A Simulation-Based Method for Aggregating Markov Chains

Kun Deng, Prashant G. Mehta, and Sean P. Meyn

***Abstract*— This paper addresses model reduction for a Markov chain on a large state space. A simulation-based framework is introduced to perform state aggregation of the Markov chain based on observations of a single sample path.**

The Kullback-Leibler (K-L) divergence rate is employed as a metric to measure the distance between two stationary Markov chains. Model reduction with respect to this metric is cast as an infinite-horizon average cost optimal control problem. In this way an optimal policy corresponds to an optimal partition of the state space with respect to the K-L divergence rate.

The optimal control problem is simplified in an approximate dynamic programming (ADP) framework: First, a relaxation of the policy space is performed, and based on this a parameterization of the set of optimal policies is introduced. This makes possible a stochastic approximation approach to compute the best policy within a given parameterized class. The algorithm can be implemented using a single sample path of the Markov chain. Convergence is established using the ODE method. Examples illustrate the theoretical results, and show remarkably low variance and fast convergence.

I. Introduction

Markov chain models are important to a number of applications in biology [1], computer science [2], economics [3] and building systems [4]. A fundamental problem for many of the Markov models that arise in applications is the large dimension of the state space. For example, Markov models of agent movement in a large building may have millions of states [5]. Aggregation of states represent perhaps the most straightforward approach to model reduction for Markov chains. It can be justified using a singular perturbation framework (see [6]) for *nearly completely decomposable* Markov chains (NCDMC) [7], [8].

In recent years, spectral methods have become popular for state aggregation problems [9]–[12]. These methods have been applied in diverse settings: [9] considers analysis of the nonlinear chaotic dynamics of Chua's circuit model, [10] concerns molecular models, and [4] treats transport phenomena in buildings. In each of these applications, the sign structure of the second eigenvector is used to obtain the partition information for defining super-states. The technique is closely related to the notion of *cut* in spectral graph theory (see [13]) and decomposition algorithms are popular in image segmentation [14], clustering [15] and graph partitioning [16].

The problem with a spectral-based solution of the aggregation problem is that it requires eigenvectors of the Markov chain, a difficult task for large dimensional problems. In many applications, the Markov chain is so large that it can only be used for simulations from an given initial condition [17]. Due to the high dimensionality of the state space, it may not be even possible to store the eigenvector let alone compute it. This motivates methods for model reduction via simulation and learning, the topic of this paper. In fact, a simulation based algorithm obtained as a variant of Oja's algorithm was introduced in [18]. Although consistent, the algorithm suffers from high variance.

The paper builds on earlier work reported in our recent paper [19]. There, we propose the use of Kullback-Leibler (K-L) divergence rate metric (see [20], [21]) for aggregating *stationary* Markov chains. A reduced order Markov model is obtained as a solution of an optimization problem with respect to this metric. Taking the bi-partition problem as an example, the solution is shown to be given by sign-structure of the second eigenvector consistent with the spectral theory of Markov models.

To confront the complexity issues, in this paper, we formulate the model reduction problem as an infinite-horizon average cost optimal control problem. The control objective is to obtain the partition function (policy) that minimizes the K-L metric (average cost). For small problems, model reduction can be obtained by directly using the methods of dynamic programming (DP). The key advantage of a DP based formulation, however, is that a number of simulation-based approximation methods exist to approximate the optimal value function and learn an optimal partition [22].

Based on these methods, the model reduction problem is simplified in an approximate dynamic programming (ADP) framework. First, a relaxation of the policy space is performed, and based on this a parameterization of the set of optimal policies is introduced. This makes possible a stochastic approximation approach to compute the best policy within a given parameterized class. The algorithm can be implemented online by simulating a single sample path. At each step in the iteration, the algorithm requires only a small number of calculations even for large problems. Convergence properties are established using standard stochastic approximation arguments (in particular, the ODE method), and illustrated with the aid of examples.

The outline of this paper is as follows. In Section II and III, the K-L metric and model reduction formulae from [19] are briefly reviewed. Section IV introduces the dynamic programming formulation of the K-L metric based model reduction problem. In Section V, a simulation-based algorithm for the DP is outlined along with a discussion of its convergence properties. In Sections VI and VII, examples and conclusions are described, respectively.

The authors are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 West Main Street, Urbana, IL 61801. Email: kundeng2@uiuc.edu, mehtapg@uiuc.edu, and meyn@control.csl.uiuc.edu

II. METRIC

A. Preliminaries and notations

We consider a first-order homogeneous Markov chain $\mathcal{X}_t$ defined on a finite dimensional state space $\mathcal{N} = \{1, 2, \ldots, n\}$ (see [23] for terminology). The following notations are adopted throughout the paper: The state value at time t is denoted as $X(t)$, the initial condition $X(0)$ is denoted as $x_0 \in \mathcal{N}$, and the sequence $\{X(1), X(2), \ldots, X(l)\}$ is denoted as $X_1^l \in \mathcal{N}^l$, where $\mathcal{N}^l \triangleq \mathcal{N} \times \cdots \times \mathcal{N}$ is the n-fold Cartesian product. $\mathcal{P}(\mathcal{N}^l)$ denotes the space of probability distribution on $\mathcal{N}^l$ and $\nu_0 \in \mathcal{P}(\mathcal{N})$ is used to denote the initial distribution of x_0. The transition probability between states is described by a $n \times n$ stochastic matrix P whose ij^{th} entry is given by

$$P_{ij} = \text{Prob}(X(t+1) = j \mid X(t) = i), \quad i, j \in \mathcal{N}.$$

Assumption 1 *All Markov chains considered in this paper are* ergodic *chains, i.e. they are irreducible and aperiodic.*

Under Assumptions 1, a Markov chain is further said to be *stationary* if it has a unique *stationary distribution* π such that

$$\pi = \pi P,$$

where $\pi_i > 0$ for all $i \in \mathcal{N}$. We use the tuple (π, P) to denote a stationary Markov chain P with a stationary distribution π.

B. K-L divergence rate for Markov chains on $\mathcal{N}$

For two stationary Markov chains (π, P) and (ϖ, Q) defined on the same state space $\mathcal{N}$, the K-L divergence rate is given by following formula (see [21]):

$$R(P \parallel Q) = \sum_{i,j \in \mathcal{N}} \pi_i P_{ij} \log\left(\frac{P_{ij}}{Q_{ij}}\right). \tag{1}$$

To ensure $R(P \parallel Q)$ is finite, we require P to be *absolutely continuous* w.r.t. Q, i.e. $Q_{ij} = 0 \Rightarrow P_{ij} = 0$.

C. K-L divergence rate for Markov chains on different state spaces

For model reduction, it is of interest to compare two Markov chains P and Q defined on $\mathcal{N}$ and $\mathcal{M}$ respectively. The relationship between $\mathcal{N}$ and $\mathcal{M}$ is described by a partition function ϕ.

Definition 1 (partition function) *Let $\mathcal{N} = \{1, 2, \ldots, n\}$ and $\mathcal{M} = \{1, 2, \ldots, m\}$ be two finite dimensional state spaces with $m \leq n$. A* partition function *$\phi : \mathcal{N} \mapsto \mathcal{M}$ is a surjective function from $\mathcal{N}$ onto $\mathcal{M}$. For $k \in \mathcal{M}$, $\phi^{-1}(k)$ denotes the k^{th} group in $\mathcal{N}$.*

Since we already have a formula for comparing two Markov chains on the same state space (see (1)), the strategy is to use the partition function ϕ to *lift* the Markov chain Q to the original state space $\mathcal{N}$. The lifted Markov chain is denoted as $\hat{Q}$.

Definition 2 (μ-lifting of Q) *Let ϕ be a partition function on $\mathcal{N}$ and μ be a probability measure on $\mathcal{P}(\mathcal{N})$. Let $\mathcal{M}$ denote the range of ϕ and Q be a Markov transition matrix on $\mathcal{M}$. Then* μ-lifting of Q under the partition function ϕ *is a Markov matrix on $\mathcal{N}$ defined as*

$$\hat{Q}_{ij}^{(\mu)}(\phi) = \frac{\mu_j}{\sum_{k \in \psi(j)} \mu_k} Q_{\phi(i)\phi(j)}, \quad i, j \in \mathcal{N} \tag{2}$$

where $\psi(j) = \phi^{-1} \circ \phi(j) \subset \mathcal{N}$ denotes the set of states belonging to the same group as the j^{th} state.

The definition of the K-L divergence rate is extended to two chains on *different state spaces* using the lifted chain:

Definition 3 *Let (π, P) denote a stationary Markov chain on $\mathcal{N}$ and (ϖ, Q) a Markov chain on $\mathcal{M}$. Then*

$$R^{(\phi)}(P \parallel Q) \triangleq \min_{\mu \in \mathcal{P}(\mathcal{N})} R(P \parallel \hat{Q}^{(\mu)}(\phi)), \tag{3}$$

where $\hat{Q}^{(\mu)}(\phi)$ denotes the μ-lifting of Q under the partition function ϕ.

The minimum of (3) can be obtained by taking $\mu = \pi$, the stationary distribution of P:

Theorem 1 (Theorem 1 in [19]) *Suppose that (π, P) is a stationary Markov chain on $\mathcal{N}$, ϕ is a partition function with range $\mathcal{M}$ with $m \leq n$, and (ϖ, Q) is a stationary Markov chain on $\mathcal{M}$. Then, there is a unique matrix $\hat{Q}^{(\mu^*)}$ that achieves the minimum in* (3)*. The optimizer μ^* can be taken as the stationary distribution of P:*

$$\pi \in \arg\min_{\mu \in \mathcal{P}(\mathcal{N})} R(P \parallel \hat{Q}^{(\mu)}(\phi)).$$

Proof: See [19]. ∎

III. OPTIMIZATION PROBLEM

A. Problem statement

Let (π, P) be a given stationary Markov chain on $\mathcal{N}$. The *m-partition problem*, is to find the partition function $\phi : \mathcal{N} \mapsto \mathcal{M}$ and the optimal aggregated Markov chain (ϖ, Q) such that $R^{(\phi)}(P \parallel Q)$ is minimized:

$$\begin{array}{ll} \min\limits_{\phi, Q} & R^{(\phi)}(P \parallel Q) \\ \text{s.t.} & \sum_{l \in \mathcal{M}} Q_{kl} = 1, \quad k \in \mathcal{M} \\ & Q_{kl} \geq 0, \quad k, l \in \mathcal{M} \end{array} \tag{4}$$

where $R^{(\phi)}(P \parallel Q) = R(P \parallel \hat{Q}^{(\pi)}(\phi))$ and constraints arise due to stochastic property of the Markov transition matrix.

The optimization problem (4) is a *mixed-integer nonlinear program*. In general, it is intractable for Markov chains with large state space.

B. Optimal solution of Q

It turns out that the main difficulty in solving (4) is in finding the optimal partition function ϕ. The following theorem shows that for a fixed (say an optimal) partition function, the solution of Q that solves (4) can be easily obtained.

Theorem 2 (Theorem 3 in [19]) *Let (π, P) be a given Markov chain on $\mathcal{N}$. For a given partition function $\phi : \mathcal{N} \mapsto \mathcal{M}$, the optimal solution of Q is given by*

$$Q^*_{kl}(\phi) = \frac{\sum_{i\in\phi^{-1}(k)} \sum_{j\in\phi^{-1}(l)} \pi_i P_{ij}}{\sum_{i\in\phi^{-1}(k)} \pi_i}, \quad k, l \in \mathcal{M} \tag{5}$$

where the stationary distribution of Q^* is given by

$$\varpi^*_k(\phi) = \sum_{i\in\phi^{-1}(k)} \pi_i, \quad k \in \mathcal{M}.$$

Proof: See [19]. ∎

Using Theorem 2, we simplify the m-partition problem to only finding the *optimal partition function* ϕ^* such that,

$$\phi^* \in \arg\min_{\phi:\mathcal{N}\mapsto\mathcal{M}} R^{(\phi)}(P \parallel Q^*(\phi)), \tag{6}$$

where $Q^*(\phi)$ is the optimal aggregated Markov chain (5). We refer to $R^{(\phi)}(P \parallel Q^*(\phi))$ as the *K-L metric* for the partition function ϕ.

C. Algorithms for finding the optimal partition function

In [19], we consider the bi-partition ($m = 2$) problem for (4). After relaxing the integer constraints on the partition function, the optimization problem is shown to lead to a spectral partition with respect to the second eigenvector of a certain symmetric matrix related to P. A recursive bi-partition algorithm is also described to solve the multi-partition problem in a sub-optimal way.

IV. A Dynamic Programming Formulation

Consider a stationary Markov chain (π, P) defined on a finite state space $\mathcal{N}$ with $n =\mid \mathcal{N} \mid$. Let $\Phi = \{\phi_1, \phi_2, \ldots, \phi_L\}$ denote the collection of all possible partition functions defined on $\mathcal{N}$. For a m-partition problem, $L = m^n$.

A. K-L metric as an average cost

At state $i \in \mathcal{N}$, we define the *one-step cost* as,

$$g_i(\phi) = \sum_{j\in\mathcal{N}} P_{ij} \log\left(\frac{P_{ij}}{\hat{Q}^{(\pi)}_{ij}(\phi)}\right), \tag{7}$$

where $\phi \in \Phi$, $\hat{Q}^{(\pi)}(\phi)$ denotes the π-lifting of $Q^*(\phi)$ (see (2)), and $Q^*(\phi)$ is the optimal aggregated Markov chain (5) for (π, P) with the partition function ϕ.

The *average cost* is defined as,

$$\lambda(x_0; \phi) = \lim_{T\to\infty} \frac{1}{T} \mathsf{E}\left[\sum_{t=0}^{T-1} g_{X(t)}(\phi)\right],$$

where $X(t) \in \mathcal{N}$ denotes the state at time t with $X(0) = x_0$.

For a stationary Markov chain (π, P), the average cost $\lambda(x_0; \phi)$ is well defined for any partition function $\phi \in \Phi$ and does not depend on the initial state x_0. It is given by,

$$\lambda(\phi) \triangleq \sum_{i\in\mathcal{N}} \pi_i g_i(\phi). \tag{8}$$

Substituting (7) into (8), we find that the average cost $\lambda(\phi)$ is just the K-L metric with the partition function ϕ (see (1) and (3)).

The optimization problem (6) now is,

$$\phi^* \in \arg\min_{\phi\in\Phi} \lambda(\phi), \tag{9}$$

where optimal partition function ϕ^* is the one that achieves the minimum, and $\lambda^* \triangleq \lambda(\phi^*)$ denotes the *optimal average cost*.

B. Dynamic programming

We recast the model reduction problem (6) as an optimal control problem: The control objective is to obtain a stationary policy, the partition function ϕ^*, that achieves the minimal average cost (see (9)).

For any state $i \in \mathcal{N}$ and policy $\phi \in \Phi$, we define the *differential cost function* $h_i(\phi)$ as,

$$h_i(\phi) = \mathsf{E}\left[\sum_{t=0}^{T-1} (g_{X(t)}(\phi) - \lambda(\phi)) \mid X(0) = i\right],$$

where $\lambda(\phi))$ is given in (8), and $T = \min\{t > 0 \mid X(t) = s\}$ is the first future time that *reference state* s is visited. With this definition, we always have $h_s(\phi) = 0$. Since (π, P) is a stationary Markov chain, the reference state s can be taken as any fixed state in $\mathcal{N}$. The differential cost function $h_i(\phi)$ captures the relative difference of starting the process in state i, rather than in the reference state s. It follows that the vector $h(\phi) \triangleq [h_1(\phi), h_2(\phi), \ldots, h_n(\phi)]$ is the unique solution to the following Poisson equation (see e.g. [23]),

$$g(\phi) = \lambda(\phi)e + (I - P)h, \tag{10}$$

where $e \triangleq [1, 1, \ldots, 1]$, and I denotes the identity matrix.

For the optimization problem (9), although there may exist several optimal polices, there only exists a unique vector h^* such that, for all optimal polices ϕ^*, we have $h(\phi^*) = h^*$. We refer to h^* as the *optimal differential cost vector*. The theory of dynamic programming asserts that the optimal differential cost vector h^* is the unique solution to the following Bellman equation,

$$h_i = \min_{\phi\in\Phi} \left\{ g_i(\phi) - \lambda^* + \sum_{j\in\mathcal{N}} P_{ij} h_j \right\}, \quad i \in \mathcal{N}\backslash\{s\}$$

and $h_s = 0$, where $\lambda^* = \lambda(\phi^*)$ is the optimal average cost.

The optimal differential cost vector as well as the optimal policies can be obtained using standard dynamic programming methods, i.e., value or policy iteration algorithms [22]. For a stationary Markov chain, these algorithms are guaranteed to converge to an optimum. However, a direct implementation of these algorithms is impractical because of the

curse of dimensionality [24]. The curse here arises not only due to the large size n of the state space $\mathcal{N}$ but also the even larger size L of the partition function space Φ.

V. Aggregation via Learning

There are two separate complexity related issues. The first issue is the large number L of partition functions. The second issue is the large size n of the state space. To confront the first issue, a parametric representation is described to represent partition functions in terms of a small number of parameters. For the second issue, a simulation-based method is described based on a single sample path of the Markov chain.

A. Parameterizations of randomized partition policy

A randomized partition policy is defined as a mapping,

$$\eta : \mathcal{N} \to [0,1]^L,$$

where component $\eta_\phi(i)$ is the probability that the partition function ϕ is assigned to state i. We have $\sum_\phi \eta_\phi(i) = 1$ for all $i \in \mathcal{N}$. The partition policy is said to be deterministic if for every state i, there is a single partition function $\phi^{(i)}$ such that $\eta_{\phi^{(i)}}(i) = 1$. If the function $\phi^{(i)}$ is the same for all i then the policy η yields a partition of the space $\mathcal{N}$.

The problem is that L is very large, i.e., $L = m^n$ for the m-partition problem. So, following the considerations of [25], we introduce a parameter vector $\theta \triangleq [\theta_1, \theta_2, \ldots, \theta_K] \in \mathbb{R}^K$ where K is of moderate size. The remainder of the discussion of this subsection follows closely after Section VI in [25].

For $\theta \in \mathbb{R}^K$, we associate a randomized partition policy $\eta_\phi(i,\theta)$ where $\sum_\phi \eta_\phi(i,\theta) = 1$. For every $i \in \mathcal{N}$ and $\theta \in \mathbb{R}^K$, the expected cost per stage is defined as,

$$g_i(\theta) = \sum_{\phi \in \Phi} \eta_\phi(i,\theta) g_i(\phi). \tag{11}$$

Then the average cost is defined as,

$$\lambda(\theta) = \lim_{T\to\infty} \frac{1}{T} \mathsf{E}\left[\sum_{t=0}^{T-1} g_{X(t)}(\theta)\right].$$

For a stationary Markov chain (P, π), we have,

$$\lambda(\theta) = \sum_i \pi_i g_i(\theta). \tag{12}$$

Then the optimization problem can be expressed as,

$$\theta^* \in \arg\min_{\theta \in \mathbb{R}^K} \lambda(\theta), \tag{13}$$

and θ^* defines an optimal randomized policy $\eta_\phi(i,\theta^*)$. Furthermore, if η_ϕ is deterministic and independent of i then a partition function can be uniquely obtained from η_ϕ.

In practice, the policy is independent of i because the Markov chain P does not depend upon the partition function ϕ. A numerical solution will in general, however, only lead to a partition function with high probability determined by $\eta_\phi(\theta)$. This is discussed in a greater detail in Section VI with the aid of numerical examples.

B. Parametric representation for the bi-partition problem

Let $\mathcal{M} = \{1, 2\}$ denote the reduced aggregated state space with two superstates. For the bi-partition problem, a partition function $\phi : \mathcal{N} \to \mathcal{M}$ can take only two values, 1 and 2 for any state i. We consider the parameter vector $\theta \triangleq [\theta_1, \ldots, \theta_n] \in \mathbb{R}^n$, where θ_i decides the group assignment for the state $i \in \mathcal{N}$. In particular, we use $\frac{1}{1+\exp(M\theta_i)}$ to reflect the probability that $\phi(i) = 1$, where $M > 0$ is some positive constant. This gives a randomized partition policy

$$\eta_\phi(i,\theta) = \frac{1}{1+\exp(M\theta_i)} \mathbb{1}_{\phi(i)=1} + \frac{\exp(M\theta_i)}{1+\exp(M\theta_i)} \mathbb{1}_{\phi(i)=2}, \tag{14}$$

where $\mathbb{1}_{\phi(i)=1}$ is 1 if $\phi(i) = 1$ and 0 otherwise, and similarly for $\mathbb{1}_{\phi(i)=2}$.

In the remainder of this paper, we will consider only the bi-partition problem with representation $\eta_\phi(i,\theta)$ in (14). The general case is similarly handled but the bi-partition case allows for a simpler discussion of the main ideas.

Lemma 3 *Assume we choose the parametric representation as in (14). Then, for any $i \in \mathcal{N}$, the function $\lambda(\theta)$ defined in (12) is bounded from below (i.e. $\lambda(\theta) \geq 0$), twice differentiable w.r.t. θ, and has bounded first and second derivatives for all $\theta \in \mathbb{R}^n$.*

Proof: For each state $i \in \mathcal{N}$, $g_i(\phi) \geq 0$ by its definition (7), and $\eta_\phi(i,\theta)$ is a positive globally Lipschitz C^∞ function with respect to θ. ■

By Lemma 3, we know that the gradient of $\lambda(\theta)$ with respect to θ is well defined (we denote $\nabla \triangleq \nabla_\theta$ for short),

$$\nabla\lambda(\theta) = \sum_{i\in\mathcal{N}} \pi_i \nabla g_i(\theta),$$

where

$$\nabla g_i(\theta) = \sum_{\phi\in\Phi} \nabla\eta_\phi(i,\theta) g_i(\phi). \tag{15}$$

C. An idealized gradient-descent algorithm

Since the gradient of $\lambda(\theta)$ can be exactly computed, we could use the gradient-based method to solve the unconstrained optimization problem (13) in the whole real space $\mathbb{R}^n$. An idealized gradient-decent algorithm is given here to update the parameter θ,

$$\theta^{(t+1)} = \theta^{(t)} - \gamma_t \nabla\lambda(\theta^{(t)}). \tag{16}$$

Using Lemma 3, we know the $\nabla\lambda(\theta)$ is a bounded globally Lipschitz function on $\mathbb{R}^n$. Under suitable conditions on the stepsize γ_t, one can show that $\lim_{t\to\infty} \nabla\lambda(\theta^{(t)}) = 0$ and $\theta^{(t)}$ converges to a finite value (see [24, Prop. 4.3.2]).

However, each iteration of the idealized gradient-descent algorithm involves a burdening computation of the averaged gradient term $\nabla\lambda(\theta^{(t)})$ based on each gradient $\nabla g_i(\theta^{(t)})$ over the entire state space (see Fig. 1). Significant computer resources must be allocated in order to store a large number of iteration values and compute this averaged gradient.

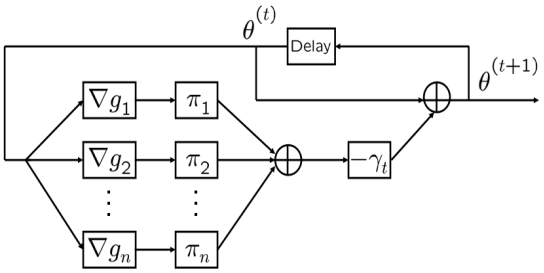

Fig. 1. Idealized gradient-descent algorithm: The parameters are updated using the gradient of the average cost evaluated on the entire state space.

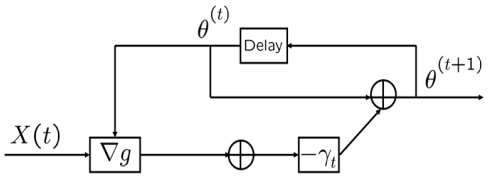

Fig. 2. Simulation-based gradient-descent algorithm: The parameters are updated using the gradient of the one-step cost only evaluated on the currently visited state.

D. A simulation-based gradient-descent algorithm

A simulation-based (stochastic-approximation) gradient-descent algorithm is obtained by dropping the averaging operation in the idealized gradient algorithm (16). In this algorithm, the true/averaging gradient $\nabla\lambda(\theta)$ is approximated by the one-step cost gradient $\nabla g_{X(t)}(\theta)$ (see Fig. 2).

Let $\{X(t)\}_{t=0}^{\infty}$ be a single sample path generated from the simulation of a stationary Markov chain (π, P). Then the simulation-based gradient-descent algorithm for updating the parameter vector θ is given as,

$$\theta^{(t+1)} = \theta^{(t)} - \gamma_t \nabla g_{X(t)}(\theta^{(t)}). \tag{17}$$

where the value $\theta^{(t)}$ is assumed to be available from the previous iteration, and $\nabla g_{X(t)}(\theta)$ is computed using (15) for any $X(t) \in \mathcal{N}$. In addition, another stochastic approximation algorithm for updating the average cost is run in parallel,

$$\tilde{\lambda}^{(t+1)} = \tilde{\lambda}^{(t)} + \gamma_t (g_{X(t)}(\theta^{(t)}) - \tilde{\lambda}^{(t)}) \tag{18}$$

where $\tilde{\lambda}^{(t)}$ is the estimated average cost and parameter $\theta^{(t)}$ comes from (17).

It is assumed throughout that the stepsize γ_t satisfies the standard stochastic approximation conditions:

Assumption 2 *The stepsize values $\{\gamma_t\}$ are nonnegative and satisfy,*

$$\sum_{t=1}^{\infty} \gamma_t = \infty, \quad \sum_{t=1}^{\infty} \gamma_t^2 < \infty.$$

In the simulations described in this paper, $\gamma_t = \frac{1}{t}$ is chosen to satisfy the above assumption.

Under the assumption that the chain is irreducible and aperiodic, it follows that $g_{X(t)}(\theta)$ and $\nabla g_{X(t)}(\theta)$ are asymptotically unbiased estimates for $\lambda(\theta)$ and $\nabla\lambda(\theta)$ respectively:

Lemma 4 *Suppose that the Markov chain (π, P) satisfies Assumption 1 and $\{X(t)\}_{t=0}^{\infty}$ is a sample path from the simulation of (π, P). Then we have*

$$\lim_{t\to\infty} \mathsf{E}\left[g_{X(t)}(\theta)\right] = \lambda(\theta), \tag{19}$$

$$\lim_{t\to\infty} \mathsf{E}\left[\nabla g_{X(t)}(\theta)\right] = \nabla\lambda(\theta), \tag{20}$$

where $g_{X(t)}(\theta)$ is computed by (11) and $\lambda(\theta)$ is given in (12),

The convergence of the simulation-based algorithm is established using an ODE method. The ODE is obtained by first considering the difference equation obtained on taking asymptotical expectations on both sides of (17) and (18),

$$\theta^{(t+1)} = \theta^{(t)} - \gamma_t \nabla\lambda(\theta^{(t)}), \tag{21}$$

$$\tilde{\lambda}^{(t+1)} = \tilde{\lambda}^{(t)} + \gamma_t(\lambda(\theta^{(t)}) - \tilde{\lambda}^{(t)}). \tag{22}$$

The ODE is then the differential equation analog,

$$\dot{\theta}_t = -\nabla\lambda(\theta_t),$$

$$\dot{\tilde{\lambda}}_t = \lambda(\theta_t) - \tilde{\lambda}_t. \tag{23}$$

By construction, $\lambda(\theta_t) \geq 0$ is bounded from below (see (12)). It is also a non-increasing function because $\dot{\lambda}(\theta_t) = -\|\nabla\lambda(\theta_t)\|^2 \leq 0$. Thus $\lambda(\theta_t)$ must converge to some non-negative limit and $\nabla\lambda(\theta_t)$ must converge to zero as $t \to \infty$. Using (23), we conclude that $\tilde{\lambda}_t$ must also converge to the same limit as $\lambda(\theta_t)$.

The analogous statements for the stochastic approximation recursion are contained in the following proposition. The proof follows from stability of the ODE and standard arguments (see e.g. Theorem 2 of Chapter 6 of [26]).

Proposition 1 *Let Assumption 2 hold, and assume that the parameter vector sequence $\{\theta^{(t)}\}$ and average cost sequence $\{\tilde{\lambda}^{(t)}\}$ are updated according to (17) and (18) respectively. Then, the sequence $\lambda(\theta^{(t)})$ converges almost surely and,*

$$\nabla\lambda(\theta^{(t)}) \xrightarrow{a.s.} \mathbf{0}, \quad \text{as } t \to \infty.$$

Moreover, $\tilde{\lambda}^{(t)}$ also converges almost surely to the same limit,

$$\tilde{\lambda}^{(t)} \xrightarrow{a.s.} \lambda(\theta^{(t)}), \quad \text{as } t \to \infty.$$

VI. Simulations and Discussion

A. 4-state Markov chain

Consider a stationary 4-state Markov chain with transition matrix,

$$P = \begin{bmatrix} 0.5 & 0.4 & 0.0 & 0.1 \\ 0.4 & 0.5 & 0.1 & 0.0 \\ 0.0 & 0.1 & 0.5 & 0.4 \\ 0.1 & 0.0 & 0.4 & 0.5 \end{bmatrix},$$

whose stationary distribution $\pi = [0.25, 0.25, 0.25, 0.25]$.

The optimal bi-partition function for this Markov chain is $\phi^* = [1, 1, 2, 2]$, i.e. the state $\{1, 2\}$ are aggregated into one group and the states $\{3, 4\}$ are aggregated into another group.

The optimal aggregated Markov chain can be computed according to (5),

$$Q^* = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}.$$

whose stationary distribution $\varpi^* = [0.5, 0.5]$.

1) Dynamic programming: Since the state space is relatively small, the dimension of bi-partition function space is 2^4. The optimal partition function ϕ^* can be obtained using standard dynamic programming methods [22]. The optimal average cost λ^* is equal to 0.0749.

2) Idealized gradient algorithm: For this small-scale example, we can efficiently compute $\nabla\lambda(\theta)$ for any given θ. We can thus implement the idealized gradient algorithm (16). The evolution of this algorithm, starting with $\theta^{(0)} = [1, 1, 1, 1]$, is shown in Fig. 3. After 100 iterations, parameter vector $\theta^{(100)} = [-6.3411, -6.3464, 6.3575, 6.3580]$, and the probabilities of states being in the first group are $\eta_{\phi=[1,1,1,1]}(\cdot, \theta^{(100)}) = [0.9982, 0.9983, 0.0017, 0.0017]$. Thus, we obtain the optimal partition function as $\phi^* = [1, 1, 2, 2]$ with high probability. The corresponding estimated average cost is given by $\tilde{\lambda}^{(100)} = 0.0757$, which is very close to the optimal value.

3) Simulation based algorithm: We consider a single sample path of the Markov process simulated according to the transition matrix P. The updating algorithms (17) and (18) are implemented at every time step along the sample path. We start with the same initial parameter vector $\theta^{(0)} = [1, 1, 1, 1]$. The evolution of parameters and the average cost are give in Fig. 4. After 500 iterations, the simulation-based algorithm has comparable performance to the idealized gradient algorithm after 100 iterations. The parameter vector $\theta^{(500)} = [-5.6893, -5.5390, 5.7600, 5.6252]$, and the probabilities of states being in the first group are $\eta_{\phi=[1,1,1,1]}(\cdot, \theta^{(500)}) = [0.9966, 0.9961, 0.0031, 0.0036]$. From this, the optimal partition function $\phi^* = [1, 1, 2, 2]$ can be determined with high probability. The corresponding estimated average cost is equal to $\tilde{\lambda}^{(500)} = 0.0766$, which is a little larger than the optimal value.

B. 100-state Markov chain

We consider a 100-state Markov chain borrowed from [16]. Fig. 5 illustrates the transition probabilities for this chain. The cold colors indicate weak interactions (small transition probability), and warm colors indicate strong interactions (large transition probability) between states. The color plot suggests that this Markov chain is nearly completely decomposable with five blocks.

In [19], it is shown that a certain relaxation of the bi-partition problem (6) gives a solution according to the sign-structure of the second eigenvector, depicted in Fig. 6. The optimal bi-partition of the state space is: States $1-60$ are aggregated as the first group and states $61-100$ are aggregated as the second group. The spectral bi-partition is shown in Fig. 5. The corresponding K-L metric (average cost) is given by $\lambda^*_{global} = 0.1321$. The subscript *"global"* refers to the fact that the spectral bi-partition gives the global

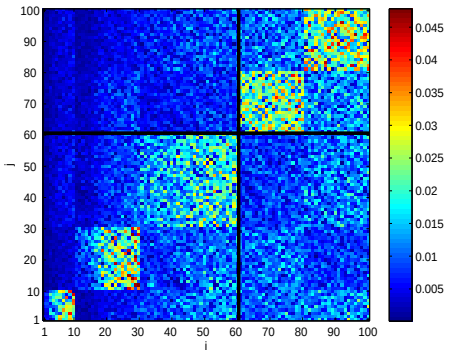

Fig. 5. The graph of transition matrix P for the 100-state nearly complete decomposable Markov chain (see [16], [19]). The optimal bi-partition of the state space is indicted by two bold lines: States $1-60$ are aggregated as the first group and states $61-100$ are aggregated as the second group.

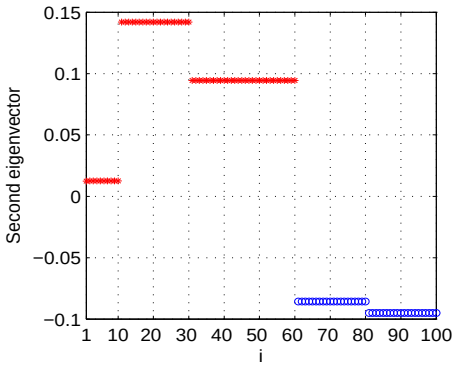

Fig. 6. A plot of the second eigenvector of the 100×100 transition matrix P.

minimum of the K-L metric for the bi-partition problem (6) (see [19] for more details).

For the simulation-based algorithm, a single sample path of Markov process was obtained according to the transition matrix P. θ is now a 100-dimensional parameter vector initialized as $\theta^{(0)} = [0, 0, \ldots, 0]$. At time t, the vector $\theta^{(t)}$ is updated according to (17) and the estimated average cost $\tilde{\lambda}^{(t)}$ is updated according to (18). Even though, the state space is large, the update at time t only requires the evaluation of

$$\sum_{j \in \mathcal{N}} P_{X(t)j} \log \left(\frac{P_{X(t)j}}{\hat{Q}^{(\pi)}_{X(t)j}(\phi)} \right)$$

for entries with $P_{X(t)j} > 0$, where $X(t)$ is the state at time t.

Fig. 7 (a) depicts $\frac{1}{1+\exp(M\theta_i^{(t)})}$ as a function of t for $i \in \mathcal{N}$. It reflects the probability that i^{th} state is assigned to the 1^{st} group (see Section V-B). For large t, each state is assigned with high probability to one of the two groups. The resulting two-group assignments at time $t = 5000$ are obtained with high probability and depicted in Fig. 7 (c): The first group is formed by states $1-10$ (first block), $31-60$ (third block) and $81-100$ (fifth block), while the second group is formed by states $11-30$ (second block) and $61-80$ (fourth block).

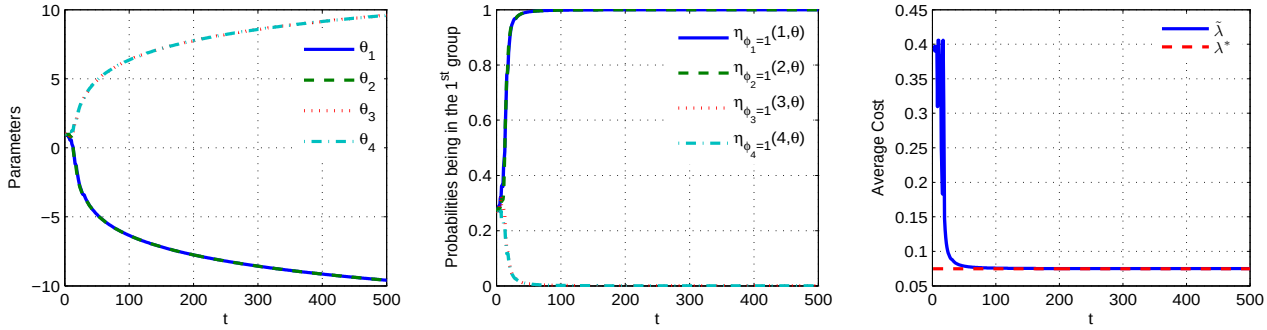

Fig. 3. Plots of (a) $\theta^{(t)}$, (b) $\eta_\phi(\cdot, \theta^{(t)})$ and (c) $\tilde{\lambda}^{(t)}$ for the 4-state Markov chain with the idealized gradient algorithm (16).

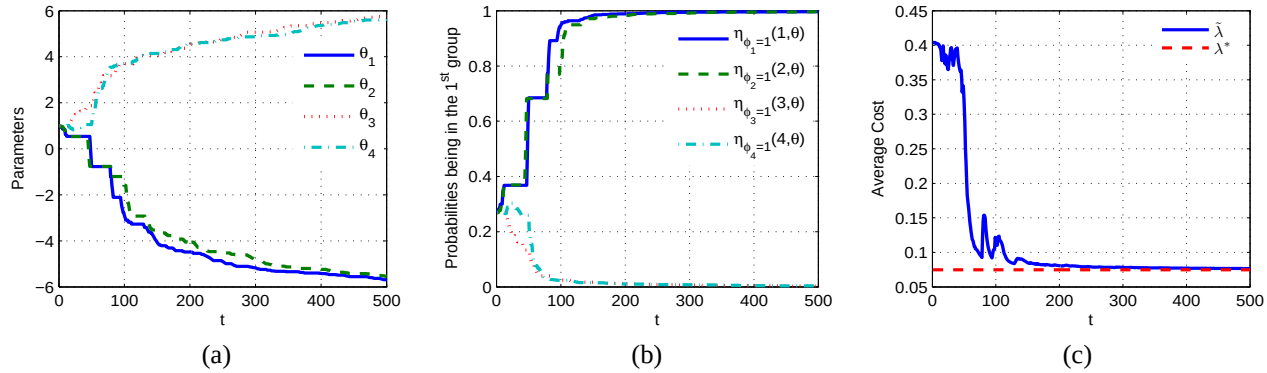

Fig. 4. Plots of (a) $\theta^{(t)}$, (b) $\eta_\phi(\cdot, \theta^{(t)})$ and (c) $\tilde{\lambda}^{(t)}$ for the 4-state Markov chain with the simulation-based algorithm (17) and (18).

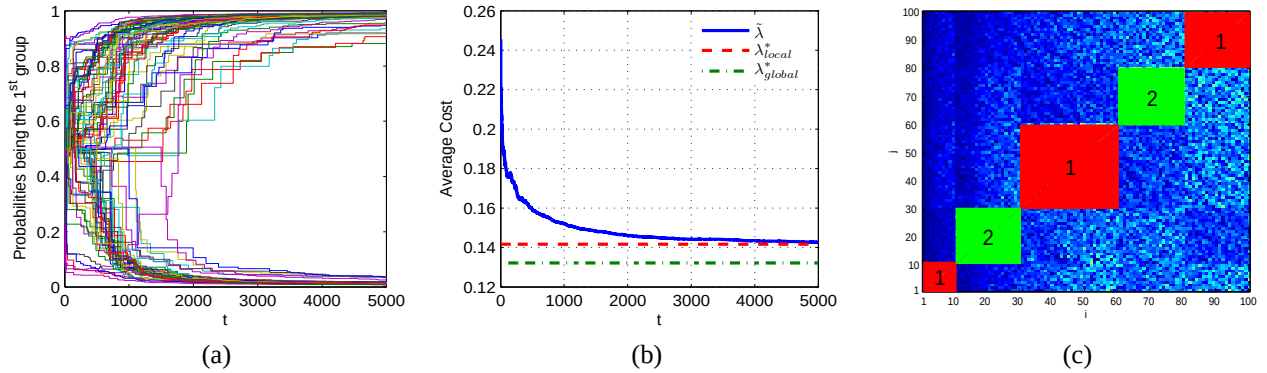

Fig. 7. Plots of (a) $\eta_\phi(\cdot, \theta^{(t)})$, (b) $\tilde{\lambda}^{(t)}$ and (c) the bi-partition obtained using $\theta^{(5000)}$ for the 100-state Markov chain with the simulation-based algorithm.

The partition found using the simulation-based algorithm is not the global optimum for the bi-partition problem (compare Fig. 5 with Fig. 7 (c)). The algorithm converges to a local minimum. In Fig. 7 (b), we depict the plots of estimated average cost $\tilde{\lambda}^{(t)}$, the average cost $\lambda^*_{local} = 0.1416$ computed for the local optimal partition shown in Fig. 7 (c), and the average cost λ^*_{global} for the global optimal partition shown in Fig. 5.

The local optimal convergence issue is due to the non-convexity of the average cost $\lambda(\theta)$ with respect to the parameter θ. In general, one finds only the local minimum with a gradient-descent scheme [24]. For the Markov chain in Fig. 5, one can find multiple partitions by recursively applying the bi-partition algorithm.

VII. Conclusions

In this paper, we proposed a simulation-based algorithm to aggregate a large-scale Markov chain to obtain a reduced-order Markov model. Three are three main concepts in the paper: 1) the use of K-L metric to compare Markov chains for model reduction purpose; 2) the dynamic programming formulation of the model reduction problem; 3) its solutions using a simulation-based (stochastic-approximation) algorithm. The proposed framework requires only a sample path

of the Markov process and has small computation burden even for very large problems.

REFERENCES

[1] A. Krogh, M. Brown, I. S. Mian, K. Sjolander, and D. Haussler, "Hidden Markov Models in computational biology: applications to protein modelling," *Journal of Molecular Biology*, vol. 235, pp. 1501–1531, 1994.

[2] M. Meila and J. Shi, "A random walks view of spectral segmentation," in *International Conference on AI and Statistics (AISTAT)*, 2001.

[3] J. Cai and D. C. M. Dickson, "Ruin probabilities with a Markov chain interest model," *Insurance: Mathematics and Economics*, vol. 35, no. 3, pp. 513–525, 2004.

[4] P. G. Mehta, M. Dorobantu, and A. Banaszuk, "Graph-based multi-scale analysis of building system transport models," in *Proceeding of American Control Conference*, Minneapolis, 2006, pp. 1110–1115.

[5] J. Niedbalski, K. Deng, P. G. Mehta, and S. Meyn, "Model reduction for reduced order estimation in traffic models," in *Proceeding of American Control Conference*, 2008, pp. 914–919.

[6] R. G. Phillips and P. V. Kokotovic, "A singular perturbation approach to modeling and control of Markov chains," *IEEE Trans. Automat. Contr.*, vol. 26, no. 5, pp. 1087–1094, 1981.

[7] H. A. Simon and A. Ando, "Aggregation of variables in dynamical systems," *Econometrica*, vol. 28, pp. 111–138, 1961.

[8] P. J. Courtois, *Decomposability: Queueing and Computer System Applications*. New York, NY: Academic Press, 1977.

[9] O. Junge and M. Dellnitz, "Almost invariant sets in Chua's circuit," *Int. J. Bif. and Chaos*, vol. 7, pp. 2475–85, 1997.

[10] C. Schütte, A. Fischer, W. Huisinga, and P. Deuflhard, "A direct approach to conformational dynamics based on hybrid Monte Carlo," *J. Comput. Phys., Special Issue on Computational Biophysics*, vol. 151, pp. 146–168, 1999.

[11] W. Huisinga, S. P. Meyn, and C. Schütte, "Phase transitions and metastability in Markovian and molecular systems," *Ann. Appl. Probab.*, vol. 14, no. 1, pp. 419–458, 2004.

[12] S. P. Meyn, G. Hagen, G. Mathew, and A. Banaszuk, "On complex spectra and metastability of Markov models," in *Proc. of the IEEE Conf. on Decision & Control*, December 2008.

[13] F. R. K. Chung, *Spectral Graph Theory*. Providence, RI: American Mathematical Society, 1997.

[14] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[15] S. X. Yu, R. Gross, and J. Shi, "Concurrent object recognition and segmentation by graph partitioning," in *Neural Information Processing Systems*, 2002.

[16] M. Meila and L. Xu, "Multiway cuts and spectral clustering," May 2003, Technical Report 442.

[17] K. Deng, W. Chen, P. G. Mehta, and S. Meyn, "Resource pooling for optimal evacuation of a large building," in *IEEE Conference on Decision and Control*, 2008, pp. 5565–5570.

[18] V. Borkar and S. P. Meyn, "Oja's algorithm for graph clustering and Markov spectral decomposition," in *Proceedings of the Third International Conference on Performance Evaluation Methodologies and Tools*, Athens, Greece, 2008.

[19] K. Deng, P. G. Mehta, and S. P. Meyn, "An information-theoretic framework to aggregate a Markov chain," in *Proceeding of American Control Conference*, St. Louis, MO, June 2009, online: https://netfiles.uiuc.edu/kundeng2/www/paper/dmmacc09.pdf.

[20] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 1st ed. New York, NY: John Wiley & Sons, Inc., 1991.

[21] Z. Rached, F. Alalaji, and L. L. Campbell, "The Kullback-Leibler divergence rate between Markov sources," *IEEE Trans. Info. Thy.*, vol. 50, no. 5, pp. 917–921, 2004.

[22] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Bemont, MA: Athena Scientific, 1995.

[23] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*, 2nd ed. London: Springer-Verlag, 1993, online: `http://black.csl.uiuc.edu/~meyn/pages/book.html`.

[24] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Bemont, MA: Athena Scientific, 1996.

[25] P. Marbach and J. N. Tsitsiklis, "Simulation-based optimization of Markov reward processes," *IEEE Transactions on Automatic Control*, vol. 46, no. 2, pp. 191–209, 2001.

[26] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Delhi, India and Cambridge, UK: Hindustan Book Agency and Cambridge University Press (jointly), 2008.