Particle Flow Bayes' Rule

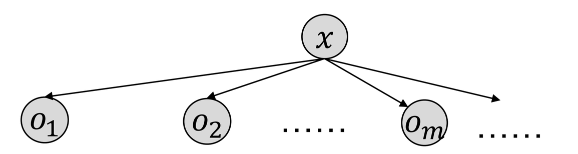
Xinshi Chen[1*], Hanjun Dai[2*], Le Song[2,3]

{[1]School of Mathematics, [2]College of Computing} at Georgia Institute of Technology, [3]Ant Financial Services Group. (* Equal contribution)

Bayes' Rule

Given

1. **Prior** distribution $\pi(\boldsymbol{x})$
2. **Likelihood** function $p(\boldsymbol{o}|\boldsymbol{x})$
3. **Observations** $\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_m$



The **posterior distribution** of unknown parameter $\boldsymbol{x}$ can by computed by Bayes' Rule:

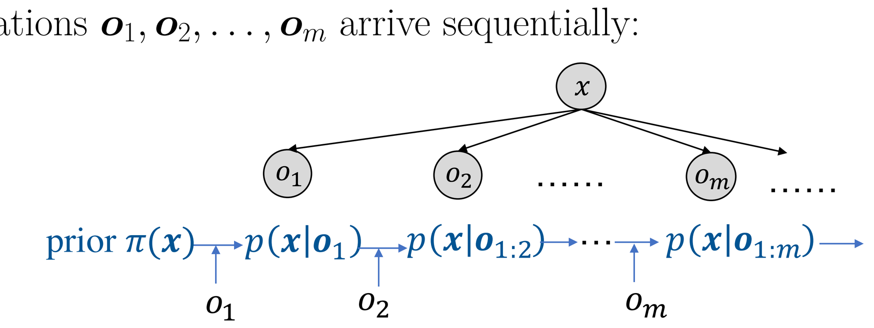
$$p(x|o_{1:m}) = \frac{1}{z}\pi(x)\prod_{i=1}^{m} p(o_i|x)$$

$$z = \int \pi(x)\prod_{i=1}^{m} p(o_i|x)\,dx$$

Challenging computational problem for high dimensional $\boldsymbol{x}$

Sequential Bayesian Inference

Observations $\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_m$ arrive sequentially:



An ideal algorithm should:

- ***Efficiently update*** $p(\boldsymbol{x}|\boldsymbol{o}_{1:m})$ to $p(\boldsymbol{x}|\boldsymbol{o}_{1:m+1})$ when $\boldsymbol{o}_{m+1}$ is observed
- ***Without storing*** all historical observations $\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_m$

$$\underbrace{p(x|o_{1:m})}_{\text{updated posterior}} \propto \underbrace{p(x|o_{1:m-1})}_{\text{current posterior}}\underbrace{p(o_m|x)}_{\text{likelihood}}$$

Related Works

MCMC

- requires a complete scan of the data

Variational Inference (VI)

- requires re-optimization for every new observation

Stochastic approximate inference

- are prescribed algorithms to optimize the final posterior $p(\boldsymbol{x}|\boldsymbol{o}_{1:M})$
- can not exploit the structure of the sequential inference problem

Sequential monte Carlo

- state of art for online Bayesian Inference
- but suffers from path degeneracy problem in high dimensions
- rejuvenation steps can help but will violate online constraints

An Operator View: Kernel Bayes' Rule

- the posterior is represented as an embedding $\mu_m = \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{o}_{1:m})}\phi(\boldsymbol{x})$
- $\underbrace{\mu_{m+1}}_{\text{updated embedding}} = \mathcal{K}(\underbrace{\mu_m}_{\text{current embedding}}, o_{m+1})$
- views the Bayes update as an operatior in RKHS

Our Method

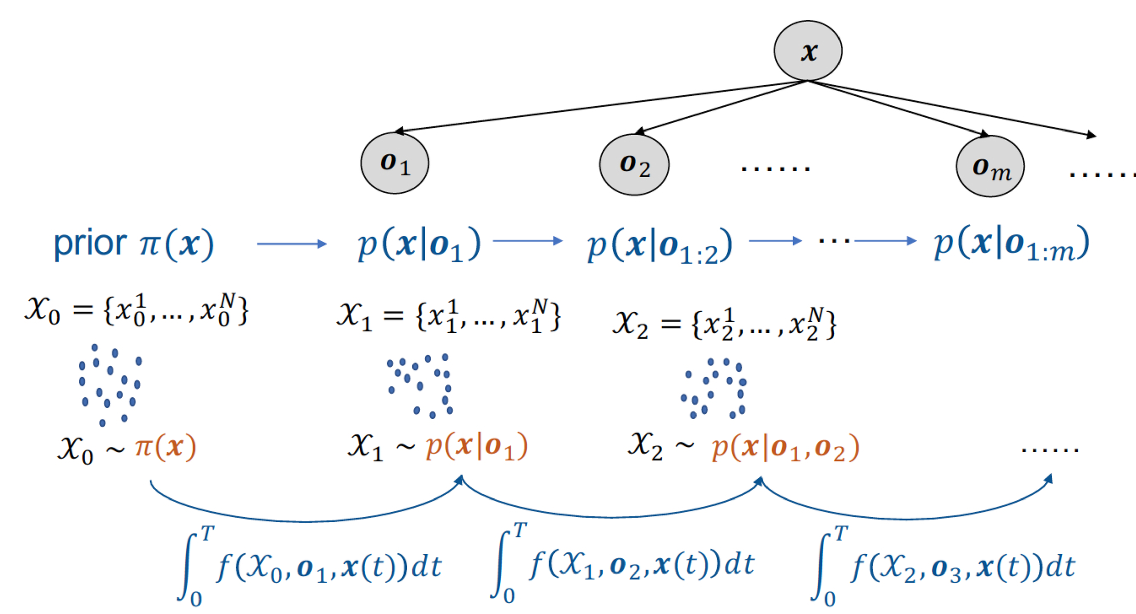
Start with N particles

$$\mathcal{X}_0 = \left\{x_0^1, x_0^2, \ldots, x_0^N\right\}, \text{ sampled i.i.d. from prior } \pi(\boldsymbol{x})$$

Transport particles to next posterior as the solution of ODEs

$$\begin{cases} \frac{d\boldsymbol{x}}{dt} = f(\mathcal{X}_0, \boldsymbol{o}_1, \boldsymbol{x}(t), t), \ \forall t \in (0, T] \\ \boldsymbol{x}(0) = \boldsymbol{x}_0^n \end{cases} \overset{\text{gives}}{\Longrightarrow} \boldsymbol{x}_1^n = \boldsymbol{x}(T)$$

Particle Flow Bayes' Rule



Particle Flow as a Bayesian Operator

$$\boldsymbol{x}_{m+1}^n = \mathcal{F}(\mathcal{X}_m, o_{m+1}, \boldsymbol{x}_m^n) := \boldsymbol{x}_m^n + \int_0^T f(\mathcal{X}_m, o_{m+1}, \boldsymbol{x}(t), t)\,dt.$$

$$\log q_{m+1}(\boldsymbol{x}_{m+1}^n) = \log q_m(\boldsymbol{x}_m^n) - \int_0^T \nabla_x \cdot f\,dt.$$

Advantages: Flow Property

There are mainly two obvious advantages of Particle Flow:

1. First, the ***location*** of the particles can be ***moved*** according to posterior distribution.
2. Second, the probability density can be computed efficiently because the ***change of log-density also follows a ODE***.

- **Continuity Equation** express the law of *local conservation of mass*: (1) Mass can neither be created nor destroyed; (2) nor can it 'teleport' from one place to another.

$$\frac{\partial q(\boldsymbol{x}, t)}{\partial t} = -\nabla_x \cdot (qf)$$

- **Theorem.** If $\frac{d\boldsymbol{x}}{dt} = f$, then the change in log-density follows

$$\frac{d\log q(\boldsymbol{x}, t)}{dt} = -\nabla_x \cdot f.$$

Does A Unified Flow Velocity f exist?

$$x(0) \sim \pi(x) \qquad x(t) \sim p(x|o_1)$$

$$x(T) = x(0) + \int_0^T f(inputs)dt$$

Does a ***unified flow velocity*** f exist for different Bayesian inference tasks involving different priors and different observations?

Existence of Flow-based Bayes' Rule

(1) Langevin dynamics is a *stochastic* processs

$$d\boldsymbol{x}(t) = \nabla_x \log \pi(\boldsymbol{x})p(\boldsymbol{o}|\boldsymbol{x})\,dt + \sqrt{2}\,d\boldsymbol{w}(t),$$

where $d\boldsymbol{w}(t)$ is a standard Brownian motion.

- The probability density $q(\boldsymbol{x}, t)$ of $\boldsymbol{x}(t)$ converges to a stationary distribution, which is the posterior $p(\boldsymbol{x}|\boldsymbol{o})$.

(2) Stochastic Flow to Deterministic Flow.

- The probability density $q(\boldsymbol{x}, t)$ of Langevin dynamcis follows a ***deterministic*** evolution according to **Fokker-Planck equation**

$$\frac{\partial q}{\partial t} = -\nabla_x \cdot (q\nabla_x \log \pi(\boldsymbol{x})p(\boldsymbol{o}|\boldsymbol{x})) + \nabla_x q(\boldsymbol{x}, t).$$

- Fokker-Planck equation can be rewritten in the form of **Continuity Equation**:

$$\frac{\partial q}{\partial t} = -\nabla_x \cdot (qf),$$

where $f = \nabla_x \log \pi(\boldsymbol{x})p(\boldsymbol{o}|\boldsymbol{x}) - \nabla_x \log q(\boldsymbol{x}, t)$.
$\Longrightarrow$ **deterministic flow!**

(3) Closed-Loop to Open-Loop: The above deterministic flow is closed-loop, which depends on flow state $q(\boldsymbol{x}, t)$. We use optimal control theory to show there exists a unified f which is independent of $q(\boldsymbol{x}, t)$.

Conclusion of a unified f. There exists a fixed and deterministic flow velocity f of the form

$$\nabla_x \log p(\boldsymbol{x}|\boldsymbol{o}_{1:m})p(\boldsymbol{o}_{m+1}|\boldsymbol{x}) - w^*(p(\boldsymbol{x}|\boldsymbol{o}_{1:m}), t),$$

which can transform $p(\boldsymbol{x}|\boldsymbol{o}_{1:m})$ to $p(\boldsymbol{x}|\boldsymbol{o}_{1:m+1})$ and in turns define a unified particle flow Bayes operator $\mathcal{F}$.

Parameterization

$$f(p(\boldsymbol{x}|\boldsymbol{o}_{1:m}), p(\boldsymbol{o}_{m+1}|\boldsymbol{x}), \boldsymbol{x}(t), t) \Rightarrow f(\mathcal{X}_m, \boldsymbol{o}_{m+1}, \boldsymbol{x}(t), t)$$

- $p(\boldsymbol{x}|\boldsymbol{o}_{1:m}) \Rightarrow \mathcal{X}_m$
  Use samples $\mathcal{X}_m$ as surrogates, feature space embedding.
- $p(\boldsymbol{o}_{m+1}|\boldsymbol{x}) \Rightarrow (\boldsymbol{o}_{m+1}, \boldsymbol{x}(t))$

Overall we parameterize the flow velocity as

$$f = \boldsymbol{h}\left(\tfrac{1}{N}\textstyle\sum_{n=1}^N \phi(\boldsymbol{x}_m^n), o_{m+1}, \boldsymbol{x}(t), t\right),$$

where $\boldsymbol{h}$ and ϕ are neural networks. Let $\theta \in \Theta$ be their parameters which are independent of t.

Learning Algorithm

Multi-task Framework

- The training set $\mathcal{D}_{\text{train}}$ contains **multiple inference tasks**
- Each task $\mathcal{T} \in \mathcal{D}_{train}$ is a tuple

$$\mathcal{T} := (\underbrace{\pi(\boldsymbol{x})}_{\text{prior}}, \underbrace{p(\cdot|\boldsymbol{x})}_{\text{likelihood}}, \underbrace{\{o_1, \ldots, o_M\}}_{M \text{ observations}})$$

Loss Function

- The loss for each $\mathcal{T}$ is $\sum_{m=1}^M \mathrm{KL}(q_m(\boldsymbol{x})||p(\boldsymbol{x}, \boldsymbol{o}_{1:m}))$, where $q_m(\boldsymbol{x})$ is the distribution transported by $\mathcal{F}$ at m-th stage.
- Equivalent to minimize negative evidence lower bound (ELBO)

$$\mathcal{L}(\mathcal{T}) = \sum_{m=1}^M \sum_{n=1}^N (\log q_m(\boldsymbol{x}_m^n) - \log p(\boldsymbol{x}_m^n, \boldsymbol{o}_{1:m})).$$

- Cumulative loss: $\mathcal{L}(\mathcal{D}_{\text{train}}) = \sum_{\mathcal{T} \in \mathcal{D}_{\text{train}}} \mathcal{L}(\mathcal{T})$.

Experiment 1: Benefits for High Dimension

Multivariate Guassian Model

- prior $\boldsymbol{x} \sim \mathcal{N}(\mu_x, \Sigma_x)$
- observation conditioned on prior $\boldsymbol{o}|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{x}, \Sigma_o)$

Experiment Setting

- Training set only contains *sequences of 10 observations*.
- Testing set contains 25 difference *sequences of 100 observations*.

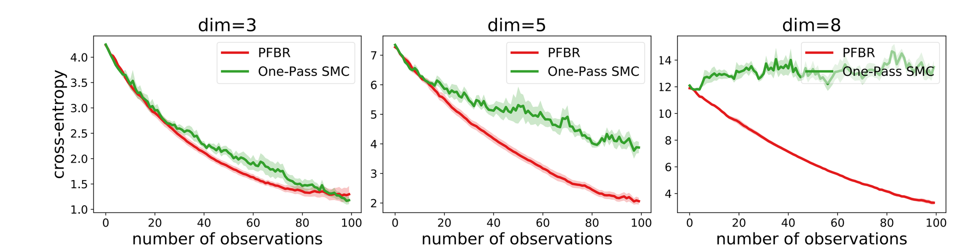


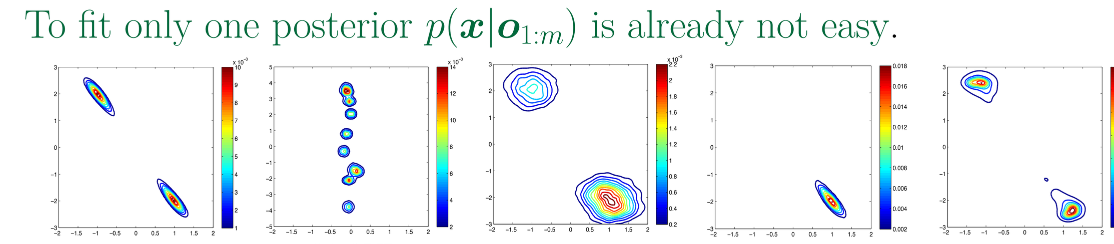
Figure: Cross entropy $\mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{o}_{1:m})} - \log q_m$

Experiment 2: Multi-Modal Posterior

Guassian Mixture Model

- prior $\boldsymbol{x}_1, \boldsymbol{x}_2 \sim \mathcal{N}(0, 1)$
- observations $\boldsymbol{o}|\boldsymbol{x}_1, \boldsymbol{x}_2 \sim \frac{1}{2}\mathcal{N}(\boldsymbol{x}, 1) + \frac{1}{2}\mathcal{N}(\boldsymbol{x}_1 + \boldsymbol{x}_2, 1)$
- With $(\boldsymbol{x}_1, \boldsymbol{x}_2) = (1, -2)$, the posterior has two modes.

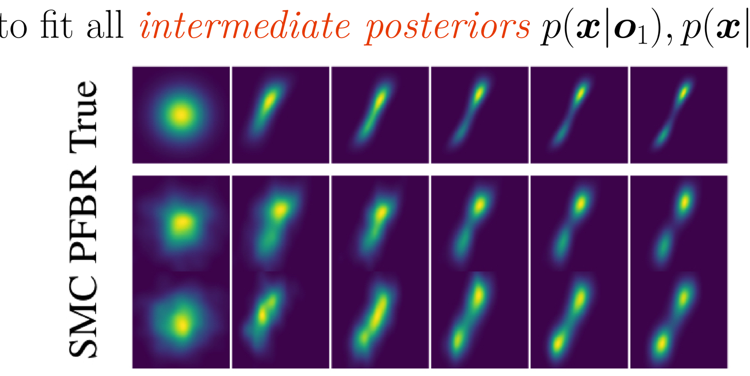
To fit only one posterior $p(\boldsymbol{x}|\boldsymbol{o}_{1:m})$ is already not easy.



(a) True posterior (b) SVI (c) SGLD (d) Gibbs Sampling (e) SVI Nonparametric

Our more challenging experimental setting:

- The learned MPF operator will be tested on sequences that are not observed in training set
- It needs to fit all *intermediate posteriors* $p(\boldsymbol{x}|\boldsymbol{o}_1), p(\boldsymbol{x}|\boldsymbol{o}_1, \boldsymbol{o}_2), \ldots$

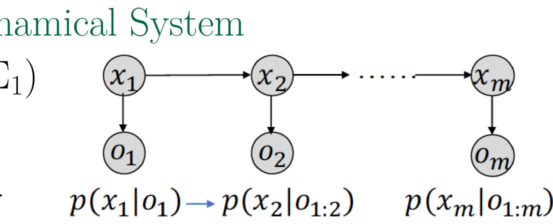


Experiment 3: Hidden Markov Model

Hidden Markov Model - Linear Dynamical System

- $\boldsymbol{x}_m = A\boldsymbol{x}_{m-1} + \epsilon_m,\ \epsilon_m \sim \mathcal{N}(0, \Sigma_1)$
- $o_m = B\boldsymbol{x}_m + \delta_m,\ \delta_m \sim \mathcal{N}(0, \Sigma_2)$

Marginal posteriors update: $\rightarrow$



Transition sampling + FPBR operator:

$$\text{(i) } \tilde{\boldsymbol{x}}_m^n = A\boldsymbol{x}_{m-1}^n + \epsilon_m, \quad \text{(ii) } \boldsymbol{x}_m^n = \mathcal{F}(\tilde{\mathcal{X}}_m, \tilde{\boldsymbol{x}}_m^n, o_{m+1}),$$

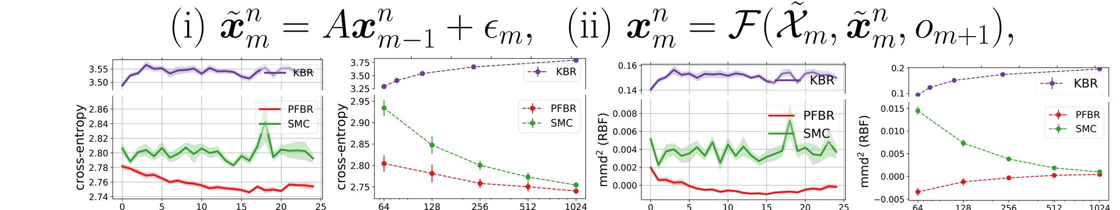


Figure: Left: cross entropy; Right: Square MMD with RBK kernel.

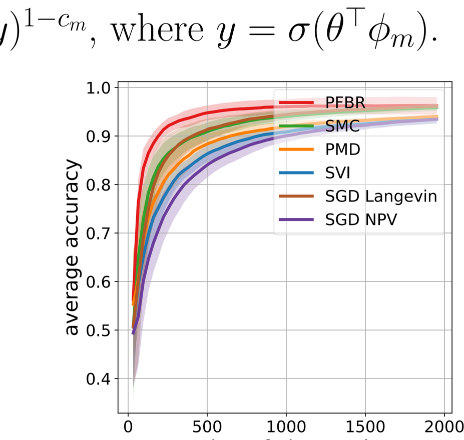
Experiment 4: Bayesian Logistic Regression

BLR on MNIST dataset 8 vs 6

- Likelihood function $p(o_m|\theta) = y^{c_m}(1-y)^{1-c_m}$, where $y = \sigma(\theta^\top \phi_m)$.

Multi-task Environment

- Reduce the dimension 50 by PCA
- Rotate the first two components by an angle $\psi \sim [-15°, 15°]$



predict $\rightarrow$ observe true labels $\rightarrow$ update particles $\rightarrow$ predict $\rightarrow$ observe true labels$\rightarrow \ldots$