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

Sean Meyn
Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory
University of Illinois

Joint work with Vivek Borkar, Tata Institute of Fundamental Research
Supported in part by J. C. Bose Fellowship of Dept. of Science and Technology, Govt. of India,
General Motors India Science Laboratory, National Science Foundation ECS-0523620

Outline

Oja's algorithm for PCA

Stability and the o.d.e. @ ∞

Markov spectral theory

Conclusions

Outline

Oja's algorithm for PCA

Stability and the o.d.e. @ ∞

Markov spectral theory

Conclusions

Issues

Compute leading eigenvalues and eigenvectors of a matrix w

In matlab: `[V,l] = eig(w)`

We want only the leading eigenvalues, with w possibly very large

Applications: Model reduction via PCA / spectral graph theory

Hyvarinen 1999
Jolliffe 2002

Scholkopf, Smola, and Muller, 1998
Weiss, 1999
Nadler, Lafon, Coifman, and Kevrekidis 2006

Model reduction via Markov spectral graph theory

Rey-Bellet and Thomas, 2000
Deuflhard, Huisinga, Fischer, and Schuette, 2000
Bovier, Eckhoff, Gayrard, and Klein, 2001, 2004, 2005
Huisinga 2001
Huisinga, M., Schuette 2004

Issues

We want only the leading eigenvalues, with w possibly very large

Applications: Model reduction via PCA / spectral graph theory

Hyvarinen 1999
Jolliffe 2002

Scholkopf, Smola, and Muller, 1998
Weiss, 1999
Nadler, Lafon, Coifman, and Kevrekidis 2006

Model reduction via Markov spectral graph theory

Rey-Bellet and Thomas, 2000
Deuflhard, Huisinga, Fischer, and Schuette, 2000
Bovier, Eckhoff, Gayrard, and Klein, 2001, 2004, 2005
Huisinga 2001
Huisinga, M., Schuette 2004

Risk sensitive control and large deviations

Whittle 1990
Borkar and Meyn 2002
Kontoyiannis andMeyn 2002 -

Oja's Algorithm

Krasulina,1970
Oja 1982
Oja and Karhunen 1984
Chen, Hua, and Yan, 1998
Yi, Ye, Lv, and Tan, 2005

Compute leading eigenvalues and eigenvectors of a matrix w

Notation: First eigenvectors expressed as a matrix m^*

$$m^* = \left[v^1 \middle| \cdots \middle| v^{N_m} \right]$$

$$wm^* = m^* \Lambda$$

$$\Lambda = \mathrm{diag}\left(\lambda_i\right)$$

Oja's Algorithm

Krasulina,1970
Oja 1982
Oja and Karhunen 1984
Chen, Hua, and Yan, 1998
Yi, Ye, Lv, and Tan, 2005

Compute leading eigenvalues and eigenvectors of a matrix w

Notation: First eigenvectors expressed as a matrix

$$m^* = \left[v^1 \middle| \cdots \middle| v^{N_m} \right]$$

Oja's o.d.e.,

$$\tfrac{d}{dt} m(t) = [I - m(t) m^T(t)] w m(t)$$

Deterministic approximation,

$$m(n+1) - m(n) = a(n) \left[I - m(n) m^T(n) \right] w m(n)$$

Oja's Algorithm

Krasulina,1970
Oja 1982
Oja and Karhunen 1984
Chen, Hua, and Yan, 1998
Yi, Ye, Lv, and Tan, 2005

Compute leading eigenvalues and eigenvectors of a matrix w

Notation: First eigenvectors expressed as a matrix

$$m^* = \left[v^1 \middle| \cdots \middle| v^{N_m} \right]$$

Oja's o.d.e.,

$$\frac{d}{dt} m(t) = [I - m(t) m^T(t)] w m(t)$$

Deterministic approximation,

$$m(n+1) - m(n) = a(n) \left[I - m(n) m^T(n) \right] w m(n)$$

Convergent for a.e. initial condition provided w is positive definite

Oja's Algorithm - SA implementation

Krasulina,1970
Oja 1982
Oja and Karhunen 1984
Chen, Hua, and Yan, 1998
Yi, Ye, Lv, and Tan, 2005

Suppose that $\boldsymbol{X}$ is an n-dimensional stochastic process

Covariance matrix: $w = \mathsf{E}[X(t)X(t)^{\mathrm{T}}]$

Oja & Karhunen 1984:

$$M(n+1) - M(n) = a(n)\left[I - M(n)M^{\mathrm{T}}(n)\right]\widehat{W}(n)M(n)$$

$$\widehat{W}(n) = X(n)X^{\mathrm{T}}(n)$$

Converence is established only under strong conditions on $\boldsymbol{X}$

Oja's Algorithm - SA implementation

Suppose that $\boldsymbol{X}$ is an n-dimensional stochastic process

Covariance matrix: $w = \mathsf{E}[X(t)X(t)^{\mathrm{T}}]$

Oja & Karhunen 1984:

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

$$\widehat{W}(n) = X(n)X^{\mathrm{T}}(n)$$

Converence is established only under strong conditions on $\boldsymbol{X}$

Difficulty: Cubic nonlinearity in recursion and in o.d.e.

Oja's Algorithm - SA implementation

Suppose that $\boldsymbol{X}$ is an n-dimensional stochastic process

Covariance matrix: $w = \mathsf{E}[X(t)X(t)^{\mathrm{T}}]$

Oja & Karhunen 1984:

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

$$\widehat{W}(n) = X(n)X^{\mathrm{T}}(n)$$

Converence is established only under strong conditions on $\boldsymbol{X}$

Difficulty: Cubic nonlinearity in recursion and in o.d.e.
Solution: Scale RHS so that it is Lipschitz

Outline

Oja's algorithm for PCA

Stability and the o.d.e. @ ∞

Markov spectral theory

Conclusions

Scaling

Difficulty: Cubic nonlinearity in recursion and in o.d.e.
Solution: Scale RHS so that it is Lipschitz

Scaled o.d.e.

$$\frac{d}{dt} m(t) = a(t)\big[I - m(t)m^T(t)\big] w m(t)$$

$$a(t) = \left(1 + \mathrm{trace}\left(m(t)m(t)^{\mathrm{T}}\right)\right)^{-1}$$

Scaling

Difficulty: Cubic nonlinearity in recursion and in o.d.e.
Solution: Scale RHS so that it is Lipschitz

Scaled o.d.e.

$$\frac{d}{dt} m(t) = a(t)\big[I - m(t)m^T(t)\big]wm(t)$$

$$a(t) = \left(1 + \mathrm{trace}\left(m(t)m(t)^{\mathrm{T}}\right)\right)^{-1}$$

Time-scaling of original o.d.e.
Convergence properties maintained

Scaled Stochastic Approximation Algorithm

Stochastic approximation algorithm

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

$\boldsymbol{W}$ is i.i.d. with mean w

Scaled Stochastic Approximation Algorithm

Stochastic approximation algorithm

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

$$a(n) = b(n)(1 + \mathrm{trace}\,(M(n)M(n)^{\mathrm{T}}))^{-1}$$

$$b(n) = (1+n)^{-1},\ n \geq 0$$

Scaled Stochastic Approximation Algorithm

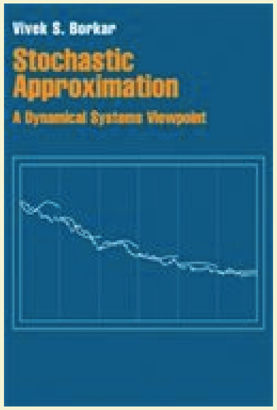
Stochastic approximation algorithm

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

$$a(n) = b(n)(1 + \mathrm{trace}\,(M(n)M(n)^{\mathrm{T}}))^{-1}$$

$$b(n) = (1+n)^{-1},\; n \geq 0$$

RHS is now Lipschitz, so that standard theory applies



Graph Partitioning

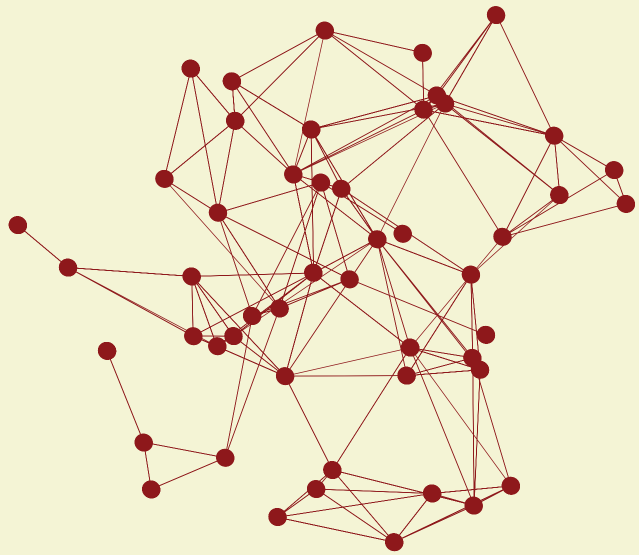
Network with fifty nodes

Sign structure of eigenvectors of graph used for decomposition

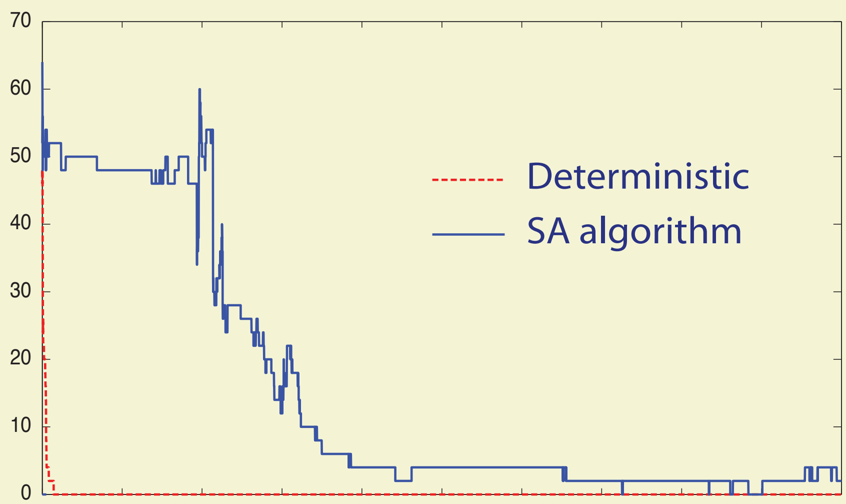
Graph Partitioning

Network with fifty nodes

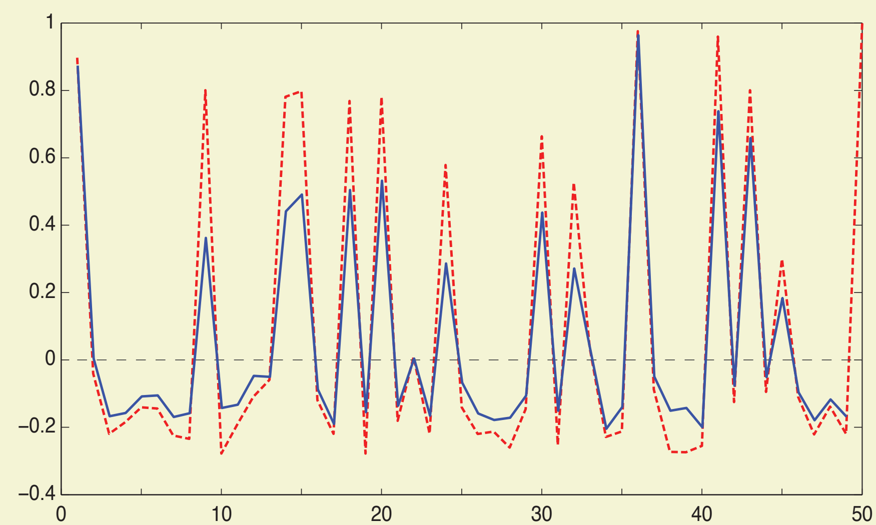
Sign structure of eigenvectors
of graph used for decomposition



$N_m = 2$



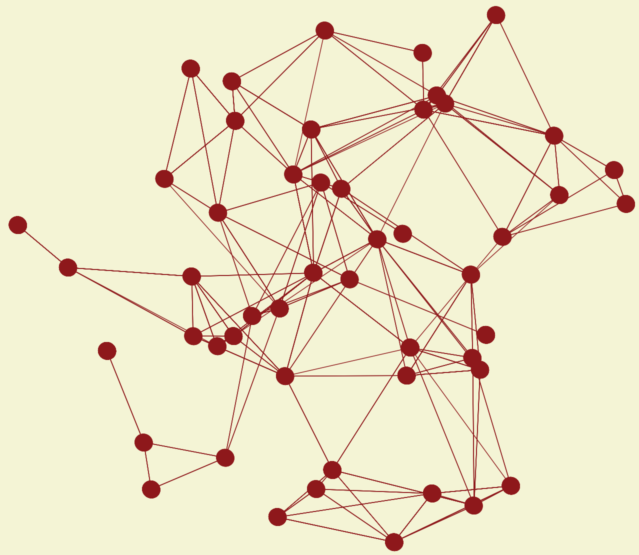
Evolution of sign error count for each algorithm



Estimates of second eigenvector after 10,000 steps

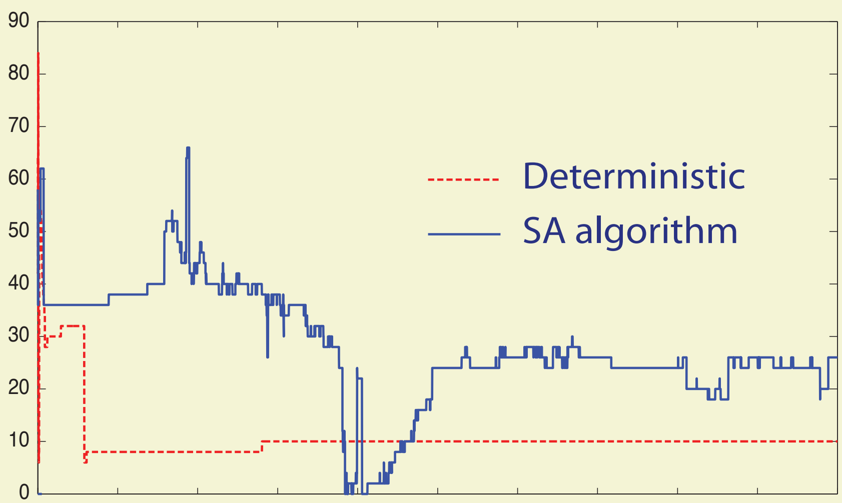
Graph Partitioning

Network with fifty nodes



Sign structure of eigenvectors of graph used for decomposition

$N_m = 3$



Evolution of sign error count for each algorithm - 10,000 steps

Convergence is slowed *significantly* when estimating the three dimensional eigenspace compared with two

Convergence of Oja's Algorithm

$$M(n+1) - M(n) = a(n)\left[I - M(n)M^{\mathrm{T}}(n)\right]W(n)M(n)$$
$$+\ a(n)\,\xi(n+1)$$

Why ξ ?

Convergence of Oja's Algorithm

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

Why ξ ?

Avoidance of traps: There are many undesirable fixed points of the o.d.e.

Convergence of Oja's Algorithm

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

Assumptions:

ξ and w are independent

ξ has a non-vanishing density

$w > 0$

Convergence of Oja's Algorithm

$$M(n+1) - M(n) = a(n)\left[I - M(n)M^{\mathrm{T}}(n)\right]W(n)M(n) + a(n)\,\xi(n+1)$$

Assumptions:

- ξ and w are independent
- ξ has a non-vanishing density
- $w > 0$

Then, any limit point of the algorithm has columns that lie in the eigenspace spanned by the first eigenvalues of w

Convergence of Oja's Algorithm: Proof of stability

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

Fluid model, or *o.d.e. at infinity*:

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

Obtained by a LLN applied to the algorithm,
letting the initial condition tend to infinity

Convergence of Oja's Algorithm: Proof of stability

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

Fluid model, or *o.d.e. at infinity*:

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

Lyapunov function:

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

Convergence of Oja's Algorithm: Proof of stability

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

Fluid model, or *o.d.e. at infinity*:

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

Lyapunov function:

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

Stability of o.d.e.

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

Convergence of Oja's Algorithm: Proof of stability

SA:

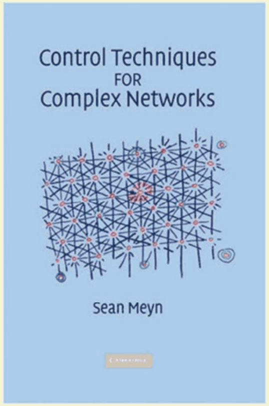
$$M(n+1) - M(n) = a(n)\big[I - M(n)M^{\mathrm{T}}(n)\big]W(n)M(n) \\ + \; a(n)\,\xi(n+1)$$

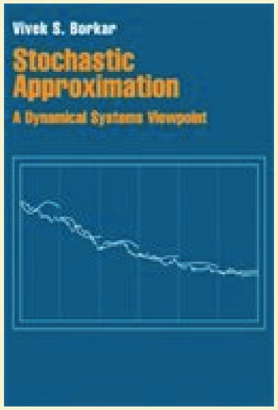
ODE:

$$\frac{d}{dt}m^{\infty}(t) = -\Big[\frac{m^{\infty}(t)m^{\infty T}(t)}{\mathrm{trace}\,(m^{\infty}(t)m^{\infty}(t)^{\mathrm{T}})}\Big]wm^{\infty}(t)$$

Stability of the o.d.e. implies that the stochastic algorithm has bounded sample paths

Borkar and Meyn, 2000





Outline

Oja’s algorithm for PCA

Stability and the o.d.e. @ ∞

Markov spectral theory

Conclusions

Markov Spectral Theory

Assumptions:

P is a Markov transition matrix
for a *reversible* Markov chain

Stationary distribution:

$$\pi P = \pi$$

Markov Spectral Theory

Assumptions:

P is a Markov transition matrix
for a *reversible* Markov chain

Reversibility: Self-adjoint in the Π-norm $\quad \Pi = \mathrm{diag}(\pi)$

$$\pi P = \pi$$

Markov Spectral Theory

Assumptions:

P is a Markov transition matrix
for a *reversible* Markov chain

Reversibility: Self-adjoint in the Π-norm $\quad \Pi = \mathrm{diag}(\pi)$

$$\pi P = \pi$$

Equivalently

$$w = \Pi P = P^T \Pi \quad \textit{symmetric}$$

Markov Spectral Theory

Assumptions:

P is a Markov transition matrix
for a *reversible* Markov chain

Reversibility: Self-adjoint in the Π-norm $\quad \Pi = \text{diag}(\pi)$

$$\pi P = \pi$$

Equivalently

$$w = \Pi P = P^T \Pi \quad \textit{symmetric}$$

Equivalently

$$w = \Pi^{\frac{1}{2}} P \Pi^{-\frac{1}{2}} \quad \textit{symmetric}$$

Shares the same eigenvalues as P

Oja's Algorithm for Markov chains

$$M(n+1) - M(n) = a(n)\big[I - M(n)M^{\mathrm{T}}(n)\,\Pi(n)\big]P(n)M(n) \\ +\ a(n)\,\xi(n+1)$$

$\Pi(n)$: Estimate of Π via Monte-Carlo

$P(n)$: Estimate of P

Oja's Algorithm for Markov chains

$$M(n+1) - M(n) = a(n)\big[I - M(n)M^{\mathrm{T}}(n)\,\boldsymbol{\Pi}(n)\big]P(n)M(n) \\ +\ a(n)\,\xi(n+1)$$

Simplest setting: $\boldsymbol{X}$ is a Markov chain

$\hat{\pi}(n)$ is the empirical distribution of $\boldsymbol{X}$

$$\hat{\Pi}(n) = \mathrm{diag}\,(\hat{\pi}(n))$$

$$[\widehat{P}(n)]_{ij} = [\widehat{W}(n)]_{ij}/[\hat{\pi}(n)]_i$$

$$[\widehat{W}(n)]_{ij} = \mathbb{I}(X(n) = i,\, X(n+1) = j)$$

Oja's Algorithm for Markov chains

$$M(n+1) - M(n) = a(n)\big[I - M(n)M^{\mathrm{T}}(n)\,\Pi(n)\big]P(n)M(n) \\ + \; a(n)\,\xi(n+1)$$

Simplest setting: $\boldsymbol{X}$ is a Markov chain

$\hat{\pi}(n)$ is the empirical distribution of $\boldsymbol{X}$

$\hat{\Pi}(n) = \mathrm{diag}\,(\hat{\pi}(n))$

$[\widehat{P}(n)]_{ij} = [\widehat{W}(n)]_{ij}/[\hat{\pi}(n)]_i$

$[\widehat{W}(n)]_{ij} = \mathbb{I}(X(n) = i, X(n+1) = j)$

Refinement: Replace P by $rI + P$

to ensure positive spectrum

Oja's Algorithm for Markov chains

$$M(n+1) - M(n)$$
$$= a(n)\big[I - M(n)M^{\mathrm{T}}(n)\,\Pi(n)\big]\big[rI + P(n)\big]M(n)$$
$$+ \; a(n)\,\xi(n+1)$$

Assumptions:

ξ and $\boldsymbol{X}$ are independent

ξ has a non-vanishing density

$rI + P$ has a positive spectrum

Then, any limit point of the algorithm has columns that lie in the eigenspace spanned by the first eigenvalues of w

Oja's Algorithm for Risk Sensitive Control

Given a cost function c, partial sums:

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

Risk-sensitive cost:

$$\Lambda(\theta) = \lim_{n\to\infty} \frac{1}{n} \log \mathsf{E}_x[\exp(\theta S_n)] \qquad \theta > 0$$

Maximum eigenvalue of $\quad P_\theta(i,j) = e^{\theta c(i)} P(i,j)$

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

Oja's Algorithm for Risk Sensitive Control

Given a cost function c, partial sums:

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

Risk-sensitive cost:

$$\Lambda(\theta) = \lim_{n\to\infty} \frac{1}{n} \log \mathsf{E}_x[\exp(\theta S_n)] \qquad \theta > 0$$

Maximum eigenvalue of $\quad P_\theta(i,j) = e^{\theta c(i)} P(i,j)$

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

Found using Oja if P is reversible

Outline

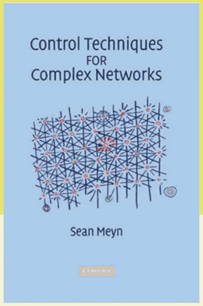
Oja's algorithm for PCA

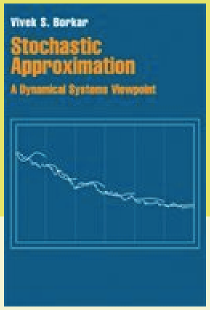
Stability and the o.d.e. @ ∞

Markov spectral theory

Conclusions

Conclusions



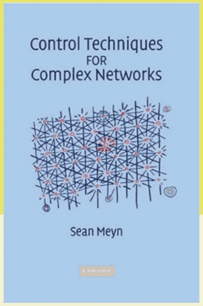


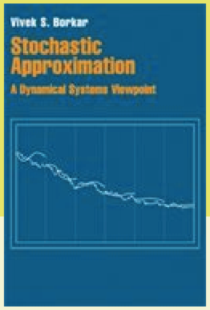
Contributions: A natural scaling ensures stability

Proof based on new fluid-model approach to stability of stochastic models

New applications to Markov models

Conclusions





Contributions: A natural scaling ensures stability

Proof based on new fluid-model approach to stability of stochastic models

New applications to Markov models

Future directions:

Stochastic algorithms prone to high variance

Control variates for these algorithms?

Is symmetry (or reversibility) truly necessary?

ODE approach shows that stability only requires

$$w + w^T > 0$$

What about convergence?