Sequential Density Estimation

Introduction

- Most density estimation techniques, including NPE, NLE and NRE, all have sequential forms as well (SNPE, SNLE, SNRE etc.) – the data is processed as it arrives and then is updated sequentially, as opposed to being collected all first and all calculations are done at once
- Particularly useful for:
  - Data that arrives in a stream
  - Getting real time updates to the density
  - Limited memory/computational resources or the data is very high dimensional/complex
- Used in a variety of places outside of density estimation, including Kalman filtering or other recursive methods

SNPE

- Specifically interested in SNPE for this project
- As a reminder – NPE is simulate some data, train an approximate posterior based on this data, then repeat and retrain etc. until some loss metric is minimised
- What returns is a trained posterior which can provide posterior estimates for any new x without retraining
- Sequential NPE is designed to increase sample efficiency and works as follows:
  - Train from an initial prior and simulated dataset to get initial estimate for posterior
  - Use the trained posterior to identify higher probability regions for θ given an observation x_0
  - Generate new simulations clustered around high probability areas in order to improve accuracy of the posterior estimates in relevant areas
  - Can either combine the new simulations with the old ones or replace them depending on the algorithm and retrain the NN
  - After fewer overall simulations than NPE, we get a fully trained posterior
- This targeted approach is especially efficient for a sharply peaked/narrowly concentrated posterior

SNPE vs NPE

Comparison: NPE vs. SNPE

Feature	NPE	SNPE
Simulation Strategy	All simulations are upfront and uniform across the prior.	Simulations are iterative, targeting high-posterior regions.
Efficiency	May require many simulations to cover the entire parameter space.	Focuses simulations on relevant areas, reducing overall cost.
Adaptivity	Non-adaptive; prior knowledge isn't updated during inference.	Adaptive; posterior guides simulations in subsequent rounds.
Use Case	Suitable for well-behaved priors or simple likelihoods.	Ideal for complex, high-dimensional, or narrow posterior regions.
Accuracy	Can struggle with regions of low simulation density.	More accurate in regions of interest due to targeted refinement.
Computational Requirements	Large initial simulation and memory requirements.	Lower overall simulation cost, but iterative retraining adds overhead.
Amortization	Fully amortized inference, works for all x including far from x_0	Still amortized, but in practice it is biased towards x_0
