markov chain P be defined on $\mathbb{S}$ such that $P(i,j) > 0$, for $(i,j) \in \mathscr{E}$, where $P(i,j)$ denotes the transition probability of going from i to j, and $P(i,j) = 0$, for $(i,j) \notin \mathscr{E}$. For the sake of technical simplicity, we equip isolated nodes with a self-loop—that is, when for $i \in \mathbb{S}$ there exists no $j \in \mathbb{S}$ such that $(i,j) \in \mathscr{E}$, then we artificially add edge (i,i) to $\mathscr{E}$. We call P a *Markovian chain* on graph $(\mathbb{S}, \mathscr{E})$, and we consider in the following the weighted directed graph $(\mathbb{S}, \mathscr{E}, P)$, called a *Markov influence graph*.

The interest in Markov influence graphs—that is, in research that elaborates on the relation between the graph structure $(\mathbb{S}, \mathscr{E})$ and the Markov chain P—stems from the analysis of hyperlink networks found on the World Wide Web and the growing interest in the analysis of social networks. For example, the acclaimed PageRank algorithm used by Google's search engine to rank websites on the internet (see Brin and Page 1998, Langville and Meyer 2011) relies on a Markov chain constructed from the actual graph constituted by the hyperlink information. In social network theory, the interaction graph between social agents provides the means for studying naive belief updating in social networks, where $P(i,j)$ models the strength of the belief that agent i has in the assessments of agent j (see DeGroot 1974). Study of the wisdom-of-crowds phenomenon relies on application of the asymptotic theory of Markov chains to social networks [see Golub and Jackson (2010) and the references therein].

In this paper, we consider general finite Markov chains. Here *general* means that the Markov chain may have several ergodic classes and transient states and may be periodic. The *deviation matrix* of a general Markov chain P is given by

$$D_P = (I - P + \Pi_P)^{-1} - \Pi_P,$$

where I is an appropriately sized identity matrix, and $\Pi_P = (\Pi_P(i,j))_{\mathbb{S}\times\mathbb{S}}$ is called the *ergodic projector* of P, given by

$$\Pi_P = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} P^n,$$

where element $\Pi_P(i,j)$ gives the long-term fraction of visits of the Markov chain to j when started at i. The fundamental role of the deviation matrix is best expressed by C. D. Meyer (1975, p.1) who wrote that "virtually everything that one would want to know about Markov chains can be determined by computing the deviation matrix." For example, the deviation matrix allows one to analyze mean first passage times, thereby providing information on the distance between states, where the *distance* of state i to state j is measured by the mean number of transitions required to go from i to j. As an illustration, a set of strongly connected nodes may be decomposable into, say, two subsets, where (1) a subset contains all nodes that are relatively close to each other and (2) nodes from different subsets have a relatively large distance between them. The phenomenon that a Markov chain P has sets of nodes that are relatively more close has been frequently observed in the literature: It is called *homophily* in social networks (Mcpherson et al. 2001) and is known as *nearly decomposability*[1] in Markov chain theory (Meyer 1989, Stewart 1994).

Evaluating D_P allows for the computation of a network connectivity measure called the *Kemeny constant*. The Kemeny constant is a weighted average of mean first passage times (to be formally defined later). A network modeled by a Markov chain with a small Kemeny constant has relatively good connectedness and vice versa. We will exploit this insight for identifying the (weak) links in a network, the removal of which would reveal the actual subsets of nodes driving the network dynamic. This is of great interest in the analysis of the limiting behavior of Markov chains and has a wide range of further applications discussed in Section 2.

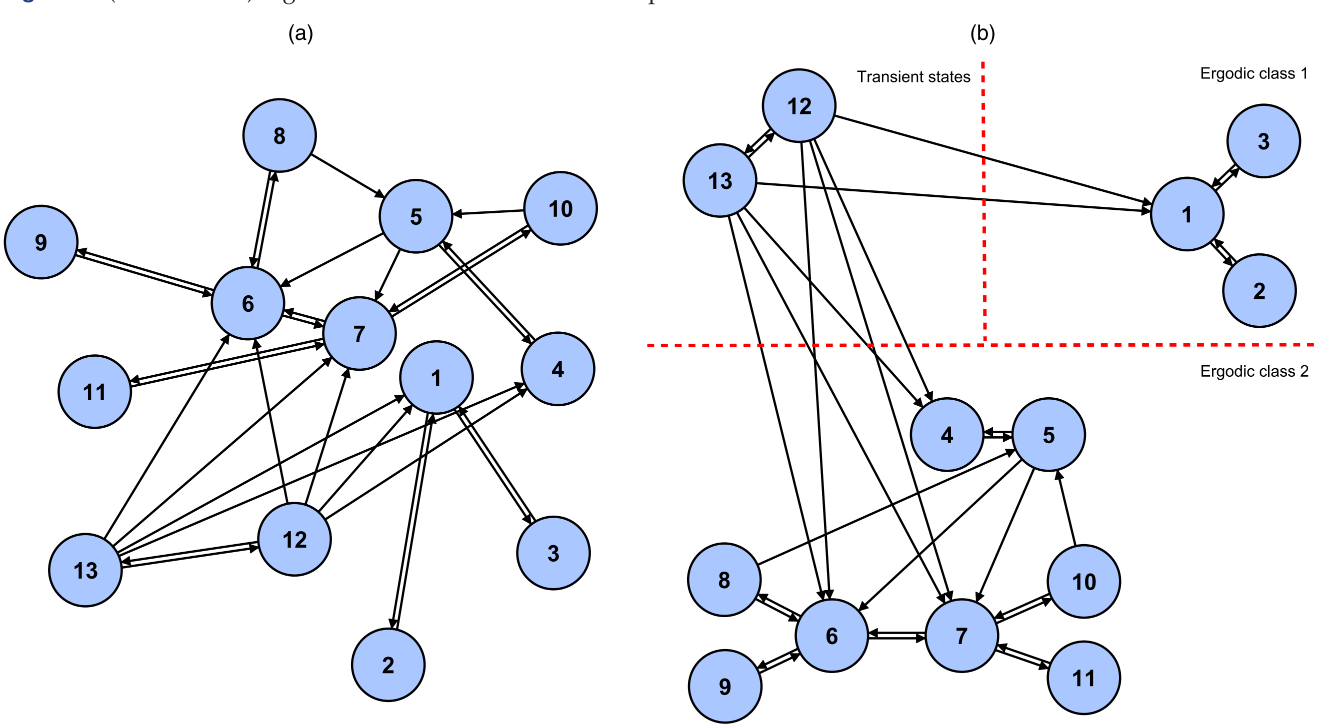
We show in this paper that one can take the derivative of the Kemeny constant with respect to any entry of P. This allows us to identify an entry (i, j) of P such that the derivative of the Kemeny constant with respect to this entry has the smallest negative value—that is, we find the link (i, j) that has the largest impact on decreasing the Kemeny constant. In other words, one can identify the edge $(i, j) \in \mathscr{E}$ such that removing the possibility of going from i to j leads to a maximal decrease in connectivity of the network modeled by the Markov chain. This leads to our *Kemeny decomposition algorithm* for Markov chains, which allows us to split a Markov chain on graphs into subgraphs such that the connectivity of the chain measured by the Kemeny constant is maximally decreased. The rationale is that this leads to a "natural" decomposition into subclasses, which brings the true network dynamics to the fore. Furthermore, our algorithm can be applied to nearly decomposable Markov chains, decomposition of a social network into subgraphs, and cluster analysis. Although the motivation of this research and our leading examples stem from the analysis of social networks, we would like to emphasize that the methods and results developed in this paper can be applied to any finite weighted (di)graph (see also Section 5).

The rest of this paper is organized as follows. In Section 2, we illustrate how mean recurrence times and the Kemeny constant lead to natural decompositions of Markov chains. Section 3 formally introduces Markov chain concepts used in our analysis and extends the concept of the Kemeny constant to multichains. Our Kemeny decomposition algorithm is presented in Section 4, and numerical experiments are provided in Section 5. Topics of further research are discussed in Section 6. Some of the proofs and additional examples are provided in the online appendix.

2. "Natural" Decomposition of Markov Chains

Consider the small social network shown in Figure 1. This network consists of 13 social agents, numbered from 1 to 13—that is, $\mathbb{S} = \{1, \ldots, 13\}$. There is a directed edge from agent i to agent j if the opinions/beliefs of

Figure 1. (Color online) Figures of the Social Network Example from Section 2



agent i are influenced by the opinions/beliefs of agent j. A (chaotic) plot of the network is given in Figure 1(a).

Figure 1(b) is obtained from analysis of the graph structure and shows that there are three strongly connected subgraphs, the first consisting of social agents 1, 2, and 3, and the second consisting of social agents $4, 5, \ldots, 10$, and 11. The third strongly connected subgraph, containing social agents 12 and 13, turns out to have no influence on other agents. In Markov chain terminology, agents 12 and 13 represent transient states, and the other two strongly connected subgraphs are ergodic classes of P (see also Figure 1(b)). Weights can be attached to the directed edges reflecting the strength of influence (see DeGroot 1974, Winkelmann and Haselmann 2010). Normalizing the weights, one obtains a transition probability matrix P for the Markov influence graph. Details on the construction of P will be provided in Section 5. For our primary social network example, we will be using the derived transition matrix P as given in Section EC.1 of the online appendix.

An interpretation of P can be given via the so-called random walk model, which models a random walk on the social network. When the walk is at agent i, the random walk is continued at agent j with probability given by entry (i, j) of P [in the context of internet networks, this was called the random surfer model in Brin and Page (1998)]. The ergodic projector for the social network from Figure 1 with P as given in Section EC.1 of the online appendix can be computed to be (rounded to two decimals):

$$\Pi_P = \begin{bmatrix}
0.5 & 0.25 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5 & 0.25 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5 & 0.25 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\text{Ergodic class 1} & & & & & & & & & & & & \\
0 & 0 & 0 & 0.1 & 0.11 & 0.26 & 0.26 & 0.05 & 0.08 & 0.05 & 0.08 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0.11 & 0.26 & 0.26 & 0.05 & 0.08 & 0.05 & 0.08 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0.11 & 0.26 & 0.26 & 0.05 & 0.08 & 0.05 & 0.08 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0.11 & 0.26 & 0.26 & 0.05 & 0.08 & 0.05 & 0.08 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0.11 & 0.26 & 0.26 & 0.05 & 0.08 & 0.05 & 0.08 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0.11 & 0.26 & 0.26 & 0.05 & 0.08 & 0.05 & 0.08 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0.11 & 0.26 & 0.26 & 0.05 & 0.08 & 0.05 & 0.08 & 0 & 0 \\
0 & 0 & 0 & 0.1 & 0.11 & 0.26 & 0.26 & 0.05 & 0.08 & 0.05 & 0.08 & 0 & 0 \\
& & & & & & & \text{Ergodic class 2} & & & & & \\
0.13 & 0.06 & 0.06 & 0.08 & 0.08 & 0.2 & 0.2 & 0.04 & 0.06 & 0.04 & 0.06 & 0 & 0 \\
0.05 & 0.03 & 0.03 & 0.09 & 0.09 & 0.24 & 0.24 & 0.05 & 0.07 & 0.05 & 0.07 & 0 & 0
\end{bmatrix},$$

which shows that P has ergodic classes $\{1, 2, 3\}$ and $\{4, \ldots, 11\}$ and transient states $\{12, 13\}$.

The ergodic projector allows for ranking agents according to their influence through the values of the ergodic projector Π_P inside the ergodic classes. Specifically, $\Pi_P(1,1) > \Pi_P(1,2) = \Pi_P(1,3)$ implies that the influence ranking inside ergodic class 1 is

$$1 \succ 2 \sim 3,$$

meaning that 1 is more influential ($\succ$) than 2 and 3, and 2 and 3 are equally influential ($\sim$). This is intuitively clear because both 2 and 3 point toward 1, meaning that 1 influences two agents, while 2 and 3 both only influence 1 (which finds 2 and 3 equally interesting/important). Similarly for ergodic class 2, it holds for the influence ranking therein that

$$6 \sim 7 \succ 5 \succ 4 \succ 9 \sim 11 \succ 8 \sim 10,$$

which is reasonable because 6 and 7 form the hub in this ergodic class, meaning that they can be seen as the most influential agents inside ergodic class 2. The ranking between the ergodic classes is not clear because there is no direct connection between these classes. In order to achieve a global ranking, the bored random surfer approach can be applied (Brin and Page 1998, Langville and Meyer 2011), or a weighted ranking of the ergodic classes together with the transient nodes can be considered (Berkhout 2016, Berkhout and Heidergott 2017).

The mean first passage times between nodes gives insight into distances between nodes. Let $M(i, j)$ denote the mean first passage time from i to j, provided i and j belong to the same ergodic class. In the social network context, entry $M(i, j)$ can be interpreted as the directed distance of social agent i to social agent j. Kemeny and Snell (1976) show that for Markov chains with a single ergodic class and no transient states, the corresponding mean first passage time matrix, denoted by M, can be calculated from the ergodic projector Π_P and the deviation matrix D_P as

$$M = (I - D_P + \bar{1}\bar{1}^\top \cdot \mathrm{dg}(D_P)) \cdot \mathrm{dg}(\Pi_P)^{-1},$$

where $\bar{1}$ is an appropriately sized vector of ones, and $\mathrm{dg}(A)$ results from A by setting off-diagonal entries equal to zero. Particularly for the social network example, calculating M for the ergodic states (i.e., the transient part is left out) gives (rounded to one decimal)

$$M = \begin{bmatrix}
2 & 3 & 3 & - & - & - & - & - & - & - & - \\
1 & 4 & 4 & - & - & - & - & - & - & - & - \\
1 & 4 & 4 & - & - & - & - & - & - & - & - \\
- & - & - & 9.6 & 1 & 400.9 & 400.9 & 416.9 & 412.5 & 416.9 & 412.5 \\
- & - & - & 8.6 & 9.5 & 399.9 & 399.9 & 415.9 & 411.5 & 415.9 & 411.5 \\
- & - & - & 1508.6 & 1500 & 3.8 & 3.8 & 16 & 11.7 & 19.8 & 15.4 \\
- & - & - & 1508.6 & 1500 & 3.8 & 3.8 & 19.8 & 15.4 & 16 & 11.7 \\
- & - & - & 1502.1 & 1493.5 & 3 & 6.8 & 19 & 14.7 & 22.8 & 18.4 \\
- & - & - & 1509.6 & 1501 & 1 & 4.8 & 17 & 12.7 & 20.8 & 16.4 \\
- & - & - & 1502.1 & 1493.5 & 6.8 & 3 & 22.8 & 18.4 & 19 & 14.7 \\
- & - & - & 1509.6 & 1501 & 4.8 & 1 & 20.8 & 16.4 & 17 & 12.7
\end{bmatrix}, \qquad (1)$$

where "–" indicates that there is no path connecting the corresponding nodes. The mean first passage time matrix reveals via the dotted rectangles that the second ergodic class can be decomposed into two subclasses because the values of M have significantly larger values for paths between the subclasses compared with paths within the subclasses. In particular, inspection shows that there exists a weak connection between 4 and 5 on the one hand and nodes 6–11 on the other hand. This illustrates the potential of using M for community detection inside ergodic classes.

Applications of the decomposition of ergodic classes into appropriate subclasses are manifold. For example, in marketing, targeting the most influential agents is of importance. Indeed, community detection provides an interesting tool for identifying the most relevant agent in a subclass. As our social network example shows, agents 6 and 7 are most influential in the second ergodic class, which makes them natural candidates to be targeted by advertisement, but a closer look at the dormant structure of the second ergodic class shows that both agents belong to the same subclass, and targeting agents 6 and 4, the two most influential agents in the respective subclasses, will lead to much better information spread. Another type of application is network security, where a given network is divided into subclasses to avoid, for example, the spread of computer viruses. Recently, the controlled growth of networks has caught interest. In this line of research, a controller can either add or remove links between agents in order to maximize connectivity or maximize the number of strongly connected subgraphs (see, e.g., D'Souza and Nagler 2015).

Evaluating Π_P together with D_P allows for the computation of M. Moreover, the *Kemeny constant* of an ergodic class can be evaluated as well. The Kemeny constant of a Markov chain P consisting of one ergodic class with no transient states, denoted by K_P, is given by

$$K_P = \sum_{j \in \mathbb{S}} M(i,j)\pi_P(j), \quad \forall i \in \mathbb{S}, \tag{2}$$

where π_P is the unique stationary distribution of P. In words, K_P gives the expected number of steps until reaching a state that is randomly chosen according to the stationary distribution of the Markov chain. Somewhat surprisingly, K_P is a constant regardless of the initial state. An alternative perspective on the Kemeny constant is that it is a π_P-weighted average over the entries in M. Consequently, it places more emphasis on the relatively more relevant nodes in the network as measured by π_P. These interpretations of K_P motivate its use as a connectivity measure of a network: The smaller (larger) is K_P, the better (worse) is the connectivity of the network. For our social network example, it can be calculated that the Kemeny constant value in the second ergodic class equals 321.5. This means that it takes 321.5 transitions in expectation before reaching a randomly drawn destination with probability distribution according to Π_P in the second ergodic class. With only eight states in ergodic class 2, one may conclude that the connectivity inside this class is rather poor.

In Section 4, the Kemeny constant derivatives will be calculated by perturbing P in the direction of existing edges. In this way, the criticality of edges with respect to the connectivity as measured by K_P can be mapped. For the second ergodic class, it turns out that the smallest derivatives of the Kemeny constant are in the direction of edges $(8,5)$, $(10,5)$ (both value –6,574), $(5,6)$, and $(5,7)$ (both value −49,491). For example, a relative increase in transition probability $P(8,5)$ with some small $\delta > 0$ leads to a connectivity improvement of $6574 \cdot \delta$. For the second ergodic class of the social network example, cutting the four mentioned edges with the smallest Kemeny constant derivatives leads to the already-identified community decomposition of nodes 4 and 5 and nodes 6–11. If we normalize the submatrices on these communities to stochastic matrices, the Kemeny constants can be calculated to be equal to 1.5 and 6.2, respectively, a significant reduction compared with the original Kemeny constant of 321.5.

To summarize, from M it becomes apparent whether P has dormant subclasses or not. However, although $M(i,j)$ represents information over *all paths* from i to j, $M(i,j) < \infty$ does not imply that there exists a link from i to j and thus offers no direct means of assessing the impact of an individual link (i,j) on connectivity. The key idea of this paper is that the impact of an individual link can be expressed via the sensitivity of the Kemeny constant (a function of M) with respect to the link; details will be provided in Section 4.

3. Basic Results for Markov Multichains

A Markov chain with only one closed irreducible set of states and a (possibly empty) set of transient states is called a *Markov unichain* (in short, *unichain*). A closed irreducible set of states is also referred to as an *ergodic class*. For unichains, the unique stationary distribution of P, denoted as $\pi_P^\top$, can be found by solving $\pi_P^\top P = \pi_P^\top$. It holds for unichains that (1) the chain will eventually be trapped in the (unique) ergodic class, independent of the initial state, and (2) all rows of ergodic projector Π_P equal $\pi_P^\top$. Note that if P is aperiodic, the stationary distribution is the unique limiting distribution of the unichain.

Markov multichains (in short, multichains) have multiple ergodic classes and a (possibly empty) set of transient states. Unlike unichains, the initial state has an impact on the long-run average number of visits in multichains, which stems from the fact that once the chain enters one of the several ergodic classes, it remains there forever. To study the long-run behavior of a multichain, one first has to uncover the ergodic classes and the transient states using, for example, the algorithm in Fox and Landi (1968). After possible relabeling of the states, the transition matrix and the ergodic projector can be written in the following canonical forms, respectively:

$$P = \begin{bmatrix} P_1 & 0 & 0 & \cdots & 0 \\ 0 & P_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & P_E & 0 \\ P_{T1} & P_{T2} & \cdots & P_{TE} & P_{TT} \end{bmatrix} \quad \text{and} \tag{3}$$

$$\Pi_P = \begin{bmatrix} \Pi_1 & 0 & 0 & \cdots & 0 \\ 0 & \Pi_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \Pi_E & 0 \\ R_1 & R_2 & \cdots & R_E & 0 \end{bmatrix},$$

where E denotes the number of ergodic classes, T the possible empty set of transient states, and Π_i the square matrix of which all rows equal the unique stationary distribution of the chain with transition matrix P_i inside ergodic class i. Furthermore, $R_i(j,k)$ is the expected long-run number of visits of the chain to state k in ergodic class i when started in transient state j. For more details, see Berkhout and Heidergott (2019) and Kemeny and Snell (1976).

Note that whether a state i is ergodic or transient can be decided from $\Pi_P(i,i)$ because the diagonal values of Π_P are zero if and only if i is a transient state. This has been called *diagonal criterion* in Berkhout and Heidergott (2019) and allows for the identification of ergodic classes and transient states through Π_P or an approximation thereof.

The deviation matrix $D_P = (I - P + \Pi_P)^{-1} - \Pi_P$, the existence of which is proven in Puterman (1994), plays a key role in many applications, including Markov decision processes (Puterman 1994, Koole and Spieksma 2001), simulation (Glynn 1984, Whitt 1992), perturbation theory (Schweitzer 1968, Hunter 2005), first-passage time analysis (Meyer 1975, Hunter 1982), control of road networks (Crisostomi et al. 2011), series expansions of Markov chains (Heidergott et al. 2007, 2010), ranking methodologies (Berkhout 2016, Berkhout and Heidergott 2017), and speed of convergence to stationarity (Coolen-Schrijner and van Doorn 2002). Via a controlled Markov decision process, the deviation matrix even plays an important role in the connection between Markov chains and the Hamiltonian cycle problem (Litvak and Ejov 2009, Borkar et al. 2012). Existence of D_P is guaranteed for finite-state aperiodic Markov chains; see Heidergott et al. (2007) for a proof. The deviation matrix is closely related to the so-called fundamental matrix (Kemeny and Snell 1976), and in fact, the fundamental matrix equals $D_P + \Pi_P$. As observed in Meyer (1975), wherever the fundamental matrix appears, it can be directly replaced by D_P. Furthermore, denoting the identity matrix of appropriate size with I, the deviation matrix is known as the *group inverse for* $I - P$ in the literature (Meyer 1975, 1982).

The deviation matrix is only explicitly known in a few special cases. For example, Koole and Spieksma (2001) provide an explicit expression for the deviation matrix of the $M/M/s/N$ and $M/M/s/\infty$ queues. In Abbas et al. (2016), the deviation matrix is computed for Markov influence graphs having a simple topology, such as a ring or star-shaped topology.

Elaborating on the canonical form of Π_P in (3), the deviation matrix can be written as

$$D_P = \begin{bmatrix} D_{P_1} & 0 & 0 & \cdots & 0 \\ 0 & D_{P_2} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & D_{P_E} & 0 \\ D_{P_{T1}} & D_{P_{T2}} & \cdots & D_{P_{TE}} & D_{P_{TT}} \end{bmatrix}, \tag{4}$$

where

$$D_{P_i} = (I - P_i + \Pi_{P_i})^{-1} - \Pi_{P_i}, \quad i = 1, 2, \ldots, E,$$

and

$$D_{P_{TT}} = (I - P_{TT})^{-1},$$

because $\Pi_{P_{TT}} = 0$. Moreover, for $i = 1, 2, \ldots, E$, it holds that

$$D_{P_{Ti}} = (I - P_{TT})^{-1} \cdot (P_{Ti} - R_i) \cdot (I - P_i + \Pi_{P_i})^{-1} - R_i.$$

An equivalent representation of the Kemeny constant $K_P = \sum_{i=1}^{n} M(i,j)\pi_P(j)$ [introduced in (2)] is

$$K_P = \mathrm{tr}(D_P) + 1, \tag{5}$$

where $\mathrm{tr}(A)$ indicates the trace of a matrix A, which is the sum of all diagonal elements; see also Kirkland (2010). The advantage of (5) over (2) is that the deviation matrix is well defined for Markov chains with several ergodic classes and transient states, whereas M can be only meaningfully defined for a single ergodic class. The Kemeny constant can be extended to Markov multichains as follows. From (5) and (4), it follows that

$$\begin{aligned} K_P &= \mathrm{tr}(D_P) + 1 \\ &= \sum_{i=1}^{E} \mathrm{tr}(D_{P_i}) + \mathrm{tr}(D_{P_{TT}}) + 1 \\ &= \sum_{i=1}^{E} K_{P_i} + \mathrm{tr}(D_{P_{TT}}) - E + 1. \end{aligned}$$

Denoting the set of states from ergodic class i by E_i, one can replace K_{P_i} with

$$K_{P_i} = \sum_{j \in E_i} M(k,j)\pi_{P_i}(j), \quad \forall k \in E_i.$$

In words, a Markov multichain Kemeny constant is equivalent to the sum of (1) the Kemeny constants per ergodic class i, (2) the total sum for $D_{P_{TT}}(j,j)$ over all transient states j, and (3) a correction factor for the number of ergodic classes $E-1$.

The trace of $D_{P_{TT}}$ occurs in K_P because $D_{P_{TT}}$ has the following probabilistic interpretation: $D_{P_{TT}}(i,j)$ is the expected total number of times that the process is in state j when starting in state i; for more details, see Kemeny and Snell (1976). Hence, the trace provides an indication of the stay duration inside the transient part and therefore provides another interpretation of K_P as a connectivity measure, this time regarding transient states. It holds for the social network example from Section 2 that

$$D_{P_{TT}} = \begin{bmatrix} 1.028 & 0.342 \\ 0.085 & 1.028 \end{bmatrix};$$

for example, (transient) state 13 is visited 0.342 times in expectation when starting in (transient) state 12. This shows that the transient part is relatively strongly connected to the ergodic classes. However, when P_{TT} is chosen such that its row sums are close to one, values in $D_{P_{TT}}$ would be significantly larger, indicating that the transient part is only weakly connected to the ergodic classes. Also, when transient subcomponents are weakly connected, relabeling will identify a similar pattern of large values in $D_{P_{TT}}$ as seen in (1) for M.

The Kemeny constant can also be written as a mapping of the eigenvalues of P (see Kirkland 2010) or as a function involving the square root and the inverse square root of Π_P (see Agharkar et al. 2014). For more background on the Kemeny constant, see Catral et al. (2010) and Hunter (2014). For an interpretation in terms of a random surfer on the internet who is disorientated, we refer to the *navigation problem* in Levene and Loizou (2002). Matrices that obtain minimal values for K_P are characterized in Kirkland (2010, 2014).

A Markov chain P is called *nearly decomposable* if P is irreducible and, after possible relabeling of states, can be written in the form

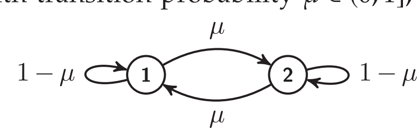
$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1k} \\ P_{21} & P_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & P_{(k-1)k} \\ P_{k1} & \cdots & P_{k(k-1)} & P_{kk} \end{bmatrix},$$

where the diagonal blocks P_{ii}, $i = 1, 2, \ldots, k$, are square and have rows that sum up to $1-\epsilon$, with $\epsilon > 0$ small (see, e.g., Stewart 1994, Meyer 1989). The multichain and nearly decomposability characteristic are not mutually exclusive. Indeed, a Markov chain may be a multichain with transient states and may have an ergodic class (or a transient part) that in itself constitutes a nearly decomposable chain.

4. Decomposition of Markov Chains

In this section, a decomposition algorithm for Markov chains is developed called the *Kemeny decomposition algorithm* (KDA). In the following example, the relation between the Kemeny constant and connectivity is illustrated.

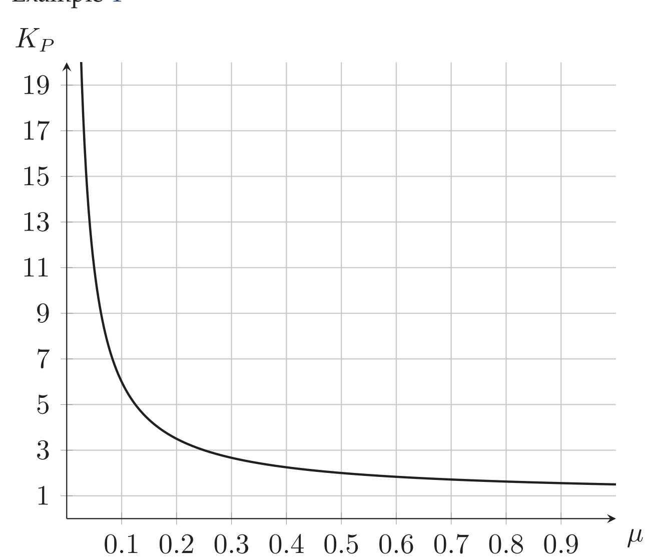
Example 1. Consider the following simple network with transition probability $\mu \in (0,1]$, that is,



$$P = \begin{bmatrix} 1-\mu & \mu \\ \mu & 1-\mu \end{bmatrix}.$$

In Figure 2, the Kemeny constant is plotted as a mapping of μ for $\mu \in (0,1]$. It shows that when μ is close to 0, the Kemeny constant becomes arbitrarily large, as path lengths from 1 to 2 and vice versa tend to infinity, representing poor connectivity—that is, the network becomes nearly decomposable into two almost isolated states. On the contrary, for $\mu = 1$, the smallest value of the Kemeny constant is obtained corresponding to best connectivity.

Figure 2. Plot of $K_P = 1 + \frac{1}{2\mu}$ for $\mu \in (0,1]$ with P as in Example 1



Note. In case $\mu = 0$, $K_P = 0$.

Because the Kemeny constant reflects connectedness of a network, it has a wide range of applications. For example, in road networks, small values of K_P indicate small average travel times (see Crisostomi et al. 2011). In Agharkar et al. (2014), K_P is the basis for stochastic surveillance strategies for quickest detection of anomalies in networks. Lastly, the Kemeny constant might provide insight in the information dissemination throughout a social network.

In Section 4.1, the derivative of the Kemeny constant with respect to (w.r.t.) the transition matrix in a convex combination model is computed. This derivative of the Kemeny constant is the main tool of analysis used in Section 4.2 for developing the KDA.

4.1. Graph Derivatives of the Kemeny Constant

In this section, we introduce a model that allows us to scale the weight of edges in P. With this model, we can define the directed derivative of the Kemeny constant, which gives insight into the effect of increasing the transition probability from i to j in terms of network connectedness as measured by the Kemeny constant.

For each edge $(i, j) \in \mathscr{E}$, we introduce a Markov chain with transition matrix

$$R_{ij} = P - e_i e_i^\top P + e_i e_j^\top = P - e_i(e_i^\top P - e_j^\top),$$

where e_i indicates the ith column of the identity matrix. In words, R_{ij} equals P but with the ith row replaced by the jth row of the identity matrix. We now perturb P by R_{ij} and consider the mixed chain $P_{ij}(\theta)$, where

$$P_{ij}(\theta) = (1-\theta)P + \theta R_{ij}, \quad \text{for } \theta \in (0, 1).$$

In words, for $\theta > 0$, the chain $P_{ij}(\theta)$ for $\theta > 0$ takes mass away from the transition probabilities from i to the neighboring states and shifts this mass to the transition probability from state i to state j. The following theorem provides an expression for the derivative of $K(P_{ij}(\theta))$ with respect to θ at $\theta = 0$. The proof of the theorem is provided in Section EC.2 of the online appendix.

Theorem 1. *For each edge $(i, j) \in \mathscr{E}$, it holds that*

$$\frac{d}{d\theta} K_{P_{ij}(\theta)} = (D_{P_{ij}(\theta)})^2(j, i) - (P(D_{P_{ij}(\theta)})^2)(i, i),$$

where e_i is the ith column of the identity matrix. In matrix form, at $\theta = 0$,

$$\left(\frac{d}{d\theta} K_{P_{ij}(\theta)}\bigg|_{\theta=0}\right)_{(i,j)\in\mathscr{E}} = ((D_P)^2)^\top - dg(P(D_P)^2)\bar{1}\bar{1}^\top,$$

where $dg(A)$ denotes matrix A with the off-diagonal entries replaced by zeros, and $\bar{1}$ is a vector of ones.

In words, $\frac{d}{d\theta} K_{P_{ij}(\theta)}$ gives insight in the network connectedness as measured by K_P when the transition probability of an edge (i, j) relatively increases. Put differently, $\frac{d}{d\theta} K_{P_{ij}(\theta)}$ measures how important existing edge (i, j) is for the connectedness of the network. This insight will be made fruitful in the Kemeny decomposition algorithm to be introduced below.

4.2. Decomposition of Markov Chains

By using the concept of the Kemeny constant derivatives, our KDA will be presented in the following. KDA relies on the following related hypotheses.

Hypothesis 1. *$K(P)$ is a suitable connectivity measure of a Markov influence graph described by P.*

Hypothesis 2. *An edge $(i, j) \in \mathscr{E}$ with a small negative value for $\frac{d}{d\theta} K(P_{ij}(\theta))\big|_{\theta=0}$ implies that increasing the transition probability from i to j has a large positive impact on connectivity. Vice versa, setting $P(i, j)$ to zero, in other words cutting edge $(i, j) \in \mathscr{E}$, and normalizing the row afterward uncouples the network most significantly.*

The idea of the decomposition algorithm is to continue cutting edges until some stopping criterion holds—for example, a predefined number of ergodic classes or strongly connected components is reached. After each cut, the network dynamics will change, yielding a new Markov chain $\tilde{P}(\theta)$ that lives only on the remaining edges. Then $\frac{d}{d\theta} K\left(\tilde{P}_{ij}(\theta)\right)\Big|_{\theta=0}$ has to be computed for the adjusted Markov chain for all remaining edges (i, j). This iteration is repeated until the stopping criterion is met. Although, intuitively, it might be worthwhile to iteratively calculate $\frac{d}{d\theta} K(P_{ij}(\theta))\big|_{\theta=0}$ after each cut in order to obtain a decomposition, this may be computationally too costly for large Markov chains. In that case, one can rely on the following experimentally verified hypothesis.

Hypothesis 3. *Cutting edges in ascending order w.r.t. $\frac{d}{d\theta} K(P_{ij}(\theta))\big|_{\theta=0}$, for all $(i, j) \in \mathscr{E}$—that is, without recomputing the Markov chain—yields a natural decomposition of the network.*

The intuition behind this observation is that the relative criticality of edges w.r.t. connectivity will not alter too much after cutting multiple edges in general. Therefore, cutting multiple edges at once based on values $\frac{d}{d\theta} K(P_{ij}(\theta))\big|_{\theta=0}$ will still lead to a reasonable decomposition.

KDA is a general approach that allows a tradeoff between iteratively calculating $\frac{d}{d\theta} K(P_{ij}(\theta))\big|_{\theta=0}$ and

Table 1. Possibilities for Conditions CO_A and CO_B in the KDA

Condition	Label	Specification
CO_A	CO_A_1(i)	Number of times performed $< i$
	CO_A_2(E)	Number of ergodic classes in P^c is $< E$
CO_B	CO_B_1(e)	Number of edges cut is $< e$
	CO_B_2(E)	Number of ergodic classes in P^c is $< E$
	CO_B_3(q)	Not all edges with $\frac{d}{d\theta} K(P^c_{ij}(\theta)) < q$ are cut

computational intensity. The pseudocode for KDA is presented below for a given Markov influence graph with transition matrix P. It leads to Markov influence graphs with transition matrix P^c.

Kemeny Decomposition Algorithm

FUNCTION KDA(P, CO_A, CO_B, SC):

INPUT:

P = Markov chain transition matrix
CO_A = condition A
CO_B = condition B
SC = True, when edges are symmetrically cut (SC), False otherwise.

START:

- Initialize cut transition matrix $P^c = P$, and set $\mathbb{E} = \mathscr{E}$.
- While CO_A: (*Derivative updating loop*)
  - For all $(i,j) \in \mathbb{E}$, calculate $\frac{d}{d\theta}K(P^c_{ij}(\theta))\big|_{\theta=0}$.
  - While CO_B: (*Edge cutting loop*)
    - ▹ Determine $(i^\star, j^\star) = \arg\min_{(i,j)\in\mathbb{E}} \frac{d}{d\theta}K(P^c_{ij}(\theta))\big|_{\theta=0}$.
    - ▹ Set $P^c(i^\star, j^\star) = 0$ and normalize the $i^\star$th row of P^c.
    - ▹ Set $\mathbb{E} = \mathbb{E} \setminus \{(i^\star, j^\star)\}$.
    - ▹ If SC = True: (*Symmetric network*)
      - ⋆ Set $P^c(j^\star, i^\star) = 0$ and normalize the $j^\star$th row of P^c.
      - ⋆ Set $\mathbb{E} = \mathbb{E} \setminus \{(j^\star, i^\star)\}$.
  - End While
- End While

OUTPUT:

P^c = Decomposed Markov chain transition matrix

The goal of the double while loop in KDA is to allow for the described tradeoff between computational intensity and accuracy of the decomposition algorithm. Possibilities for conditions A and B in the KDA—that is, CO_A and CO_B—are given (and labeled) in Table 1.

Remark 1. Instead of the number of ergodic classes, one might consider the number of strongly connected components. This allows decomposing a transient part into several strongly connected components. It is a topic of further research whether this can be fruitful in applications.

Some general recommendations regarding the input of the KDA in case of different instances will be given in the following. The choice for SC depends on whether symmetrically cutting makes sense for the instance at hand. For example, in case the transition probabilities are based on distances, SC = TRUE might be suitable and will ensure faster convergence for the KDA. The data-clustering examples are examples of distance-based transition probabilities. Another example for which SC = TRUE is appropriate is the social network of Zachary's karate club data below. In case a graph is clearly asymmetric regarding the directed weights—for example, the Courtois matrix later on—SC = FALSE is more appropriate. For the rest of the input, the following recommendations can be given per instance type given by P:

- Large unknown instance: KDA(P, CO_A = CO_A_1(1), CO_B = CO_B_3(0), SC).
- Large instance where the natural/required number of ergodic classes is given by E: KDA(P, CO_A = CO_A_1(1), CO_A_2(E), SC).
- Medium/small unknown instance: iteratively perform KDA(P, CO_A = CO_A_1(1), CO_B = CO_B_1(1), SC) until satisfied. Alternatively: KDA(P, CO_A = CO_A_1(1), CO_B = CO_B_3(0), SC).
- Medium/small where the natural/required number ergodic classes is given by E: KDA(P, CO_A_2(E), CO_B = CO_B_1(1), SC)

5. Applications

In this section, the application of our KDA introduced in the preceding section is illustrated by a series of numerical examples. Four examples are considered: the Courtois matrix from Stewart (1994), a social network from Zachary (1977), two weakly connected social networks, and an application to data clustering including a comparison with different common clustering techniques.

5.1. Nearly Decomposable Markov Chains

The Courtois matrix is a notorious transition matrix that is often used in the context of nearly decomposability (see Stewart 1994). It is given by

$$P = \begin{bmatrix} .85 & 0 & .149 & .0009 & 0 & .00005 & 0 & .00005 \\ .1 & .65 & .249 & 0 & .0009 & .00005 & 0 & .00005 \\ .1 & .8 & .0996 & .0003 & 0 & 0 & .0001 & 0 \\ 0 & .0004 & 0 & .7 & .2995 & 0 & .0001 & 0 \\ .0005 & 0 & .0004 & .399 & .6 & .0001 & 0 & 0 \\ 0 & .00005 & 0 & 0 & .00005 & .6 & .2499 & .15 \\ .00003 & 0 & .00003 & .00004 & 0 & .1 & .8 & .0999 \\ 0 & .00005 & 0 & 0 & .00005 & .1999 & .25 & .55 \end{bmatrix},$$

with stationary distribution

$$\pi_P^\top = \left[0.089\ 0.093\ 0.04\ 0.159\ 0.119\ 0.12\ 0.278\ 0.102\right].$$

Applying KDA(P, CO_A_2(2), CO_B_1(1), FALSE)—that is, updating the Kemeny constant derivatives after each cut and stopping when two ergodic classes are found—cuts the edges in the following order before termination:

$$(7,3),(7,1),(7,4),(4,7),(5,6),(6,5),(6,2),(8,5),$$
$$(8,2),(2,6),(2,8),(1,8),(1,6),(3,7),$$

which leads to

$$\Pi_{P^c} = \begin{bmatrix} .175 & .182 & .08 & .322 & .241 & 0 & 0 & 0 \\ .175 & .182 & .08 & .322 & .241 & 0 & 0 & 0 \\ .175 & .182 & .08 & .322 & .241 & 0 & 0 & 0 \\ .175 & .182 & .08 & .322 & .241 & 0 & 0 & 0 \\ .175 & .182 & .08 & .322 & .241 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & .241 & .556 & .204 \\ 0 & 0 & 0 & 0 & 0 & .241 & .556 & .204 \\ 0 & 0 & 0 & 0 & 0 & .241 & .556 & .204 \end{bmatrix}.$$

After the first 13 edges are cut (all edges in the list above except for $(3,7)$), the graph as shown in Figure 3(a) can be drawn, where the ergodic class is given by circles and the transient states by rectangles. Afterward, edge $(3,7)$ is cut, leading to Figure 3(b). This illustrates that the KDA is also suitable when transient states are present. Furthermore, it is worth noting that KDA(P, CO_A_1(1), CO_B_2(2), FALSE) —that is, evaluating the Kemeny constant derivatives once and cutting until two ergodic classes arise—cuts exactly the same edges (only three edge-pairs in a different order). This numerically verifies Hypothesis 3 from the preceding section.

Further cutting—that is, choosing CO_A_2($E = 3$) instead of choosing CO_A_2($E = 2$) in the KDA—cuts additionally

$$(4,2),(5,3),(5,1),(2,5),(1,4),(3,4),$$

leading to Figure 3(c) and

$$\Pi_{P^c} = \begin{bmatrix} .402 & .417 & .182 & 0 & 0 & 0 & 0 & 0 \\ .402 & .417 & .182 & 0 & 0 & 0 & 0 & 0 \\ .402 & .417 & .182 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .571 & .429 & 0 & 0 & 0 \\ 0 & 0 & 0 & .571 & .429 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & .241 & .556 & .204 \\ 0 & 0 & 0 & 0 & 0 & .241 & .556 & .204 \\ 0 & 0 & 0 & 0 & 0 & .241 & .556 & .204 \end{bmatrix}.$$

Again, Hypothesis 3 can be verified because KDA(P, CO_A_1(1), CO_B_2(3), FALSE) cuts the same edges. Normalizing $\pi_P^\top$ on the states of the found ergodic classes in case of $E = 2$ and $E = 3$ illustrates that the long-term dynamics inside the classes are not significantly affected by the cutting algorithm. It is worth noting that the algorithm only cuts edges *between* the eventually found ergodic classes. Moreover, not necessarily the smallest edges of P are cut first—for example, edge $P(4,7) = .0001$ was cut before $P(1,8) = .00005$. This shows that the decoupling algorithm takes the network dynamics into account instead of superficial indications. Lastly, it holds that KDA(P, CO_A_2(3), CO_B_1(1), FALSE) = KDA(P, CO_A_1(1), CO_B_3(0), FALSE).

In words, iteratively cutting edges until $E = 3$ ergodic classes arise (the natural number of ergodic classes in the Courtois matrix) leads to the same decomposition as cutting all edges with a negative Kemeny constant derivative at once. This suggests that the sign of the Kemeny constant derivative might be used as an indicator to determine the natural decomposition of a network.

5.2. Social Network in a Karate Club

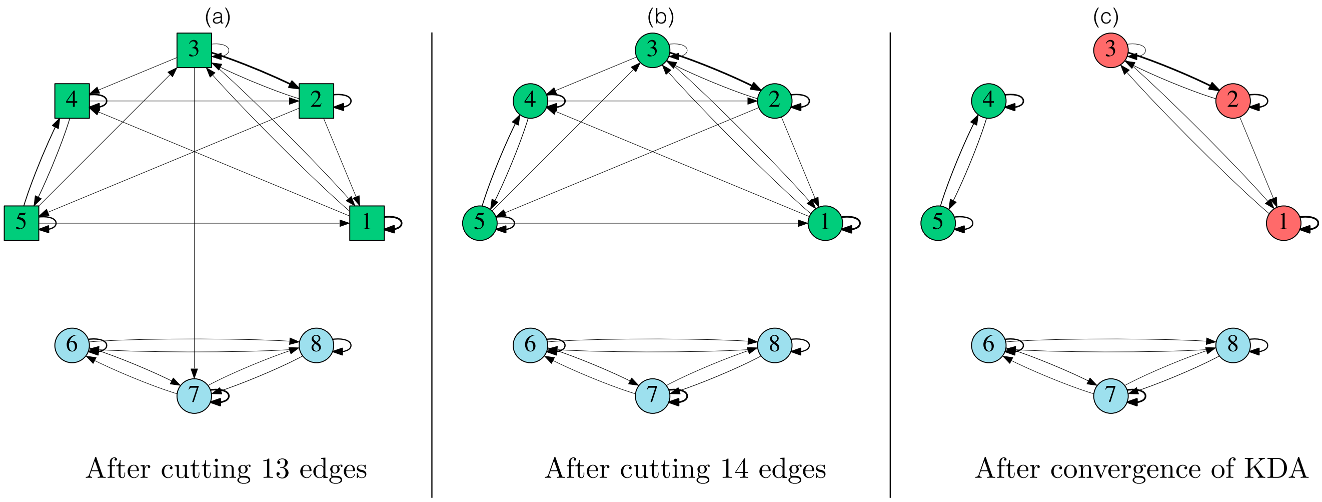
An interesting question is whether the KDA does indeed find a natural decomposition. To that end, data describing a social network from Zachary (1977) is considered. The paper provides a case study investigating fission in small groups: a university-based karate club is considered in which a factional division led to a formal separation of the club into two organizations. In other words, the "natural" decomposition of the social network consisting of the members of the karate club is known, a unique feature. Over time, the number of contact moments, such as joint training, participation in tournaments, etc., between the karate club members is counted. This leads to a positive symmetrically weighted adjacency matrix; see also figure 3 in Zachary (1977). For the following experiments, this adjacency matrix is normalized, so all rows sum up to one, to obtain a transition matrix P.

Figure 4 shows the results of KDA(P, CO_A_1(1), CO_B_3(0), FALSE)—that is, nonsymmetrically cutting all edges for which the Kemeny constant derivatives of the original P are negative. Inspecting the weakly connected components, we see that the KDA correctly identifies the factions after the club fission. The individual club member represented by node 9 indeed belonged more strongly to the faction represented by the nodes with black font. However, he joined the faction with white font to obtain his black belt more easily. See also the discussion in Zachary (1977). Again, this is an indication that the sign of the Kemeny constant derivatives is a suitable measure to determine the natural decomposition. Results for KDA(P, CO_A_2(E), CO_B_1(1), TRUE), for $E = 2$ and $E = 3$, can be found in Section EC.3.1 of the online appendix. This leads to comparable conclusions as for the experiment in Figure 4.

5.3. Decomposition of Social Networks

We now perform a similar but larger experiment in the context of an abstract social network. In particular, we set up two social networks: a preferential attachment (PA) network of 500 states as described in Barabási and Albert (1999) and a 400-state Kleinberg (Kl) network

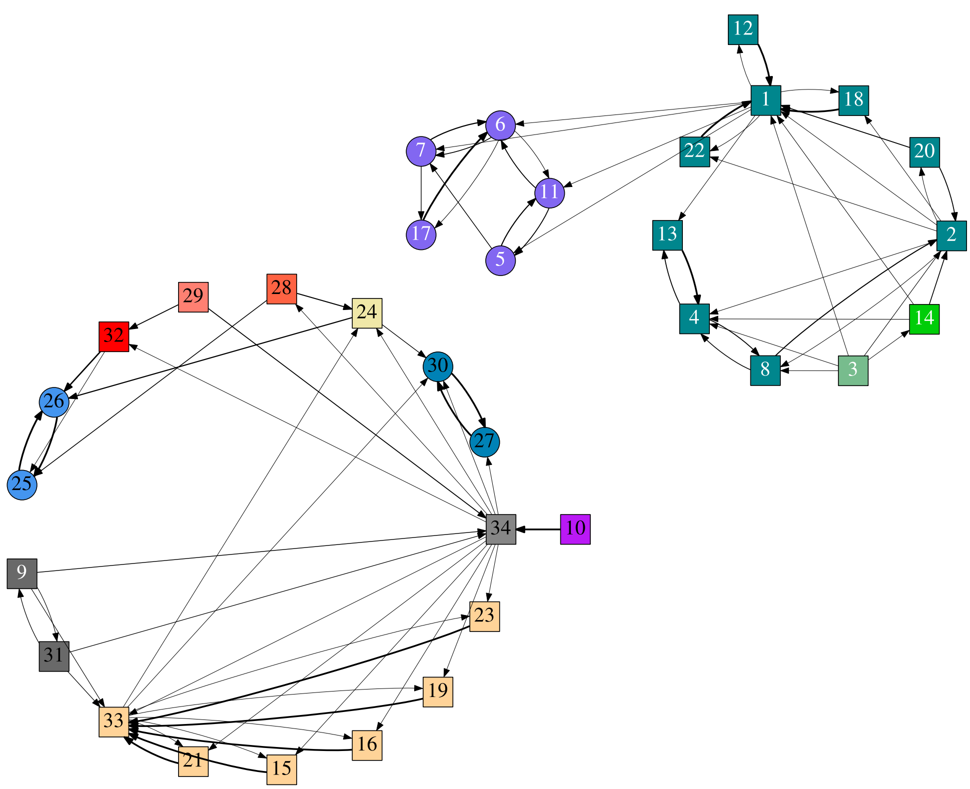
Figure 3. Convergence Progress KDA(P, CO_A_2(3), CO_B_1(1), FALSE)



Notes. Ergodic states are represented by circles and transient states by squares. The colors correspond with strongly connected components.

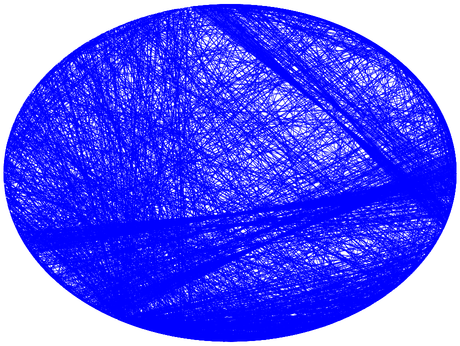
from Kleinberg (2000). The CONTEST Toolbox from Taylor and Higham (2009) was used for computation, where we used the default degree parameter of 2 for the PA network, and for the Kl network we used a distance parameter of 1, two large range connections per state, and exponent parameter 1.5. Then we randomly add 400 directed edges in the adjacency matrix to connect the two social subnetworks: 200 edges from the PA network to the Kl network and 200 edges vice versa. The randomly added edges are assigned weight 0.02 instead of 1, and the adjacency matrix is afterward normalized to obtain P. The objective, then, is to identify the original PA and Kl networks in the strongly connected social network by using our KDA.

Figure 4. KDA(P, CO_A_1(1), CO_B_3(0), FALSE)

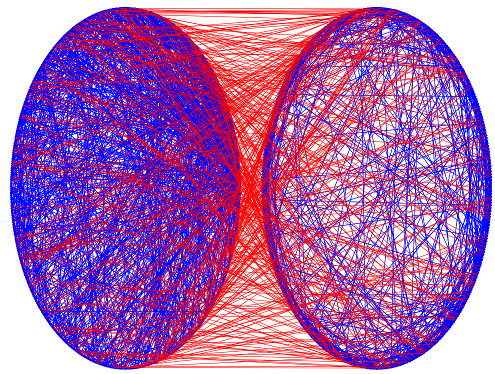


Note. Node colors correspond to strongly connected components, and its font colors (black and white) indicate the two formed factions after the fission.

Figure 5. Results for KDA(P, 10^{-7}, CO_A_1(1), CO_B_2(2), False) Where P Describes a Social Network of 900 Nodes Consisting of the Weakly Connected PA and Kl Subnetworks



Plot of original network described by P.



Effect of KDA after restructuring:

red edges are cut, blue edges remain.

Figure 5 presents the results for KDA(P, CO_A_1(1), CO_B_2(2), False). On the left side, the 900-state network is plotted as described by the original P, and the right sides gives a version of the cut network P^c where nodes have been arranged in such a way that the original PA and Kl networks emerge, with the cut edges indicated in red.

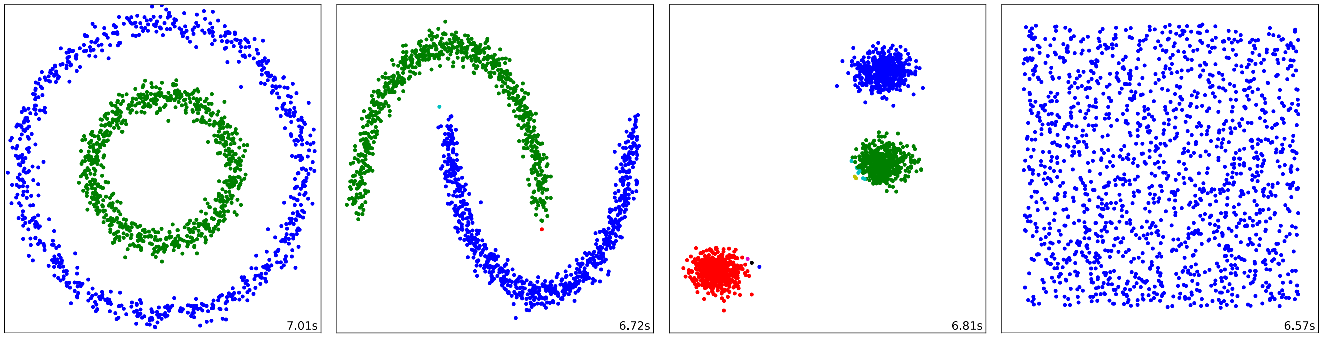
As can be seen from the results, the decomposition algorithm successfully identifies the two subnetworks without cutting many edges inside the remaining subnetworks. By varying parameters in this example, we observed that there are different aspects that determine whether the decomposition algorithm uncovers the PA and Kl subnetworks (i.e., whether the decomposition algorithm leads to a natural uncoupling). These aspects include the number of randomly added edges (fewer edges means that the subnetworks come forward more easily), the probability of the added edges (smaller probabilities stimulate detection of added edges by the algorithm), the connectivity degree inside subnetworks, and the location of randomly added edges (if key players are connected from multiple subnetworks, the connectivity between the subnetworks could be large and consequently may complicate original subnetwork detection).

5.4. Data Clustering

A natural application of the KDA is to the clustering technique used for exploratory data analysis. More specifically, consider n data vectors $x_1, x_2, \ldots, x_n$ of arbitrary dimension. The main objective in data clustering is to divide the data into C clusters, where C is given. In applied data clustering, the problem is that it is unusual for the user to know the value of C beforehand. A related question is that of the most natural choice of C.

Before applying the KDA, the transition matrix P has to be set up according to the data. This is done by constructing the so-called similarity matrix, whose goal is to model the local neighborhood relationships between the data vectors. In particular, a similarity matrix S is constructed such that its (i, j)th element gives the similarity between data points x_i and x_j. The similarity matrix S afterward can be normalized to obtain the transition probabilities of a random walk induced over the data points; mathematically, $P = (\mathrm{dg}(S\bar{1}))^{-1}S$, where $\mathrm{dg}(S\bar{1})$

Figure 6. KDA(P, CO_A_1(1), CO_B_3(0), True) Applied to Four Different Data Sets ($\delta = 6.5$)



is a zeros matrix of appropriate size where the diagonal is replaced by vector $S\bar{1}$. Transition matrix P is then used in the KDA.

There are several popular constructions to transform a given set $x_1, x_2, \ldots, x_n$ of data points into a similarity matrix. For an overview and discussion of common similarity matrices used in spectral clustering, see section 2.2 in Luxburg (2006). For the following experiments, the so-called Gaussian similarity function is used to construct S, which is a common approach in the literature, according to Meyer et al. (2013). In particular, an $n \times n$ symmetric similarity matrix S is considered with elements

$$S(i,j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2/\delta}\right), \quad \text{for all } i,j = 1,2,\ldots,n,$$

where

$$\delta > 0, \quad \sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}\|x_i - \mu\|^2 \quad \text{and} \quad \mu = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

Here $\|\cdot\|$ represents the Euclidean norm, and $\delta > 0$ is a user-defined scale that controls the width of the data vector neighborhoods determined by σ^2/δ. Larger values of δ mean smaller neighborhoods, and vice versa.

In Figure 6, we apply the KDA to four generated data sets consisting of 1,500 data points each. The natural clusters in the three data sets on the left-hand side are visible to the eye, and the data set on the right-hand side has no cluster because the data are generated in a uniform manner. For an appropriate choice of δ, the KDA identifies the present clusters without having to specify the number of clusters beforehand. Figure 6 shows the results for KDA(P, CO_A_1(1), CO_B_3(0), TRUE) with P corresponding to the four data sets, respectively, and $\delta = 6.5$. In words, the KDA cuts all edges with a negative Kemeny constant derivative and returns the decomposition.

It is worth noting that the KDA identifies two extra clusters consisting of one node each between the "moons" from the second data set and some small clusters between the two lowest bulbs in the third data set. This seems to be a specific feature for this KDA setting. Other settings and other similarity matrices most likely lead to different results.

The data sets are taken from the Scikit-learn (2017) web page, which is part of scikit-learn from Pedregosa et al. (2011), a machine learning package written in the programming language Python. On the Scikit-learn (2017) web page, eight common clustering techniques are plotted, and we compared the KDA with these data clustering techniques; the KDA shows similar results as DBSCAN from Ester et al. (1996) for the considered data sets and settings. For details, we refer to Section EC.3.2 in the online appendix.

6. Conclusion

This paper introduces the KDA, which allows one to decompose Markov influence graphs to uncover the true underlying network dynamics. The KDA is based on a structural perturbation of the Kemeny constant that can be computed via the fundamental deviation matrix from Markov chain theory. Relying on classical Markov chain concepts, the KDA succeeds in uncovering the true underlying network dynamics in a wide range of applications. By taking all possible weighted paths into account, the KDA accurately measures the criticality of existing edges. We therefore believe that it provides a new, promising way of decomposing/clustering networks in an increasingly data-driven society. Future research includes the possible application of the sign switch of the Kemeny constant derivative as a (network-based) natural KDA stopping criterion and elaboration of the KDA ideas in an algorithmic framework suitable for really large real-life networks for which a matrix inversion is impracticable.

Endnote

[1] Some authors use the terminology nearly completely decomposable or nearly completely reducible.

References

Abbas K, Berkhout J, Heidergott BF (2016) A critical account of perturbation analysis of Markov chains. *Markov Processes Related Fields*. 22(2):227–266.

Agharkar P, Patel R, Bullo F (2014) Robotic surveillance and Markov chains with minimal first passage time. *53rd IEEE Conf. Decision Control* (IEEE, Piscataway, NJ), 6603–6608.

Barabási A, Albert R (1999) Emergence of scaling in random networks. *Science* 286(5439):509–512.

Berkhout J (2016) Google's PageRank algorithm for ranking nodes in general networks. *Proc. 13th Internat. Workshop Discrete Event Systems (WODES 2016)* (IEEE, Piscataway, NJ), 153–158.

Berkhout J, Heidergott BF (2019) The jump start power method: A new approach for computing the ergodic projector of a finite Markov chain. *J. Sci. Comput.* 78(3):1691-1723.

Berkhout J, Heidergott BF (2017) Ranking nodes in general networks: A Markov multi-chain approach. *Discrete Event Dyn. Systems* 28(1):3–33.

Borkar VS, Ejov V, Filar JA, Nguyen GT (2012) *Hamiltonian Cycle Problem and Markov Chains* (Springer-Verlag, New York).

Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. *Comput. Networks ISDN Systems* 30(1-7): 107–117.

Catral M, Kirkland S, Neumann M, Sze NS (2010) The Kemeny constant for finite homogeneous ergodic Markov chains. *J. Sci. Comput.* 45(1-3):151–166.

Coolen-Schrijner P, van Doorn EA (2002) The deviation matrix of a continuous-time Markov chain. *Probab. Engrg. Inform. Sci.* 16(3): 351–366.

Crisostomi E, Kirkland S, Shorten R (2011) A Google-like model of road network dynamics and its application to regulation and control. *Internat. J. Control*. 84(3):633–651.

DeGroot MH (1974) Reaching a consensus. *J. Amer. Statist. Assoc.* 69(345):118–121.

D'Souza RM, Nagler J (2015) Anomalous critical and supercritical phenomena in explosive percolation. *Nature Phys.* 11(7):531–538.

Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. *Proc. 2nd Internat. Conf. Knowledge Discovery Data Mining* (AAAI Press, Palo Alto, CA), 226–231.

Fox BL, Landi DM (1968) An algorithm for identifying the ergodic subchains and transient states of a stochastic matrix. *Comm. ACM*. 11(9):619–621.

Glynn PW (1984) Some asymptotic formulas for Markov chains with applications to simulation. *J. Statist. Comput. Simulation* 19(2):97–112,

Golub B, Jackson MO (2010) Naïve learning in social networks and the wisdom of crowds. *Amer. Econom. J. Microeconom.* 2(1):112–149.

Heidergott BF, Hordijk A, Leder N (2010) Series expansions for continuous-time Markov processes. *Oper. Res.* 58(3):756–767.

Heidergott BF, Hordijk A, van Uitert M (2007) Series expansions for finite-state Markov chains. *Probab. Engrg. Inform. Sci.* 21(3):381–400.

Hunter JJ (1982) Generalized inverses and their application to applied probability problems. *Linear Algebra Appl.* 45:157–198.

Hunter JJ (2005) Stationary distributions and mean first passage times of perturbed Markov chains. *Linear Algebra Appl.* 410:217–243,

Hunter JJ (2014) The role of Kemeny's constant in properties of Markov chains. *Comm. Statist. Theory Methods* 43(7):1309–1321.

Kemeny JG, Snell JL (1976) *Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix"* (Springer, New York).

Kirkland S (2010) Fastest expected time to mixing for a Markov chain on a directed graph. *Linear Algebra Appl.* 433(11–12):1988–1996.

Kirkland S (2014) On the Kemeny constant and stationary distribution vector for a Markov chain. *Electron. J. Linear Algebra* 27(1):354–372.

Kleinberg JM (2000) Navigation in a small world. *Nature* 406(6798):845.

Koole GM, Spieksma FM (2001) On deviation matrices for birth-death processes. *Probab. Engrg. Inform. Sci.* 15(2):239–258.

Langville AN, Meyer CD (2011) *Google's PageRank and Beyond: The Science of Search Engine Rankings* (Princeton University Press, Princeton, NJ).

Levene M, Loizou G (2002) Kemeny's constant and the random surfer. *Amer. Math. Monthly* 109(8):741–745.

Litvak N, Ejov V (2009) Markov chains and optimality of the Hamiltonian cycle. *Math. Oper. Res.* 34(1):71–82.

Luxburg UV (2006) A tutorial on spectral clustering a tutorial on spectral clustering. *Statist. Comput.* 17(4):395–416.

Mcpherson M, Smith-lovin L, Cook JM (2001) Birds of a feather: Homophily in social networks. *Annual Rev. Soc.* 27(1):415–444.

Meyer CD (1975) The role of the group generalized inverse in the theory of finite Markov chains. *SIAM Rev.* 17(3):443–464.

Meyer CD (1982) Analysis of finite Markov chains by group inversion techniques. Campbell SL, ed. *Recent Applications of Generalized Inverses, Research Notes in Math*, vol. 66 (Pitman, Boston), 50–81.

Meyer CD (1989) Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Rev.* 31(2):240–272.

Meyer CD, Race S, Valakuzhy K (2013) Determining the number of clusters via iterative consensus clustering. *Proc. 2013 SIAM Internat. Conf. Data Mining* (Society for Industrial and Applied Mathematics, Philadelphia), 94–102.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, et al. (2011) Scikit-learn machine learning in Python. *J. Machine Learn. Res.* 12:2825–2830.

Puterman ML (1994) *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (Wiley, New York).

Schweitzer PJ (1968) Perturbation theory and finite Markov chains. *J. Appl. Probab.* 5(2):401–413.

Scikit-learn (2017) Comparing different clustering algorithms on toy datasets. Retrieved May 9, 2017, http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html,

Stewart WJ (1994) *Introduction to the Numerical Solution of Markov Chains* (Princeton University Press, Princeton, NJ).

Taylor A, Higham DJ (2009) CONTEST. *ACM Trans. Math. Software* 35(4):1–17.

Whitt W (1992) Asymptotic formulas for Markov processes with applications to simulation. *Oper. Res.* 40(2):279–291.

Winkelmann A, Haselmann T (2010) Ein konzeptionelles model auf basis von Markov-ketten zur berechnung und bewertung von persönlichen kompetenzen in sozialen netzwerken. *Proc. Multikonferenz Wirtschaftsinformatik (MKWI)* (Leuphana Universität Lüneburg, Lüneburg, Germany), 1317–1328.

Zachary WW (1977) An information flow model for conflict and fission in small groups. *J. Anthropological Res.* 33(4):452–473.

Joost Berkhout is postdoctoral researcher at the Centrum Wiskunde & Informatica, which is the national research institute for mathematics and computer science in the Netherlands. He received his PhD from Vrije Universiteit Amsterdam in 2016. He is recipient of the Best Student Paper Award from 13th Workshop on Discrete Event Systems in 2016. His research interests are optimization of production scheduling in modern industrial plants, simulation optimization, and Markov chain theory with a focus toward applications (such as social networks analysis and Google's PageRank).

Bernd F. Heidergott is the professor of stochastic optimization at the Department of Econometrics and Operations Research at the Vrije Universiteit Amsterdam, Netherlands. He received his PhD from the University of Hamburg, Germany, in 1996, and held postdoc positions at various universities before joining the Vrije Universiteit. He is research fellow of the Tinbergen Institute and Amsterdam Business Research Institute. His research interests are optimization and control of discrete event systems, perturbation analysis, Markov chains, max-plus algebra, and social networks analysis.