ODE Filtering

A Gaussian Decision Agent for Forward Problems

Hans Kersting

Alan Turing Institute

12 April 2018

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Max Planck Institute for Intelligent Systems

erc

some of the presented work is supported by the European Research Council.

$$\dot{x}(t) = f(x(t)),\ t \in [0, T] \qquad x(0) = x_0 \in \mathbb{R}^d \tag{1}$$

We model all unkown (even if deterministic) objects, i.e.

- solution $x \in C^1([0, T]; \mathbb{R}^d)$,
- vector field $f \in C^0([0, T]; \mathbb{R}^d)$

by random variables or stochastic processes (**prior information**), and define which information we obtain in the course of the numerical computation of the solution (**measurement model**).

Prior information + measurement model $\longrightarrow$ application of Bayes rule

Choice of Prior information

Prior information on x

For prior information on f see our publications.

We a priori model x and $\dot{x}$ with an arbitrary Gauss-Markov process, i.e. with linear SDE

$$dX_t = FX_t dt + L dB_t,$$

with Gaussian initial condition $X_0 \sim \mathcal{N}(m_0, P_0)$. For an Integrated Brownian motion (Wiener process)

$$\begin{pmatrix} x(t) \\ \dot{x}(t) \end{pmatrix} \sim \begin{pmatrix} \mathrm{d}X_t \\ \mathrm{d}\dot{X}_t \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} X_t \\ \dot{X}_t \end{pmatrix} \mathrm{d}t + \begin{pmatrix} 0 \\ \sigma \end{pmatrix} \mathrm{d}B_t, \tag{2}$$

the ODE filter coincides with Runge-Kutta and Nordsieck methods in a certain sense [SSH18]. An Ornstein Uhlenbeck prior

$$\begin{pmatrix} x(t) \\ \dot{x}(t) \end{pmatrix} \sim \begin{pmatrix} \mathrm{d}X_t \\ \mathrm{d}\dot{X}_t \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -\theta \end{pmatrix} \begin{pmatrix} X_t \\ \dot{X}_t \end{pmatrix} \mathrm{d}t + \begin{pmatrix} 0 \\ \sigma \end{pmatrix} \mathrm{d}B_t, \tag{3}$$

has also been studied [MKSH17].

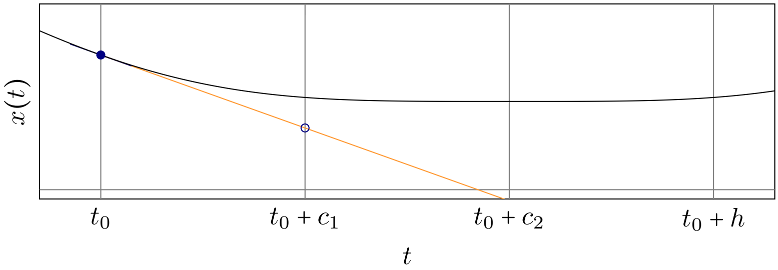
Numerical solutions of IVPs

plots: Runge-Kutta of order 3

How classical solvers extrapolate forward from time t_0 to $t_0 + h$:

- Estimate $\dot{x}(t_i)$, $t_0 \leq t_1 \leq \cdots \leq t_n \leq t_0 + h$ by evaluating $y_i \approx f(\hat{x}(t_i))$, where $\hat{x}(t)$ is itself an estimate for $x(t)$
- Use this data $y_i := \dot{x}(t_i)$ to estimate $x(t_0 + h)$, i.e.

$$\hat{x}(t_0 + h) \approx x(t_0) + h \sum_{i=1}^{b} w_i y_i.$$



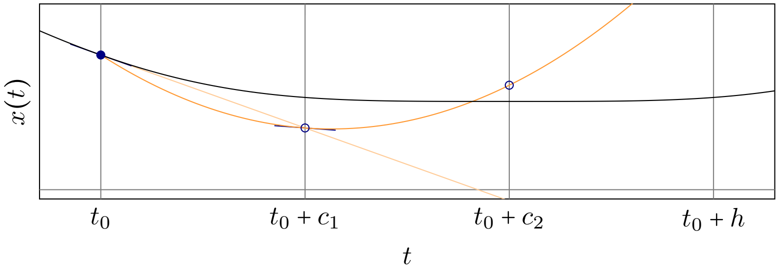
Numerical solutions of IVPs

plots: Runge-Kutta of order 3

How classical solvers extrapolate forward from time t_0 to $t_0 + h$:

- Estimate $\dot{x}(t_i)$, $t_0 \leq t_1 \leq \cdots \leq t_n \leq t_0 + h$ by evaluating $y_i \approx f(\hat{x}(t_i))$, where $\hat{x}(t)$ is itself an estimate for $x(t)$
- Use this data $y_i := \dot{x}(t_i)$ to estimate $x(t_0 + h)$, i.e.

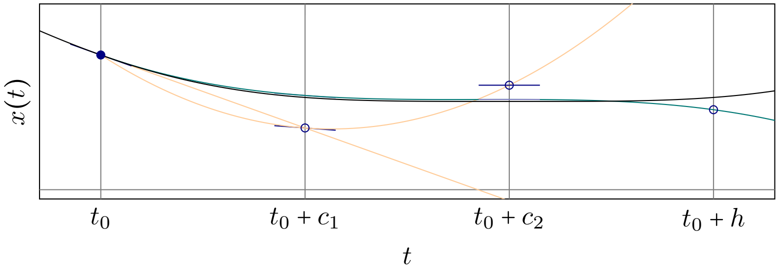
$$\hat{x}(t_0 + h) \approx x(t_0) + h \sum_{i=1}^{b} w_i y_i.$$



How classical solvers extrapolate forward from time t_0 to $t_0 + h$:

- Estimate $\dot{x}(t_i)$, $t_0 \leq t_1 \leq \cdots \leq t_n \leq t_0 + h$ by evaluating $y_i \approx f(\hat{x}(t_i))$, where $\hat{x}(t)$ is itself an estimate for $x(t)$
- Use this data $y_i := \dot{x}(t_i)$ to estimate $x(t_0 + h)$, i.e.

$$\hat{x}(t_0 + h) \approx x(t_0) + h \sum_{i=1}^{b} w_i y_i.$$



Numerical solutions of IVPs

plots: Runge-Kutta of order 3

How classical solvers extrapolate forward from time t_0 to $t_0 + h$:

- Estimate $\dot{x}(t_i)$, $t_0 \leq t_1 \leq \cdots \leq t_n \leq t_0 + h$ by evaluating $y_i \approx f(\hat{x}(t_i))$, where $\hat{x}(t)$ is itself an estimate for $x(t)$
- Use this data $y_i := \dot{x}(t_i)$ to estimate $x(t_0 + h)$, i.e.

$$\hat{x}(t_0 + h) \approx x(t_0) + h \sum_{i=1}^{b} w_i y_i.$$

Information in these calculations:

$$\dot{x}(t) = f(x(t)) \approx f(\hat{x}(t)) \tag{4}$$

For information, f is evaluated at (or around) the current numerical estimate $\hat{x}$ of x.

In principle, given a Gaussian believe

$$\begin{pmatrix} x(t) \\ \dot{x}(t) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m(t) \\ \dot{m}(t) \end{pmatrix}, \begin{pmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{pmatrix} \right), \tag{5}$$

the 'true' information on $\dot{x}(t)$ would be the pushforward measure $f_* \mathcal{N}(\dot{m}(t), P_{00})$. For computational speed, we want a Gaussian, with matched moments

$$y = \int f(\xi) \, \mathrm{d}\mathcal{N}(\xi; m(t), P_{00}), \tag{6}$$

and covariance

$$R = \int f(\xi) f^T(\xi) \, \mathrm{dd}\mathcal{N}(\xi; m(t), P_{00}). \tag{7}$$

Suitable ways to approximate these integrals have been studied in [KH16]. For maximum speed, we can just use $y = f(m(t))$ and $R = 0$ as proposed in [SSH18].

This yields a (Kalman) filtering algorithm for ODEs.

We interpret $(x, \dot{x}, x^{(2)}, \dots, x^{(q-1)})$ as a draw from a q-times-integrated Wiener process $(X_t)_{t\in[0,T]} = (X_t^{(1)}, \dots, X_t^{(q)})^T_{t\in[0,T]}$ given by a linear time-invariant SDE:

$$
\begin{aligned}
dX_t &= FX_t dt + Q dW_t, \\
X_0 &= \xi, \quad \xi \sim \mathcal{N}(m(0), P(0)).
\end{aligned}
$$

Filtering–based probabilistic ODE solvers

Gaussian filtering

We interpret $(x, \dot{x}, x^{(2)}, \ldots, x^{(q-1)})$ as a draw from a q-times-integrated Wiener process $(X_t)_{t \in [0,T]} = (X_t^{(1)}, \ldots, X_t^{(q)})^T_{t \in [0,T]}$ given by a linear time-invariant SDE:

$$
\begin{aligned}
dX_t &= FX_t dt + QdW_t, \\
X_0 &= \xi, \quad \xi \sim \mathcal{N}(m(0), P(0)).
\end{aligned}
$$

Calculation of Posterior by Gaussian filtering

We interpret $(x, \dot{x}, x^{(2)}, \dots, x^{(q-1)})$ as a draw from a q-times-integrated Wiener process $(X_t)_{t\in[0,T]} = (X_t^{(1)}, \dots, X_t^{(q)})^T_{t\in[0,T]}$ given by a linear time-invariant SDE:

$$
\begin{aligned}
dX_t &= FX_t dt + QdW_t, \\
X_0 &= \xi, \quad \xi \sim \mathcal{N}(m(0), P(0)).
\end{aligned}
$$

Calculation of Posterior by Gaussian filtering

Prediction step:

$$
\begin{aligned}
m^-_{t+h} &= A(h)m_t, \\
P^-_{t+h} &= A(h)P_tA(h)^T + Q(h),
\end{aligned}
$$

We interpret $(x, \dot{x}, x^{(2)}, \dots, x^{(q-1)})$ as a draw from a q-times-integrated Wiener process $(X_t)_{t\in[0,T]} = (X_t^{(1)}, \dots, X_t^{(q)})^T_{t\in[0,T]}$ given by a linear time-invariant SDE:

$$\begin{aligned} dX_t &= FX_t dt + QdW_t, \\ X_0 &= \xi, \quad \xi \sim \mathcal{N}(m(0), P(0)). \end{aligned}$$

Calculation of Posterior by Gaussian filtering

Prediction step:

$$\begin{aligned} m^-_{t+h} &= A(h)m_t, \\ P^-_{t+h} &= A(h)P_tA(h)^T + Q(h), \end{aligned}$$

Gradient prediction at $t+h$: Approximate

$$\begin{aligned} y &\approx \int f(\xi)\, \mathrm{d}\mathcal{N}\left(\xi; m(t), P_{00}\right), \\ R &\approx \int f(\xi) f^T(\xi)\, \mathrm{d}\mathcal{N}\left(\xi; m(t), P_{00}\right) \end{aligned}$$

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

[SDH14]

We interpret $(x, \dot{x}, x^{(2)}, \dots, x^{(q-1)})$ as a draw from a **q**-times-integrated Wiener process $(X_t)_{t\in[0,T]} = (X_t^{(1)}, \dots, X_t^{(q)})^T_{t\in[0,T]}$ given by a linear time-invariant SDE:

$$
\begin{aligned}
dX_t &= FX_t dt + QdW_t, \\
X_0 &= \xi, \quad \xi \sim \mathcal{N}(m(0), P(0)).
\end{aligned}
$$

Calculation of Posterior by Gaussian filtering

Prediction step:

$$
\begin{aligned}
m^-_{t+h} &= A(h)m_t, \\
P^-_{t+h} &= A(h)P_tA(h)^T + Q(h),
\end{aligned}
$$

Gradient prediction at $t + h$: Approximate

$$
\begin{aligned}
y &\approx \int f(\xi)\,\mathrm{d}\mathcal{N}\left(\xi; m(t), P_{00}\right), \\
R &\approx \int f(\xi)f^T(\xi)\,\mathrm{d}\mathcal{N}\left(\xi; m(t), P_{00}\right)
\end{aligned}
$$

Update step:

$$
\begin{aligned}
z &= y - e_n^T m^-_{t+h}, \\
S &= e_n^T P^-_{t+h} e_n + R, \\
K &= P^-_{t+h} e_n S^{-1}, \\
m_{t+h} &= m^-_{t+h} + Kz, \\
P_{t+h} &= P^-_{t+h} - K e_n^T P^-_{t+h},
\end{aligned}
$$

Research Questions

1. worst-case convergence rates vs. average convergence rates (over a measure on $\boldsymbol{f}$),
2. trade-off between computational speed (with Gaussians) and statistical accuracy (with samples),
3. properties of different priors on $\boldsymbol{x}$,
4. in which sense are 'Bayesian' algorithms (like the above) approximations of Bayesian algorithms in the sense of [COSG17],
5. can PN algorithms for ODEs be extended to SDEs?,
6. Bayesian inverse problems—inner loop vs outer loop trade-off like in Bayesian optimization?
7. different filters (particle filter, ensemble Kalman filter)?

Thank you for listening!

Bibliography

- J. Cockayne, C.J. Oates, T. Sullivan, and M.A. Girolami.
  Bayesian probabilistic numerical methods.
  arXiv:1702.03673 [stat.ME], February 2017.
- Hans Kersting and P. Hennig.
  Active Uncertainty Calibration in Bayesian ODE Solvers.
  Uncertainty in Artificial Intelligence (UAI), 2016.
- E. Magnani, H. Kersting, M. Schober, and P. Hennig.
  Bayesian Filtering for ODEs with Bounded Derivatives.
  arXiv:1709.08471 [cs.NA], September 2017.
- M. Schober, D. Duvenaud, and P. Hennig.
  Probabilistic ODE Solvers with Runge–Kutta Means.
  Advances in Neural Information Processing Systems (NIPS), 2014.
- Michael Schober, Simo Särkkä, and Philipp Hennig.
  A probabilistic model for the numerical solution of initial value problems.
  Statistics and Computing, January 2018.