Coarse-grained projective schemes for particle filters

Dror Givon

Princeton University

Coworkers

- Coarse-grained projective schemes
  - Ioannis Kevrekidis - Princeton
  - Bill Gear - Princeton
  - Thomas Frewen and Michail Kavousanakis - Princeton
- Dimension reduction, Variance reduction for Particle Filters
  - Panagiotis Stinis - Stanford, Minnesota
  - Jonathan Weare - Courant

Outline

- Stochastic dynamical systems that exhibit time-scale separation
- Coarse projective schemes (Equation Free and Averaging principle)
- Particle Filters

Stochastic dynamical systems

- Deterministic dynamical system

$$dx = a(x)dt$$
$$x(t+\Delta t) - x(t) = a(x(t))\Delta t$$

- Stochastic dynamical system
  - Stochastic differential equation

$$dx = a(x)dt + \theta(x)dW$$
$$x(t+\Delta t) - x(t) = a(x(t))\Delta t + \theta(x(t))\mathcal{N}(0, \Delta t)$$

  - Jump process

$$dx = a(x)dP(\lambda)$$
$$x(t+\tau) - x(t) = a(x(t)), \tau \propto Exp(1/\lambda)$$

Scale separation

$$\begin{cases} dx &= a(x,y)dt \\ dy &= \frac{1}{\epsilon}b(x,y)dt + \frac{1}{\sqrt{\epsilon}}\sigma(x,y)dW \end{cases}$$

where $\epsilon \ll 1$.

Scale separation

$$\begin{cases} dx &= a(x,y)dt \\ dy &= \frac{1}{\epsilon}b(x,y)dt + \frac{1}{\sqrt{\epsilon}}\sigma(x,y)dW \end{cases}$$

where $\epsilon \ll 1$.

We sample the dynamical system during a small time interval, small enough to ensure that the slow variable has not changed significantly.

Time scales:

i time scale of the detailed model/of the fast dynamics

ii the time needed for the y dynamics to fill up a distribution

iii the time scale for the slow variable

Averaging Principle

If for fixed x

$$dy = \frac{1}{\epsilon} b(x,y)dt + \frac{1}{\sqrt{\epsilon}} \sigma(x,y)dW$$

generates an invariant measure $\mu_x(dy)$ then $x(t) \approx X(t)$ where

$$dX(t) = \bar{a}(X(t))dt$$

$$\bar{a}(x) = \int a(x,y)\mu_x(dy)$$

or

$$(x(t), y(t)) \sim (X(t), \mu_{X(t)}(dy))$$

Averaging Principle

If for fixed x

$$dy = \frac{1}{\epsilon} b(x,y)dt + \frac{1}{\sqrt{\epsilon}} \sigma(x,y)dW$$

generates an invariant measure $\mu_x(dy)$ then $x(t) \approx X(t)$ where

$$dX(t) = \bar{a}(X(t))dt$$

$$\bar{a}(x) = \int a(x,y)\mu_x(dy)$$

or

$$(x(t), y(t)) \sim (X(t), \mu_{X(t)}(dy))$$

- In order to approximate the slow variable we need to know $a(x,y)$, the Right Hand Side
- It may happen that the detailed model is given as a “black box”

Equation Free

Black Box:

- RHS which consists of complicated functions
- Random walks, Stochastic processes
- Legacy code, Proprietary software, Closed source software
  - Input - (initial value at time t_0, integration time t_i)
  - Output - value/sample at time $t_0 + t_i$

Equation Free

Black Box:

- RHS which consists of complicated functions
- Random walks, Stochastic processes
- Legacy code, Proprietary software, Closed source software
  - Input - (initial value at time t_0, integration time t_i)
  - Output - value/sample at time $t_0 + t_i$

How do we integrate/simulate such a black box?

$$dX(t) = \bar{a}(X(t))dt$$
$$X(t + \Delta t) - X(t) = \bar{a}(X(t))\Delta t$$

without the knowledge of the function $\bar{a}(x)$.

Equation Free

$$\bar{a}(X(t)) = \lim_{h \to 0} \frac{X(t+h) - X(t)}{h}$$

Equation Free

- $$\bar{a}(X(t)) = \lim_{h \to 0} \frac{X(t+h) - X(t)}{h}$$

which suggests

- $$X(t + \Delta t) - X(t) \approx \frac{X(t+h) - X(t)}{h} \Delta t$$

Equation Free

- $$\bar{a}(X(t)) = \lim_{h \to 0} \frac{X(t+h) - X(t)}{h}$$

which suggests

- $$X(t+\Delta t) - X(t) \approx \frac{X(t+h) - X(t)}{h} \Delta t$$

- Short bursts enable a projective step/big jump in time

Equation Free for Multiscale dynamics

Recall that $(x(t), y(t)) \sim (X(t), \mu_{X(t)}(dy))$.

In this case our black box simulates $x(t)$ and not $X(t)$ hence

$$x(t+\Delta t) - x(t) \neq \frac{x(t+h) - x(t)}{h}\Delta t$$

$$\bar{a}(x(t))\Delta t \qquad\qquad a(x(t), y(t))\Delta t$$

Equation Free for Multiscale dynamics

Recall that $(x(t), y(t)) \sim (X(t), \mu_{X(t)}(dy))$.

In this case our black box simulates $x(t)$ and not $X(t)$ hence

$$\underset{\bar{a}(x(t))\Delta t}{x(t+\Delta t) - x(t)} \neq \underset{a(x(t),y(t))\Delta t}{\frac{x(t+h)-x(t)}{h}\Delta t}$$

Solution:

1. Given $X_{n\Delta t}$, choose $Y_{n\Delta t}$ (almost surely any choice is good)
2. Evolve the detailed model starting at $(X_{n\Delta t}, Y_{n\Delta t})$ for time length δt, to obtain $(X_{n\Delta t+\delta t}, Y_{n\Delta t+\delta t})$
3. Evolve the system for time length ηt, to obtain $\left(X_{n\Delta t+\delta t+\eta t}, Y_{n\Delta t+\delta t+\eta t}\right)$
4. Estimate $\bar{a}\left(x^{\epsilon}_{n\Delta t}\right)$ by evaluating

$$\frac{X_{n\Delta t+\delta t+\eta t} - X_{n\Delta t+\delta t}}{\eta t} =: A(X_{n\Delta t}) \approx \bar{a}(x^{\epsilon}_{n\Delta t})$$

5. Projective step:

$$X_{(n+1)\Delta t} = X_{n\Delta t} + A(X_{n\Delta t})\Delta t$$

Coarse variable effectively stochastic

What if $\quad X(t+\Delta t) - X(t) = \bar{a}(X(t))\Delta t + \bar{\theta}(X(t))\Delta W$

In that case different realizations will give different values for $X(t+\Delta t)$.

So how do we estimate $\bar{a}, \bar{\theta}$?

Solution:

1.
$$\bar{a}(X(t)) = \lim_{h\to 0} \frac{\mathbb{E}\left[X(t+h) - X(t)\right]}{h} \approx \frac{\frac{1}{N}\sum_{i=0}^{N}\left(X^i(t+h) - X(t)\right)}{h}$$

2.
$$\bar{\theta}\bar{\theta}^\top(X(t)) = \lim_{h\to 0} \frac{\mathbb{E}\left[X(t+h) - X(t)\right]^2}{h} \approx \frac{\frac{1}{N}\sum_{i=0}^{N}\left(X^i(t+h) - X(t)\right)^2}{h}$$

Multiscale case:

3.
$$\bar{a}(X(t)) \approx \frac{\mathbb{E}\left[x(t+\eta t) - X(t)\right]}{\eta t} \approx \frac{\frac{1}{N}\sum_{i=0}^{N}\left(x^i(t+\eta t) - X(t)\right)}{\eta t}$$

4.
$$\bar{\theta}\bar{\theta}^\top(X(t)) \approx \frac{\mathbb{E}\left[x(t+\eta t) - X(t)\right]^2}{\eta t} \approx \frac{\frac{1}{N}\sum_{i=0}^{N}\left(x^i(t+\eta t) - X(t)\right)^2}{\eta t}$$

Coarse variable effectively stochastic

!!! In order to make a Δt time step we use (instead) N short bursts of length ηt.

Coarse variable effectively stochastic

!!! In order to make a Δt time step we use (instead) N short bursts of length ηt.

Since all realizations/short bursts are independent we can run all N simulations in parallel (embarrassingly parallel)

- We dramatically reduce the wall-clock time while not increasing the overall processor usage
- We can generate the entire distribution rather than just obtaining one long (Δt) realization

Similar ideas are applied on pure jump processes, chemical reactions, where we can use short bursts to estimate jumps intensity and jumps amplitude.

Simple example - Chemical reaction

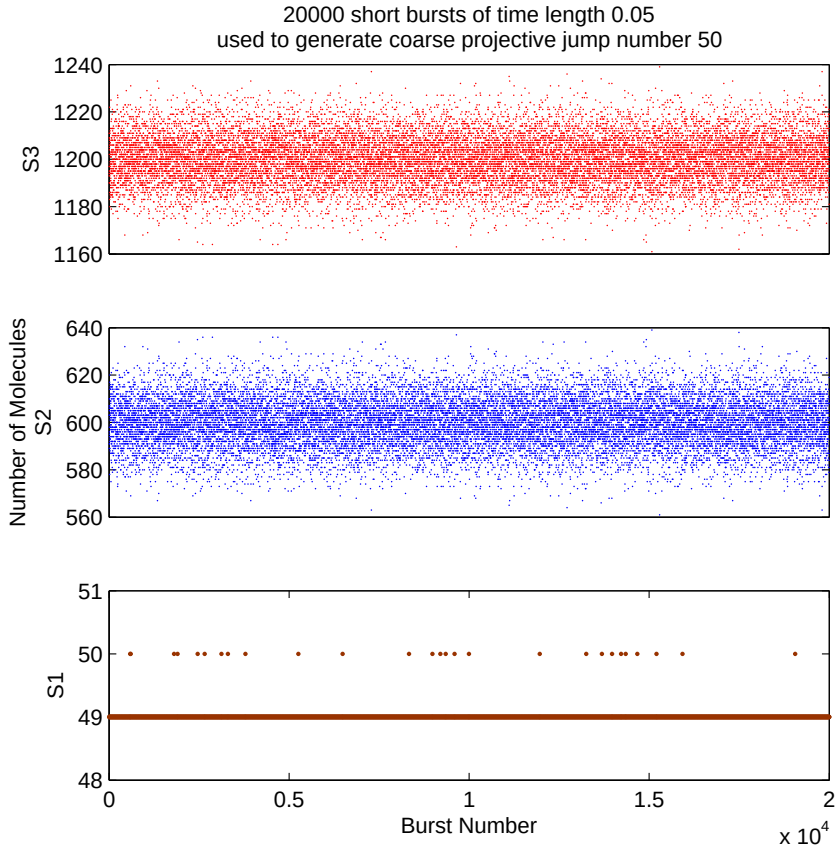
We demonstrate the scheme on the multiscale jump process given by

$$\begin{pmatrix} dS_1 \\ dS_2 \\ dS_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} dP_1(S_2) \\ dN_1(S_3) \\ dN_2(S_2) \end{pmatrix}, \tag{1}$$

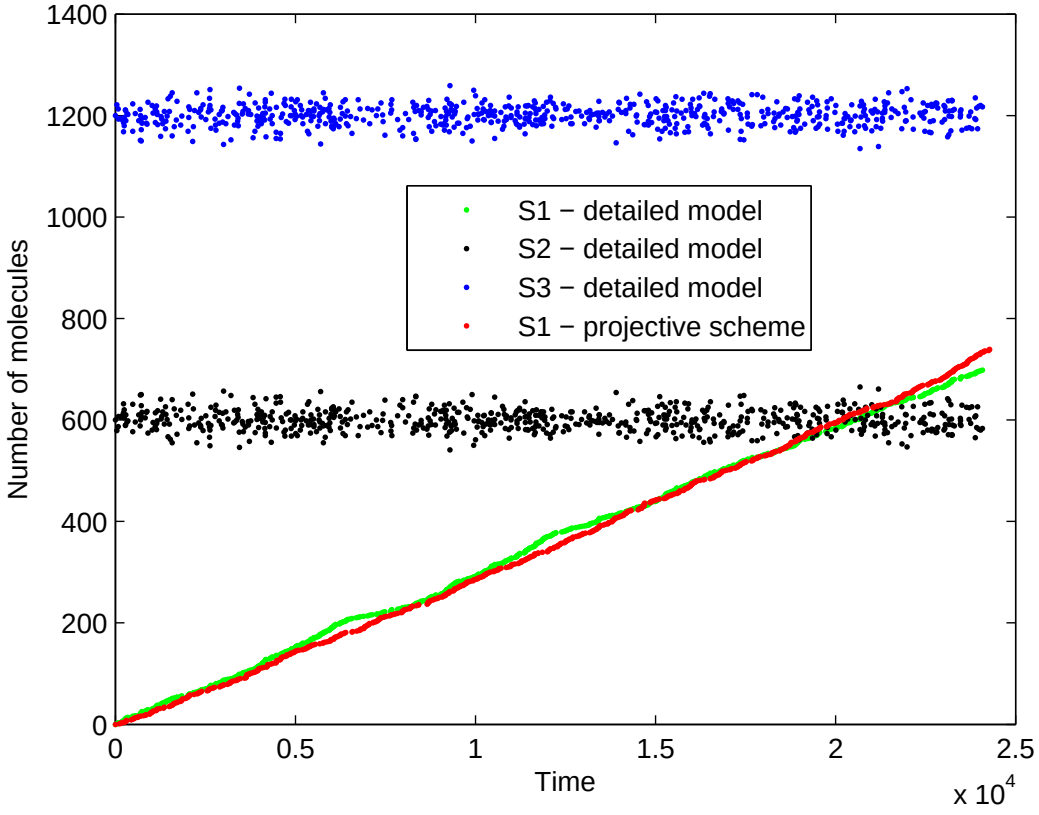
with initial value $(0, 600, 1200)$ and with intensity functions

$$\begin{pmatrix} a_1\left(S_1, S_2, S_3\right) \\ a_2\left(S_1, S_2, S_3\right) \\ a_3\left(S_1, S_2, S_3\right) \end{pmatrix} = \begin{pmatrix} 5 \cdot 10^{-5} S_2 \\ S_3 \\ 2S_2 \end{pmatrix}.$$

Simple example - Chemical reaction



Simple example - Chemical reaction



Filtering Problem

- Tracking the state of a system as it evolves over time - *Hidden Process*
- We have: Sequentially arriving (noisy or ambiguous) *observations*
- We want to know: Best possible estimate of the hidden variables

Example:

- $\rightsquigarrow$ Tracking of aircraft positions from radar
- £ Predicting economical data

Filtering Problem

- V_t is a Markov process with transition probability $Q_t(v, dv')$ where $Q_0(dv)$ is a known distribution - *Hidden Process*

Filtering Problem

- V_t is a Markov process with transition probability $Q_t(v, dv')$ where $Q_0(dv)$ is a known distribution - *Hidden Process*
- $S_1, \ldots, S_N$ are noisy discrete *observations* at N regularly spaced times t_i. The *likelihood* of S_{i+1} conditional on (V_{i+1}) is known and given by $s \mapsto g(V_{i+1}, s)$

Filtering Problem

- V_t is a Markov process with transition probability $Q_t(v, dv')$ where $Q_0(dv)$ is a known distribution - *Hidden Process*
- $S_1, \ldots, S_N$ are noisy discrete *observations* at N regularly spaced times t_i. The *likelihood* of S_{i+1} conditional on (V_{i+1}) is known and given by $s \mapsto g(V_{i+1}, s)$
- The filtering problem consists of computing the conditional expectations

$$\Pi_i f = \mathbb{E}\left[f(V_{t_i}) | S_1, \ldots, S_i\right]$$

for some function f.

Example - Filtering Problem

- $Q_t(v, dv')$ Predict next state from current estimate
- $g(V_{i+1}, S_{i+1})$ Update the prediction using sequentially arriving new measurements
- The filtering problem consists of computing the conditional expectations

$$\Pi_i f = \mathbb{E}\left[f(V_{t_i})|S_1, \ldots, S_i\right]$$

For example:

- $$dV_t = a(V_t)\,dt + b(V_t)\,dW_t$$
$$S_t = h(V_t) + \chi$$

  where χ is a random variable, noise term.

Particle Filter

1. Sample n i.i.d. vectors (particles) $v_0^j,\ j = 1, \ldots, n$ from $Q_0(dv)$. Set $k = 1$.
2. For $j = 1, \ldots, n$ evolve v_{k-1}^j according to $Q(v_{k-1}^j, dv')$ to get v_k^j.
3. Calculate
$$W_k^n\, f = \frac{1}{n} \Sigma_{j=1}^n f(v_k^j) \qquad ,$$
4.
5. If $k < N$: Set $k \to k+1$, and go to Step 2.

We stop at the end of step N, and our approximation of $\Pi_N\, f$ is,

$$\Psi_N^n\, f = \frac{W_N^n\, f}{W_N^n\, 1}$$

Particle Filter

1. Sample n i.i.d. vectors (particles) $v_0^j,\ j = 1, \ldots, n$ from $Q_0(dv)$. Set $k = 1$.
2. For $j = 1, \ldots, n$ evolve v_{k-1}^j according to $Q(v_{k-1}^j, dv')$ to get v_k^j.
3. Calculate
$$W_k^n\, f = \frac{1}{n} \Sigma_{j=1}^n f(v_k^j)\, g(v_k^j, s_k),$$
4.
5. If $k < N$: Set $k \to k+1$, and go to Step 2.

We stop at the end of step N, and our approximation of $\Pi_N\, f$ is,

$$\Psi_N^n\, f = \frac{W_N^n\, f}{W_N^n\, 1}$$

Particle Filter

1. Sample n i.i.d. vectors (particles) $v_0^j,\ j = 1, \ldots, n$ from $Q_0(dv)$. Set $k = 1$.
2. For $j = 1, \ldots, n$ evolve v_{k-1}^j according to $Q(v_{k-1}^j, dv')$ to get v_k^j.
3. Calculate
$$W_k^n\, f = \frac{1}{n} \Sigma_{j=1}^n f(v_k^j)\, g(v_k^j, s_k),$$
4. Define the probability measure Ψ_k^n on $\mathbb{R}^d$ as
$$\Psi_k^n = \frac{1}{n\, W_k^n\, 1} \Sigma_{j=1}^n g(v_k^j, s_k) \delta(v_k^j).$$
5. If $k < N$: Set $k \to k+1$. Sample n i.i.d. variables $x_k^j,\ j = 1, \ldots, n$ from Ψ_k^n, and go to Step 2.

We stop at the end of step N, and our approximation of $\Pi_N\, f$ is,

$$\Psi_N^n\, f = \frac{W_N^n\, f}{W_N^n\, 1} = \mathbb{E}_{\Psi_N^n} f$$

Reminder of Goal

- Present a particle filter construction for a system that exhibits time-scale separation

$\surd$ Particle filter construction

$\leftrightsquigarrow$ time-scale separation

Combining the projective scheme for mulitscale filtering problem

- Underline model - Hidden variable,

$$dx_t = a(x_t, y_t)\, dt$$

$$dy_t = \frac{1}{\epsilon} b(x_t, y_t)\, dt + \frac{1}{\sqrt{\epsilon}} \sigma(x_t, y_t)\, dW_t$$

- Observations - Likelihood,

$S_t = h(x_t, y_t, \chi_t)$ with density/likelihood $g(x, y, s)$

Combining the projective scheme for mulitscale filtering problem

- We cannot naively apply the coarse projective scheme since it evolves only the slow/coarse variable while the observations are a function of the entire phase space.

Combining the projective scheme for mulitscale filtering problem

- We cannot naively apply the coarse projective scheme since it evolves only the slow/coarse variable while the observations are a function of the entire phase space.
- Instead we use the approximation for the original/higher dimensional space

$$(x(t), y(t)) \sim (X(t), \mu_{X(t)}(dy))$$

  - Once an observation arrives, we "lift" the X variable to many copies of (X, Y) by simulating for a very short time and *recording* the entire ensemble of M outputs.

$$\{X(t), y(t + k\delta t)\}\, k = 1, \ldots, M$$

Projective Particle filter

1. Sample n i.i.d. vectors X_0^j $j = 1, \ldots, n$ from $Q_0(dx, dy)$. Set $k = 1$.
2. For each $j = 1, \ldots, n$ evolve X_{k-1}^j until the next observation time using the projective scheme to generate samples X_k^j
3. When an observation arrives *lift* X_k^j to $\left\{ \left(X_k^j, Y_{k,m}^j \right) \right\}$:
   (a) Evolve the detailed model/full simulator M times with time step δt to generate $\left\{ \left(X_k^j, Y_{k,m}^j \right) \right\}, m = 1, \ldots, M$
4. For each $j = 1, \ldots, n$:
   (a) Calculate $W\left[f(X_k^j)\right] = \frac{1}{M} \sum_{m=1}^{M} f(X_k^j, Y_{k,m}^j)\, g(X_k^j, Y_{k,m}^j, s_k)$
   (b) Calculate $W_k^n\, f = \frac{1}{n} \sum_{j=1}^{n} W\left[f(X_k^j)\right]$
5. Define the probability measure Ψ_k^n on $\mathbb{R}^{d_x}$ as

$$\Psi_k^n = \frac{1}{n\, W_k^n\, 1} \Sigma_{j=1}^{n} W\left[1(X_k^j)\right] \delta(X_k^j)$$

6. If $k < N$: Set $k \to k+1$. Sample n i.i.d. variables $X_k^j,\ j = 1, \ldots, n$ from Ψ_k^n, and go to Step 2.

Summary

- Stochastic dynamical systems with scale separation
- Coarse projective scheme that enable to integrate the slow/coarse variable under the assumption of a black box (Equation Free)
- Wall-clock time reduction when using parallel architecture
- The use of projective schemes for particle filters

The end

- Thank you
- http://www.princeton.edu/ dgivon/