Moment-Based Variational Inference for Markov Jump Processes

Christian Wildner and Heinz Koeppl

Department of Electrical Engineering and Information Technology
Technische Universität Darmstadt, Germany
Introduction

**Model Class:** Markov jump process / continuous time Markov chain

- Applications in many domains (finance, social networks, healthcare, systems biology, etc.)
- Data-driven modelling requires latent state estimation
Introduction

**Model Class:** Markov jump process / continuous time Markov chain

- Applications in many domains (finance, social networks, healthcare, systems biology, etc.)
- Data-driven modelling requires latent state estimation

**Problem:** Hard/intractable for large state spaces
Introduction

Model Class: Markov jump process / continuous time Markov chain

- Applications in many domains (finance, social networks, healthcare, systems biology, etc.)
- Data-driven modelling requires latent state estimation

Problem: Hard/intractable for large state spaces

Proposed solution: new variational inference approach based on
- transition space partitioning
- gradient-based optimization
Markov Jump Processes

An MJP $X$ is fully defined by

- an initial distribution $\rho_0$
- a transition function $Q^X$ with
Markov Jump Processes

An MJP $X$ is fully defined by

- an initial distribution $\rho_0$
- a transition function $Q^X$ with

\[
Pr(X(t + h) = y \mid X(t) = x) = \delta(x, y) + Q^X(x, y, t)h + o(h)
\]

\[=: P_h\]
Markov Jump Processes

An MJP $X$ is fully defined by

- an initial distribution $\rho_0$
- a transition function $Q^X$ with

$$Pr(X(t + h) = y \mid X(t) = X) = \delta(x, y) + Q^X(x, y, t)h + o(h)$$

$$=: P_h$$

[Diagram showing discretized representation of hidden MJP]
**Exact Inference**

**Goal:** Compute posterior *path* distribution $P(X_{[0,T]} \mid Y_1, \ldots, Y_n)$

![Diagram](image)

discretized representation of hidden MJP
**Exact Inference**

**Goal:** Compute posterior path distribution $P(X_{[0,\tau]} \mid Y_1, \ldots, Y_n)$

Discretized representation of hidden MJP

Posterior paths are realized by smoothing process $\tilde{X}$ with modified transition function
Exact Inference

**Goal:** Compute posterior path distribution $P(X_{[0,T]} \mid Y_1, \ldots, Y_n)$

Discretized representation of hidden MJP

Posterior paths are realized by smoothing process $\tilde{X}$ with modified transition function

$$\tilde{Q}(x, y, t) = \frac{\sigma(y, t)}{\sigma(x, t)} Q^X(x, y)$$
Variational Inference

Minimize path level KL divergence $D_{KL}[P^Z||P^\tilde{x}]$

exact smoothing process
Variational Inference

Minimize path level KL divergence $D_{KL}[P^Z || P^\hat{x}]$

The usual decomposition applies

$$D_{KL}[P^Z || P^\hat{x}] = D_{KL}[P^Z || P^X] - \sum_{k=1}^{N} \mathbb{E}[\log p(y_k | Z(t_k))] + \log p(y_1, \ldots, y_n)$$

exact smoothing process

negative ELBO

variational to prior

marginal likelihood (evidence)
Variational Inference

Minimize path level KL divergence

\[ D_{KL}[P^Z \| P^{\tilde{X}}] \]

The usual decomposition applies

\[ D_{KL}[P^Z \| P^{\tilde{X}}] = D_{KL}[P^Z \| P^X] - \sum_{k=1}^{N} E[\log p(y_k \mid Z(t_k))] + \log p(y_1, \ldots, y_n) \]

Hard to construct suitable variational process class

exact smoothing process

negative ELBO

variational to prior

marginal likelihood (evidence)
Transition Space Partitioning

Smoothing process is in the class of controlled MJP with

\[ Q^Z(x, y, t) = \lambda(x, y, t)Q^X(x, y) \]

- time and state dependent control factor
- prior transition function
Transition Space Partitioning

Smoothing process is in the class of controlled MJP with

\[ Q^Z(x, y, t) = \lambda(x, y, t)Q^X(x, y) \]

\[ \quad \text{time and state dependent control factor} \quad \text{prior transition function} \]

Partition transitions into groups \( \Pi_i \) and set

\[ Q^Z(x, y, t) = \lambda_i(t)Q^X(x, y), \quad (x, y) \in \Pi_i \]
Transition Space Partitioning

Smoothing process is in the class of controlled MJP with

\[ Q^Z(x, y, t) = \lambda(x, y, t)Q^X(x, y) \]

time and state dependent prior transition
control factor function

Partition transitions into groups \( \Pi_i \) and set

\[ Q^Z(x, y, t) = \lambda_i(t)Q^X(x, y) , \quad (x, y) \in \Pi_i \]

Example: random walk
Example: random walk

1 \quad 2 \quad 3 \quad 4

q \quad p \quad q \quad p \quad p

q \quad q \quad q
Partitioning Example

**Example:** random walk

\[ Q^X = \begin{pmatrix} -p & p & 0 & 0 \\ q & -(p + q) & p & 0 \\ 0 & q & -(p + q) & p \\ 0 & 0 & q & -q \end{pmatrix} \]
**Partitioning Example**

**Example:** random walk

\[
Q^x = \begin{pmatrix}
-p & p & 0 & 0 \\
q & -(p + q) & p & 0 \\
0 & q & -(p + q) & p \\
0 & 0 & q & -q
\end{pmatrix}
\]

\[\lambda_1(t) : \text{common scaling for rightward transitions}\]

\[\lambda_2(t) : \text{common scaling for leftward transitions}\]
Partitioning Example

**Example**: random walk

$$Q^X = \begin{pmatrix} -p & p & 0 & 0 \\ q & -(p+q) & p & 0 \\ 0 & q & -(p+q) & p \\ 0 & 0 & q & -q \end{pmatrix}$$

$$\lambda_1(t)$$: common scaling for rightward transitions

$$\lambda_2(t)$$: common scaling for leftward transitions
Complexity Reduction

\[ D_{KL}[P^Z \mid \mid P^X] = \int_0^T \sum_x p^Z(x, t) \sum_{y \neq x} \left[ Q^X(x, y) \right. \]

\[ - Q^Z(x, y, t) - Q^Z(x, y, t) \log \left( \frac{Q^Z(x, y, t)}{Q^X(x, y)} \right) \left. \right] \, dt \]
Complexity Reduction

\[ D_{KL}[P^Z \mid P^X] = \int_0^T \sum_x p^Z(x, t) \sum_{y \neq x} [Q^X(x, y) - Q^Z(x, y, t) - Q^Z(x, y, t) \log \left( \frac{Q^Z(x, y, t)}{Q^X(x, y)} \right)] \, dt \]

transition space partitioning into \( r \) classes
**Complexity Reduction**

\[
D_{KL}[P^Z \| P^X] = \int_0^T \sum_x p^Z(x, t) \sum_{y \neq x} \left[ Q^X(x, y) - Q^Z(x, y, t) - Q^Z(x, y, t) \log \left( \frac{Q^Z(x, y, t)}{Q^X(x, y)} \right) \right] dt
\]

transition space partitioning into \( r \) classes

\[
D_{KL}[P^Z \| P^X] = \sum_{i=1}^r \int_0^T \varphi_i(t) (1 - \lambda_i(t) + \lambda_i(t) \log \lambda_i(t)) dt
\]
**Complexity Reduction**

\[
D_{KL}[P^Z \parallel P^X] = \int_0^T \sum_x p^Z(x, t) \sum_{y \neq x} \left[ Q^X(x, y) - Q^Z(x, y, t) - Q^Z(x, y, t) \log \left( \frac{Q^Z(x, y, t)}{Q^X(x, y)} \right) \right] dt
\]

transition space partitioning into r classes

\[
D_{KL}[P^Z \parallel P^X] = \sum_{i=1}^r \int_0^T \varphi_i(t) (1 - \lambda_i(t) + \lambda_i(t) \log \lambda_i(t)) dt
\]

expected summary statistic
Control Problem

Using the Markov property, derive moment equations for $\varphi$. 
Control Problem

Using the Markov property, derive moment equations for $\varphi$

Obtain non-linear, deterministic optimal control problem

\[
\begin{align*}
\text{minimize} & \quad L[\lambda, \varphi] - F[\varphi] \\
\text{subject to} & \quad \frac{d}{dt} \varphi(t) = f(\lambda(t), \varphi(t))
\end{align*}
\]
Control Problem

Using the Markov property, derive moment equations for $\varphi$

Obtain non-linear, deterministic optimal control problem

\[
\begin{align*}
\text{minimize} & \quad L[\lambda, \varphi] - F[\varphi] \\
\text{subject to} & \quad \frac{d}{dt} \varphi(t) = f(\lambda(t), \varphi(t))
\end{align*}
\]

Solve via natural gradient descent in the controls $\lambda(t)$
Example Application

Scenario:
Stochastic gene expression
Studied by fluorescence microscopy

Goals:
Parameter estimation
Model comparison
Optimal experiment design