Background: Mixed flows (MixFlows)

4

problem: have to optimize λ

solution: construct a map that automatically adapts to target

intuition: normalizing flows only learn target through optimization—too black-box

defn: a function T is **measure-preserving** for p if

$$T_\sharp p = p$$

intuition: $X \sim p \implies T(X) \sim p$

example: $p = \mathrm{Unif}[0,1]$

$$T(x) = x + \xi \mod 1$$

$$\xi \in \mathbb{R}$$

defn: a function T is **ergodic** for p if for all msrble A

$$T(A) = A \implies p(A) \in \{0, 1\}$$

intuition: $(T^n(x))_{n=1}^{\infty}$ explores support of p

(doesn't get stuck)

example: $p = \mathrm{Unif}[0, 1]$

$$T(x) = x + \xi \mod 1$$

only if ξ is irrational!

MixFlow: q is an averaged pushforward:

$$q_{N,\lambda} = \frac{1}{N} \sum_{n=0}^{N-1} T^n_{\lambda \sharp} q_0$$

theory?

theorem [XCC23]: if $q_0 \ll p$ and T_λ is an e.m.p. diffeomorphism then

$$\forall\lambda \quad \mathrm{D}_{\mathrm{TV}}(q_{N,\lambda}, p) \xrightarrow{N\to\infty} 0$$

sampling?

$$X_0 \sim q_0, \qquad K \sim \mathrm{Unif}\{0, \ldots, N-1\}$$
$$\implies T_\lambda^K(X_0) \sim q_{N,\lambda}$$

density?

$$q_{N,\lambda}(x) = \frac{1}{N} \sum_{n=0}^{N-1} \frac{q_0(T_\lambda^{-n}(x))}{\prod_{j=0}^{n} |J_\lambda(T_\lambda^{-j}(x))|}$$

only for real x!

Background: Mixed flows (MixFlows)

problem: have to optimize λ

solution: construct a map that automatically adapts to target

intuition: normalizing flows only learn target through optimization—too black-box

defn: a function T is **ergodic** for p if for all msrble A

$$T(A) = A \implies p(A) \in \{0, 1\}$$

intuition: $(T^n(x))_{n=1}^{\infty}$ explores support of p (doesn't get stuck)

example: $p = \mathrm{Unif}[0,1]$

$$T(x) = x + \xi \mod 1$$

only if ξ is irrational!

MixFlow: q is an averaged pushforward:

$$q_{N,\lambda} = \frac{1}{N}\sum_{n=0}^{N-1} T^n_{\lambda\sharp} q_0$$

density?

$$q_{N,\lambda}(x) = \frac{1}{N}\sum_{n=0}^{N-1} \frac{q_0(T_\lambda^{-n}(x))}{\prod_{j=0}^{n} |J_\lambda(T_\lambda^{-j}(x))|}$$

only for real x!

Background summary

normalizing flows:

- simple, general framework for continuous posterior distributions
- limited work for discrete posteriors—embed in continuous space and move on
- hard to optimize—many parameters, difficult optimization landscape