Zuse Institute Berlin

Takustrasse 7
D-14195 Berlin-Dahlem
Germany

ANDREAS BITTRACHER[1] AND CHRISTOF SCHÜTTE[1,2]

[1] *Department of Mathematics and Computer Science, Freie Universität Berlin, Germany*

[2] *Zuse Institute Berlin, Germany*

A PROBABILISTIC ALGORITHM FOR AGGREGATING VASTLY UNDERSAMPLED LARGE MARKOV CHAINS

ZIB Report 20-21 (June 2020)

Zuse Institute Berlin
Takustrasse 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

A probabilistic algorithm for aggregating vastly undersampled large Markov chains

Andreas Bittracher[1] and Christof Schütte[1,2]

[1]Department of Mathematics and Computer Science, Freie Universität Berlin, Germany
[2]Zuse Institute Berlin, Germany

Abstract

Model reduction of large Markov chains is an essential step in a wide array of techniques for understanding complex systems and for efficiently learning structures from high-dimensional data. We present a novel aggregation algorithm for compressing such chains that exploits a specific low-rank structure in the transition matrix which, e.g., is present in metastable systems, among others. It enables the recovery of the aggregates from a vastly undersampled transition matrix which in practical applications may gain a speedup of several orders of magnitude over methods that require the full transition matrix. Moreover, we show that the new technique is robust under perturbation of the transition matrix. The practical applicability of the new method is demonstrated by identifying a reduced model for the large-scale traffic flow patterns from real-world taxi trip data.

1. Introduction

Large-scale time- and space-discrete Markov chains are ubiquitous in many areas of quantitative science, where they arise as discretizations of continuous models [41, 36, 37, 24], as formalization of network-based models [8, 40, 35], or as models of many other types of complex dynamics. However, the number of states in these Markov chains (denoted in the following by N) can be orders of magnitude larger than what contemporary computer systems can process, or sometimes even represent. Efficient model reduction methods are thus required to enable numerical analysis, prediction and control of these systems, and to gain understanding of the underlying mechanisms.

Luckily, the essential dynamics of these systems often indeed possesses an underlying less complex mechanism that operates on a significantly smaller state space—one may argue this is what makes the system relevant to study in the first place. For example, Markov chains arising from discretized biomolecular systems often exhibit *metastability*, the phenomenon that on long time scales, the dynamics is determined by rare jumps between almost-invariant subsets of states [17, 41, 43]. Another example are complex traffic networks, whose transition matrices often exhibit a low-rank structure, which can be explained by patterns in the large-scale traffic flow between neighborhoods of a city [31, 6].

The reduced models in these systems arise from the observation that states can be grouped into certain *aggregates* based on similarities in their dynamic behavior. Under certain conditions, the reduced models can be shown to again be Markovian, which is highly favorable due to their simplicity. The developments of data-driven algorithms for the extraction of the reduced Markov models is an area of intense activity ever since Markov chains were studied computationally, a small selection is presented in Section 2. The output of such algorithms is again a Markov chain, whose states now correspond to the aggregates.

The identification of these aggregates is typically based on the analysis of the original system's transition matrix (denoted by P in the following) and in most cases requires knowledge of all

entries of this matrix, i.e., global knowledge of the transition probabilities. As the number of entries of P is N^2, for large systems, analyzing or even storing P becomes nontrivial. An even bigger problem is that, for many systems, the entries of P must be approximated from expensive numerical computations. Typical examples are Markov chains arising from the Ulam discretization method for space-continuous systems, where the continuous state space is partitioned into N discretization elements that form the states of the Markov chain [49]. The transition probabilities are then computed by starting many numerical simulations in each discretization element and counting the transitions to each other element. As the number of states N may depend exponentially on the dimension of the continuous system (a phenomenon known as the *curse of dimensionality* [5]), and the simulation may require specialized hard- and software [44], this procedure quickly becomes prohibitively expensive. Many of the aforementioned methods acknowledge the difficulty of globally sampling the dynamics, and various solutions have been suggested, including adaptive sampling [9], accelerated dynamics [33, 30] or statistical reweighting methods [4, 11].

In this work however, we will take a different approach. Instead of improving the sampling of global data, we will instead use the existence of an underlying reduced Markov chain in order to show that extensive global sampling is not required in the first place. To be specific, we discuss two crucial properties that when combined guarantee the existence of a reduced Markov chain and make this chain recoverable from sparse, i.e., vastly incomplete dynamical information. Informally, the properties can be written as follows: (i) The probability to transition from any state to an aggregate must depend only on the aggregate of the starting state, and (ii) the probability to transition from some starting to some ending state must essentially depend only on the aggregate of the ending state. Consequently, measuring the transition probabilities from and to all states of one aggregate would mostly generate redundant information. In this case it suffices to measure the transition probabilities from only one state of each aggregate in order to capture the full dynamic behavior. Hence, the amount of dynamical information required to describe the full model depends only on the size of the reduced model, not of the full model.

The properties (i) and (ii), called *lumpability* and *deflatability*, induce a special form of low-rankedness in both the row and the column space of P. This low-rank structure is robust, in that small violations of the two properties cause P to still be close to a low-rank matrix, and the deviation is again independent of the full system's size. Also, lumpability and deflatability, which to the best of our knowledge have not been investigated together before, appear to be the minimal requirement for the described low-rank structure. While there exist a number of related concepts in the literature, we will see that none of them imply this structure in the same generality.

The main contribution of this work is the development of a probabilistic aggregation algorithm that exploits this low-rank structure. To be specific, the algorithm starts by randomly and sparsely sampling the column space of P in order to estimate the range of P. It therefore is similar in spirit to probabilistic low-rank approximation and matrix decomposition methods from randomized linear algebra [26, 32]. The number of required columns of P hereby depends only on the expected number of aggregates, as well as a certain "confidence parameter". In particular, both of these quantities are independent of the size N of the original model. The algorithm proceeds by computing the singular value decomposition (SVD) of the subsampled transition matrix. This reveals the aggregates, similar to how the SVD of the (full) transition matrix of a metastable system reveals the metastable sets [21]. Finally, even the reduced transition matrix can be computed from the subsampled transition matrix, using only elemental algebraic calculations.

In summary, due to its probabilistic nature, the algorithm is able to exploit the low-rank structure of the full transition matrix without detailed knowledge of it. This gives our method a computational advantage of several orders of magnitude over methods that require the full transition matrix. This advantage grows with the size of the full Markov chain, as long as the size of the underlying reduced Markov chain remains constant.

The paper is organized as follows: Section 2 introduces the requirements of aggregatable Markov chains and discusses the resulting low-rank structure. Section 3 contains the derivation of the low-rank algorithm, with the method to identify the aggregates in Section 3.1, and the method to compute the reduced transition matrix in Section 3.2. In Section 4 the algorithm is demonstrated by three numerical examples. These include a generic, randomly-generated aggregatable Markov

chain, a benchmark metastable system, as well as a traffic network derived from real-world taxi trip data. Section 5 contains the conclusions and remarks on future work.

Notation

This article makes use of some special notation, mostly regarding the entries of matrices. For $N \in \mathbb{N}$, denote $[N] := \{1, \dots, N\}$. For a matrix $A \in \mathbb{R}^{M \times N}$, denote by $A_{[:,j]}$ the j-th column vector of A. Likewise, denote by $A_{[i,:]}$ the i-th row vector of A. For $\mathcal{J} \subset [N]$, let $A_{\mathcal{J}}$ denote the column subsampled matrix with respect to $\mathcal{J}$:

$$\left(A_{\mathcal{J}}\right)_{[:,j]} = \begin{cases} A_{[:,j]}, & \text{if } j \in \mathcal{J} \\ 0, & \text{otherwise} \end{cases}, \tag{1}$$

where 0 here denotes the zero-vector in $\mathbb{R}^N$. For $R \in [N]$, the matrix consisting of the leading R columns and all rows of P is denoted by $P_{[:,1:R]}$. Analogously, the matrix consisting of the leading R rows and all columns of P is denoted by $P_{[1:R,:]}$. As usual, the entry of the i-th row and j-th column of A is denoted by A_{ij}.

2. Aggregatable Markov chains

We consider an N-state time- and space-discrete Markov chain $(X_n)_{n\in\mathbb{N}}$, or short (X_n). Without loss of generality, its state space is $[N]$. Let $\Omega := \{\Omega_1, \dots, \Omega_R\}$, $\Omega_r \subset [N]$ be a partition of $[N]$. Let $\omega : [N] \to [R]$ be the function assigning the states to their respective partition element: $\omega(i) = r$ if $i \in \Omega_r$. The number of states in the r-th partition element is denoted by $m_r := |\Omega_r|$. The time-evolution of probability distributions under (X_n) is described by the transition matrix[1] $P \in \mathbb{R}^{N \times N}$ of (X_n):

$$P_{ij} = \mathbb{P}[X_{n+1} = i \mid X_n = j].$$

As the process (X_n) is homogeneous, P does not depend on the step n. Similarly, we can describe the transition probabilities from individual states to the partition elements by a matrix $\widetilde{P} \in \mathbb{R}^{R \times N}$:

$$\widetilde{P}_{rj} = \mathbb{P}[X_{n+1} \in \Omega_r \mid X_n = j] = \sum_{i \in \Omega_r} P_{ij}.$$

Now, given (X_n) and Ω, we can define the *aggregated* stochastic process $(Y_n)_{n\in\mathbb{N}}$, or short (Y_n), on state space $[R]$ by

$$Y_n = r \;:\Longleftrightarrow\; X_n \in \Omega_r \quad \text{for } n \in \mathbb{N}.$$

In contrast to (X_n), the process (Y_n) is in general non-homogeneous, and furthermore depends on the initial distribution of (X_n). Hence, the transition matrix $\widehat{P} \in \mathbb{R}^{R \times R}$ of (Y_n), for now only symbolically defined by

$$\widehat{P}_{rs} = \mathbb{P}[X_{n+1} \in \Omega_r \mid X_n \in \Omega_s]$$

is not well-defined.

The purpose of this article is now essentially to answer the following questions:

1. When is (Y_n) again a Markov process, i.e., when is the matrix $\widehat{P}$ well-defined?
2. Is (Y_n) equivalent to the full process, i.e., can P be restored from $\widehat{P}$, and if so, how?
3. How much knowledge (data) about the full process is required to construct the reduced process, i.e., can $\widehat{P}$ and Ω be computed from just a sparse sample of P?

The conditions on P and Ω under which all three questions can be answered positively are presented in this section.

[1] Note that we use the definition of the transition matrix from [43], hence P is the transposed of what is more commonly known as the transition matrix (see, e.g., [29]). This way, the space of accessible distributions of the Markov chain coincides with the (column) span of P, and invariant distributions are (right) eigenvectors of P.

2.1. Lumpability and deflatability

There are two central conditions a transition matrix P along with a partition Ω must fulfill in order to be sparsely compressible into $\widehat{P}$, called *lumpability* and *deflatability*. These conditions impose strong restrictions on the admissible transition probabilities from and to the partition elements Ω_i, and in this way on the column- and row structure of P. We will show that these two conditions are fundamental for making P low rank and thus for the construction of a sparse approximation algorithm. We will also see later (Section 2.3) that other common properties of Markov chains related to model reduction are *not* equivalent in inducing said low-rank structure.

Definition 2.1. *Let $\Omega = \{\Omega_1, \dots, \Omega_R\}$ be a partition of $[N]$ and $\pi = \{\pi_1, \dots, \pi_R\}$ be a collection of distribution vectors over $[N]$, where π_r has support in Ω_r. Let Π be the matrix*

$$\Pi = \begin{bmatrix} | & & | \\ \pi_1 & \cdots & \pi_R \\ | & & | \end{bmatrix} \in \mathbb{R}^{N \times R}.$$

We call the transition matrix P lumpable *with respect to Ω if*

$$\widetilde{P}_{[:,j]} = \widetilde{P}_{[:,k]} \qquad \text{if } \omega(j) = \omega(k). \tag{2}$$

We call P deflatable *with respect to (Ω, π) if for all $j \in [N]$ holds*

$$P_{[:,j]} = \Pi \cdot \widetilde{P}_{[:,j]}. \tag{3}$$

We call P aggregatable *with respect to (Ω, π) if P is lumpable and deflatable with respect to (Ω, π). In this case we call the partition elements $\Omega_1, \dots, \Omega_R$ the* aggregates *of (X_n).*

The two properties lumpability and deflatability have very different historical backgrounds. While the former is well-established and the basis for many model reduction techniques, the latter, to the best of our knowledge, seems to be a new concept and uninvestigated (the term "deflatability" is introduced herein for the first time). Still, we prefer to see the two properties complementary to each other, in the following way:

Lumpability means that the probability to transition into a certain partition element Ω_r depends only on the partition element $\omega(j)$ of the starting state j, not on the exact starting state:

$$\widetilde{P}_{rj} = \widetilde{P}_{rk} \quad \text{if } \omega(j) = \omega(k).$$

Hence, lumpability describes a sort of "starting state similarity" of the transition probabilities within the aggregates.

In contrast to lumpability, deflatability describes a sort of "end state similarity". By re-writing (3) as

$$P_{ij} = \widetilde{P}_{\omega(i),j} \cdot \pi_{\omega(i)}(i), \tag{4}$$

one sees that deflatability means that the transition probabilities between states essentially depend only on the aggregate of the end state, up to factors that do not depend on the starting state:

$$P_{ij}\pi_r(k) = P_{kj}\pi_r(i) \quad \text{if } \omega(i) = \omega(k) = r.$$

Alternatively, we can describe deflatability as the property that after a jump into a partition element, the selection of one specific next state from this partition element is, independent of where the jump started, decided by randomly choosing from the distribution $\pi_{\omega(i)}$ on that partition element.

Still, the lumpability property alone, first introduced by Kenemy and Snell in [29], already ensures that (Y_n) is a Markov process and independent of the initial distribution of (X_n):

Theorem 2.2 ([29], Theorem 6.3.2)**.** *The matrix P is lumpable with respect to Ω if and only if the aggregated process (Y_n) is homogeneous and its transition probabilities*

$$\widehat{P}_{sr} = \mathbb{P}[X_{n+1} \in \Omega_s \mid X_n \in \Omega_r]$$

are independent of the choice of the probability distribution Ω_r.

Question 1 from the beginning of this section can therefore be answered by assuming lumpability of the underlying chain. Since its inception in the 70s, there have been numerous numerical algorithms that exploit lumpability for model reduction [16, 46, 45, 10], where recently the connection to metastability has come more into focus [27, 51]. All the cited algorithms are robust, in the sense that under the assumption of an appropriate notion of only *approximate* lumpability (such as weak lumpability [39] or quasi-lumpability [12]), they allow for the recovery of approximate reduced models. In situations where not even approximate lumpability may be assumed, multilevel aggregation methods [14, 13, 15] may be applicable, that do not require lumpability with respect to any predetermined collection of aggregates, but successively construct the "best possible" lumping of states on each level.

However, all the mentioned algorithms in general require full knowledge of the transition matrix P, and indeed, without further assumptions on P, a successful deduction of $\widehat{P}$ cannot be performed without it. This is why the additional property of deflatability is required.

Combining lumpability and deflatability immediately implies that P admits a very simple structure:

Lemma 2.3. *Let P be aggregatable with respect to (Ω, π). Then*

$$P_{[:,j]} = P_{[:,k]} \quad \text{if } \omega(j) = \omega(k). \tag{5}$$

Proof. For $\omega(j) = \omega(k)$ we have

$$P_{ij} \overset{(4)}{=} \widetilde{P}_{\omega(i),j} \cdot \pi_{\omega(i)}(i) \overset{(2)}{=} \widetilde{P}_{\omega(i),k} \cdot \pi_{\omega(i)}(i) \overset{(4)}{=} P_{ik}. \qquad \square$$

In words, an aggregatable transition matrix P consists of exactly R pairwise distinct columns, hence $\operatorname{rank}(P) = R$. The assumption of aggregability, which restricts P to matrices of form (5), is therefore a very strong requirement. However, we will argue in Section 2.3.1, as well as the example Section 4, that many real-world Markov transition matrices indeed possess at least an approximate form of this property.

Remark 2.4. We call the property (5) *state-wise lumpability* of P with respect to Ω. Note that not every transition matrix of rank R is state-wise lumpable, hence aggregatable. Moreover, not every state-wise lumpable matrix is deflatable, so state-wise lumpability is not equivalent to aggregability.

Another important consequence of aggregability is that the transition probabilities from aggregates to states are independent of the starting distribution:

Lemma 2.5. *Let P be aggregatable with respect to (Ω, π). Let $i \in [N]$, $r \in [R]$, and $\rho_r^{(1)}, \rho_r^{(2)}$ be two arbitrary distributions with support on Ω_r. Then*

$$\mathbb{P}[X_{n+1} = i \mid X_n \sim \rho_r^{(1)}] = \mathbb{P}[X_{n+1} = i \mid X_n \sim \rho_r^{(2)}].$$

Proof. We have

$$\mathbb{P}[X_{n+1} = i \mid X_n \sim \rho_r^{(1)}] = \sum_{j \in \Omega_r} P_{ij}\, \rho_r^{(1)}(j) = \sum_{j \in \Omega_r} P_{ik}\, \rho_r^{(1)}(j),$$

where the last identity holds for any $k \in \Omega_r$ due to (5). Hence,

$$\begin{aligned}\mathbb{P}[X_{n+1} = i \mid X_n \sim \rho_r^{(1)}] &= P_{ik} \underbrace{\sum_{j\in\Omega_r} \rho_r^{(1)}(j)}_{=1} = P_{ik} \underbrace{\sum_{j\in\Omega_r} \rho_r^{(2)}(j)}_{=1} \\ &= \sum_{j\in\Omega_r} P_{ik}\rho_r^{(2)}(j) = \sum_{j\in\Omega_r} P_{ij}\rho_r^{(2)}(j) \\ &= \mathbb{P}[X_{n+1} = i \mid X_n \sim \rho_r^{(2)}]. \qquad \square\end{aligned}$$

Note that Lemma 2.5 is stronger than Theorem 2.2. Hence, for aggregatable Markov chains, the reduced transition matrix $\widehat{P}$ can be defined by

$$\widehat{P}_{rs} := \sum_{i\in\Omega_r} \mathbb{P}[X_{n+1} = i \mid X_n \sim \rho_s], \tag{6}$$

where ρ_s is any distribution with support in Ω_s. Clearly this $\widehat{P}$ is stochastic and thus induces a Markov chain $(Y_n)_{n\in\mathbb{N}}$ whose states are the aggregates $\Omega_1, \dots, \Omega_R$ (or equivalently, $[R]$).

Finally, the following central result describes the exact relation of $\widehat{P}$ to the original transition matrix P, hence can be seen as the answer to question 2 from the beginning of this section. It will also play a major role in the latter algorithmic procedure.

Proposition 2.6. *Let P be aggregatable with respect to (Ω, π). Then P admits the decomposition*

$$P = \Pi \widehat{P} \Lambda, \tag{7}$$

where

$$\Lambda = \begin{bmatrix} — & \mathbb{1}_{\Omega_1} & — \\ & \vdots & \\ — & \mathbb{1}_{\Omega_R} & — \end{bmatrix} \in \mathbb{R}^{R\times N}$$

and $\mathbb{1}_{\Omega_r} \in \mathbb{R}^N$ is the indicator vector of Ω_r.

Proof. Condition (3) implies $P = \Pi\widetilde{P}$. On the other hand, we have

$$\begin{aligned}\big(\widehat{P}\Lambda\big)_{ri} &= \sum_{s=1}^{R} \widehat{P}_{rs}\Lambda_{si} = \widehat{P}_{r\omega(i)} \\ &= \mathbb{P}[X_{n+1} \in \Omega_r \mid X_n \in \Omega_{\omega(i)}].\end{aligned}$$

As by Lemma 2.5 this probability is independent of the starting distribution on $\Omega_{\omega(i)}$, we may assume $X_n \sim \mathbb{1}_i$, and hence

$$\big(\widehat{P}\Lambda\big)_{ri} = \mathbb{P}[X_{n+1} \in \Omega_r \mid X_n = i] = \widetilde{P}_{ri}. \qquad \square$$

Thus, P can be restored under knowledge of $\Pi, \widehat{P}$ and Λ.

Remark 2.7. We can interpret the decomposition (7) as follows: For a distribution vector $u \in \mathbb{R}^N$ over $[N]$, $Pu = \Pi\widehat{P}\Lambda u$ describes the pushforward of u under the dynamics. In a first step, Λu averages u over the aggregates, i.e., $\Lambda u \in \mathbb{R}^R$ is a distribution vector over $[R]$. In a second step, this distribution is pushed forward by the reduced transition matrix $\widehat{P}$, i.e., transformed according to the probabilities to transition between the aggregates. The result is again a distribution vector over $[R]$. Finally, Π extends this vector again to a distribution vector over the individual states $[N]$, by multiplying each entry with the appropriate distribution π_r. The procedure is illustrated in Figure 1.

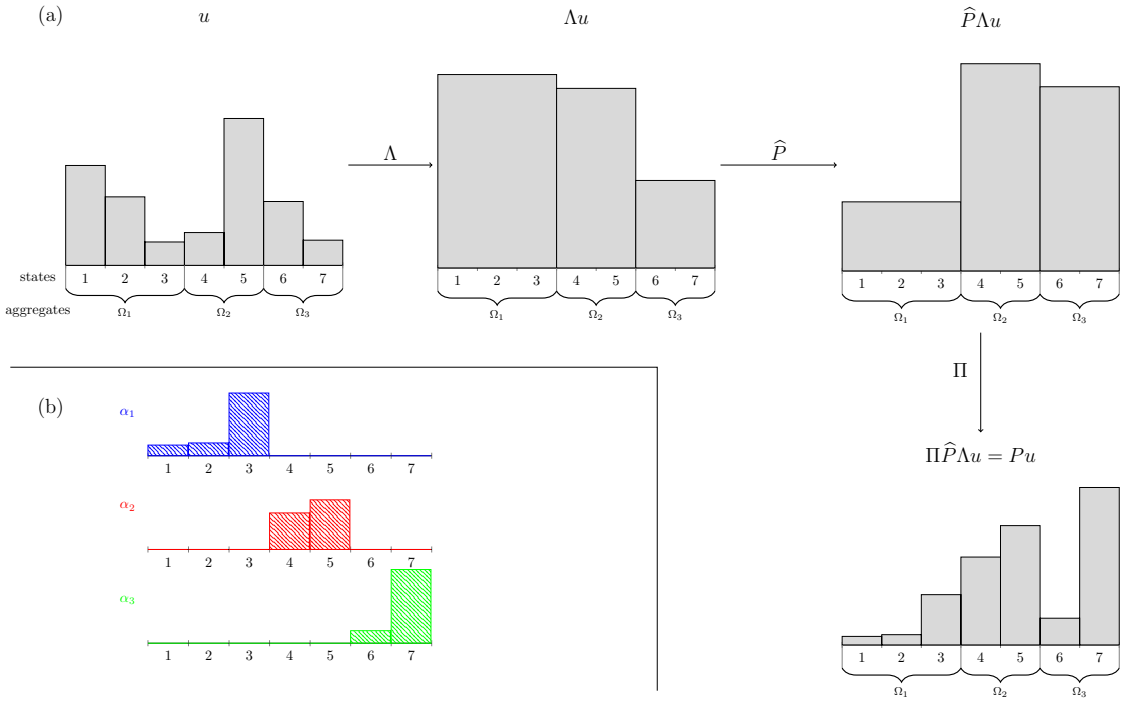

Figure 1: (a) Illustration of the distribution transport under an aggregatable matrix P for $N = 7$ and $R = 3$. The distribution vector u gets first averaged over the aggregates $\{\Omega_1, \ldots, \Omega_R\}$ by the matrix Λ, subsequently pushed forward by the reduced transition matrix $\widehat{P}$, and finally "inflated" again to a distribution vector over the states by the matrix Π. (b) Illustration of the distribution vectors $\pi_1, \ldots, \pi_R$ used in the last step. For $r = 1, \ldots, R$, the r-th entry of the vector $\widehat{P}\Lambda u$ gets multiplied by π_r to form the vector $Pu = \Pi\widehat{P}\Lambda u$.

2.2. Almost aggregability

Markov chains encountered in real-life applications rarely fulfill the lumpability and deflatability conditions exactly. We therefore introduce appropriate notions of "almost lumpability" and "almost deflatability", and investigate in what sense such transition matrices are close to truly aggregatable matrices.

Definition 2.8. *Let $\Omega = \{\Omega_1, \ldots, \Omega_R\}$ be a partition of $[N]$, let $\pi = \{\pi_1, \ldots, \pi\}$ be a collection of distribution vectors over $[N]$, where π_r has support in Ω_r, and let*

$$\Pi = \begin{bmatrix} | & & | \\ \pi_1 & \cdots & \pi_R \\ | & & | \end{bmatrix} \in \mathbb{R}^{R \times N}.$$

*For $\varepsilon > 0$, we call the transition matrix P ε-*almost lumpable with respect to Ω *if for all $r \in [R]$ holds*

$$\left\| \widetilde{P}_{[:,j]} - \widetilde{P}_{[:,k]} \right\|_1 \leq \varepsilon \quad \text{if } \omega(j) = \omega(k) \tag{8}$$

*We call P ε-*almost deflatable with respect to (Ω, π) *if for all $j \in [N]$ holds*

$$\left\| P_{[:,j]} - \Pi \cdot \widetilde{P}_{[:,j]} \right\|_1 \leq \varepsilon \tag{9}$$

*We call P ε-*almost aggregatable with respect to (Ω, π) *if P is ε-almost lumpable and ε-almost deflatable with respect to (Ω, π).*

Remark 2.9. A comment on the choice of the norm: Lumpability (2) is the equality of two columns of the matrix $\widetilde{P}$, which are distribution vectors over $[R]$. A natural definition for ε-almost lumpability is therefore the ε-closeness of these distribution vectors, for which the natural distance measure is the L^1-norm in $\mathbb{R}^R$.

Likewise, deflatability (3) is the equality of a column of the matrix P and a specific vector in $\mathbb{R}^N$. These are both distribution vectors over $[N]$, and thus a natural definition for ε-almost deflatability is the ε-closeness of these vectors in the L^1-norm in $\mathbb{R}^N$.

Almost aggregability now implies that P is close to an aggregatable transition matrix in the L^1 norm:

Theorem 2.10. *Let P be ε-almost aggregatable with respect to (Ω, π). Then there exists an aggregatable transition matrix $\overline{P} \in \mathbb{R}^{N\times N}$ and a matrix $E \in \mathbb{R}^{N\times N}$ with $\|E\|_1 \leq 4\varepsilon$ such that*

$$P = \overline{P} + E. \tag{10}$$

Proof. See Appendix A. □

Note that $\|E\|_1 \leq 4\varepsilon$ implies $|E_{ij}| \leq 4\varepsilon$ for all $i, j \in [N]$. We from now on assume that the perturbation matrix $E = E(\varepsilon)$ is element-wise analytic in ε. Then P admits a Taylor expansion

$$P = \overline{P} + \varepsilon E^{(1)}(0) + \frac{\varepsilon^2}{2} E^{(2)}(0) + \dots, \tag{11}$$

where $E^{(k)}$ denotes the k-th element-wise derivative of E with respect to ε. We shorthand write (11) as

$$P = \overline{P} + \varepsilon L + \mathcal{O}(\varepsilon^2), \tag{12}$$

where for $L := E^{(1)}(0) \in \mathbb{R}^{N\times N}$ holds $\|L\|_1 \leq 4$. Although the element-wise perturbation result (12) is somewhat weaker than the perturbation with respect to the $\|\cdot\|_1$-norm (10), it will prove more useful when performing perturbation analysis on the spectrum of P (Section 3.1.3).

2.3. Comparison to other properties of compressible Markov chains

(Almost) lumpability and (almost) deflatability should be seen as fundamental, abstract properties that Markov chains from different areas of applications may or may not have. To the best of our knowledge, there exists no concept in the literature that is equivalent to our definition of (almost) aggregability. In this section, we compare almost aggregability to two other properties of Markov chains that are commonly investigated for the purpose of model compression, namely metastability and near completely-decomposability, and show that they are indeed not equivalent to our definition of almost aggregability.

2.3.1. Metastable Markov chains

Metastable Markov chains are almost aggregatable. We show this for the special case of reversible Markov chains.

For a subset of states $\mathcal{M} \subset [N]$ consider the first exit time from $\mathcal{M}$

$$\tau_{\mathcal{M}} := \inf\{n \in \mathbb{N},\ X_n \notin \mathcal{M}\},$$

which is a random variable in $\mathbb{N}$. Let $\pi_{\mathcal{M}}$ be the quasi-sationary density (QSD) of $\mathcal{M}$, defined as the long-time limit of the law of (X_n) conditioned to stay on $\mathcal{M}$:

$$\pi_{\mathcal{M}} := \lim_{n\to\infty} \mathrm{Law}(X_n \mid \tau_{\mathcal{M}} > n). \tag{13}$$

Following [25], we now call the set $\mathcal{M}$ *metastable* if the time to observe almost-convergence in (13) is small compared to the mean exit time $\mathbb{E}[\tau_{\mathcal{M}}]$. This definition can be made precise by relating the

convergence rate of (13) and the rate of exit events to the dominant eigenvalues of the infinitesimal generator of the process [25].

Based on the above understanding of metastability, we can assume that for the unrestricted process (X_n), at the time when the first exit event from $\mathcal{M}$ happens, $\mathrm{Law}(X_n)$ is already close to $\pi_{\mathcal{M}}$, without loss of generality in the 1-norm. Hence, if $\mathcal{M}$ is metastable, there exists a step count $\eta \ll \mathbb{E}[\tau_{\mathcal{M}}]$, and a small number $\varepsilon > 0$ such that

$$\left\|(P^\eta)_{[:,j]} - \pi_{\mathcal{M}}\right\|_1 \leq \varepsilon \quad \text{for all } j \in \mathcal{M}, \tag{14}$$

where P^η denotes the η-th power of P. The lag time η in (14) should be thought of as being long enough to observe local equilibration to the QSD, but not long enough to likely experience exit events and observe global equilibration to the stationary density.

Now suppose that $\Omega = \{\Omega_1, \dots, \Omega_R\}$ is a metastable partition of $[N]$, and that there exists an $\eta > 0$ and a small $\varepsilon > 0$ such that (14) holds for all $\mathcal{M} = \Omega_r$. Under this assumption, we can now show that the η-step transition matrix P^η is almost aggregatable:

Proposition 2.11. *Let Ω be a partition of $[N]$ and let (14) hold with parameters η, ε for all $\Omega_r \in \Omega$. Let π_r denote the QSD of Ω_r, $r = 1, \dots, R$. Then P^η is 2ε-almost aggregatable with respect to (Ω, π), where $\pi = \{\pi_1, \dots, \pi_R\}$.*

Proof. The matrix P^η is state-wise 2ε-almost lumpable: for $\omega(i) = \omega(j) = r$ we have

$$\left\|(P^\eta)_{[:,i]} - (P^\eta)_{[:,j]}\right\|_1 \leq \left\|(P^\eta)_{[:,i]} - \pi_r\right\|_1 + \left\|(P^\eta)_{[:,j]} - \pi_r\right\|_1 \leq 2\varepsilon.$$

In particular P^η is 2ε-almost lumpable.

Now let e_r be the r-th unit vector in $\mathbb{R}^R$. Because the transition probability into other aggregates is low, the j-th column of $\widetilde{P^\eta}$ is ε-close to $e_{\omega(j)}$:

$$\left\|(\widetilde{P^\eta})_{[:,j]} - e_{\omega(j)}\right\|_1 = \sum_{r \in [R]} \left|(\widetilde{P^\eta})_{rj} - e_{\omega(j)}(r)\right| =: (\star).$$

As $\pi_{\omega(j)}$ is a distribution with support in $\Omega_{\omega(j)}$, we have $\sum_{i \in \Omega_r} \pi_{\omega(j)}(i) = e_{\omega(j)}(r)$. Thus,

$$\begin{aligned}
(\star) &= \sum_{r \in [R]} \Big| \sum_{i \in \Omega_r} (P^\eta)_{ij} - \sum_{i \in \Omega_r} \pi_{\omega(j)}(i) \Big| \\
&\leq \sum_{r \in [R]} \sum_{i \in \Omega_r} \left|(P^\eta)_{ij} - \pi_{\omega(j)}(i)\right| \\
&= \sum_{i \in [N]} \left|(P^\eta)_{ij} - \pi_{\omega(j)}(i)\right| \\
&= \left\|(P^\eta)_{[:,j]} - \pi_{\omega(j)}\right\|_1 \overset{(14)}{\leq} \varepsilon.
\end{aligned}$$

Using this and (14), we can show 2ε-almost deflatability:

$$\begin{aligned}
\left\|(P^\eta)_{[:,j]} - \Pi \cdot (\widetilde{P^\eta})_{[:,j]}\right\|_1 &\leq \left\|(P^\eta)_{[:,j]} - \Pi \cdot e_{\omega(j)}\right\|_1 + \left\|\Pi \cdot e_{\omega(j)} - \Pi \cdot (\widetilde{P^\eta})_{[:,j]}\right\|_1 \\
&\leq \underbrace{\left\|(P^\eta)_{[:,j]} - \pi_{\omega(j)}\right\|_1}_{\leq \varepsilon} + \underbrace{\|\Pi\|_1}_{=1} \underbrace{\left\|e_{\omega(j)} - (\widetilde{P^\eta})_{[:,j]}\right\|_1}_{\leq \varepsilon}.
\end{aligned}$$

□

Remark 2.12. On the other hand, not every almost aggregatable Markov chain is metastable, hence the two concepts are not equivalent. See Section 4.1 for a counterexample. Loosely speaking, an almost aggregatable transition matrix P is metastable if its reduced transition matrix $\widehat{P}$ is almost diagonal.

2.3.2. Nearly completely decomposable Markov chains

Nearly completely decomposable Markov chains [3, 48], also called nearly uncoupled [18], in general are not almost aggregatable. To be specific, they do not fulfill the almost deflatability property (9).

Let $\Omega = \{\Omega_1, \dots, \Omega_R\}$ again be a partition of state space $[N]$, and assume the states are ordered by partition element, i.e.,

$$\forall\, r, s \in [R], \forall\, i \in \Omega_r, j \in \Omega_s : r < s \Rightarrow i < j.$$

A transition matrix P is then called *completely decomposable* (CD) with respect to Ω, if it has block-diagonal form, i.e., if there exist matrices $D_1 \in \mathbb{R}^{m_1 \times m_1}, \dots, D_R \in \mathbb{R}^{m_R \times m_R}$, such that [12]

$$P = \begin{bmatrix} D_1 & 0 & \cdots & 0 \\ 0 & D_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & D_R \end{bmatrix}.$$

A transition matrix P is called *nearly completely decomposable* (NCD), if $P = \overline{P} + E$ for an uncoupled matrix $\overline{P}$, and small $\varepsilon := \|E\|_\infty$ [12].

Uncoupled matrices are in general not deflatable with respect to Ω. One easily sees that all columns of the individual D_i need to be equal in order for P to be deflatable. Consider for example the CD matrix

$$P = \left[\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right] \tag{15}$$

with the two partition elements $\Omega_1 = \{1, 2\},\ \Omega_2 = \{3, 4\}$ and the sub-matrices $D_1 = D_2 = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. The matrices D_1, D_2 are themselves not CD, hence P cannot be decomposed further. For (15), the transition probability matrix between the individual states and the partition elements becomes

$$\widetilde{P} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

In particular, the columns of $\widetilde{P}$ in each partition element are equal (this is a universal property of CD matrices). However, as the two columns of P in each partition element are *not* equal, no matrix $\Pi \in \mathbb{R}^{N \times R}$ can exist such that $P = \Pi\widetilde{P}$, as would be required by the deflatability condition (3).

The matrix (15) is not even ε-almost deflatable for a small ε, as

$$\operatorname*{arg\,min}_{\overline{P} \text{ deflatable}} \|P - \overline{P}\|_1 = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{bmatrix},$$

hence $\min_{\overline{P} \text{ deflatable}} \|P - \overline{P}\|_1 = 1$. As every CD matrix is NCD, this demonstrates that NCD transition matrices in general are not almost deflatable.

Remark 2.13. On the other hand, it is easy to see that every CD matrix is lumpable. It also has been shown that NCD matrices are quasi-lumpable [12], a slight relaxation of lumpability.

Finally, note that metastability is a special case of nearly complete decomposability. Here, the internal homogeneity in the sub-matrices D_i that is required for deflatability is present due to the rapid equilibration inside the metastable sets.

3. A probabilistic aggregation algorithm

Now let P be an ε-almost aggregatable matrix, which in particular implies

$$P = \overline{P} + \varepsilon L + \mathcal{O}(\varepsilon^2),$$

where $\overline{P}$ is aggregatable and $\|L\|_1 \leq 4$. Our goal in this section is to compute the aggregates $\Omega_1, \ldots, \Omega_R$ as well as the matrix $\widehat{P}$ of the aggregatable matrix $\overline{P}$. We will derive an algorithm that achieves this using only a vastly incomplete subset of the entries of P. This algorithm will therefore be the answer to question 3 posed at the beginning of Section 2.

3.1. Sparse recovery of the aggregates

Assume for the moment that P is aggregatable, i.e., $P = \overline{P}$. Let $\mathcal{J} \subset [N]$ be any index set in which all aggregates are "represented", i.e., $\omega(\mathcal{J}) = [R]$[2]. Consider the column-subsampled transition matrix $P_{\mathcal{J}}$, as defined in (1). As $\omega(\mathcal{J}) = [R]$ and P is state-wise lumpable, the vector spaces spanned by the columns of P and $P_{\mathcal{J}}$ are identical. Furthermore, the R leading left singular vectors of $P_{\mathcal{J}}$ are linear combinations of the π_r, as shown by the following theorem. Note that this does *not* simply follow from $\operatorname{range}(P_{\mathcal{J}}) = \operatorname{span}\{\pi_1, \ldots, \pi_R\}$.

Theorem 3.1. *Let P be an aggregatable matrix admitting the decomposition $P = \Pi \widehat{P} \Lambda$ from Proposition 2.6, and let $P_{\mathcal{J}}$ as defined in (1). For $r \in [R]$, let*

$$j_r := |\mathcal{J} \cap \Omega_r|,$$

i.e., the number of indices in $\mathcal{J}$ that belong to Ω_r. Define the diagonal matrices

$$D_\Pi = \operatorname{diag}\left(\|\pi_1\|_2, \ldots, \|\pi_R\|_2\right), \quad D_{\Lambda_{\mathcal{J}}} = \operatorname{diag}\left(\sqrt{j_1}, \ldots, \sqrt{j_R}\right).$$

Let $\widehat{U}\,\widehat{S}\,\widehat{V}$ be the singular value decomposition of $D_\Pi \widehat{P} D_{\Lambda_{\mathcal{J}}}$. Then there exists a singular value decomposition $U\,S\,V$ of $P_{\mathcal{J}}$, with

$$S_{[1:R,1:R]} = \widehat{S},$$

and

$$U_{[:,1:R]} = \Pi D_\Pi^{-1} \widehat{U}, \quad V_{[1:R,:]} = \widehat{V} D_\Lambda^{-1} \Lambda_{\mathcal{J}}.$$

Proof. We can write the column-subsampling of P as $P_{\mathcal{J}} = P \cdot I_{\mathcal{J}}$, where I is the $N \times N$ identity matrix. Plugging in the decomposition of P and the SVD of $D_\Pi \widehat{P} D_{\Lambda_{\mathcal{J}}}$ yields

$$P_{\mathcal{J}} = \left(\Pi \widehat{P} \Lambda\right) I_{\mathcal{J}} = \underbrace{\Pi D_\Pi^{-1} \widehat{U}}_{=:U^{(1)}}\, \widehat{S}\, \underbrace{\widehat{V} D_{\Lambda_{\mathcal{J}}}^{-1} \Lambda I_{\mathcal{J}}}_{=:V^{(1)}}.$$

The columns of $U^{(1)}$ are orthonormal:

$$U^{(1)\intercal} U^{(1)} = \left(\Pi D_\Pi^{-1} \widehat{U}\right)^\intercal \left(\Pi D_\Pi^{-1} \widehat{U}\right) = \widehat{U}^\intercal D_\Pi^{-1} \underbrace{\Pi^\intercal \Pi}_{=(D_\Pi)^2} D_\Pi^{-1} \widehat{U} = \widehat{U}^\intercal \widehat{U} = I.$$

The rows of $V^{(1)}$ are also orthonormal:

$$V^{(1)} V^{(1)\intercal} = \left(\widehat{V} D_{\Lambda_{\mathcal{J}}} \Lambda I_{\mathcal{J}}\right)\left(\widehat{V} D_{\Lambda_{\mathcal{J}}} \Lambda I_{\mathcal{J}}\right)^\intercal = \widehat{V} D_{\Lambda_{\mathcal{J}}} \underbrace{\Lambda_{\mathcal{J}} \Lambda_{\mathcal{J}}^\intercal}_{=(D_{\Lambda_{\mathcal{J}}})^2} D_{\Lambda_{\mathcal{J}}} \widehat{V}^\intercal = \widehat{V} \widehat{V}^\intercal = I.$$

Let $U^{(2)}$ be a completion of the columns of $U^{(1)}$ to an orthonormal basis of $\mathbb{R}^N$, and analogously $V^{(2)}$ be a completion of the rows of $V^{(1)}$ to an orthonormal basis of $\mathbb{R}^N$. We then can write $P_{\mathcal{J}}$ as

$$P = \underbrace{\left[\begin{array}{c|c} U^{(1)} & U^{(2)} \end{array}\right]}_{=:U} \underbrace{\left[\begin{array}{c|c} \widehat{S} & 0 \\ \hline 0 & 0 \end{array}\right]}_{=:S} \underbrace{\left[\begin{array}{c} V^{(1)} \\ \hline V^{(2)} \end{array}\right]}_{=:V},$$

which by definition is a singular value decomposition of $P_{\mathcal{J}}$. □

[2] We will describe later how to find such an index set without a priori knowledge of the aggregates.

The fact that the leading R left singular vectors $\{u_1, \dots, u_R\}$ of $P_{\mathcal{J}}$ are linear combinations of the columns of Π, i.e., the vectors $\{\pi_1, \dots, \pi_R\}$, now can be exploited to compute the aggregates under knowledge of only the matrix $P_{\mathcal{J}}$. As the π_r do not change sign within the aggregates, neither do the u_r. Furthermore, there exist no two aggregates on which the sign structure of all $u_r,\ r \in [R]$, is identical:

Lemma 3.2. *Let P be aggregatable with respect to (Ω, π). Let $u_1, \dots, u_R$ be the leading orthonormal left singular vectors of $P_{\mathcal{J}}$, and let $\sigma : [N] \to \{-1, 0, 1\}^N$ be given by*

$$\sigma(i) := \big[\operatorname{sgn}\big(u_1(i)\big), \dots, \operatorname{sgn}\big(u_R(i)\big)\big], \quad i = 1, \dots, N$$

where $\operatorname{sgn} : \mathbb{R} \to \{-1, 0, 1\}$ denotes the sign function. Then for any two i, j with $\omega(i) \neq \omega(j)$ holds

$$\sigma(i) \neq \sigma(j) \quad \textit{and} \quad \sigma(i) \neq -\sigma(j).$$

Proof. The proof is identical to that of [21, Theorem 3.1], where instead of aggregatable matrices, block stochastic matrices were considered. We repeat the short argument for completenes' sake.

Since the u_r do not change sign within the aggregates, we may assume that each aggregate conists of only one state, i.e., $N = R$. Then $U = [u_1, \dots, u_R] \in \mathbb{R}^{R \times R}$ is a square matrix with orthonormal columns, the rows of U are also orthogonal. Hence, no two row vectors, which are the vectors $\big(u_1(i), \dots, u_R(i)\big)$, can have the same sign structure. □

Thus, once the left singular vectors of $P_{\mathcal{J}}$ have been computed, the aggregates can be recovered by grouping the states according to the values of the vectors $\sigma(i),\ i \in [N]$, i.e.,

$$\omega(i) \overset{!}{=} \omega(j) \quad \text{if } \sigma(i) = \sigma(j).$$

Remark 3.3. The technique of grouping states via the sign structure of eigen- or singular vectors of some propagator matrix is not new, and in general is known as *spectral clustering*. In particular, similar to us, Fritzsche et al. [21] find the metastable sets of a metastable system by analyzing the dominant singular vectors of the transition matrix P. The fundamental idea however goes back to Dellnitz, Junge, Deuflhard, Schütte and coworkers [17, 18], who originally identified metastable sets from the dominant eigenvectors of discretizations of transfer operators. Fritzsche et al. state as the main reason to compute the singular- over the eigendecomposition of P the applicability to non-reversible systems. Similarly, Froyland [22] computes metastable sets via eigenvectors of a certain "reversibilized" transition matrix of a in general non-reversible Markov chain. Compared to all these methods, the innovation of our method is that only aggregability of the system must be assumed (of which metastability is a special case), and, crucially, only the vastly incomplete matrix $P_{\mathcal{J}}$ instead of the full matrix P is required.

Also note that the sign structure of eigen- or singular vectors is in general unstable under perturbation of the underlying matrix [38] (see also Section 3.1.3). Therefore, the assignment to the aggregates based purely on the sign structure is unstable as well. However, there exist advanced and robust spectral clustering techniques, for example PCCA+ [19, 38] and SEBA [23], that consider not only the signs, but also the magnitude of the eigenvector entries and are less susceptible to these instabilities. These methods are fully compatible with our setting.

3.1.1. Probabilistic column sampling.

While there exist minimal index sets $\mathcal{J}$ with $|\mathcal{J}| = R$ and $\omega(\mathcal{J}) = [R]$, it is in general impossible to select such a set without a priori knowledge of the aggregates, or analyzing all columns of P. Our algorithmic strategy will therefore rely on *randomly* sampling the column space of P. We will show that under a sensible assumption regarding the sizes of the aggregates, the number of samples required to fulfill the condition $\omega(\mathcal{J}) = [R]$ with a certain high probability does not depend on N.

Let $J = |\mathcal{J}|$ denote the number of indices. When drawing the indices from $[N]$ uniformly and independently, the probability to "hit" all aggregates is

$$\mathbb{P}\big[\omega(\mathcal{J}) = [R]\big] = \mathbb{P}\big[1 \in \omega(\mathcal{J}) \wedge \ldots \wedge R \in \omega(\mathcal{J})\big].$$

Note that in general, assuming that $R-1$ aggregates are hit will decrease the probability to hit the remaining aggregate, as there are now at most $J-R+1$ chances for the remaining aggregate to be hit. However, if we choose J much bigger than R, this effect is negligible, as then the probability to hit an individual aggregate with J draws is close to the probability to hit it with $J-R+1$ draws. The probabilities $\mathbb{P}\big[r \in \omega(\mathcal{J})\big]$ then are approximately independent, and we have

$$\mathbb{P}\big[\omega(\mathcal{J}) = [R]\big] \approx \prod_{r=1}^{R} \mathbb{P}\big[r \in \omega(\mathcal{J})\big] = \prod_{r=1}^{R} \left(1 - \frac{\binom{N-m_r}{J}}{\binom{N}{J}}\right),$$

where, as a reminder, $m_r = |\Omega_r|$.

In the last equation, the factor $\mathbb{P}\big[r \in \omega(\mathcal{J})\big]$ can be thought of as the probability to draw at least one red ball from an urn with m_r red and $N-m_r$ black balls within J draws without replacement. If $J \ll N$, as we would require in practical applications, drawing with replacement results in approximately the same probability. Hence in that case, we get

$$\mathbb{P}\big[\omega(\mathcal{J}) = [R]\big] \approx \prod_{r=1}^{R} \mathbb{P}\big[r \in \omega(\mathcal{J})\big] \approx \prod_{r=1}^{R} \left(1 - \left(1 - \frac{m_r}{N}\right)^J\right). \tag{16}$$

Now let $p \in (0,1)$ be a "confidence parameter", i.e., minimal probability for which we want to find the smallest J such that

$$\mathbb{P}\big[\omega(\mathcal{J}) = [R]\big] \geq p. \tag{17}$$

From Formula (16), we see that if there exists only a single lowly-populated aggregate Ω_r, i.e., one with $m_r/N \approx 0$, then a large J is required to achieve (17) for any satisfactory p. On the other hand, the best case scenario is when all R aggregates are approximately evenly populated, i.e., $m_r \approx N/R$ for all $r \in [R]$, as this maximizes the right hand side of (16). In this case, (16) approximately takes the form

$$\mathbb{P}\big[\omega(\mathcal{J}) = [R]\big] \approx \left(1 - \left(1 - \frac{1}{R}\right)^J\right)^R. \tag{18}$$

Crucially, this probability does not depend on the number of states N, and hence the number of draws J required to achieve (17) also does not depend on N. For reasonable values of p and small R, J can thus be chosen much smaller than N, which means that only a small subset of the columns of P has to be computed. Referring to question 3 from the beginning of Section 2, it is therefore indeed possible to compute $\widehat{P}$ from a vastly undersampled data matrix P, if one is willing to accept the (qualitative) uncertainty (17) of the result.

We will from now on always assume that all aggregates are of approximately equal size in order to use the simple formula (18) to estimate the required number of column draws. However, this assumption is actually not strictly required to achieve (17) with a low number of draws J. Denoting the ratio of the smallest to the largest aggregate by $\theta \in (0,1)$, i.e.,

$$\theta := \frac{\min_r m_r}{\max_r m_r},$$

one gets

$$\mathbb{P}\big[\omega(\mathcal{J}) = [R]\big] \gtrapprox \left(1 - \left(1 - \frac{\theta}{R}\right)^J\right)^R. \tag{19}$$

For moderate values of θ that do not depend on N (say, $\theta = 0.5$), (19) still allows one to choose moderate values of J in order to guarantee (17) for a sensible p.

Remark 3.4. As a side remark, in the case where all m_r are perfectly equal, the described problem is equal to the so-called *coupon collector's problem* [34, Section 3.6]. It estimates the expectation of the number of randomly drawn columns in order to hit all aggregates as

$$\mathbb{E}\big[\ |\mathcal{J}|\ \big|\ \omega(\mathcal{J}) = [R]\big] = R\log R + \Theta(R),$$

and its variance as

$$\mathrm{Var}\ \big[\ |\mathcal{J}|\ \big|\ \omega(\mathcal{J}) = [R]\big] = \frac{\pi^2}{6}R^2 + \Theta(R\log R),$$

where Θ is the Landau symbol for asymptotically equal growth. The expectation and variance are again independent of N. Thus, for large N, the expected number of columns of P that has to be computed in order to hit all aggregates is again much smaller than N.

3.1.2. The probabilistic aggregation algorithm

In summary, the random column sampling strategy, combined with the singular value decomposition and spectral clustering method leads to our main algorithm:

Algorithm 3.1 Probabilistic aggregation of large Markov chains.

Input: Ability to compute individual columns of the transition matrix P,
Upper bound of the number R of aggregates,
Confidence parameter $p \in (0, 1)$

1: Using Equation (18), randomly draw an index set $\mathcal{J} \subset [N]$, so that

$$\mathbb{P}\big[\omega(\mathcal{J}) = [R]\big] \geq p.$$

2: Compute the columns of P with indices in $\mathcal{J}$ and assemble the matrix $P_{\mathcal{J}}$.
3: Compute the singular value decomposition $P_{\mathcal{J}} = USV$. Let the leading R left singular vectors be denoted by $u_1, \ldots, u_R$.
4: Apply a spectral clustering algorithm such as PCCA+ or SEBA to $u_1, \ldots, u_R$

Output: Aggregates $\Omega_1, \ldots, \Omega_R$ of P.

Multiple remarks are in order:

Remark 3.5. The main attractiveness of this aggregation algorithm is that only J columns of the $N \times N$ transition matrix P need to be computed. As we have discussed in Section 3.1.1, J depends only on R and p, thus for large N this represents enormous savings in numerical effort. The actual method of computing the columns varies from case to case, but typically they need to be computed individually by expensive Monte Carlo sampling methods, see the example in Section 4.2 for details.

Remark 3.6. The requirement of an upper bound of the number R of expected aggregates is not as harsh as it may seem. For one, a ballpark estimate of R is often available in practice. One knows for example that in Markov chains that describe the folding of small proteins, the number of metastable conformations (which here represent the aggregates) typically is in the order of 10^1 to 10^2. For the other, overestimating R does not degrade the quality of the end result, but only leads to additional numerical effort. However, in many practical applications with thousands or millions of states but only a handful of aggregates, even overestimating R by one or two orders of magnitude still leads to vastly improved performance over computing the full transition matrix P. A detailed error analysis of Algorithm 3.1 with respect to R will be subject of future research.

Step 1 of the Algorithm 3.1 again requires that all aggregates are approximately equal-size. In case where this is not a reasonable assumption, we can conduct a different strategy: For aggregatable matrices, $\omega(\mathcal{J})$ is equivalent to $\mathrm{rank}(P_{\mathcal{J}}) = R$, hence $\sigma_R > 0$, where σ_R is the R-th largest singular value of $P_{\mathcal{J}}$. For almost aggregatable matrices, this condition becomes $\sigma_R \gg \sigma_{R+1}$,

i.e., the existence of a *spectral gap* after σ_R. The computational strategy is therefore to randomly add columns to $P_{\mathcal{J}}$ and compute its SVD until such a gap appears.

Remark 3.7. Algorithm 3.1 is related to a class of probabilistic algorithms designed to compute an orthonormal basis of the range of a generic rank-R matrix $A \in \mathbb{R}^{N\times N}$, see for example [26, p. 224]. These algorithms are based on "measuring" A by a randomly-drawn test matrix $T \in \mathbb{R}^{N\times R}$, i.e. the product $Y = A \cdot T$ is computed. As the R randomly-drawn columns of T are almost surely linearly independent, and almost surely do not fall into $\ker(A)$, the columns of Y are also almost surely linearly independent, and it holds $\operatorname{range}(Y) = \operatorname{range}(A)$. The leading left singular vectors of Y hence form an orthonormal basis of $\operatorname{range}(A)$. Note however that the computation of Y here requires the full matrix A.

In Algorithm 3.1, the selection of the columns $\mathcal{J}$ from P is equivalent to the multiplication to P with the matrix $I_{\mathcal{J}}$ (see the proof of Proposition 3.1), i.e., $P_{\mathcal{J}} = P \cdot I_{\mathcal{J}}$, thus $I_{\mathcal{J}}$ can be considered a test matrix in the above context. Crucially however, although the columns of $I_{\mathcal{J}}$ have not been randomly drawn and contain only one non-zero element, we still have $\operatorname{range}(P_{\mathcal{J}}) = \operatorname{range}(P)$, due to the equality of the columns due to state-wise lumpability (5).

3.1.3. Applicability to almost aggregatable Markov chains

We now shift our focus to Markov chains that are only almost instead of exactly aggregatable. Assume that P is ε-almost aggregatable with respect to (Ω, π), which by (12) implies

$$P = \overline{P} + \varepsilon L + \mathcal{O}(\varepsilon^2), \quad \|L\|_1 \leq 4.$$

Our goal is to recover the aggregates of the exactly aggregatable but unknown matrix $\overline{P}$, and we again assume that the computation of individual columns of P is possible. Our strategy is to apply Algorithm 3.1 to the perturbed matrix P, in the hope that the above ε-perturbation will only result in an error for the aggregates of order of magnitude ε. We will illuminate under which conditions this holds true, and for this perform a perturbation analysis of the singular vectors of $\overline{P}$. In particular, we will investigate how the sign structure of the individual components of the singular vectors respond to perturbation, as they are used for the aggregate assignments.

It turns out that the same techniques that have been used by Fritzsche et al. [21] in the analysis of almost block-stochastic transition matrices can also be applied to our setting of almost aggregatable transition matrices. Most of the following arguments have therefore been borrowed from [21, Section 4.1].

Let $\mathcal{J} \subset [N]$ be again an index set with $\omega(\mathcal{J}) = [R]$. Define the matrix

$$T(\varepsilon) := P_{\mathcal{J}} P_{\mathcal{J}}^{\intercal}.$$

We will study how the eigenvectors of $T(\varepsilon)$ depend on ε, as they are identical to the left singular vectors of $P_{\mathcal{J}}$. The matrix $T(\varepsilon)$ admits a Taylor expansion in ε,

$$T(\varepsilon) = T + \varepsilon T^{(1)} + \mathcal{O}(\varepsilon^2), \tag{20}$$

with $T = \overline{P}_{\mathcal{J}} \overline{P}_{\mathcal{J}}^{\intercal}$ and $T^{(1)} = \overline{P}_{\mathcal{J}} L_{\mathcal{J}}^{\intercal} + L_{\mathcal{J}} \overline{P}_{\mathcal{J}}^{\intercal}$. Therefore, $T(\varepsilon)$ is analytic in ε, and also symmetric. By applying the perturbation theory for symmetric matrices from [28, Section 6.2], we get that $T(\varepsilon)$ possesses an orthonormal basis of eigenvectors $\varphi_1(\varepsilon), \ldots, \varphi_N(\varepsilon)$ that are also analytic in ε and thus admit a Taylor expansion in ε:

$$\varphi_k(\varepsilon) = \varphi_k + \varepsilon \varphi_k^{(1)} + \mathcal{O}(\varepsilon^2). \tag{21}$$

Here the φ_k are the eigenvectors of the unperturbed matrix T, i.e., the left singular vectors of $\overline{P}_{\mathcal{J}}$ (the singular vectors used for the aggregate assignments). The first order perturbation error of the k-th left singular vector is therefore given by $\varphi_k^{(1)}$, for which an expression is given by the following theorem:

Theorem 3.8. *Let $\lambda_k(\varepsilon)$ be the k-th largest eigenvalue of the perturbed operator $T(\varepsilon)$, counting multiplicity. Let $Q_{1,\dots,R} : \mathbb{R}^N \to \mathbb{R}^N$ denote the orthogonal projection onto* $\operatorname{span}(\varphi_1, \dots, \varphi_R)$*, and let β_{kj} be coefficients such that*

$$Q_{1,\dots,R}\varphi_k^{(1)} = \sum_{j=1}^{R} \beta_{kj}\pi_j.$$

Then, for $k = 1, \dots, R$, the eigenvector $\varphi_k(\varepsilon)$ corresponding to $\lambda_k(\varepsilon)$ is of the form

$$\varphi_k(\varepsilon) = \varphi_k + \varepsilon\left(\sum_{j=1}^{R} \beta_{kj}\pi_j + \sum_{j=R+1}^{N} \langle \varphi_j, \varphi_k^{(1)}\rangle \varphi_j\right) + \mathcal{O}(\varepsilon^2). \tag{22}$$

Proof. The proof of this theorem is very similar to that of [21, Theorem 4.7].

For $k = 1, \dots, R$, let Q_k be projection onto the eigenspace of T corresponding to the eigenvalue λ_k. Note that this eigenspace may be multi-dimensional. Under perturbation, the eigenvalue λ_k will in general split into multiple eigenvalues of $T(\varepsilon)$, which we call the λ_k-group of eigenvalues (see [28, Sec. II.1.8]). According to [28, Sec. II.2.1], the perturbed projection operator $Q_k(\varepsilon)$ onto the combined eigenspaces of $T(\varepsilon)$ corresponding to the λ_k-group is analytic in ε and admits the Taylor expansion

$$Q_k(\varepsilon) = Q_k + \varepsilon Q_k^{(1)} + \mathcal{O}(\varepsilon^2).$$

According to [28, Sec. II.2.1 (2.14)], the first order error coefficient can be written as

$$Q_k^{(1)} = \sum_{\substack{j\in\{1,\dots,N\}\\ j\neq k}} \frac{1}{\lambda_k - \lambda_j}\big(Q_k T^{(1)} Q_j + Q_j T^{(1)} Q_k\big), \quad k = 1, \dots, R.$$

Let $Q_{1,\dots,R}$ be the orthogonal projection onto the eigenspace of T to the distinct eigenvalues $\lambda_1, \dots, \lambda_R$. Then for the corresponding perturbed projection holds

$$\begin{aligned} Q_{1,\dots,R}(\varepsilon) &= \sum_{i=1}^{R} Q_i(\varepsilon) \\ &= \sum_{i=1}^{R} Q_i + \varepsilon \sum_{i=1}^{R} \sum_{\substack{j\in\{1,\dots,N\}\\ j\neq i}} \frac{1}{\lambda_i - \lambda_j}\big(Q_i T^{(1)} Q_j + Q_j T^{(1)} Q_i\big) + \mathcal{O}(\varepsilon^2) \\ &= Q_{1,\dots,R} + \varepsilon \sum_{i=1}^{R} \sum_{j=R+1}^{N} \frac{1}{\lambda_i}\big(Q_i T^{(1)} Q_j + Q_j T^{(1)} Q_i\big) + \mathcal{O}(\varepsilon^2), \end{aligned} \tag{23}$$

where in the last line we used that the terms for $j \le R$ cancel out, and $\lambda_j = 0,\ j = R+1, \dots, N$. For the eigenvectors $\varphi_1(\varepsilon), \dots, \varphi_R(\varepsilon)$ of $T(\varepsilon)$, we have that

$$\varphi_k(\varepsilon) = Q_{1,\dots,R}(\varepsilon)\varphi_k(\varepsilon), \quad k = 1, \dots, R. \tag{24}$$

Combining (21), (23) and (24) and using $Q_j\varphi_k = 0$ for $j = R+1, \dots, N$, we obtain

$$\varphi_k(\varepsilon) = Q_{1,\dots,R}\big(\varphi_k + \varepsilon\varphi_k^{(1)}\big) + \varepsilon \sum_{j=R+1}^{N} \frac{1}{\lambda_k} Q_j T^{(1)} \varphi_k + \mathcal{O}(\varepsilon^2).$$

Since $Q_{1,\dots,R}\varphi_k = \varphi_k$, and

$$Q_{1,\dots,R}\varphi_k^{(1)} = \sum_{j=1}^{R} \tilde{\beta}_{kj}\varphi_j = \sum_{j=1}^{R} \beta_{kj}\pi_j,$$

for some coefficients $\tilde{\beta}_{kj}, \beta_{kj} \in \mathbb{R}$, we can write the perturbed eigenvector as

$$\varphi_k(\varepsilon) = \sum_{j=1}^{R} \big(\gamma_{kj} + \varepsilon\beta_{kj}\big)\pi_j + \varepsilon \sum_{j=R+1}^{N} \frac{1}{\lambda_k} Q_j T^{(1)} \varphi_k + \mathcal{O}(\varepsilon^2).$$

This confirms the first sum in (22). The second summand, we can rewrite using the Euclidean inner product as

$$\sum_{j=R+1}^{N} \frac{1}{\lambda_k} Q_j T^{(1)} \varphi_k + \mathcal{O}(\varepsilon^2) = \sum_{j=R+1}^{N} \frac{1}{\lambda_k} \langle \varphi_j, T^{(1)} \varphi_k \rangle \varphi_j. \tag{25}$$

To derive an expression for $\langle \varphi_j, T(1)\varphi_k \rangle$, we combine the perturbation expansions (20) of $T(\varepsilon)$, (21) of $\varphi(\varepsilon)$, and the expansion of the eigenvalue $\lambda_k(\varepsilon)$,

$$\lambda_k(\varepsilon) = \lambda_k + \varepsilon\lambda_k^{(1)} + \mathcal{O}(\varepsilon^2). \tag{26}$$

Together, we obtain

$$\Big(T + \varepsilon T^{(1)} + \mathcal{O}(\varepsilon)\Big)\big(\varphi_k + \varepsilon\varphi_k^{(1)} + \mathcal{O}(\varepsilon)\big) = \big(\lambda_k + \varepsilon\lambda^{(1)} + \mathcal{O}(\varepsilon^2)\big)\big(\varphi_k + \varepsilon\varphi_k^{(1)} + \mathcal{O}(\varepsilon)\big). \tag{27}$$

Comparing the zero-th and first order terms in (27) yields

$$\begin{aligned} T\varphi_k &= \lambda_k \varphi_k, \\ T^{(1)}\varphi_k &= (\lambda_k I - T)\varphi_k^{(1)} + \lambda_k^{(1)}\varphi_k. \end{aligned}$$

Plugged into the above scalar product we get

$$\begin{aligned} \langle \varphi_j, T^{(1)}\varphi_k \rangle &= \langle \varphi_j, (\lambda_k I - T)\varphi_k^{(1)} + \lambda_k^{(1)}\varphi_k \rangle \\ &= \langle \varphi_j, (\lambda_k I - T)\varphi_k^{(1)} \rangle + \lambda_k^{(1)} \underbrace{\langle \varphi_j, \varphi_k \rangle}_{=0}, \end{aligned}$$

where the last term vanishes due to the φ being orthonormal as eigenvectors of a symmetric operator. Since $(\lambda_k I - T)$ is symmetric, we can rewrite the first term as

$$\begin{aligned} \langle \varphi_j, (\lambda_k I - T)\varphi_k^{(1)} \rangle &= \langle (\lambda_k I - T)\varphi_j, \varphi_k^{(1)} \rangle \\ &= (\lambda_k - \lambda_j)\langle \varphi_j, \varphi_k^{(1)} \rangle. \end{aligned}$$

Plugged into (25) and using $\lambda_j = 0$ for $j > R$, this finally yields

$$\varphi_k(\varepsilon) = \sum_{j=1}^{R} (\gamma_{kj} + \varepsilon\beta_{kj})\pi_j + \varepsilon \sum_{j=R+1}^{N} \langle \varphi_j, \varphi_k^{(1)} \rangle \varphi_j + \mathcal{O}(\varepsilon).$$

□

We will now further investigate the individual summands of Equation (22) in order to better understand their significance for the sign structure perturbation of the φ_k. As $\varphi_k \in \operatorname{span}(\pi_1, \ldots, \pi_R)$, we can write

$$\varphi_k = \sum_{j=1}^{R} \gamma_{kj}\pi_j$$

for some coefficients $\gamma_{kj} \in \mathbb{R}$. With that (22) can be written as

$$\varphi_k(\varepsilon) = \sum_{j=1}^{R} (\gamma_{kj} + \varepsilon\beta_{kj})\pi_j + \varepsilon \sum_{j=R+1}^{N} \langle \varphi_j, \varphi_k^{(1)} \rangle \varphi_j + \mathcal{O}(\varepsilon^2). \tag{28}$$

Hence, the β_{kj} represent the perturbation in the coefficients γ_{kj} of the unperturbed eigenvector φ_k. As the β_{kj} are independent of ε, this perturbation is small for small enough ε.

In any case however, even for large ε, the first summand in (28) is a linear combination of the π_j. Hence, as the π_j are non-negative and have support on the respective Ω_j, the first summand does again not change sign within the aggregates (although the particular sign structure may differ from that of φ_k if γ_{kj} and β_{kj} have different signs and $\varepsilon|\beta_{kj}| > |\gamma_{kj}|$.) Also, the perturbed eigenvectors $\varphi_k(\varepsilon)$ again form an orthonormal system. Hence, upon neglecting the remaining two summands in (28), the $\varphi_1(\varepsilon), \dots, \varphi_R(\varepsilon)$ fulfill the same prerequisites as the singular vectors $u_1, \dots, u_R$ in Lemma 3.2 (which are the unperturbed eigenvectors $\varphi_1, \dots, \varphi_R$). Therefore, if we could expect the second and third summand in (28) to not perturb the sign structure of the first summand, applying a spectral clustering algorithm to the perturbed eigenvectors $\varphi_1(\varepsilon), \dots, \varphi_R(\varepsilon)$ would reveal the aggregates perfectly as per Remark 3.3.

However, the second sum in (22) potentially does perturb the sign structure within the individual aggregates, as the non-dominant eigenfunctions φ_j, $j = R+1, \dots, N$ are in general not linear combinations of the π_i. To be precise, the second sum induces a sign change at position $i \in [N]$, if at that position the second sum has a different sign than the first sum, and

$$\varepsilon \left| \Big[\sum_{j=R+1}^{N} \left\langle \varphi_j, \varphi_k^{(1)} \right\rangle \varphi_j \Big]_i \right| > \left| \Big[\sum_{j=1}^{R} (\gamma_{kj} + \varepsilon\beta_{kj})\pi_j \Big]_i \right| \tag{29}$$

Again, (29) cannot hold true if ε is small enough, hence in this situation the sign structure of the $\varphi_k(\varepsilon)$ is again equal to that of the φ_k. For only moderately small ε, however, (29) needs to be checked on a case-by-case basis. Unfortunately, as the unperturbed eigenvectors φ_j and hence the coefficients γ_{kj}, β_{kj} are unknown, this condition cannot be checked numerically in practice. We are however able to state two easily-interpretable circumstances under which (29) is fulfilled, and argue that these circumstances are avoided automatically in many practically relevant systems:

1. A first condition that implies (29) is if the singular value λ_k is close to zero, because, due to (25),
$$\sum_{j=R+1}^{N} \left\langle \varphi_j, \varphi_k^{(1)} \right\rangle \varphi_j = \sum_{j=R+1}^{N} \frac{1}{\lambda_k} Q_j T^{(1)} \varphi_k,$$
where Q_j is the orthogonal projection onto the j-th distinct eigenspace of T. In many real-world settings, however, for example if the aggregates correspond to metastable sets, a spectral gap after the last dominant singular value is present, which then implies $\lambda_k \gg \lambda_{R+1} = 0$.

2. A second condition that implies (29) is if all π_j are approximately zero at position i, as
$$\Big[\sum_{j=1}^{R} (\gamma_{kj} + \varepsilon\beta_{kj})\pi_j \Big]_i = \sum_{j=1}^{R} (\gamma_{kj} + \varepsilon\beta_{kj})\pi_j(i).$$
For metastable systems, the π_j are the quasi-stationary densities on the Ω_j (see Section 2.3.1). These take near-zero values only in the statistically irrelevant border regions of the aggregates. For i in the "core aggregates", it holds $\pi_j(i) \gg 0$, hence no sign change will occur in these regions.

Of course, other circumstances such as particular combinations of γ_{kj} and β_{kj} with opposite sign may lead to (29) being fulfilled and have to be examined on a case-by-case basis. Overall, however, we can expect reasonable stability of Algorithm 3.1 with respect to perturbation of the transition matrix of form (12) if ε is small.

3.2. Recovery of the reduced transition matrix

Assume once more that P is an exactly aggregatable matrix, i.e., $P = \Pi \widehat{P} \Lambda$. Also assume that we now have knowledge of the aggregates $\Omega_1, \ldots, \Omega_R$, typically obtained by applying Algorithm 3.1 to P. We now explain how to compute the reduced transition matrix $\widehat{P}$, again using only a sparse, randomly-selected subset of the columns of P.

Knowing the index sets $\Omega_k \subset [N]$, we first assemble the aggregation matrix Λ via

$$\Lambda_{ri} := \begin{cases} 1 & \text{if } i \in \Omega_r \\ 0 & \text{otherwise} \end{cases},$$

as well as the diagonal matrix $M \in \mathbb{R}^{R \times R}$ of aggregate cardinalities:

$$M := \operatorname{diag}(m_1, \ldots, m_R), \quad m_r = |\Omega_r|.$$

Note that $\Lambda\Pi = I \in \mathbb{R}^{R \times R}$, and $\Lambda\Lambda^\intercal = M$. Hence, if the full transition matrix P were known, one could recover $\widehat{P}$ by applying Λ and $\Lambda^\intercal M^{-1}$ from left and right to P:

$$\Lambda P \Lambda^\intercal M^{-1} \overset{(7)}{=} \underbrace{\Lambda\Pi}_{=I} \widehat{P} \underbrace{\Lambda\Lambda^\intercal}_{=M} M^{-1} = \widehat{P}. \tag{30}$$

Alternatively, $\widehat{P}$ can also be recovered by column-normalizing the matrix $\Lambda P \Lambda^\intercal$.

3.2.1. Probabilistic matrix recovery

Now suppose that we are again in the situation where assembling the whole matrix P is numerically infeasible, i.e., only a small subset of the columns of P can be computed. Let $\mathcal{K} \subset [N]$ be an index set with $\omega(\mathcal{K}) = [R]$ and let the column-subsampled matrices $P_\mathcal{K}$ and $\Lambda_\mathcal{K}$ be defined as in (1). While $\mathcal{K}$ does not need to coincide with $\mathcal{J}$ from Section 3.1 and can trivially be chosen to fulfill $\omega(\mathcal{K}) = [R]$ if the Ω_r are known, choosing $\mathcal{K} = \mathcal{J}$ has the advantage that no additional computations need to be performed to compute $P_\mathcal{K}$. Applying Λ from the left and $\Lambda_\mathcal{K}^\intercal$ from the right to the column-sparse matrix $P_\mathcal{K}$ instead of P also recovers $\widehat{P}$:

Proposition 3.9. *Let k_r be number of indices in $\mathcal{K}$ that belong to aggregate Ω_r, i.e.,*

$$k_r = |\mathcal{K} \cap \Omega_r|.$$

Define the diagonal matrix $K := \operatorname{diag}(k_1, \ldots, k_R)$. Then

$$\Lambda P_\mathcal{K} \Lambda_\mathcal{K}^\intercal K^{-1} = \widehat{P}. \tag{31}$$

Proof. Consider the identity matrix $I \in \mathbb{R}^{N \times N}$, and $I_\mathcal{K}$ as defined by (1). The column-sampled matrix $P_\mathcal{K}$ can then be written as $P_\mathcal{K} = P \cdot I_\mathcal{K}$. With that and (7), the product in (31) becomes

$$\Lambda P_\mathcal{K} \Lambda_\mathcal{K}^\intercal K^{-1} = \underbrace{\Lambda\Pi}_{=I} \hat{P} \Lambda I_\mathcal{K} \Lambda_\mathcal{K}^\intercal K^{-1}.$$

The assertion follows from the identity $\Lambda I_\mathcal{K} \Lambda_\mathcal{K}^\intercal = \operatorname{diag}(k_1, \ldots, k_r)$. □

Thus, if $k_r \neq 0$ for all $r \in [R]$, then $\widehat{P}$ can be recovered from $\mathcal{P}_\mathcal{K}$. Again, in the case where the numerical effort is dominated by the computation of the entries of P, being able to restore $\widehat{P}$ from (31) instead of (30) provides a computational speedup of factor $N/|\mathcal{K}|$, as to assemble $P_\mathcal{K}$, only $|\mathcal{K}|$ columns of P have to be computed.

This leads to the following algorithm for the recovery of $\widehat{P}$:

Algorithm 3.2 Recovery of the reduced transition matrix $\widehat{P}$.

Input: Aggregates $\Omega_1, \dots, \Omega_R$ of P,
Ability to compute individual columns of the transition matrix P

1: Choose an index set $\mathcal{K} \subset [N]$, so that $\omega(\mathcal{K}) = [R]$.
2: Compute the columns of P with indices in $\mathcal{K}$ and assemble the matrix $P_{\mathcal{K}}$.
3: Assemble the matrices $\Lambda, \Lambda_{\mathcal{K}}$ and K via

$$\Lambda_{ri} := \begin{cases} 1 & \text{if } i \in \Omega_r \\ 0 & \text{otherwise} \end{cases}, \qquad \left(\Lambda_{\mathcal{K}}\right)_{ri} := \begin{cases} 1 & \text{if } i \in \Omega_r \text{ and } r \in \mathcal{K} \\ 0 & \text{otherwise} \end{cases},$$

and

$$K := \operatorname{diag}(k_1, \dots, k_R), \quad k_r := |\mathcal{K} \cap \Omega_r|.$$

4: Compute $\widehat{P} := \Lambda P_{\mathcal{K}} \Lambda_{\mathcal{K}}^{\intercal} K^{-1}$.

Output: Aggregated transition matrix $\widehat{P}$

3.2.2. Applicability to almost aggregatable Markov chains

We now again consider the case where P is only ε-almost aggregatable, i.e.,

$$P = \overline{P} + E, \quad \|E\|_1 \leq 4\varepsilon$$

where $\overline{P}$ is aggregatable with $\overline{P} = \Pi \widehat{P} \Lambda$. Like for the recovery of the aggregates, we want to compute an approximation to the matrix $\widehat{P}$ by applying Algorithm 3.2 to the perturbed matrix P (or rather, the column-subsampled matrix $P_{\mathcal{K}}$). The error in that approximation, dependent on the perturbation ε, is invesitaged in this section.

We limit the investigation to the situation where the aggregates, hence the matrices $\Lambda_{\mathcal{K}}$ and K, are known exactly. In this situation, the difference between the computed and the true reduced transition matrix is given by

$$\left\|\Lambda P_{\mathcal{K}} \Lambda_{\mathcal{K}}^{\intercal} K^{-1} - \widehat{P}\right\|_1 = \left\|\Lambda E_{\mathcal{K}} \Lambda_{\mathcal{K}}^{\intercal} K^{-1}\right\|_1 \leq \underbrace{\left\|\Lambda\right\|_1}_{=1} \left\|E_{\mathcal{K}}\right\|_1 \left\|\Lambda_{\mathcal{K}}^{\intercal} K^{-1}\right\|.$$

For the second factor on the right hand side holds $\left\|E_{\mathcal{K}}\right\|_1 \leq \left\|E\right\|_1 \leq 4\varepsilon$. The r-th column of the matrix $\Lambda_{\mathcal{K}}^{\intercal} K^{-1}$ contains k_r-times the value $\frac{1}{k_r}$, and only zeros otherwise, hence $\left\|\Lambda_{\mathcal{K}}^{\intercal} K^{-1}\right\|_1 = 1$. Overall, we get

$$\left\|\Lambda P_{\mathcal{K}} \Lambda_{\mathcal{K}}^{\intercal} K^{-1} - \widehat{P}\right\|_1 \leq 4\varepsilon.$$

Hence, by applying Algorithm 3.2 to an ε-almost aggregatable P, we can expect an L^1-error in $\widehat{P}$ of order ε. Notably, this error is again independent of the typically large size N of the original model.

4. Numerical experiments

4.1. A generic almost aggregatable process

We consider a 500-state Markov jump process with transition matrix P that we explicitly construct to be almost aggregatable, i.e., that is close to an aggregatable transition matrix. To this end, we choose the matrices $\Lambda, \Pi, \widehat{P}$ and E so that $\overline{P} := \Pi \widehat{P} \Lambda$ is aggregatable (see Proposition 2.6), E is a matrix with column-sum zero and $\|E\|_1 \leq \varepsilon := 0.1$, and $P := \overline{P} + E$ is a stochastic matrix.

We first subdivide the state space $\{1, \dots, 500\}$ randomly into 10 aggregates of equal size 50. This defines the matrix Λ. For the distribution vectors $\pi_r \in \mathbb{R}^{500}$, $r = 1, \dots, 10$, we choose random

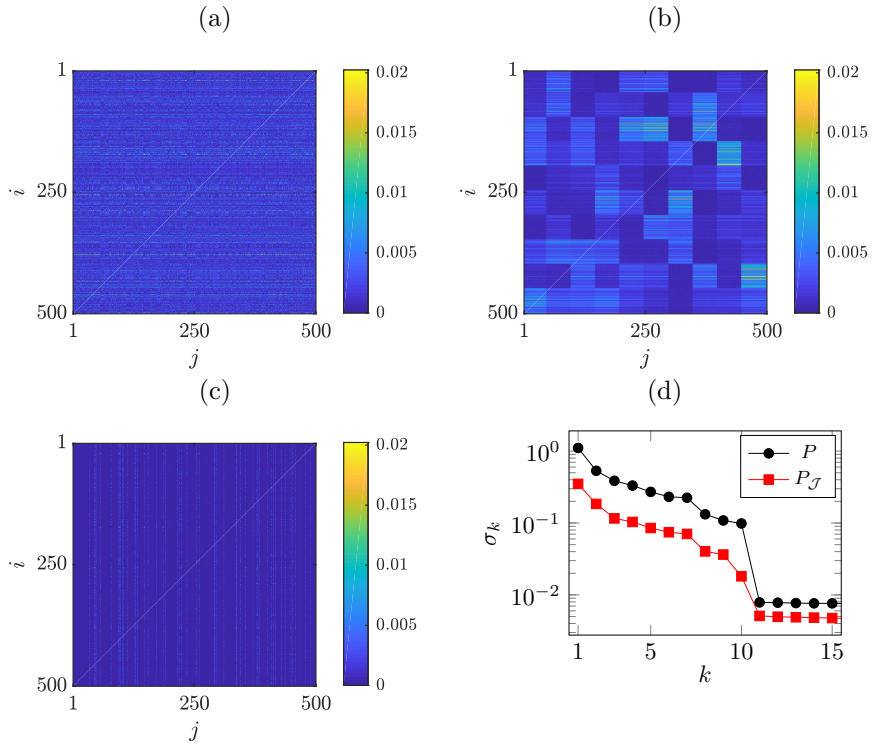

Figure 2: (a) Full transition matrix P of the almost aggregatable process. (b) Full transition matrix P permuted so that each 50 consecutive columns and rows correspond to one aggregate. (c) Subsampled transition matrix $P_{\mathcal{J}}$. (d) Leading singular values of P and $P_{\mathcal{J}}$. The spectral gaps after σ_{10} indicate that both matrices are approximately of rank 10.

distributions with support on the respective Ω_r using the method detailed in [47]. However, in order to avoid the perturbation effects discussed in Section 3.1.3 , we prohibit entries of π_r that are too small and whose signs are thus perturbed too easily. Specifically, we enforce $\pi_r(i) \geq 0.01$ for $i \in \Omega_r$. This defines the matrix Π.

The reduced transition matrix $\widehat{P}$ is constructed by randomly drawing a stochastic 10×10 matrix. We make sure here that the smallest non-zero singular value σ_{10} of $\overline{P} := \Pi\widehat{P}\Lambda$ is not too small, specifically $\sigma_{10} \geq 0.1$. The reason is again to avoid the perturbation effects discussed in Section 3.1.3. The matrix is shown in Figure 4 (a).

Finally, we randomly draw a matrix E with row-sum zero and scale it such that $\|E\|_1 = 0.1$. As E contains negative entries, the matrix $P := \Pi\widehat{P}\Lambda + E$ may not be positive, hence no stochastic matrix. We correct this fact by shifting the entries of P into the interval $[0, 1]$ and re-normalizing the columns.

The final transition matrix P used for our computations can be seen in Figure 2 (a). On close inspection, one can see that certain columns are equal. Indeed, upon sorting the rows and columns by aggregate number i.e., permuting P such that states of one aggregate appear in consecutive order, a block pattern becomes visible (Figure 2 (b)). This pattern bears similarity to the reduced transition matrix $\widehat{P}$ (Figure 4 (a)). Note that the sorted transition matrix serves only illustratory purposes, and will not be used in the following computations.

Computation of the aggregates. We employ Algorithm 3.1 in order to compute the aggregates of the (unpermuted) transition matrix P. For this we assume that the number $R = 10$ of aggregates is known in advance. We randomly choose an index set $\mathcal{J} \subset [500]$ with $|\mathcal{J}| = 50$ (Step 1 of the algorithm). Formula (18) then predicts a probability of 95% that at least one column from each

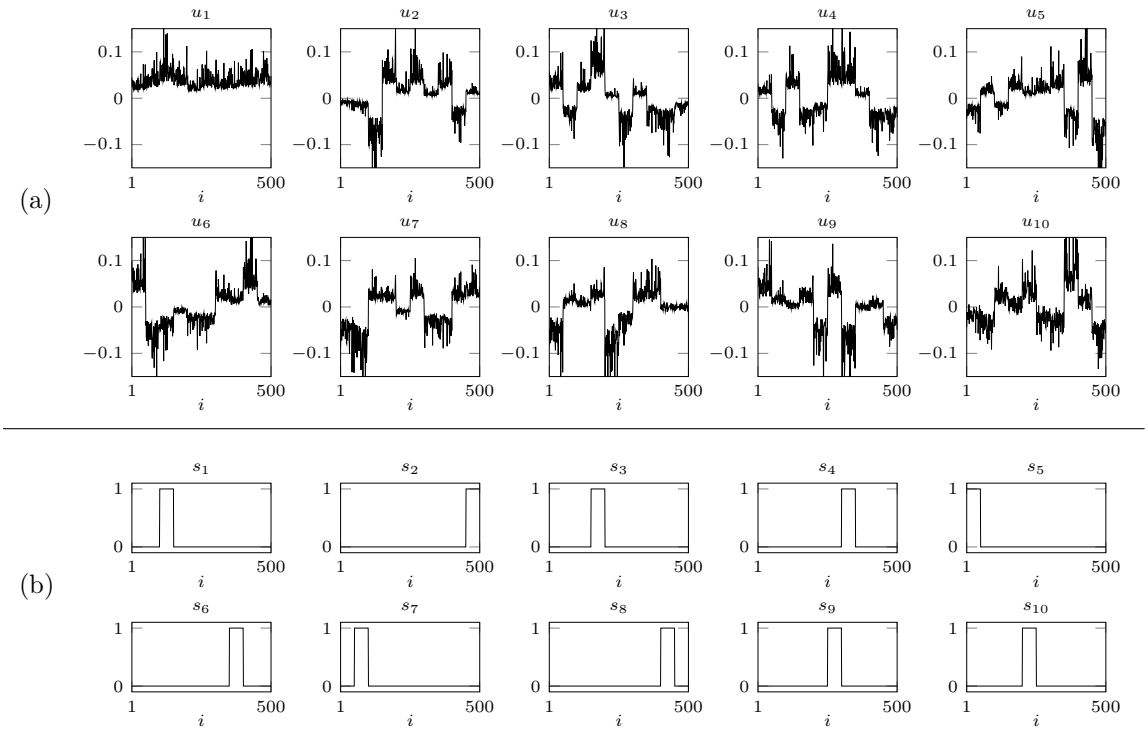

Figure 3: (a) Leading 10 left singular vectors $u_1, \ldots, u_{10}$ of the matrix $P_{\mathcal{J}}$, sorted by aggregate. We see a characteristic jumping pattern between the aggregates. (b) Output vectors $s_1, \ldots, s_{10}$ of the SEBA algorithm applied to $u_1, \ldots, u_{10}$, sorted by aggregate number. We see that they correspond to the indicator functions of the aggregates, albeit in no particular order. Hence, SEBA is able to correctly identifies the aggregates.

of the 10 aggregates is included in $\mathcal{J}$. Indeed, the gap after the singular value σ_{10} of $P_{\mathcal{J}}$ indicates that all ten aggregates have been hit.

Next, we assemble the matrix $P_{\mathcal{J}}$ (Step 2 of the algorithm). As in this example P is fully known in advance its columns do not have to be computed individually, and we can simply extract the columns with indices in $\mathcal{J}$ from P. The matrix $P_{\mathcal{J}}$ can be seen in Figure 2 (c). Note that in practical applications, each column of the transition matrix typically has to be computed individually, often from costly numerical simulations. In the present example, the ability to compute the aggregates based on $P_{\mathcal{J}}$, consisting of 50 non-zero columns instead of P, which consists of 500 non-zero columns, would therefore yield a computational speedup of factor 10.

Next we compute the leading $R = 10$ left singular vectors $u_1, \ldots, u_{10}$ of $P_{\mathcal{J}}$ (Step 3 of the algorithm). These vectors, with the entries sorted again by aggregate number for illustration purposes, are shown in Figure 3 (a). We observe sharp transitions between the aggregates, indicating that the singular vectors can indeed distinguish the individual aggregates. The irregular pattern within the individual aggregates is due to the randomly-chosen distributions π_r, and not (primarily) an effect of the random perturbation E.

Finally, we apply the SEBA spectral clustering algorithm to $u_1, \ldots, u_{10}$ (Step 4 of the algorithm). The output of SEBA are indicator vectors $s_1, \ldots s_{10} \in \mathbb{R}^N$, where $s_r(i) = 1$ indicates that $i \in \Omega_r$. We see that SEBA is able to correctly identify the aggregate affiliation of all states (Figure 3 (b)), despite the perturbation E.

Computation of the reduced transition matrix. Once we have computed the aggregates $\Omega_1, \ldots, \Omega_{10}$, Algorithm 3.2 lets us compute the reduced transition matrix $\widehat{P}$.

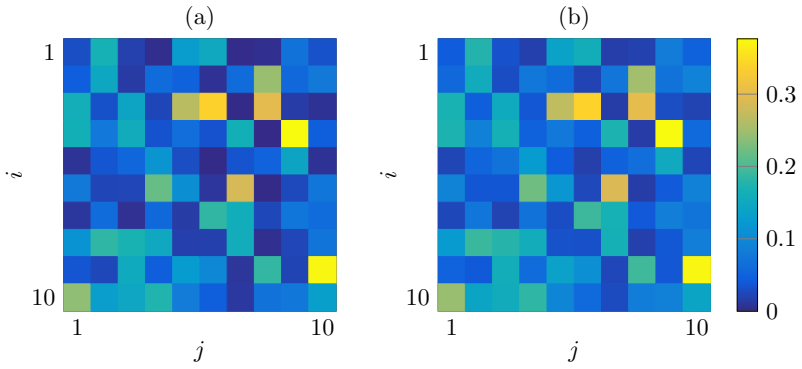

Figure 4: (a) Original reduced transition matrix $\widehat{P}$ that was used to construct the generic almost aggregatable transition matrix P. (b) Reduced transition matrix recovered from the subsampled transition matrix $P_{\mathcal{J}}$ via Formula (31).

In the first step of the algorithm, we use as the index set $\mathcal{K}$ the same index set that was used for the computation of the aggregates, i.e., $\mathcal{K} = \mathcal{J}$. The reason is that this way the already-computed matrix $P_{\mathcal{J}}$ can be re-used and, due to the success of Algorithm 3.1 in recovering the aggregates, we can be sure that $\omega(\mathcal{J}) = [R]$. Hence, the second step of the algorithm, computation of the non-zero columns of $P_{\mathcal{K}}$, is performed by simply setting $P_{\mathcal{K}} = P_{\mathcal{J}}$.

In the third step, the matrices $\Lambda, \Lambda_{\mathcal{K}} \in \mathbb{R}^{10\times 500}$ and $K \in \mathbb{R}^{10\times 10}$ are assembled, which is trivial under knowledge of the aggregates. Finally, in the fourth step, the reduced transition matrix is computed via the matrix product $\widehat{P}_{\text{restored}} := \Lambda P_{\mathcal{K}} \Lambda_{\mathcal{K}}^{\intercal} K^{-1}$. This matrix is shown in Figure 4 (b). We observe excellent quantitative agreement with the original reduced transition matrix (Figure 4 (a)).

4.2. A discretized metastable Langevin process

As shown in Section 2.3.1, metastable Markov processes represent an important special case of almost aggregatable processes. The associated coarse graining procedure is demonstrated by the following example.

We first consider the time- and space-continuous overdamped Langevin dynamics following the SDE

$$d\mathbf{X}_t = -\nabla V(\mathbf{X}_t)\, dt + \sqrt{\frac{2}{\beta}}\, d\mathbf{W}_t,$$

with inverse temperature β, Brownian motion $d\mathbf{W}_t$, and the two-dimensional potential energy function V depicted in Figure 5 (a). The system's unique stationary density is the Boltzmann density $\pi(x) = \frac{1}{Z} e^{-\beta V(x)}$, where Z is a normalization constant. We consider the system in the region $[-2, 2]^2$.

The potential has five local "energy wells", and at low enough temperature every point except those very close to the saddle points are attracted by exactly one of the wells. Typical trajectories are "trapped" in these wells for long times before eventually receiving enough energy through the stochastic part of the dynamics to jump out. The five segments shown in Figure 5 (a) are thus metastable[3].

To make this system accessible to our framework, we discretize it in time and space. We first fix an inverse temperature β and a lag time $\tau > 0$ that is long enough to observe local equilibration for each starting point (except the saddle points). Specifically, we choose $\beta = 1$, $\tau = 0.5$. Furthermore,

[3]Note that it is in fact debatable here whether the full segments or only the regions directly around the wells, the so-called *core metastable sets* [43], should be considered metastable. While our (see Section 2.3.1) and the original [17] definition of metastable systems requires a strict partition of the state space into metastable sets, other approaches accept the existence of so-called *transition states* that do not belong to any metastable set and that may possess significant statistical weight [42, 38].

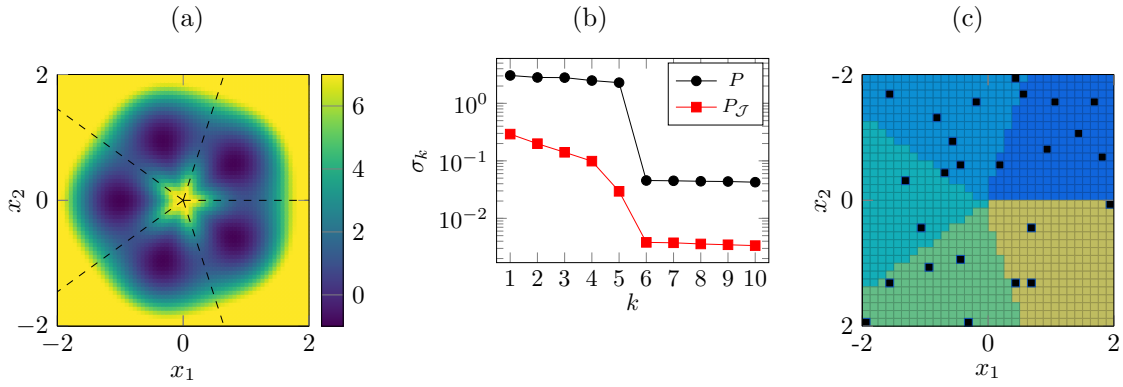

Figure 5: (a) Potential energy function V with five local minima. The dashed lines indicate the borders of the five metastable sets. (b) Leading singular values of the full transition matrix P. We see a spectral gap after the fifth singular value, indicating that P is approximately of rank 5. (c) Discretization boxes, forming the states of the discrete Markov process, along with the discrete metastable sets (colored background). The boxes marked in black correspond to the randomly-selected indices $\mathcal{J}$ used to generate the column-sparse matrix $P_{\mathcal{J}}$.

we subdivide the state space $\mathbb{X} = [-2, 2]^2$ into $N = 32 \cdot 32 = 1024$ boxes $B_1, \dots, B_N$ of equal size and consider the transition matrix $P \in \mathbb{R}^{N \times N}$

$$P_{ij} = \mathbb{P}\big[\mathbf{X}_\tau \in B_i \;\big|\; \mathbf{X}_0 \sim \mathbb{1}_{B_j}\big].$$

The Markov jump process induced by P is called the *Ulam discretization* of $\mathbf{X}_t$. For a discussion on how well it approximates the original process, see [20]. We expect the metastable sets of this Markov chain to correspond to the metastable sets of the continuous process that have been "discretized" over the boxes (shown in Figure 5 (c)).

We approximate the full transition matrix P column-wise, by starting $M = 10^5$ numerical simulations of length τ in each box B_j, and counting the transitions to each other box B_i. This is known as Ulam's method [49]. To be precise, the (i, j)-th entry of P is approximated by

$$P_{ij} \approx \frac{1}{M} \sum_{m=1}^{M} \mathbb{1}_{B_i}\Big(\Phi^\tau_{(m)}\big(x_j^{(m)}\big)\Big), \tag{32}$$

where the $x_j^{(m)}$, $m = 1, \dots, M$ are starting points that are uniformly randomly distributed in B_j, and $\Phi^\tau_{(m)}(x)$ is the endpoint of a numerically-realized trajectory with starting point x, length τ and random seed m. Of course, the full matrix P is only computed for comparison and benchmark purposes. For the computation of the aggregates via Algorithm 3.1, only a sparse subset of the columns needs to be computed (see below).

The leading singular values of P are shown in Figure 5 (b). The spectral gap after σ_5 indicates that P is approximately of rank 5, i.e. approximated well by a matrix of rank 5. This is expected, as the columns of P should consist only of approximations to the five quasi-stationary densities of the discretized metastable sets.

Computation of the aggregates. We again use Algorithm 3.1 to approximate the aggregates of P. For the number of randomly drawn column indices $\mathcal{J}$, we choose $J = 25$. By Formula (16), this will guarantee a probability of 95% to sample all five metastable sets, i.e., ensure $\operatorname{span}(P_{\mathcal{J}}) \approx \operatorname{span}(P)$. The boxes corresponding to our randomly-drawn indices are illustrated in Figure 5 (c).

The leading five left singular vectors $u_1, \dots, u_5$ of $P_{\mathcal{J}}$ are shown in Figure 6 (a). Applying the SEBA spectral clustering algorithm to $u_1, \dots, u_5$ identifies aggregates that correspond very well to the core metastable sets of the original continuous process (Figure 6 (b)).

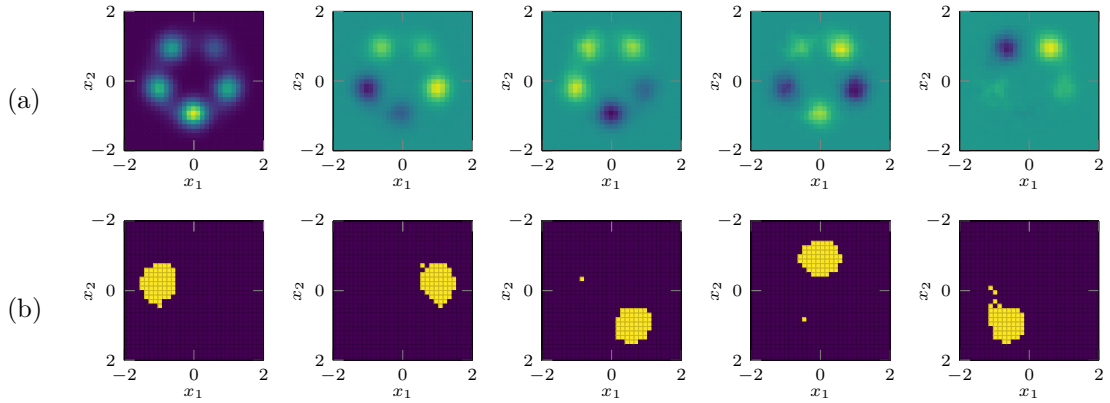

Figure 6: (a) Leading five singular vectors of $P_{\mathcal{J}}$. (b) Output vectors of SEBA clustering applied to the singular vectors. The cores of the metastable sets are clearly recognizable.

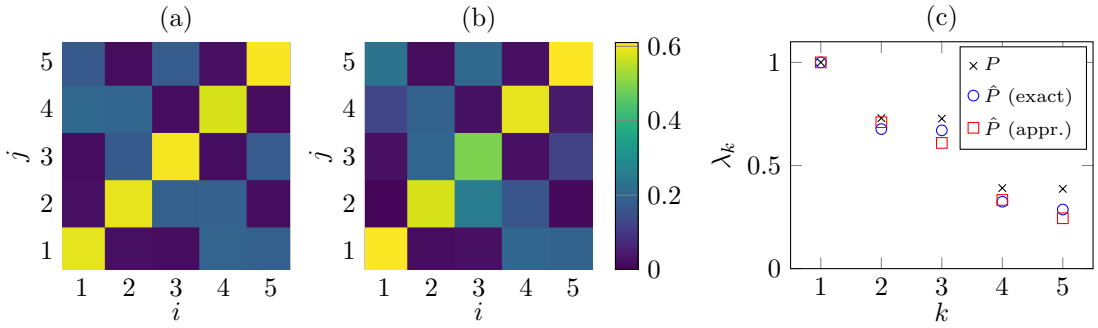

Figure 7: (a) Exact reduced transition matrix $\widehat{P}$, computed with exactly-known aggregates and the full transition matrix P via (30). (b) Our approximation to $\widehat{P}$, computed via (31) with aggregates approximated via Algorithm 3.1 and with the sparse transition matrix $P_{\mathcal{K}}$. (c) Leading five eigenvalues of the full transition matrix, the exactly-computed reduced transition matrix, and our approximation to the reduced transition matrix.

Note however that the outer and the transition regions have not been included in the metastable sets. This is an effect of the singular vectors being almost zero in these regions, and the SEBA algorithm omitting such regions for stability reasons. These almost-zero regions are of little statistical importance, and we will see in the next section that omitting them has practically no consequence when recovering the reduced transition matrix. Moreover, the recovery of only the core metastable sets can also be seen as advantageous, due to the aforementioned ambiguity in the definition of metastability.

Computation of the reduced transition matrix. As a benchmark, we first compute the exact reduced transition matrix $\widehat{P}$ via Formula (30), i.e., using the full transition matrix P, and an aggregation matrix Λ that was assembled using the analytically-known metastable sets from Figure 5 (c). The result is shown in Figure 7 (a).

We compare it to $\widehat{P}$ computed via Formula (31). For this we use only the column-subsampled transition matrix $P_{\mathcal{K}}$ (where we again choose $\mathcal{K} = \mathcal{J}$). Furthermore, the aggregation matrix Λ was assembled using the aggregates that were computed in the previous section. The resulting matrix is shown in Figure 7 (b). We observe good element-wise agreement. Likewise, a comparison of the eigenvalues of the two matrices (Figure 7 (c)) shows very good quantitative agreement.

Computational effort. For Algorithm 3.1, only the J non-zero columns of the matrix $P_{\mathcal{J}}$ are required, thus MJ simulations have to be performed. As we have chosen $\mathcal{K} = \mathcal{J}$ no additional simulations have to be performed for the computation of the reduced transition matrix. Compared to the MN simulations necessary to assemble and analyze the full transition matrix P, this represents a speedup of factor N/J, or about 41 in our example.

4.3. Aggregation of Manhattan taxi trips

We now demonstrate that the method can be successfully applied to real-life data sets that only loosely fulfill the analytic requirements of aggregability. For this, we analyze a record of $1.1 \cdot 10^7$ taxi trips performed in and around Manhattan island in January 2016. The data was released by the NYC Taxi & Limousine Commission and is freely available under [1]. In particular, the data contains the start- and end time of trips, as well as the geographical coordinates of the entry and exit point. We investigate whether our method can aggregate Manhattan into disjoint regions based on patterns in the destination of trips taken in the morning. The same data set was analyzed in [52] with the same goal but using a different methodology.

We divide the relevant region into a square grid of a total of 3150 boxes and sort the data points into the boxes according to the trip starting location. Hereby, only trips beginning between 6:00 AM and 11:59 AM are considered. Subsequently, boxes with less than 1000 points are discarded for stability reasons. On the remaining 601 boxes, we assemble a transition matrix $P \in \mathbb{R}^{601\times 601}$. The duration of the individual trips is ignored, i.e., in our model, each trip takes one unit of time to complete.

Following the argumentation of Liu et al. [31], we assume that the Markov chain induced by P is a good model for the underlying "taxi commuter dynamics" of Manhattan. Furthermore, we conjecture that this dynamics indeed fulfills the lumpability and deflatability prerequisites of our method, at least approximately, and we will give speculative justifications below. Note however that we will neither analytically nor empirically verify the justifications; rather, they should only provide a sufficient reason for heuristically applying our new method.

For one, it is plausible that for two starting boxes inside a sufficiently homogeneous district, the probabilities to journey to another district are almost identical. For example, for starting boxes from a specific residential district, the probabilities to journey to a nearby commercial district may be uniformly high, whereas the probabilities to journey to some other residential district may be uniformly low. This would imply (almost) *lumpability* of the Markov chain, with the aggregates being the respective districts.

At the same time, one can conjecture that the probability to journey to a specific box inside a destination district is determined mainly by some distribution on the destination district itself, and not so much be the exact starting box. Again using the example of a morning commute from a residential to a commercial district, workers from each starting box (which at our resolution covers multiple blocks) may "spread out" over the entire commercial district according to a certain distribution that reflects the density of businesses inside the commercial district. Hence the probability to journey from box i in the residential district to box j in the commercial district is given by the probability to journey from i to the commercial district in general, multiplied by the value of the aforementioned distribution at j. This is the alternate definition of (almost) *deflatability* (4), again with the districts as the aggregates.

Note that metastability on the other hand is not necessarily a reasonable assumption here, as very short trips within districts could be covered by other means of transportation, such as walking or biking. Figure 8 (a) shows the leading singular values of P. While P is far from low-rank, we do observe a spectral gap after σ_4, indicating the existence of four aggregates. We thus choose $R = 4$ for the following aggregation procedure. Note however, since this estimation is based on the SVD of the full transition matrix, we essentially require $R = 4$ to be known in advance.

Computation of the aggregates. We now employ Algorithm 3.1 in order to compute the leading singular vectors of P based on a sparse column sampling. Subsequently, we apply the SEBA algorithm to the sign structure of the four leading singular vectors in order to extract the aggregates.

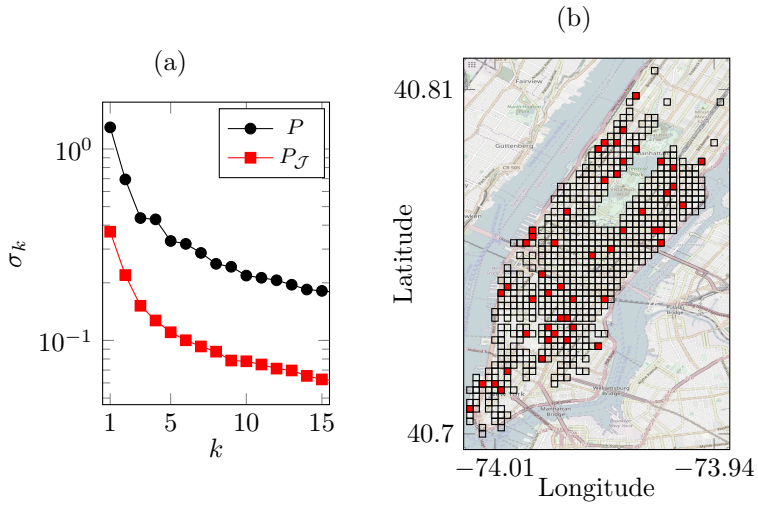

Figure 8: (a) Leading singular values of the full transition matrix P and the subsampled matrix $P_{\mathcal{J}}$. Note that the slight spectral gap of P after σ_4 essentially vanished under subsampling. (b) Boxes of the discretization with more than 1000 trips, forming the states of our Markov chain. Boxes selected for the aggregation algorithm are marked in red.

Application of SEBA to the sign structure instead of the raw singular vectors counteracts the tendency of SEBA to omit the border regions of aggregates. However, it also results in slighly more noisy aggregates, as boxes with low singular vector absolute value, i.e., high potential for erroneous assignment, are "forced" an assignment.

For the column sample size, we choose $J = 50$, which by (18) results in a near 100% chance to hit all aggregates ($p > 0.99$). However, in this practical example it is somewhat questionable whether the a-priori assumption that all aggregates have exactly the same size really holds. Hence, a comparatively large value of J was used here to compensate for possible inaccuracies of Equation (18). The boxes corresponding to the uniformly randomly-selected columns $\mathcal{J}$ are shown in Figure 8 (b). Figure 9 (a) shows the result of the algorithm. The individual aggregates are (for the most part) connected and approximately correspond to Lower Manhattan (Aggregate 1), Midtown Manhattan (Aggregate 2), Upper West Side (Aggregate 3) and Upper East Side (Aggregate 4). The former two consist of mostly commercial and manufacturing districts, but also contain smaller residential neighbourhoods, whereas the latter two contain mainly residential districts [2]. Despite being based on less that one-tenth of the data, our results are in excellent qualitative agreement with the analysis of Zhu et al. [52, Figure 3]. We also observe good agreement with the aggregates computed from the full transition matrix P (Figure), although there appears to be one systematic difference (Aggregate 4 seems to have "shifted down" into Aggregate 2). This may however again be an artifact of the unconventional application of SEBA, as the difference disappears when applying SEBA to the singular vectors directly.

Computation of the reduced transition matrix. We proceed to compute the reduced transition matrix $\widehat{P}$ using the method detailed in Section 3.2, where we use the same randomly-selected columns of P as for the aggregate computation, i.e., $\mathcal{K} = \mathcal{J}$. The transition matrix, shown in Figure 10 (a), confirms the suspected roles of the identified zones: We see that taxi trips from commercial areas (Aggregates 1 & 2) to the residential areas (Aggregates 3 & 4) are very rare in the morning. Midtown on the other hand appears to be the primary destination for commuters from Lower Manhattan and Upper East Side, which is explained by its status as the central business district of the city. The only surprise here is that Upper East Side seems to be a (slightly) more popular commuting destination than Midtown for residents of Upper West Side. One possible explanation is that the commercial north eastern parts of Midtown were assigned to the Upper

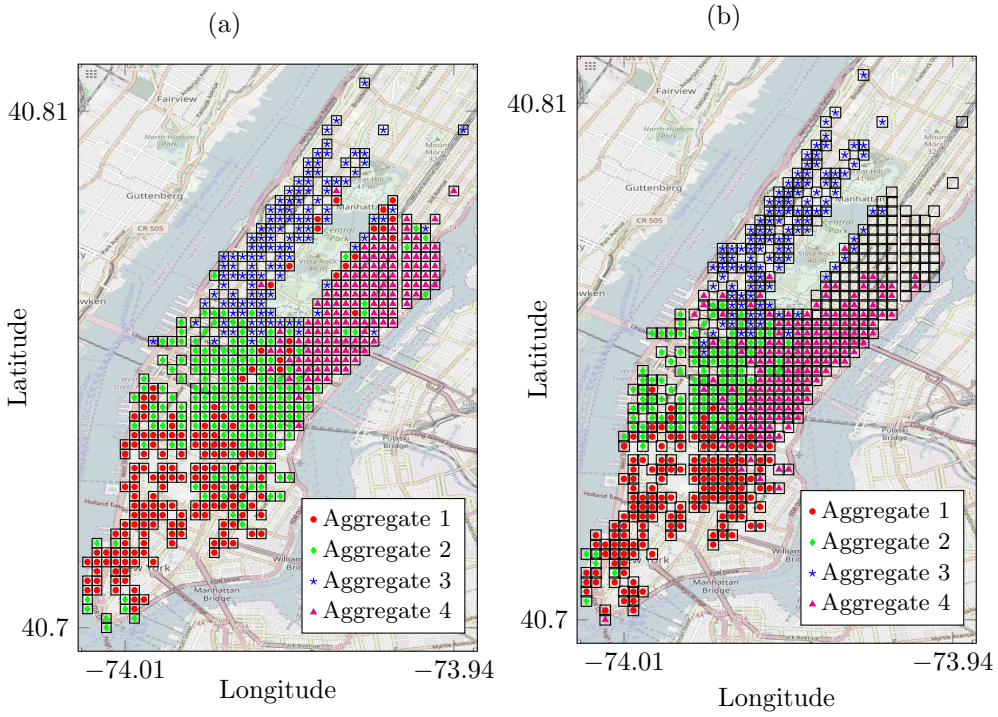

Figure 9: (a) Aggregates identified by Algorithm 3.1. Note that the SEBA algorithm may fail to assign a state to an aggregate, so the aggregates do not form a full partition. (b) Aggregates identified by spectral clustering of the singular vectors of the full transition matrix P.

(a)

$$\widehat{P} = \begin{bmatrix} 0.44 & 0.28 & 0.13 & 0.12 \\ 0.33 & 0.49 & 0.19 & 0.21 \\ 0.10 & 0.08 & 0.42 & 0.14 \\ 0.13 & 0.15 & 0.25 & 0.53 \end{bmatrix}$$

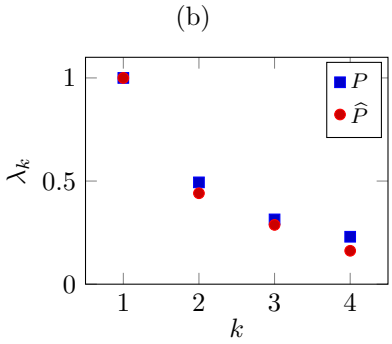

Figure 10: (a) Reduced transition matrix computed via (31). (b) Leading eigenvalues of the full vs. the reduced transition matrix.

East Side aggregate by our algorithm, and that Upper West Side residents might be commuting to this area. The reader should also be aware that this interpretation describes taxi commuters only, which may follow different and possibly unintuitive dynamics compared to general commuters.

Moreover, we observe moderately high metastability of all the aggregates (i.e., many trips begin and end in the same aggregate) which indicates that in Manhattan, even short journeys are often performed by taxi.

Finally, the comparison of the leading four eigenvalues of the full (i.e., 601×601) and the reduced (i.e., 4×4) transition matrices confirms that the reduced Markov chain captures the dominant processes of the full chain very well (Figure 10 (b)).

5. Conclusions

In this article, we derived a data-driven model reduction algorithm for large-scale Markov chains. Crucially, the number of columns of the transition matrix required by the algorithm, i.e., the number of states for which the outgoing transition probabilities have to be known, depends only on the size of the reduced model, not the full model. We have demonstrated that in applications where the computation of the transition matrix is the computational bottleneck, this can easily lead to a speedup of factor 10 or more over conventional model reduction algorithms. In a certain sense, the new method is able to circumvent the *curse of dimensionality* in model reduction in a similar way that methods from compressed sensing can circumvent the Nyquist–Shannon sampling theorem in signal processing and -compression.

In order to achieve this, the algorithm exploits a specific low-rank structure in the system's transition matrix. This low-rank structure has been shown to be induced by two natural similarity conditions in the inflow and outflow probabilities of the states. We have argued that these conditions are readily justifiable for a broad range of Markov chains for which the existence of a reduced chain can expected. Importantly, the class of *metastable* Markov chains fulfills these conditions. Moreover, in case where the formal conditions are only approximately fulfilled, we have shown that the error in the reduced model is of the same order of magnitude as the perturbation, and again is independent of the size of the full chain.

Future work. We expect the new method to be applicable to a wide variety of Markov chains that are suspected to possess an underlying low-rank structure. Moreover, the central requirements of our method, lumpability and deflatability, seem to be readily transferable to time- or space-continuous Markov models. For example, the recently-introduced class of continuous dynamical systems that possess a so-called *transition manifold* is characterized by the fact that its transition probability functions cluster around a low-dimensional manifold in a certain function space [7]. We expect this defining property to be connectable to a continuous version of lumpability and deflatability.

One tempting application of the new method is the conformation analysis of large biomolecules. However, as the dimension of the underlying continuous system ranges in the order of 10^2 to 10^5, the simple box-based Ulam discretization from Section 4.2 leads to difficulties. The first hurdle is to represent and address the sheer amount of boxes numerically, which however can be overcome by clever indexing. A bigger problem is that in this scenario, the number of boxes forming even the core metastable sets is higher than any practicable number M of numerical simulations one would be able to perform. Thus, the simple Monte Carlo procedure detailed in (32) is unsuited to accurately approximate the transition distributions $P_{[:,j]}$, as many boxes of the core metastable sets would not get hit by trajectories. A possible solution would be to utilize smooth ansatz functions with global support instead of characteristic functions over boxes in (32), for example, via a meshfree Galerkin approximation method [50]. This way, each performed simulation contributes to estimating the prefactor of multiple (or possibly all) ansatz functions. It is however unclear if the Markov chain arising as discretization on such ansatz functions still exhibits lumpability and deflatability.

Acknowledgements

This research has been funded by Deutsche Forschungsgemeinschaft (DFG) through grant CRC 1114 "Scaling Cascades in Complex Systems", Project Number 235221301, Project B03 "Multi-level coarse graining of multiscale problems" and supported by Deutsche Forschungsgemeinschaft (DFG) through grant EXC 2046 "MATH+", Project Number 390685689, Project AA1-2 "Learning Transition Manifolds and Effective Dynamics of Biomolecules".

The authors would like to thank Stefan Klus for the reference to the lumpability property, Péter Koltai for the reference to the coupon collector's problem, as well as the anonymous reviewers for helpful comments and suggestions.

References

[1] New York City TLC taxi trip data. https://www1.nyc.gov/site/tlc/about/data.page. Accessed: 2019-10-02.

[2] New York City ZoLa zoning & land use map. https://zola.planning.nyc.gov/about#12.31/40.73531/-73.94643. Accessed: 2019-10-22.

[3] A. Ando and F. M. Fisher. Near-Decomposability, Partition and Aggregation, and the Relevance of Stability Discussions. *International Economic Review*, 4(1):53, Jan. 1963.

[4] C. Bartels and M. Karplus. Multidimensional adaptive umbrella sampling: Applications to main chain and side chain peptide conformations. *Journal of Computational Chemistry*, 18(12):1450–1462, 1997.

[5] R. Bellman. *Dynamic Programming*. Dover Books on Computer Science Series. Dover Publications, 2003.

[6] A. R. Benson, D. F. Gleich, and L.-H. Lim. The spacey random walk: A stochastic process for higher-order data. *SIAM Review*, 59(2):321–345, 2017.

[7] A. Bittracher, P. Koltai, S. Klus, R. Banisch, M. Dellnitz, and C. Schütte. Transition Manifolds of Complex Metastable Systems: Theory and Data-driven Computation of Effective Dynamics. *Journal of Nonlinear Science*, 28(2):471–512, 2017.

[8] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. Wiley-Interscience, New York, NY, USA, 1998.

[9] G. R. Bowman, D. L. Ensign, and V. S. Pande. Enhanced modeling via network theory: adaptive sampling of Markov state models. *Journal of chemical theory and computation*, 6(3):787–794, 2010.

[10] P. Buchholz. Lumpability and aggregation of Markovian submodels. 02 1995.

[11] J. D. Chodera, W. C. Swope, F. Noé, J.-H. Prinz, M. R. Shirts, and V. S. Pande. Dynamical reweighting: Improved estimates of dynamical properties from simulations at multiple temperatures. *The Journal of Chemical Physics*, 134(24):244107, 2011.

[12] T. Dayar and W. J. Stewart. Quasi Lumpability, Lower-Bounding Coupling Matrices, and Nearly Completely Decomposable Markov Chains. *SIAM Journal on Matrix Analysis and Applications*, 18(2):482–498, Apr. 1997.

[13] H. De Sterck, T. A. Manteuffel, S. F. McCormick, K. Miller, J. Pearson, J. Ruge, and G. Sanders. Smoothed Aggregation Multigrid for Markov Chains. *SIAM Journal on Scientific Computing*, 32(1):40–61, Jan. 2010. Publisher: Society for Industrial and Applied Mathematics.

[14] H. De Sterck, T. A. Manteuffel, S. F. McCormick, Q. Nguyen, and J. Ruge. Multilevel Adaptive Aggregation for Markov Chains, with Application to Web Ranking. *SIAM Journal on Scientific Computing*, 30(5):2235–2262, June 2008.

[15] H. De Sterck, K. Miller, G. Sanders, and M. Winlaw. Recursively Accelerated Multilevel Aggregation for Markov Chains. *SIAM Journal on Scientific Computing*, 32(3):1652–1671, Jan. 2010. Publisher: Society for Industrial and Applied Mathematics.

[16] F. Delebecque. A Reduction Process for Perturbed Markov Chains. *SIAM Journal on Applied Mathematics*, 43(2):325–350, Apr. 1983.

[17] M. Dellnitz and O. Junge. On the approximation of complicated dynamical behavior. *SIAM J. Num. Anal.*, 36(2):491–515, 1999.

[18] P. Deuflhard, W. Huisinga, A. Fischer, and C. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra Appl.*, 315(13):39–59, 2000.

[19] P. Deuflhard and M. Weber. Robust Perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications*, 398:161 – 184, 2005. Special Issue on Matrices and Mathematical Biology.

[20] J. Ding, T. Y. Li, and A. Zhou. Finite approximations of Markov operators. *Journal of Computational and Applied Mathematics*, 147(1):137 – 152, 2002.

[21] D. Fritzsche, V. Mehrmann, D. B. Szyld, and E. Virnik. An SVD approach to identifying metastable states of Markov chains. *Electronic Transactions on Numerical Analysis*, 29:46–69, 2008.

[22] G. Froyland. Statistically optimal almost-invariant sets. *Physica D: Nonlinear Phenomena*, 200(3):205 – 219, 2005.

[23] G. Froyland, C. P. Rock, and K. Sakellariou. Sparse eigenbasis approximation: Multiple feature extraction across spatiotemporal scales with application to coherent set identification. *Communications in Nonlinear Science and Numerical Simulation*, 77:81 – 107, 2019.

[24] S. Gerber, L. Pospisil, M. Navandar, and I. Horenko. Low-cost scalable discretization, prediction and feature selection for complex systems. *Science Advances*, 2019.

[25] G. D. Gesùa, T. Lelièvre, D. L. Peutreca, and B. Nectouxa. Jump Markov models and transition state theory: the quasi-stationary distribution approach. *Faraday Discussions*, 195:469–495, 2016.

[26] N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[27] M. N. Jacobi. A robust spectral method for finding lumpings and meta stable states of non-reversible Markov chains. *Electronic Transactions on Numerical Analysis*, 37(1):296–306, 2010.

[28] T. Kato. *Perturbation theory for linear operators*. Springer Verlag Berlin, 1995, Reprint of the 1980 edition.

[29] J. Kemeny and J. L. Snell. *Finite Markov Chains*. Springer-Verlag New York, 1976.

[30] A. Laio and F. L. Gervasio. Metadynamics: a method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science. *Reports on Progress in Physics*, 71(12):126601, 2008.

[31] Y. Liu, C. Kang, S. Gao, Y. Xiao, and Y. Tian. Understanding intra-urban trip patterns from taxi trajectory data. *Journal of Geographical Systems*, 14(4):463–483, 2012.

[32] M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

[33] L. Maragliano and E. Vanden-Eijnden. A temperature accelerated method for sampling free energy and determining reaction pathways in rare events simulations. *Chemical physics letters*, 426(1):168–175, 2006.

[34] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1995.

[35] J. S. Niedbalski, Kun Deng, P. G. Mehta, and S. Meyn. Model reduction for reduced order estimation in traffic models. In *2008 American Control Conference*, pages 914–919, June 2008.

[36] F. Noé and S. Fischer. Transition networks for modeling the kinetics of conformational change in macromolecules. *Current Opinion in Structural Biology*, 18(2):154 – 162, 2008. Theory and simulation / Macromolecular assemblages.

[37] S. B. Ozkan, K. A. Dill, and I. Bahar. Computing the transition state populations in simple protein models. *Biopolymers*, 68(1):35–46, 2003.

[38] S. Röblitz and M. Weber. Fuzzy spectral clustering by PCCA+: Application to Markov state models and data classification. *Adv. Data Anal. Classif.*, 7(2):147–179, June 2013.

[39] G. Rubino and B. Sericola. On weak lumpability in Markov chains. *Journal of Applied Probability*, 26(3):446–457, 1989.

[40] R. R. Sarukkai. Link prediction and path analysis using Markov chains. *Computer Networks*, 33(1):377 – 386, 2000.

[41] C. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard. A direct approach to conformational dynamics based on hybrid Monte Carlo. *J. Comput. Phys.*, 151(1):146–168, 1999.

[42] C. Schütte, F. Noé, J. Lu, M. Sarich, and E. Vanden-Eijnden. Markov state models based on milestoning. *J. Chem. Phys.*, 134(20), 2011.

[43] C. Schütte and M. Sarich. *Metastability and Markov State Models in Molecular Dynamics*. Courant Lecture Notes in Mathematics, 2013.

[44] D. E. Shaw, M. M. Deneroff, R. O. Dror, J. S. Kuskin, R. H. Larson, J. K. Salmon, C. Young, B. Batson, K. J. Bowers, J. C. Chao, M. P. Eastwood, J. Gagliardo, J. P. Grossman, C. R. Ho, D. J. Ierardi, I. Kolossváry, J. L. Klepeis, T. Layman, C. McLeavey, M. A. Moraes, R. Mueller, E. C. Priest, Y. Shan, J. Spengler, M. Theobald, B. Towles, and S. C. Wang. Anton, a Special-purpose Machine for Molecular Dynamics Simulation. *Commun. ACM*, 51(7):91–97, jul 2008.

[45] T. Shioyama and K. Tanaka. A new aggregation-disaggregation algorithm. *European Journal of Operational Research*, 83(3):655 – 669, 1995.

[46] W. M. Spears. A Compression Algorithm for Probability Transition Matrices. *SIAM Journal on Matrix Analysis and Applications*, 20(1):60–77, Jan. 1998.

[47] R. Stafford. Random vectors with fixed sum. MATLAB Central File Exchange, https://mathworks.com/matlabcentral/fileexchange/9700-random-vectors-with-fixed-sum, 2006.

[48] W. J. Stewart. *Introduction to the numerical solution of Markov chains*. Princeton Univ. Press, Princeton, NJ, 1994.

[49] S. M. Ulam. *A Collection of Mathematical Problems*. Interscience Publishers, New York, 1960.

[50] M. Weber. *Meshless Methods in Conformation Dynamics*. PhD thesis, FU Berlin, 2006.

[51] H. Wu and F. Noé. Probability distance based compression of hidden Markov models. *Multiscale Modeling & Simulation*, 8(5):1838–1861, 2010.

[52] Z. Zhu, X. Li, M. Wang, and A. Zhang. Learning Markov models via low-rank optimization. *arXiv preprint arXiv:1907.00113*, 2019.

A. Proof of Theorem 2.10

In order to show that almost aggregatable matrices are close to aggregatable matrices in the L^1 matrix norm, we proceed as follows: first, we show that almost aggregatable matrices fulfill an approximate version of state-wise lumpability. We then proceed to show that P is indeed close to a lumpable matrix $\mathcal{L}(P)$, and close to a deflatable matrix $\mathcal{D}(P)$. The final part of the proof then consists of showing that the matrix $\mathcal{D}\big(\mathcal{L}(P)\big)$ is both lumpable and deflatable, and ε-close to P.

Lemma A.1. *Let P be ε-almost aggregatable with respect to (Ω, π). Then*

$$\big\|P_{[:,j]} - P_{[:,k]}\big\|_1 \leq 3\varepsilon \quad \textit{for all } j,k \in [N] \textit{ with } \omega(j) = \omega(k). \tag{33}$$

Proof. We have

$$\begin{aligned}\sum_{i\in[N]} \big|P_{ij} - P_{ik}\big| \leq \sum_{i\in[N]} \Big|P_{ij} - \pi_{\omega(i)}(i) \sum_{l\in\Omega_{\omega(i)}} P_{lj}\Big| \\ + \sum_{i\in[N]} \Big|\pi_{\omega(i)}(i) \sum_{l\in\Omega_{\omega(i)}} P_{lj} - \pi_{\omega(i)}(i) \sum_{l\in\Omega_{\omega(i)}} P_{lk}\Big| \qquad (\star) \\ + \sum_{i\in[N]} \Big|\pi_{\omega(i)}(i) \sum_{l\in\Omega_{\omega(i)}} P_{lk} - P_{ik}\Big|\end{aligned}$$

The first and third summand are each less than ε, due to P being ε-almost deflatable. For the second summand we get, by splitting the outer sum into the sums over the individual aggregates,

$$\begin{aligned}(\star) &= \sum_{r\in[R]} \sum_{p\in\Omega_r} \Big|\pi_{\omega(p)}(p) \sum_{l\in\Omega_{\omega(p)}} \big(P_{lj} - P_{lk}\big)\Big| \\ &= \sum_{r\in[R]} \sum_{p\in\Omega_r} \Big(\pi_r(p)\Big|\sum_{l\in\Omega_r} \big(P_{lj} - P_{lk}\big)\Big|\Big) \\ &= \sum_{r\in[R]} \Big|\sum_{l\in\Omega_r} \big(P_{lj} - P_{lk}\big)\Big| \cdot \underbrace{\sum_{p\in\Omega_r} \pi_r(p)}_{=1} \\ &\leq \sum_{i\in[N]} \big|P_{ij} - P_{ik}\big| \leq \varepsilon,\end{aligned}$$

where the last inequality holds due to P being ε-almost lumpable. □

We call condition (33) 3ε-*almost state-wise lumpability* of P with respect to Ω.

Lemma A.2. *Let P be ε-almost state-wise lumpable with respect to Ω. Define the* lumping operator $\mathcal{L} : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ *by*

$$\mathcal{L}(A)_{ij} := \frac{1}{m_{\omega(j)}} \sum_{l\in\Omega_{\omega(j)}} A_{il}.$$

Then $\mathcal{L}(P)$ is a transition matrix that is state-wise lumpable with respect to Ω and it holds

$$\big\|P - \mathcal{L}(P)\big\|_1 \leq \varepsilon.$$

Proof. All columns of $\mathcal{L}(P)$ that belong to one aggregate are identical, hence $\mathcal{L}(P)$ is state-statewisewise state-wise lumpable.

Moreover, $\mathcal{L}(P)$ is a column-stochastic matrix:

$$\sum_{i\in[N]} \mathcal{L}(P)_{ij} = \sum_{i\in[N]} \frac{1}{m_{\omega(j)}} \sum_{l\in\Omega_{\omega(j)}} P_{il} = \frac{1}{m_{\omega(j)}} \sum_{l\in\Omega_{\omega(j)}} \underbrace{\sum_{i\in[N]} P_{il}}_{=1} = \frac{1}{m_{\omega(j)}} \sum_{l\in\Omega_{\omega(j)}} 1 = 1.$$

Finally, it holds for all $j \in [N]$

$$\begin{aligned}
\left\|P_{[:,j]} - \mathcal{L}(P)_{[:,j]}\right\|_1 &= \sum_{i\in[N]} \left| P_{ij} - \frac{1}{m_{\omega(j)}} \sum_{l\in\Omega_{\omega(j)}} P_{il} \right| \\
&= \sum_{i\in[N]} \left| \frac{1}{m_{\omega(j)}} \sum_{l\in\Omega_{\omega(j)}} \left(P_{ij} - P_{il}\right) \right| \\
&\leq \max_{l\in\Omega_{\omega(j)}} \sum_{i\in[N]} \left| P_{ij} - P_{il} \right| \\
&\leq \varepsilon.
\end{aligned}$$

In the last inequality we used the definition of ε-almost state-wise lumpability. This implies $\|P - \mathcal{L}(P)\|_1 \leq \varepsilon$. □

Lemma A.3. *Let P be ε-almost deflatable with respect to (Ω, π). Define the* deflating operator *$\mathcal{D} : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ by*

$$\mathcal{D}(A)_{ij} := \Big(\sum_{l\in\Omega_{\omega(i)}} A_{lj} \Big) \cdot \pi_{\omega(i)}(i).$$

Then $\mathcal{D}(P)$ is a transition matrix that is deflatable respect to (Ω, π), and it holds

$$\left\|P - \mathcal{D}(P)\right\|_1 \leq \varepsilon.$$

Proof. By construction, $\mathcal{D}(P)$ fulfills condition 4, hence is deflatable. Further, as condition (9) holds for P, we have

$$\left\|P_{[:,j]} - \mathcal{D}(P)_{[:,j]}\right\|_1 \leq \varepsilon \quad \text{for all } j \in [N].$$

This in turn implies $\left\|P - \mathcal{D}(P)\right\|_1 \leq \varepsilon$.

It remains to show is that $\mathcal{D}(P)$ is indeed a column stochastic matrix. This follows from

$$\begin{aligned}
\sum_{i\in[N]} \mathcal{D}(P)_{ij} &= \sum_{i\in[N]} \Big(\sum_{l\in\Omega_{\omega(i)}} P_{lj} \Big) \cdot \pi_{\omega(i)}(i) \\
&= \sum_{r\in[R]} \sum_{k\in\Omega_r} \Big(\sum_{l\in\Omega_{\omega(k)}} P_{lj} \Big) \cdot \pi_{\omega(k)}(k) \\
&= \sum_{r\in[R]} \sum_{k\in\Omega_r} \Big(\sum_{l\in\Omega_r} P_{lj} \Big) \cdot \pi_r(k) \\
&= \sum_{r\in[R]} \Big(\sum_{l\in\Omega_r} P_{lj} \Big) \underbrace{\sum_{k\in\Omega_r} \pi_r(k)}_{=1} \\
&= \sum_{i\in[N]} P_{lj} = 1.
\end{aligned}$$

□

Combining these three auxiliary results allows us to show Theorem 2.10:

Proof of Theorem 2.10. Because P is ε-almost aggregatable, P is 3ε-almost state-wise lumpable (Lemma A.1). Thus, due to Lemma A.2, $\mathcal{L}(P)$ is a 3ε-almost state-wise lumpable transition matrix, and

$$\left\|P - \mathcal{L}(P)\right\|_1 \leq 3\varepsilon.$$

Moreover, $\mathcal{L}(P)$ is ε-almost deflatable, i.e., (9) is fulfilled for $\mathcal{L}(P)$:

$$\begin{aligned}
&\sum_{i\in[N]} \Big|\mathcal{L}(P)_{ij} - \Big(\sum_{l\in\Omega_{\omega(i)}} \mathcal{L}(P)_{lj}\Big)\pi_{\omega(i)}(i)\Big| \\
&\qquad = \sum_{i\in[N]} \Big|\frac{1}{m_{\omega(j)}} \sum_{k\in\Omega_{\omega(j)}} P_{ik} - \Big(\sum_{l\in\Omega_{\omega(i)}} \frac{1}{m_{\omega(j)}} \sum_{k\in\Omega_{\omega(j)}} P_{lk}\Big)\pi_{\omega(i)}(i)\Big| \\
&\qquad = \sum_{i\in[N]} \Big|\frac{1}{m_{\omega(j)}} \sum_{k\in\Omega_{\omega(j)}} P_{ik} - \frac{1}{m_{\omega(j)}} \sum_{k\in\Omega_{\omega(j)}} \Big(\sum_{l\in\Omega_{\omega(i)}} P_{lk}\Big)\pi_{\omega(i)}(i)\Big| \\
&\qquad \leq \frac{1}{m_{\omega(j)}} \sum_{i\in[N]} \sum_{k\in\Omega_{\omega(j)}} \Big|P_{ik} - \Big(\sum_{l\in\Omega_{\omega(i)}} P_{lk}\Big)\pi_{\omega(i)}(i)\Big| \\
&\qquad = \frac{1}{m_{\omega(j)}} \sum_{k\in\Omega_{\omega(j)}} \underbrace{\sum_{i\in[N]} \Big|P_{ik} - \Big(\sum_{k\in\Omega_{\omega(i)}} P_{lk}\Big)\pi_{\omega(i)}(i)\Big|}_{=:(\star\star)} = (\star).
\end{aligned}$$

Because P is ε-almost deflatable, we have $(\star\star) \leq \varepsilon$, hence

$$(\star) \leq \frac{\varepsilon}{m_{\omega(j)}} \sum_{k\in\Omega_{\omega(j)}} 1 = \varepsilon.$$

Therefore, due to Lemma A.3, $\mathcal{D}\big(\mathcal{L}(P)\big)$ is a ε-almost deflatable transition matrix, and it holds

$$\big\|\mathcal{L}(P) - \mathcal{D}\big(\mathcal{L}(P)\big)\big\|_1 \leq \varepsilon.$$

Define $\overline{P} := \mathcal{D}\big(\mathcal{L}(P)\big)$ and $E := P - \overline{P}$. Then

$$\|E\|_1 = \big\|P - \mathcal{D}\big(\mathcal{L}(P)\big)\big\|_1 \leq \underbrace{\big\|P - \mathcal{L}(P)\big\|_1}_{\leq 3\varepsilon} + \underbrace{\big\|\mathcal{L}(P) - \mathcal{D}\big(\mathcal{L}(P)\big)\big\|_1}_{\leq \varepsilon} \leq 4\varepsilon.$$

□