Oja's Algorithm for Graph Clustering, Markov Spectral Decomposition, and Risk Sensitive Control

V. Borkar [a] and S.P. Meyn [b]

[a] *School of Technology and Comp. Science, Tata Institute of Fundamental Research, Homi Bhabha Rd., Mumbai 400005, India*

[b] *Department of Electrical and Computer Engg. and the Coordinated Sciences Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

Abstract

Given a positive definite matrix M and an integer $N_m \geq 1$, Oja's subspace algorithm will provide convergent estimates of the first N_m eigenvalues of M along with the corresponding eigenvectors. It is a common approach to principal component analysis. This paper introduces a normalized stochastic-approximation implementation of Oja's subspace algorithm, as well as new applications to the spectral decomposition of a *reversible* Markov chain. Recall that this means that the stationary distribution satisfies the detailed balance equations [26]. Equivalently, the statistics of the process in steady state do not change when time is reversed. Stability and convergence of Oja's algorithm are established under conditions far milder than assumed in previous work. Applications to graph clustering, Markov spectral decomposition, and multiplicative ergodic theory are surveyed, along with numerical results.

2000 AMS Subject Classification:

05C85, 94C15, 68W20, 62L20, 60J22, 60J10, 37A30, 92B20

Key words: Graph algorithms, Oja's algorithm, stochastic approximation, Markov chains, spectral theory of Markov chains, multiplicative ergodic theory, risk sensitive control.

1 Introduction

Spectral decomposition is a classical approach to model reduction for systems that are complex due to dimension or randomness. This technique is known as principal component analysis or the Karhunen-Loève decomposition, depending on the context [20,21,24]. The same technique has been developed more recently for network decomposition [27,32,34], which in particular provides an appealing alternative to the min-cut max-flow theorem.

Given a symmetric $N \times N$ matrix w, its spectral decomposition amounts to the computation of its N real eigenvalues and corresponding eigenvectors. In the Karhunen-Loève decomposition the matrix w is a covariance matrix, and the decomposition leads to a representation of a stationary process as a moving-average of white noise. In the graph clustering problem the elements of this matrix represent positive edge weights: $w_{ij} = w_{ji}$ is the weight of the link connecting nodes i and j. The first decomposition of a connected graph is obtained by computation of the eigenvector corresponding to the second eigenvalue. It can be shown that the eigenvector possesses positive and negative entries, and this sign structure is used to define a generalized network cut in [27,32,34].

Oja's subspace algorithm is one approach to computation of the leading eigenvalues and eigenvectors of the matrix w [11,28,33]. Fix an integer $N_m \leq N$, and let $m(t)$ denote an $N \times N_m$ matrix whose columns are intended to approximate an N_m-dimensional eigenspace corresponding to the N_m largest of the N eigenvalues of w. A deterministic version of Oja's algorithm is ex-

⋆ S.M. was partially supported by the National Science Foundation under grant ECS-0523620, and by AFOSR grant FA9550-09-1-0190. V.B. was supported in part by a J. C. Bose Fellowship of Dept. of Science and Technology, Govt. of India, and a grant from General Motors India Science Lab. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of their sponsors.

Email addresses: borkar@tifr.res.in (V. Borkar), meyn@illinois.edu (S.P. Meyn).

pressed as the polynomial differential equation:

$$\tfrac{d}{dt}m(t) = [I - m(t)m^T(t)]wm(t), \tag{1}$$

where $m(0)$ is given as initial condition. If the matrix w is positive definite then the analysis of [11] establishes convergence of $\boldsymbol{m}$ for almost every initial condition.

This paper introduces a normalized implementation of Oja's algorithm that is also a multi-dimensional generalization of the one-dimensional algorithm of Krasulina [23]. Stability and convergence of the normalized algorithm are established under conditions far milder than assumed in previous work. Applications to graph clustering are surveyed, as well as new applications to the spectral decomposition of a reversible Markov chain.

In the following section we introduce the stochastic approximation algorithm, and present the main result establishing convergence of the algorithm. Applications to spectral graph theory are surveyed in Section 3, and Section 4 contains extensions of the algorithm to compute the spectrum of a reversible Markov chain. Section 5 shows connections to multiplicative ergodic theory and risk-sensitive control where the interest is in the top eigenvalue and eigenvector. Examples are contained in Section 6, and conclusions may be found in Section 7.

2 Stochastic Approximation & Oja's Algorithm

Oja's 1985 paper [29] introduces a stochastic approximation algorithm based on the o.d.e. (1). Suppose that $\boldsymbol{X}$ is an $\mathbb{R}^n$-valued stationary process with covariance matrix $w = \mathsf{E}[X(t)X(t)^{\mathrm{T}}]$. We can express Oja's stochastic approximation algorithm as the matrix recursion:

$$M(n+1) - M(n) = a(n)\big[I - M(n)M^{\mathrm{T}}(n)\big]\widehat{W}(n)M(n), \tag{2}$$

where $\widehat{W}(n) = X(n)X^{\mathrm{T}}(n)$, and $a(n)$ is a decreasing parameter — the step-size for the algorithm [5]. Specific assumptions will be imposed later. Almost sure convergence to the appropriate dimensional dominant eigenspace was established by applying stochastic approximation techniques that were available at the time. These techniques require Lipschitz continuity of the right hand side of the recursion in the variable $M(n)$, which is violated in this recursion. This issue is addressed in [29] and in [33], by imposing additional conditions on $\boldsymbol{X}$.

The lack of Lipschitz continuity presents problems even in deterministic approximations of (1) in discrete time. One such algorithm is introduced in [36] through sampling the o.d.e. to obtain the deterministic recursion,

$$m(n+1) - m(n) = a(n)\big[I - m(n)m^T(n)\big]wm(n) \tag{3}$$

While convergence is established for the deterministic algorithm, the proof is complex. Complexity is due in large part to the cubic nonlinearity seen here just as in the stochastic approximation algorithm.

To obtain an algorithm that satisfies the Lipschitz continuity and thereby place the algorithm within the framework of [4–6] we introduce a normalization. The normalized o.d.e. is given by:

$$\begin{aligned}\tfrac{d}{dt}m(t) &= a(t)\big[I - m(t)m^T(t)\big]wm(t),\\ a(t) &= [1 + \operatorname{trace}(m(t)m(t)^{\mathrm{T}})]^{-1}.\end{aligned} \tag{4}$$

The right hand side of the differential equation is Lipschitz in the variable $m(t)$. Solutions to this differential equation are simply time-scaled versions of the solutions to (1). In particular, from each initial condition the set of limit points are identical.

The stochastic approximation algorithm considered in this paper is again of the form (2) in which the gain sequence is modified as in the o.d.e. (4), with an additional scaling as follows:

$$a(n) = b(n)(1 + \operatorname{trace}(M(n)M(n)^{\mathrm{T}}))^{-1}. \tag{5}$$

It is assumed throughout that the following assumptions hold for the sequence $\boldsymbol{b} = \{b(n) : n \ge 0\}$: It is non-negative, with

$$\begin{aligned}\sum_{n=0}^{\infty} b(n) = \infty, \quad \sum_{n=0}^{\infty} b(n)^2 < \infty,\\ \sup_{n\ge 0}\left(\frac{\sum_{k\ge n} b(k)^2}{b(n)}\right) < \infty.\end{aligned} \tag{6}$$

An example is $b(n) = (1+n)^{-1}$, $n \ge 0$.

Under these conditions the algorithm is stable. To guarantee consistency we modify the algorithm slightly through the introduction of white noise:

$$\begin{aligned}&M(n+1) - M(n) =\\ &\quad a(n)\big(\big[I - M(n)M^{\mathrm{T}}(n)\big]\widehat{W}(n)M(n) + \xi(n+1)\big),\end{aligned} \tag{7}$$

where $\boldsymbol{\xi}$ is an i.i.d. $N(0, I)$ sequence. Proposition 2.1 states that this recursion shares the best possible convergence properties observed in the o.d.e. (1). While the deterministic algorithm can become trapped in an arbitrary eigenspace of w, the stochastic algorithm (7) is strongly consistent from each initial condition.

While the above result is stated for i.i.d. $\boldsymbol{X}$, Proposition 2.1 extends to ergodic Markov $\boldsymbol{X}$ as well, see, e.g., Corollary 8 and Theorem 9, p. 74–75, of [5].

Proposition 2.1 *Consider the algorithms* (2) *or* (7)*, where* $\boldsymbol{a}$ *is given in* (5)*, and with* $\boldsymbol{b}$ *satisfying the conditions in* (6)*. Suppose that the process* $\boldsymbol{X}$ *is i.i.d., with covariance* $w > 0$*, and that it is independent of the i.i.d.* $N(0, I)$ *sequence* $\boldsymbol{\xi}$*.*

Then, the following conclusions hold for each initial $M(0)$*:*

(i) Stability*: For either of the algorithms* (2) *or* (7)*,*

$$\limsup_{n\to\infty} \|M(n)\| < \infty \qquad a.s.$$

(ii) Convergence*: For the algorithm* (7)*, with probability one, any limit point* $M(\infty)$ *of the sequence of matrices* $\{M(n)\}$ *has columns that lie in the eigenspace spanned by the first* m *eigenvalues of* w*.*

Proof First we establish that the solutions to either stochastic approximation recursion are bounded a.s. by applying Theorem 7 of [5, Ch. 3] (see also [6]). This result constructs an "o.d.e. at infinity" that approximates the behavior of the recursion for large initial conditions. Based on the recursion (2) or (7) we obtain the o.d.e.,

$$\tfrac{d}{dt} m^\infty(t) = -\Big[\frac{m^\infty(t) m^{\infty T}(t)}{\text{trace}\,(m^\infty(t) m^\infty(t)^{\mathrm{T}})}\Big] w m^\infty(t), \quad (8)$$

where $m^\infty(0) \in \mathbb{R}^{N\times N_m}$ is given as initial condition. Define the real valued function $V : \mathbb{R}^{N\times N_m} \to \mathbb{R}_+$ as the quadratic:

$$V(m) := \text{trace}\,(m^{\mathrm{T}} w m), \qquad m \in \mathbb{R}^{N\times N_m}.$$

Under the positivity assumption on w this function vanishes only when m is identically zero. This property combined with the following drift condition implies that V serves as a Lyapunov function:

$$\tfrac{d}{dt} V(m^\infty(t)) = -2\Big[\frac{\text{trace}\,([m^{\infty T}(t) w m^\infty(t)]^2)}{\text{trace}\,(m^\infty(t) m^\infty(t)^{\mathrm{T}})}\Big] < 0,$$

with $m^\infty(t) \neq 0$. It follows that the origin is the unique asymptotically stable equilibrium for (8). Theorem 7 of [5, Ch. 3] completes the proof of (i).

We now restrict to the algorithm (7). From the analysis of [11] it follows that the eigenspace spanned by the first m eigenvectors of w is a locally stable invariant set for (4), whereas the remaining eigenvectors are unstable invariant sets. The introduction of the i.i.d. process $\boldsymbol{\xi}$ combined with the assumptions on the gain sequence ensure that the results of Section 4.3 of [5] apply, and the iterates avoid these unstable invariant sets with probability one. In turn, Theorem 19 of [5, Ch. 4] then ensures the desired convergence with probability one.

3 Spectral Graph Clustering

We now show how these methods can be adapted to spectral graph clustering, following [27,32,34]. The algorithms described here are variants of stochastic approximation based on the construction of a Markov chain evolving on the nodes of the graph.

Suppose that w is a symmetric matrix with non-negative entries that defines weights in a graph with adjacency matrix $A_{ij} = A_{ji} = \mathbf{1}\{w_{ij} > 0\}$. Throughout this section we impose the following assumptions on the matrix w:

(i) Symmetry: $w = w^{\mathrm{T}}$
(ii) Probabilistic normalization: $\sum_{i,j} w_{ij} = 1$.
(iii) Irreducibility: $\sum_{k=1}^\infty w_{ij}^k > 0$ for each i, j, where w^k denotes the k-fold matrix product.

The normalization can be assumed without loss of generality by scaling, and irreducibility is equivalent to connectedness of the graph.

Oja's technique is not directly applicable because w is not necessarily positive definite. One approach to enforce positivity is to add a scaled identity matrix to obtain $w^{(r)} := w + rI$. This matrix is positive definite for $r \geq 0$ sufficiently large. The relationship between the spectrum of w and $w^{(r)}$ is obvious, and the eigenvectors coincide. We henceforth assume that this scaling has been performed so that the matrix w is positive definite.

A stochastic approximation algorithm is obtained by constructing a Markov chain on the state space $\mathsf{X} := \{1, \ldots, N\}$. Under the normalization assumption, the matrix w can be interpreted as a probability measure on the product space $\mathsf{X} \times \mathsf{X}$. Its common marginal distribution is denoted $\pi(i) = \sum_j w_{ij}$, and a transition matrix is defined as the ratio,

$$P(i, j) = \frac{w_{ij}}{\pi(i)}.$$

Denote the Markov chain with this transition matrix by $\boldsymbol{X} = \{X(n) : n \geq 0\}$.

The detailed balance equations hold, $\pi(i)P(i, j) = \pi(j)P(j, i)$, $1 \leq i, j \leq N$, so that π is invariant for P. The detailed balance also implies that the stationary chain corresponding to initial law π will be reversible, i.e., the probabilistic law of $\boldsymbol{X}$ is preserved under time reversal. The transition matrix is irreducible since the graph is connected, which implies that the invariant measure π is unique.

In the applications considered in this section we redefine the matrix $\widehat{W}$ by: for $1 \leq i, j \leq N$,

$$\widehat{W}_{ij}(n) = r\mathbf{1}\{i = j\} + \mathbf{1}\{X(n) = i,\, X(n+1) = j\}, \quad (9)$$

so that we obtain $\mathsf{E}[\widehat{W}(n)] = w^{(r)}$ for each n. (Here $\mathbf{1}\{\cdots\}$ is the indicator function, 1 if the argument is true and 0 if not.) A stochastic approximation algorithm is obtained by applying (2) using this matrix sequence.

If the second eigenvalue of P is close to unity then the mixing rate of the Markov chain $\boldsymbol{X}$ will be slow, and this may adversely affect the convergence rate of (2) (see [15], [26, Ch. 20], and the discussion in Section 4.) In this case the following variant can be used, known as *split sampling* [4]. Let $\boldsymbol{X}^1$ denote an i.i.d. sequence with marginal π. Construct a second stochastic process as follows: For each $n = 1, 2, \ldots$ the random variable $X^2(n)$ is chosen in two stages. First, the value $j = X^1(n-1)$ is observed. Next, the value of $X^2(n)$ is chosen according to the distribution $P(j, \cdot)$, independent of $\{X^1(r), X^2(k) : r \in \mathbb{Z}_+, k \le n-1\}$. Based on this pair of stochastic processes, the algorithm is then defined by (2) using

$$\widehat{W}_{ij}(n) = r\mathbf{1}\{i = j\} + \mathbf{1}\{X^1(n) = i,\, X^2(n+1) = j\}. \tag{10}$$

Analogs of Proposition 2.1 can be formulated for each of these algorithms. Once again we can establish global consistency only for a perturbed algorithm, as in (7).

4 Spectral Decomposition of a Markov Chain

It is known that the rate of convergence to equilibrium for a finite state-space Markov chain is determined by the second largest eigenvalue of its transition matrix. Based on this observation, there is a large and growing literature on rates of convergence of Markov chains based on spectral theory and related methods.

For a reversible chain with finite state-space each of the eigenvalues is real. Diaconis and Stroock in [15] obtain bounds on the second largest eigenvalue in this setting. A striking conclusion is the following explicit bound on the rate of convergence, as defined by the total-variation norm distance (See [30] for another related estimate based on Prohorov's coefficient of ergodicity.):

$$\|P^n(x, \cdot) - \pi\| \le \sqrt{\tfrac{1-\pi(x)}{\pi(x)}}\, \lambda_*^n, \tag{11}$$

where λ_* is the magnitude of the second largest eigenvalue, $\lambda_*^2 = \max\{\lambda^2 : \lambda \ne 1\}$, and $\|\cdot\|$ denotes the total-variation norm. Bounds on the rate of convergence for chains that are not necessarily reversible are obtained in [17], again in the finite state-space case. The bounds are based on spectral theory, but the spectrum of the symmetrized kernel $PP^\dagger$ is considered, where $P^\dagger$ is the transition kernel for the time-reversed chain.

Just as eigenvectors are used for clustering in graph models, the use of *eigenvectors* [1] can be used to decompose a Markov model. This is a component of the classical Wentzell–Freidlin theory for model reduction. Much of this work concerns Markov processes that are reversible [8–10,14,18,31]. Extensions to non-reversible processes appeared for the first time in [19]. The foundation of this paper is the theory of quasi-stationarity, building on the work of [16]. Numerical methods are described in this aforementioned work and in [12,13].

In this section we restrict to the simpler reversible setting. Our goal is to obtain a variant of the stochastic approximation algorithm that will provide estimates of the spectrum of P rather than a symmetric matrix w.

Our starting point is a finite state space Markov chain $\boldsymbol{X}$ on the state space $\{1, \ldots, N\}$ with transition matrix P, and invariant measure π. It is assumed that $\boldsymbol{X}$ is irreducible and reversible. We write $\Pi = \operatorname{diag}(\pi)$ and $w := \Pi P$. Recall that reversibility implies that w is symmetric:

$$w = \Pi P = P^{\mathrm{T}}\Pi = w^{\mathrm{T}}. \tag{12}$$

Consider the matrix defined by the transformation:

$$w^\circ = \Pi^{-\frac{1}{2}} w \Pi^{-\frac{1}{2}} = \Pi^{\frac{1}{2}} P \Pi^{-\frac{1}{2}}. \tag{13}$$

This matrix remains symmetric. Suppose that v° is an eigenvector, i.e.,

$$w^\circ v^\circ = \lambda v^\circ.$$

Then by definition the vector $v = \Pi^{-\frac{1}{2}} v^\circ$ is an eigenvector of P.

The Oja o.d.e. to compute the spectrum of w° is given by:

$$\tfrac{d}{dt} m(t) = \Pi^{\frac{1}{2}} P \Pi^{-\frac{1}{2}} m(t) - m(t) m^T(t) \Pi^{\frac{1}{2}} P \Pi^{-\frac{1}{2}} m(t).$$

This is not attractive from the point of view of stochastic approximation. Letting $g = \Pi^{-\frac{1}{2}} m$ we eliminate the square-root in the o.d.e.:

$$\tfrac{d}{dt} g(t) = [I - g(t) g^T(t) \Pi] P g(t) \tag{14}$$

This is very similar to the original Oja o.d.e. using the matrix P. The point of all this is that this construction implies that $g(t)$ converges to the maximal eigenspace of P even though P is not symmetric.

To ensure convergence of the o.d.e. (1) or its stochastic approximation counterparts we must also assume that w° is positive definite. In this setting we are interested in calculating the eigenvalues that are maximal in modulus, so that adding the matrix rI will not solve the problem of interest. Instead, we work with the two-step transition matrix P^2. Its eigenvalues are the square of those of P, so that they are non-negative. We can then replace P with the the transition matrix $P_\varepsilon := \varepsilon I + (1-\varepsilon)P^2$ where $\varepsilon \in (0, 1)$ is arbitrary. This matrix has strictly

[1] *eigenfunctions* in general state space models

positive eigenvalues, which implies that w° is positive definite. To simplify notation we assume that P has been transformed in this way so that it is positive.

A discrete-time implementation of (14) is given by:

$$G(n+1) = G(n) + a(n)[I - G(n)G(n)^{\mathrm{T}}\Pi]PG(n), \quad (15)$$

where $\boldsymbol{a}$ is redefined by:

$$a(n) = b(n)(1 + \operatorname{trace}(G(n)G(n)^{\mathrm{T}}))^{-1}. \quad (16)$$

A stochastic approximation algorithm is obtained once more by mimicking the deterministic recursion. One algorithm is expressed in matrix form by:

$$G(n+1) - G(n) = a(n)\big[I - G(n)G(n)^{\mathrm{T}}\hat{\Pi}(n)\big]\widehat{P}(n)G(n), \quad (17)$$

based on the following definitions: $\hat{\pi}(n)$ is the empirical distribution of $\boldsymbol{X}$ based on the first n samples, $\hat{\Pi}(n) = \operatorname{diag}(\hat{\pi}(n))$, and $[\widehat{P}(n)]_{ij} = [\widehat{W}(n)]_{ij}/[\hat{\pi}(n)]_i$, with $[\widehat{W}(n)]_{ij} = \mathbf{1}(X(n) = i, X(n+1) = j)$. To avoid division by zero (and high variance), projected estimates should be employed for π:

$$\hat{\pi}(n+1) = [\hat{\pi}(n) + b(n)(I(n) - \hat{\pi}(n))]_S, \quad (18)$$

where $b(n)$ is a stepsize satisfying (6), $I_i(n) = \mathbf{1}(X(n) = i)$, and $[\,\cdot\,]_S$ is the projection onto a convex subset of the simplex satisfying $x_i > 0$ for each i when $x \in S$.

To obtain a version of the split sampling algorithm we recall the notation introduced in Section 2: $\boldsymbol{X}^1$ is i.i.d. with marginal π, and $\boldsymbol{X}^2$ is constructed based on the transition matrix P. We then apply the recursion (17) in which the random quantities are redefined by $[\widehat{W}(n)]_{ij} = \mathbf{1}(X^1(n) = i, X^2(n+1) = j)$, and $\hat{\pi}(n)$ is the true marginal π. There is no need to estimate the marginal since it is required in the construction of $\boldsymbol{X}^1$.

In experiments it is found that the multidimensional algorithm in which $N_m \geq 2$ is slow. To compute the second eigenvector an alternative algorithm is given as follows: The N-dimensional vector sequence $\boldsymbol{G}$ is constructed recursively:

$$\begin{aligned} G(n+1) - G(n) = {} & \\ & a(n)\big[I - G(n)G(n)^{\mathrm{T}}\hat{\Pi}(n)\big]\widetilde{P}_\varrho(n)G(n), \quad (19) \\ \widetilde{P}_\varrho(n) = {} & \widehat{P}(n) - \varrho 1 \otimes \hat{\pi}(n) \end{aligned}$$

where $\varrho \in (0,1)$ is chosen near unity, and we adopt the same conventions as above in the split sampling or Markovian versions. The associated o.d.e. is given by:

$$\tfrac{d}{dt}g(t) = [I - g(t)g^T(t)\Pi]P_\varrho g(t), \quad (20)$$

with $P_\varrho = P - \varrho 1 \otimes \pi$. This is a transformation of the o.d.e. (1) using the positive-definite matrix:

$$w^\circ = \Pi^{-\frac{1}{2}}(w - \varrho\pi \otimes \pi)\Pi^{-\frac{1}{2}} = \Pi^{\frac{1}{2}}P_\varrho\Pi^{-\frac{1}{2}}$$

Once again, analogs of Proposition 2.1 can be formulated for each of these algorithms, subject to the same caveats stated at the end of Section 3.

5 Risk Sensitive Control & Multiplicative Ergodicity

Assume that $\boldsymbol{X}$ is an irreducible Markov chain as in the previous section. Suppose that $c\colon \mathsf{X} \to \mathbb{R}$ is given, interpreted as a cost function in applications to control. The partial sums are denoted:

$$S_n = \sum_{t=0}^{n-1} c(X(t)).$$

The log moment generating function is defined as the limit (known to exist for aperiodic irreducible Markov chains [22]):

$$\Lambda(\theta) = \lim_{n\to\infty} \frac{1}{n} \log \mathsf{E}[\exp(\theta S_n)].$$

In applications to control, this is known as the *risk-sensitive cost* when $\theta > 0$. It is also a foundation of large deviations theory for Markov chains. Characterization of the limit is again based on spectral theory. Define the matrix

$$P_\theta(i,j) = e^{\theta c(i)}P(i,j). \quad (21)$$

For any i and any vector g, the n-fold product acting on g can be expressed as:

$$P_\theta^n g\,(i) = \mathsf{E}[\exp(\theta S_n)g(X(n)) \mid X(0) = i].$$

It follows that $\Lambda(\theta)$ coincides with the logarithm of the spectral radius of the matrix P_θ. Letting λ_θ denote the largest eigenvalue, we have:

$$\Lambda(\theta) = \log(\lambda_\theta).$$

There is an eigenvector h_θ with non-negative entries satisfying $P_\theta h_\theta = \lambda_\theta h_\theta$ — this is the Perron-Frobenious eigenvector. For any n,

$$\log(P_\theta^n h_\theta\,(i)) = n\Lambda(\theta) + \log(h_\theta(i)).$$

The function h_θ is a component of the risk-sensitive optimal control dynamic programming equation [35,7], and this eigenvector equation also characterizes the limiting log moment generating function in large deviations theory for Markov chains [22]. Stochastic approximation algorithms designed to estimate $\Lambda(\theta)$ and h_θ are described in [1–3].

To place this problem within the setting of this paper we proceed as in the previous section. Consider how (14) was derived: We have a matrix u that is self-adjoint in an inner-product space defined by a positive definite matrix Ξ. In the previous section $u = P$ and $\Xi = \Pi$. Using the same arguments as above we arrive at the o.d.e.:

$$\tfrac{d}{dt} g(t) = [I - g(t)g^T(t)\Xi]ug(t), \tag{22}$$

and we can conclude that $g(t)$ converges to the maximal eigenspace of u, provided the eigenvalues of u are all positive. We now take $u_{ij} = P_\theta(i,j) := e^{\theta c(i)} P(i,j)$. If P is reversible, then this matrix is self-adjoint with respect to the diagonal matrix $\Xi = \Pi_\theta := \text{diag}\,(e^{-\theta c(i)}\pi(i))$. Indeed,

$$\Pi_\theta P_\theta = \Pi P = P^{\mathrm{T}}\Pi = P_\theta^{\mathrm{T}}\Pi_\theta.$$

The o.d.e. (22) becomes:

$$\tfrac{d}{dt} g(t) = [I - g(t)g^T(t)\Pi_\theta]P_\theta g(t). \tag{23}$$

The solution $\{g(t)\}$ converges to the maximal eigenspace of P_θ for a.e. initial condition, provided P_θ has only non-negative eigenvalues.

In this application we typically take $N_m = 1$ since our interest is computation of the maximum eigenvalue and corresponding Perron-Frobenious eigenvalue. Hence we can add the scaled identity rI, with $r \geq 0$ sufficiently large to ensure that $rI + P_\theta$ has positive eigenvalues. A discrete-time deterministic algorithm is given by:

$$G(n+1) = G(n) + a(n)(I - G(n)G(n)^{\mathrm{T}}\Pi_\theta)[rI + P_\theta]G(n). \tag{24}$$

Stochastic approximation algorithms can be constructed based on observations of the Markov chain, or using split sampling, exactly as in the previous two settings. Convergence may require the introduction of an i.i.d. sequence $\boldsymbol{\xi}$, as in (7).

6 Examples

In most of the applications envisioned we are primarily interested in the sign structure of eigenvectors rather than their values. In such cases we judge the value of an estimate $\hat{v}$ of an eigenvector v by the error criterion:

$$\mathcal{E}(\hat{v}) = \min \|\text{sign}(v) - \text{sign}(r\hat{v})\|_1, \tag{25}$$

where the sign is computed pointwise, $\|\cdot\|_1$ denotes the ℓ_1 norm, and the minimum is over $r = \pm 1$.

We did not introduce the noise term $\boldsymbol{\xi}$ in any of our experiments. We found that the algorithms were globally convergent without this modification. The inherent randomness in the scheme was sufficient to push the iterates away from unstable attractors though it lacks the theoretical guarantee that explicit addition of extraneous noise ξ would have provided.

In our first set of examples we apply Oja's subspace algorithm for network decomposition.

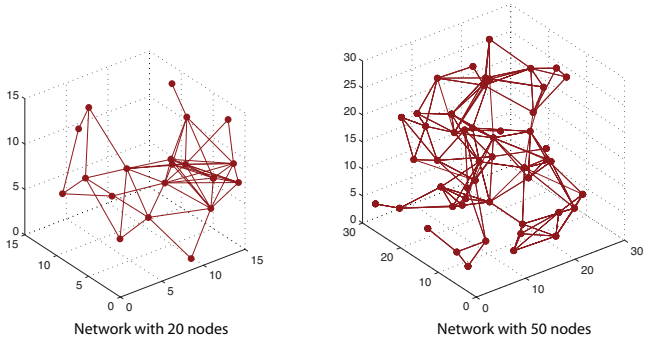

Fig. 1. The two networks considered in experiments.

6.1 *Spectral graph clustering*

Figure 1 shows the two graphs used in experiments using the deterministic algorithm, and its stochastic counterpart based on split sampling. The weighting matrix was chosen to coincide with the adjacency matrix, so that each weight was either one or zero.

Figure 2 shows results from several experiments using the normalized deterministic algorithm (3) and its stochastic approximation counterpart (2). These plots illustrate the transient behavior of the algorithm for each of the two graphs. In each plot, the vertical axis shows the error $\mathcal{E}(\hat{v}(n))$ for $n = 0, 2, \ldots$, where $\hat{v}$ is the estimate of the second eigenvector of w obtained from $m(n)$ defined by (3) (red dashed line), and $M(n)$ defined in (2) (blue solid). In these experiments the algorithm was run using $N_m = 2$. In each case the algorithm was run for 100,000 iterations. For $N = 20$ the initial 10,000 samples are shown together with the eigenvector approximation obtained after this many samples. Two sets of plots are shown for $N = 50$. These results are based on the initial 10,000 iterations, and also the final results after 100,000 iterations.

For either graph, the sign structure of the eigenvector is identified after approximately 3,000 iterations in the stochastic approximation algorithm. Convergence of the sign structure for the deterministic algorithm was nearly instantaneous.

Note that the slower rate of convergence in the stochastic algorithm is misleading since the required computation in each iteration is much smaller in the stochastic algorithm.

Convergence slowed considerably when N_m was increased. Figure 3 shows a comparison of the algorithm

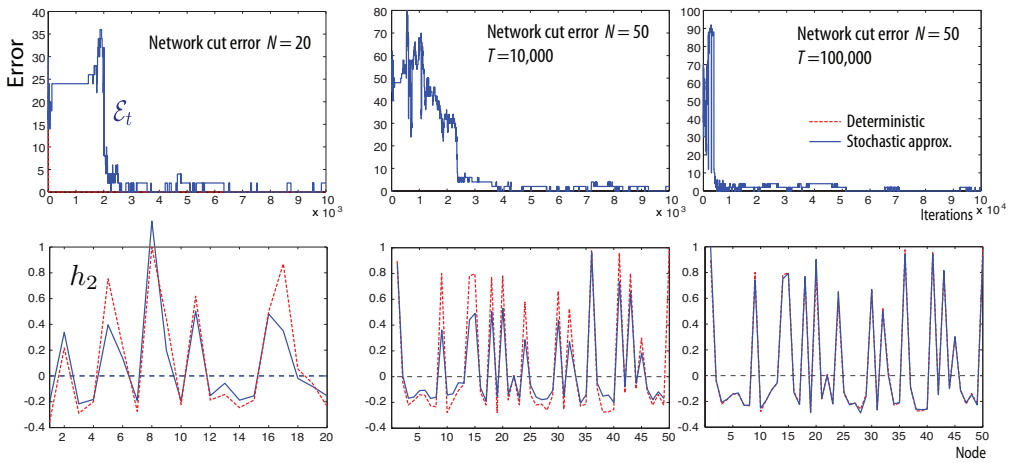

Fig. 2. Computation of the first spectral cut for the two networks shown in Figure 1. The first set of plots shows the error $\mathcal{E}(\hat{v}(n))$ as a function of time, and the second set of plots shows a comparison of h_2 and its approximation.

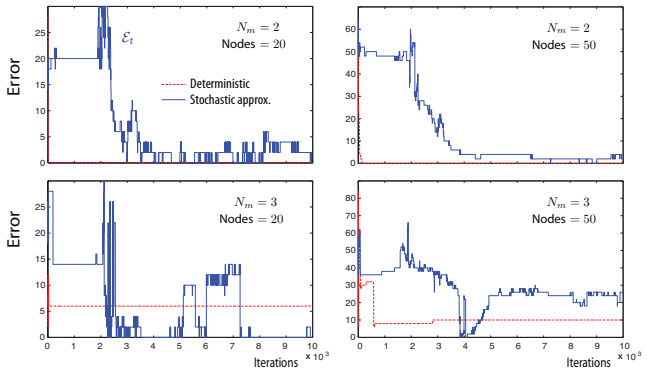

Fig. 3. Computation of the first and spectral cut for the 20 and the 50-node network after 10,000 samples. The rate of convergence is slowed significantly when N_m is increased from 2 to 3.

using $N_m = 2$ and $N_m = 3$. The convergence rate might be improved by first applying the algorithm with $N_m = 2$ to find the second eigenvector v^2, normalized so that its L_2-norm is unity. Replacing w by $w' = w - \lambda_2 v^2 v^{2\mathrm{T}}$, the algorithm can be re-run with $N_p = 2$ to compute v^3.

The next examples illustrate computation of the spectrum of a Markov transition matrix.

6.2 *Markovian spectral clustering*

To compute eigenvectors of the transition matrix we applied three approaches, each based on the recursion (19):

(i) The deterministic algorithm in which $\hat{\Pi}(n) \equiv \Pi$, $\widehat{P}(n) \equiv P$, and $\hat{\pi}(n) \equiv \pi$
(ii) The Markovian algorithm.
(iii) The algorithm based on split sampling (see (10)).

In each case the gain sequence was taken of the form (5) in which $b(n) = (1+n)^{-1}$ for $n \geq 0$.

In experiments we found that the split sampling approach converges much more quickly than the Markovian approach. Note however that the Markovian algorithm can be run using observations of the process $\boldsymbol{X}$, without knowledge of the model.

6.2.1 *Queueing model*

Following uniformization, the M/M/1/b model is a doubly reflected random walk:

$$Q(t+1) = [Q(t) + \Delta(t+1)]_{0,b} \tag{26}$$

where $[x]_{0,b} = \min(\max(x, 0), b)$ is a projection onto the interval $[0, b]$, and $\boldsymbol{\Delta}$ is an i.i.d. process. Letting α denote the arrival rate, and μ the service rate, scaled so that $\alpha + \mu = 1$, the marginal distribution of $\boldsymbol{\Delta}$ is given by,

$$\mathsf{P}\{\Delta(t) = 1\} = \alpha, \qquad \mathsf{P}\{\Delta(t) = -1\} = \mu.$$

Hence its Markov transition matrix is given by: for $x \in \mathsf{X}$,

$$\begin{aligned} P(x, \min(x+1, b)) &= \alpha, \\ P(x, \max(x-1, 0)) &= \mu \end{aligned} \tag{27}$$

To ensure that the matrix (13) is positive semi-definite we chose the Markov chain to be sampled at even integer values $X(k) = Q(2k)$, $k \geq 0$, in the stochastic approximation algorithm (19) based on Markovian observations. In the split sampling algorithm, the i.i.d. process $\boldsymbol{X}^1$ was constructed with geometric marginal distribution on X. For $n = 1, 2, \ldots$ the random variable $X^2(n)$ was chosen based on $X^1(n-1)$ using P^2, with P defined in (27).

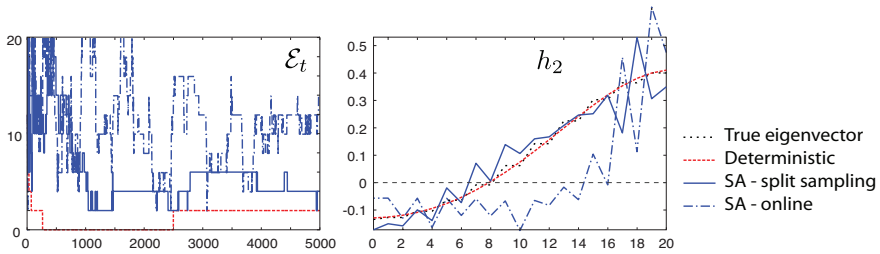

Fig. 4. Error trajectories defined by (25) and the final second eigenvalue estimates the M/M/1/20 queue. Shown on the left are error trajectories over a window of length 5,000 based on three algorithms: deterministic, stochastic split sampling algorithm, and the online algorithm based on observations of the queue. The resulting eigenvector approximations are shown on the right.

Figure 4 shows results obtained in the special case $b = 20$, and with $\rho{:=}\alpha/\mu = 0.9$. The on-line algorithm based on the observations of the queue length process required 500,000 samples to provide a useful approximation. Convergence using the deterministic or split-sampling algorithm was much faster.

We now turn to computation of the risk sensitive cost for this model, with $c(x) \equiv x$. Experiments were performed using $b = 20$ and $\rho := \alpha/\mu = 0.9$ as in the previous experiments. Note that $\Lambda(\theta) = \infty$ for every $\theta > 0$ in the unconstrained queue with $b = \infty$ [25]. Hence high variances can be expected for large values of b.

Figure 5 compares results obtained for the algorithm based on split sampling to those obtained for the deterministic algorithm in which (24) is replaced by:

$$G(n+1) = G(n) + a(n)(I - G(n)G(n)^{\mathrm{T}}\Pi)[rI + P_\theta]G(n)$$

The (fixed and deterministic) matrices Π and P_θ are defined in Section 5. In the deterministic and stochastic algorithms the offset rI was used, as shown in (24), with $r = e^\theta$.

On the plot shown on the left $\theta = 0.1$, and on the right $\theta = -0.1$. The deterministic and stochastic approximation algorithms were run for just 5,000 iterations.

6.2.2 Statistical mechanics model

A running example in [19] is the Smoluchowski equation, defined by the Itô equation:

$$dX(t) = -\nabla U(X(t))\, dt + \sigma dN(t),$$

where $\boldsymbol{N}$ is standard Brownian motion on $\mathbb{R}$, and the function $U\colon \mathbb{R} \to \mathbb{R}$ is the polynomial:

$$U(x) = \frac{1}{200}\Big(\tfrac{1}{2}x^6 - 15x^4 + 119x^2 + 28x + 50\Big).$$

Eigenfunctions of this diffusion were used to construct metastable subsets of $\mathbb{R}$.

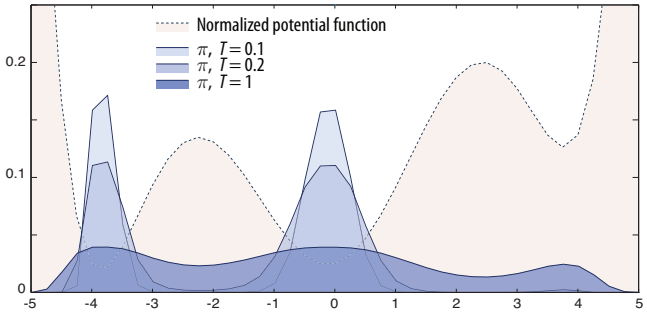

Fig. 6. Plot of the invariant measure π for $N = 41$ and for three different temperatures, $T_e = 0.1, 0.2$ and 1. Also shown is a plot of the normalized potential function $U/10$.

Here we consider a related discrete-time Markov chain, and compute the spectrum of the transition matrix using the algorithms introduced in Section 4.

The Markov chain is constructed by restricting to a finite subset of $\mathbb{R}$: x restricted to N equally spaced values between -5 and 5, denoted $\mathsf{X} = \{-5, -5+\delta, -5+2\delta, \ldots, 5-\delta, 5\}$ where $\delta = 10/(N-1)$. We fix a scalar $T_e > 0$ called the *temperature*, and define a bivariate distribution w on $\mathsf{X} \times \mathsf{X}$ as follows:

$$w(x,y) = \frac{1}{\zeta}\exp(-(\max(U(x), U(y))/T_e)),$$

where ζ is the normalizing factor, defined so that $\sum_{x,y} w(x,y) \equiv 1$. As in the general construction described in Section 2, we define $\pi(x) = \sum_y w(x,y)$, $x \in \mathsf{X}$, and a transition matrix is defined by $P(x,y) = w(x,y)/\pi(x)$, $x, y \in \mathsf{X}$. The discretized Smoluchowski equation is then defined as the Markov chain with transition matrix:

$$P(x,y) = \frac{1}{\alpha(x)}\exp\Big(-(\max(U(y) - U(x), 0)/T_e)\Big),$$

with $\alpha(x) := \sum_{y'} \exp\Big(-(\max(U(y') - U(x), 0)/T_e)\Big)$.

Shown on the right in Figure 7 is the resulting eigenvector approximation after 5,000 iterations using the deterministic and split sampling algorithms. Shown on the

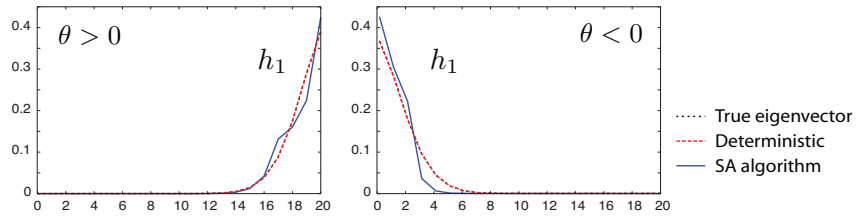

Fig. 5. Approximations of the second eigenvector for the matrix (21) for the M/M/1/b model with $b = 20$ and $c(x) \equiv x$, following 5,000 iterations of the algorithm deterministic and split sampling stochastic approximation algorithms.

left is the error process (25) using this algorithm. Convergence of the SA algorithm is slow, but note that a time horizon of 5,000 steps is very short. In this model, convergence to within 1% occurred after approximately 100,000 iterations.

7 Conclusions

We have introduced several stochastic-approximation variants of Oja's subspace algorithm for principal component analysis, Markov spectral theory, and spectral graph clustering. Convergence of these algorithms has been established through recent stochastic approximation techniques combined with stability theory from [11] that establishes convergence of the associated o.d.e..

Questions in current research include extensions to Markov chains that are not reversible, decentralized implementation, variance reduction techniques for these algorithms, and variants of the scheme in Section 4; in particular, alternate choices of functions of P instead of P^2.

References

[1] A. Basu, T. Bhattacharyya, and V. S. Borkar. A learning algorithm for risk-sensitive cost. *Math. Oper. Res.*, 33(4):880–898, 2008.

[2] V. S. Borkar. A sensitivity formula for risk-sensitive cost and the actor-critic algorithm. *Systems Control Lett.*, 44:339–346(8), 2001.

[3] V. S. Borkar. Q-learning for risk-sensitive control. *Math. Oper. Res.*, 27(2):294–311, 2002.

[4] V. S. Borkar. Reinforcement learning — a bridge between numerical methods and Markov Chain Monte Carlo. In N. S. N. Sastry, B. Rajeev, Mohan Delampady, and T. S. S. R. K. Rao, editors, *Perspectives in Mathematical Sciences*. World Scientific, 2008.

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

[6] V. S. Borkar and S. P. Meyn. The ODE method for convergence of stochastic approximation and reinforcement learning. *SIAM J. Control Optim.*, 38(2):447–469, 2000. (also presented at the *IEEE CDC*, December, 1998).

[7] V. S. Borkar and S. P. Meyn. Risk-sensitive optimal control for Markov decision processes with monotone cost. *Math. Oper. Res.*, 27(1):192–209, 2002.

[8] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein. Metastability in stochastic dynamics of disordered mean-field models. *Probab. Theory Related Fields*, 119(1):99–161, 2001.

[9] A. Bovier, M. Eckhoff, V. Gayrard, and M. Klein. Metastability in reversible diffusion processes. I. Sharp asymptotics for capacities and exit times. *J. Eur. Math. Soc. (JEMS)*, 6(4):399–424, 2004.

[10] A. Bovier, V. Gayrard, and M. Klein. Metastability in reversible diffusion processes. II. Precise asymptotics for small eigenvalues. *J. Eur. Math. Soc. (JEMS)*, 7(1):69–99, 2005.

[11] T. Chen, Y. Hua, and W.-Y. Yan. Global convergence of Oja's subspace algorithm for principal component extraction. *IEEE Trans. Neural Networks*, 9(1):58–67, Jan. 1998.

[12] K. Deng, P.G. Mehta, and S.P. Meyn. Optimal Kullback-Leibler aggregation via the spectral theory of Markov chains. To appear as a Regular Paper in the IEEE Transactions on Automatic Control, 2011.

[13] K. Deng, Y. Sun, P.G. Mehta, and S.P. Meyn. An information-theoretic framework to aggregate a Markov chain. In *Proc. of the 2009 American Control Conference (ACC)*, pages 731–736, 2009.

[14] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schütte. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra Appl.*, 315(1-3):39–59, 2000.

[15] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov chains. *Ann. Appl. Probab.*, 1:36–61, 1991.

[16] P. A. Ferrari, H. Kesten, and S. Martínez. R-positivity, quasi-stationary distributions and ratio limit theorems for a class of probabilistic automata. *Ann. Appl. Probab.*, 6:577–616, 1996.

[17] J. A. Fill. Eigenvalue bounds on convergence to stationarity for nonreversible Markov chains, with an application to the exclusion process. *Ann. Appl. Probab.*, 1(1):62–87, 1991.

[18] W. Huisinga. *Metastability of Markovian Systems: A transfer operator approach to molecular dynamics*. PhD thesis, Free University Berlin, 2001.

[19] W. Huisinga, S. P. Meyn, and C. Schütte. Phase transitions and metastability in Markovian and molecular systems. *Ann. Appl. Probab.*, 14(1):419–458, 2004. Presented at the 11TH INFORMS Applied Probability Society Conference, NYC, 2001.

[20] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[21] I. T. Jolliffe. *Principal component analysis*. Springer Series in Statistics. Springer-Verlag, New York, second edition, 2002.

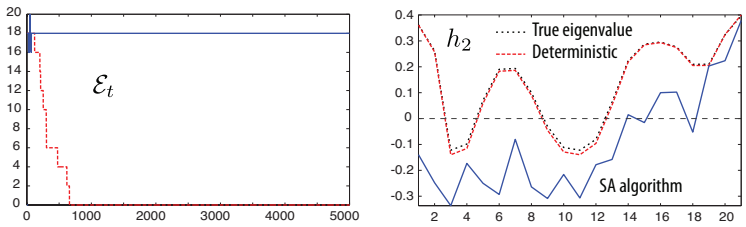

Fig. 7. Results for the discretized Smoluchowski equation based on the potential (6) using $T_e = 1$. Shown on the right is a plot of the second eigenvector for P and the approximation obtained from 5,000 iterations of the deterministic and split sampling algorithms. Shown on the left is the error process (25).

[22] I. Kontoyiannis and S. P. Meyn. Spectral theory and limit theorems for geometrically ergodic Markov processes. *Ann. Appl. Probab.*, 13:304–362, 2003. Presented at the INFORMS Applied Probability Conference, NYC, July, 2001.

[23] T. P. Krasulina. Method of stochastic approximation in the determination of the largest eigenvalue of the mathematical expectation of random matrices. *Automat. Remote Control*, 2:215–221, 1970.

[24] M. Loève. *Probability Theory II*. Springer-Verlag, New York, Heidelberg, Berlin, 4th edition, 1978.

[25] S. P. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, Cambridge, 2007.

[26] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Cambridge University Press, Cambridge, second edition, 2009. Published in the Cambridge Mathematical Library. 1993 edition online: `http://black.csl.uiuc.edu/~meyn/pages/book.html`.

[27] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press, Cambridge, MA, 2006.

[28] E. Oja. A simplified neuron model as a principal component analyzer. *J. Math. Biol.*, 15(3):267–273, 1982.

[29] E. Oja and J. Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *J. Math. Anal. Appl.*, 106(1):69–84, 1985.

[30] A.S. Poznyak, K. Najim, and E. Gómez-Ramirez. *Self-Learning Control of Finite Markov Chains*. Control Engineering. Marcel Dekker, New York, first edition, 2000.

[31] L. Rey-Bellet and L. E. Thomas. Asymptotic behavior of thermal nonequilibrium steady states for a driven chain of anharmonic oscillators. *Comm. Math. Phys.*, 215(1):1–24, 2000.

[32] B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[33] R. Sikora and W. Skarbek. On stability of Oja algorithm. In Lech Polkowski and Andrzej Skowron, editors, *Rough Sets and Current Trends in Computing*, volume 1424 of *Lecture Notes in Computer Science*, pages 354–360. Springer Berlin / Heidelberg, 2009.

[34] Y. Weiss. Segmentation using eigenvectors: a unifying view. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 2:975–982 vol.2, 1999.

[35] P. Whittle. *Risk-Sensitive Optimal Control*. John Wiley and Sons, Chichester, NY, 1990.

[36] Z. Yi, M. Ye, J. C. Lv, and Kok Kiong Tan. Convergence analysis of a deterministic discrete time system of Oja's PCA learning algorithm. *IEEE Trans. Neural Networks*, 16(6):1318–1328, Nov. 2005.