Machine learning under physical constraints
Introduction to DAN

Sixin Zhang
(sixin.zhang@toulouse-inp.fr)

Outline

From Bayesian DA to Machine learning

Introduction to Data Assimilation Networks (DAN)

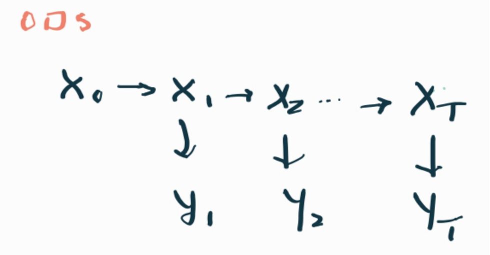
Outline

Bayesian Data Assimilation (DA)

- Assume $x_t \in \mathbb{X} = \mathbb{R}^n$, $y_t \in \mathbb{Y} = \mathbb{R}^d$
- Observed Dynamical Systems (ODS)

$$
\begin{aligned}
x_t &= Mx_{t-1} + \eta_t \\
y_t &= Hx_t + \epsilon_t
\end{aligned}
$$

- M: dynamics, H: observation process, η_t and ϵ_t: noise



Bayesian Data Assimilation (DA)

- ODS induces a joint probability density on $\mathbb{X}^{T+1} \times \mathbb{Y}^{T}$

$$p(x_0, x_1, \cdots, x_T, y_1, \cdots, y_T)$$

- Dynamical process is represented by $p(x_t | x_{t-1})$
- Observation process is represented by $p(y_t | x_t)$
- Problem: For each $t \leq T$, obtain conditional density

$$p(x_t | y_1, \cdots, y_t)$$

Bayesian Data Assimilation

- Compute $p(x_t|y_1, \cdots, y_t)$ recursively.
- Analysis by Bayes rule : $p(x|y) = \frac{p(y|x)p(x)}{\int p(y|x)p(x)dx}$
- Let $Y_t = (y_1, \cdots, y_t)$, analyze conditional densities:

$$p_t^{\mathbf{a}}(x_t|y_t) := p(x_t|y_t, Y_{t-1}),$$
$$p_t^{\mathbf{b}}(x_t) := p(x_t|Y_{t-1})$$

- Analyse step: transform $p_t^{\mathbf{b}}$ to $p_t^{\mathbf{a}}$ by Markov property and Bayes rule (time invariance: p does not change with t).

$$p_t^{\mathbf{a}}(x_t|y_t) = \frac{p(y_t|x_t)p_t^{\mathbf{b}}(x_t)}{\int p(y_t|x)p_t^{\mathbf{b}}(x)dx}$$

Bayesian Data Assimilation

- Propagate conditional densities:

$$p_t^{\mathbf{a}}(x_t|y_t) := p(x_t|Y_{t-1}, y_t),$$
$$p_{t+1}^{\mathbf{b}}(x_{t+1}) := p(x_{t+1}|Y_t)$$

- Propagation step: transform $p_t^{\mathbf{a}}$ to $p_{t+1}^{\mathbf{b}}$ by Markov property.

$$p_{t+1}^{\mathbf{b}}(x_{t+1}) = \int p(x_{t+1}|x_t) p_t^{\mathbf{a}}(x_t|y_t) dx_t$$

  (Again time invariance: p does not change with t)

- Example: Kalman Filter

Why Machine learning?

- Observed Dynamical Systems (ODS)

$$x_t = Mx_{t-1} + \eta_t$$
$$y_t = Hx_t + \epsilon_t$$

- When M or H are not linear, KF is not optimal. Can we improve DA using Machine learning?

- Problem reformulation: given sequences of ODS, can we approximate $p_t^{\mathbf{a}}$ and $p_t^{\mathbf{b}}$ for $t \leq T$?

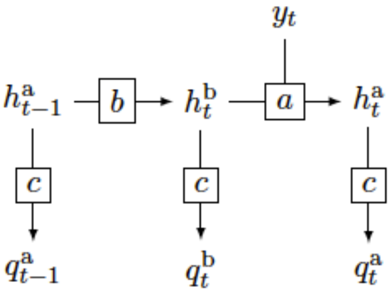
Outline

DAN framework

- Supervised learning of $p_t^{\mathbf{a}}$ and $p_t^{\mathbf{b}}$ from sequences of $\{(x_t, y_t)\}_{t \leq T}$.
- Unsupervised learning: from sequences of $\{y_t\}_{t \leq T}$.

DAN framework

- Supervised learning from sequences of $\{(x_t, y_t)\}_{t \leq T}$
- Approximate $p_t^{\mathbf{a}}$ by $q_t^{\mathbf{a}} : \mathbb{Y}^t \to \mathrm{Prob}(\mathbb{X})$, and $p_t^{\mathbf{b}}$ by $q_t^{\mathbf{b}} : \mathbb{Y}^{t-1} \to \mathrm{Prob}(\mathbb{X})$.
- Impose **Markov structures** on $q_t^{\mathbf{a}}$ and $q_t^{\mathbf{b}}$ using memory $(h_t^{\mathbf{a}}, h_t^{\mathbf{b}})$.



DAN framework: analyzer $\mathbf{a}$

- $\mathbb{Y}$: space of observation, $\mathbb{X}$: space of true state
- $\mathbb{H}$: space of memory (hidden state)
- Analyzer $\mathbf{a} \in \mathbb{H} \times \mathbb{Y} \to \mathbb{H}$

$$h_t^{\mathbf{a}} = \mathbf{a}(h_t^{\mathbf{b}}, y_t)$$

- Example of Kalman Filter: Update $h_t^{\mathbf{b}} := (\mu_t^{\mathbf{b}}, \Sigma_t^{\mathbf{b}})$ by y_t

DAN framework: propagator $\mathbf{b}$

- Propagator $\mathbf{b} \in \mathbb{H} \to \mathbb{H}$

$$h_{t+1}^{b} = \mathbf{b}(h_t^{a})$$

- Recursion of memory from t to $t+1$,

$$h_{t+1}^{\mathbf{a}} = \mathbf{a}(h_{t+1}^{\mathbf{b}}, y_{t+1}) = \mathbf{a}(\mathbf{b}(h_t^{a}), y_{t+1})$$

- Example of Kalman Filter: Update $h_t^{\mathbf{a}} := (\mu_t^{\mathbf{a}}, \Sigma_t^{\mathbf{a}})$.

DAN framework: procoder $\mathbf{c}$

- Procoder $\mathbf{c} \in \mathbb{H} \to \mathrm{Prob}(\mathbb{X})$

$$q_t^{\mathbf{a}} = \mathbf{c}(h_t^{\mathbf{a}}), \quad q_t^{\mathbf{b}} = \mathbf{c}(h_t^{\mathbf{b}})$$

- $q_t^{\mathbf{b}} = \mathbf{c}(h_t^{\mathbf{b}})$ approximates $p(x_t | Y_{t-1})$.
- $q_t^{\mathbf{a}} = \mathbf{c}(h_t^{\mathbf{a}})$ approximates $p(x_t | Y_t)$.
- Example of Kalman Filter: $h := (\mu, \Sigma) \in \mathbb{H}$, $\mathbf{c}(h) := \mathcal{N}(\mu, \Sigma)$.
- Role of memory in Ensemble Kalman Filter (ETKF).

DAN as Elman RNN

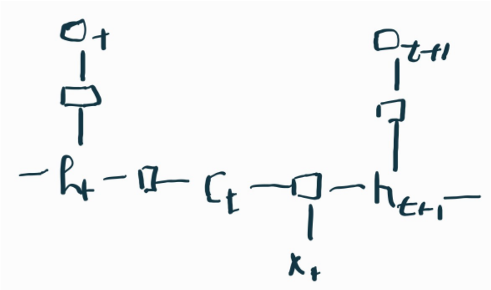


Figure: Unroll Elman RNN over time

- Relation with Elman RNN and DAN
  - Hidden h_t as $h_t^{\mathbf{a}}$: estimation of x_t given Y_t.
  - Context c_t as $h_t^{\mathbf{b}}$: prediction of x_t given Y_{t-1}.
  - Input x_t as y_t: observed state at t in ODS

DAN framework: objective function

- Maximum-likelihood estimation of $p_t^{\mathbf{a}}$ by $q_t^{\mathbf{a}}$: densities of x_t conditioned on Y_t.

- Introduce a series of objectives of $L_t(q_t^a)$

$$
\begin{aligned}
L_t(q_t^{\mathbf{a}}) &= -\int \log q_t^{\mathbf{a}}(x_t|y_t) p(x_t, Y_t) dx_t dY_t \\
&= -\int \log q_t^{\mathbf{a}}(x_t|y_t) p_t^{\mathbf{a}}(x_t|y_t) p(Y_t) dx_t dY_t
\end{aligned}
$$

- The global minimizer of L_t is $q_t^{\mathbf{a}}(\cdot|y_t) = p_t^{\mathbf{a}}(\cdot|y_t)$, p-a.s,

$$
\int \log \frac{p_t^{\mathbf{a}}(x_t|y_t)}{q_t^{\mathbf{a}}(x_t|y_t)} p_t^{\mathbf{a}}(x_t|y_t) dx_t \geq 0
$$

DAN framework: objective function

- Maximum-likelihood estimation of $p_t^{\mathbf{b}}$ by $q_t^{\mathbf{b}}$: densities of x_t conditioned on Y_{t-1}.

- Introduce a sequence of objectives for $t \leq T$,

$$\begin{aligned} L_t(q_t^{\mathbf{b}}) &= -\int \log q_t^{\mathbf{b}}(x_t) p(x_t, Y_{t-1}) dx_t dY_{t-1} \\ &= -\int \log q_t^{\mathbf{b}}(x_t) p_t^{\mathbf{b}}(x_t) p(Y_{t-1}) dx_t dY_{t-1} \end{aligned}$$

- The global minimizer of L_t is $q_t^{\mathbf{b}}(\cdot) = p_t^{\mathbf{b}}(\cdot)$, p-a.s,

$$\int \log \frac{p_t^{\mathbf{b}}(x_t)}{q_t^{\mathbf{b}}(x_t)} p_t^{\mathbf{b}}(x_t) dx_t \geq 0$$

DAN framework: objective function

- **Objective function**:

$$\min_{q_0^{\mathbf{a}},(q_t^{\mathbf{a}},q_t^{\mathbf{b}})_{t=1}^T} \frac{1}{T}\sum_{t\leq T}(L_t(q_t^{\mathbf{a}})+L_t(q_t^{\mathbf{b}}))+L_0(q_0^{\mathbf{a}})$$

- The initial density $q_0^{\mathbf{a}}(x_0)$ aims to approximate $p(x_0)$.

- Equivalently to optimize $(\mathbf{a},\mathbf{b},\mathbf{c})$ in a Recurrent Neural Network (RNN).

DAN framework: summary

- Bayesian Data Assimilation defines a sequence of conditional probability densities to learn.
- Supervised learning of ODS with DAN by respecting Markov structures.
- Define optimal objective functions from the maximum likelihood principle.