Validation of SBI (contd.)

- posterior checks: - calibration of true parameters Y^* relative to predicted $p(Y \mid X^{obs})$
  - compare $p(Y \mid X)$ to $p^*(Y \mid X)$
  - check diversity of $p(Y \mid X)$
- posterior predictive checks:

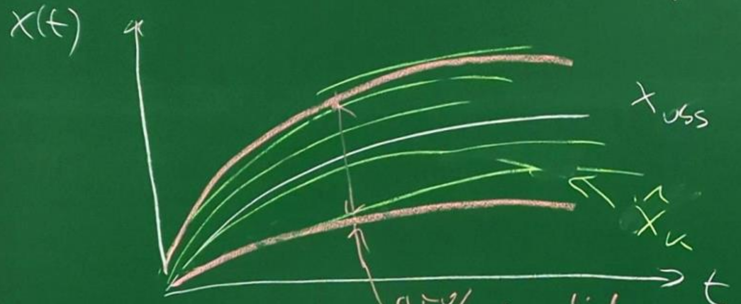
- sample from posterior $\quad \{\hat{Y}_k \sim p(Y \mid X^{obs})\}_{k=1}^{M}$
- use predicted parameters as simulation inputs $\quad \{\hat{X}_k = \phi(\hat{Y}_k, \eta)\}_{k=1}^{M}$
- compare $\hat{X}_k$ to true X^{obs} $\Rightarrow$ should be similar
  - traditional: $\|\hat{X}_k - X^{obs}\|_2^2$

[also used to solve inverse problem: $\hat{Y} = \arg\min_Y \|\phi(Y) - X^{obs}\|_2^2$

disadvantages: - non-linear optimization finds local optimum
- disregards uncertainty / ambiguity of solution
- may be ill-posed $\Rightarrow$ add a regularizer to make well-posed $\Rightarrow$ possible bias]

not good when $\hat{X}_k$ are (correctly!) very diverse

$\Rightarrow$ check calibration of X^{obs} relative to $\{\hat{X}_k\}$



95% confidence region of $\hat{x}_k$ should contain X^{obs} most of the time

[$X_j \equiv$ time points of a dynamic system]

[$Y \equiv$ parameters of dynamic system, e.g. virus: infection rate, duration of disease } later]

if posterior has lower variance than prior (= data X^{obs} gave us information)

posterior predictive scenarios are more accurate than prior predictive ones

$\Rightarrow$ better predictions of future outcomes

does not work so easily when behavior is <u>non-stationary</u>, i.e. Y is time-dependent

gets worse if outcomes X influence Y (feedback)

- **self-consistency error**   train two networks - $p(X|Y)$ (simulation surrogate)
 - $p(Y|X)$ (posterior)

if networks are correct, they fullfill Bayes rule

fix $Y_1 \sim p^s(Y)$ and $X_1 = \phi(Y_1, \eta)$

for true distributions, we have $p^s(Y) \cdot p^s(X|Y) = p^s(X)\, p^s(Y|X)$

same should hold for our models $p^s(Y)\, p(X|Y) = p(X)\, p(Y|X)$

with $p(x) = \int p(X|Y)\, p^s(Y)\, dY$

rearrange Bayes and take logarithms

$$p(X) = \frac{p^s(Y)\, p(X|Y)}{p(Y|X)} \Rightarrow$$

for X fixed.

$$\log p(X) = \underbrace{\log p^s(Y) + \log p(X|Y) - \log p(Y|X)}_{\text{must be constant for \underline{all} } Y \sim p(Y|X)} = \text{const}$$

if models are consistent, for fixed X we have

$$\mathrm{Var}_{Y \sim p(Y|X)} \left[\log p^S(Y) + \log p(X|Y) - \log p(Y|X) \right] = 0$$

$\rightarrow$ measure variance as a quality score

$\Rightarrow$ even better: use Variance as an additional loss [Schmitt et al. 2023]

$\Rightarrow$ can also be used to determine $\log p(X)$ <u>without</u> solving integral

$$\log p(X) = \mathbb{E}_{Y \sim p(Y|X)} \left[\log p^S(Y) + \log p(X|Y) - \log p(Y|X) \right]$$

$\Rightarrow$ it is beneficial to train surrogate $p(X|Y)$ and posterior $p(Y|X)$ <u>jointly</u>

JANA method [Radev et al. 2023]

- sensitivity analysis

- how much do the solutions change, when
  - training set
  - simulation
  - learned model
  - observed data

  } change

- example: if we have a lot of prior knowledge $\hat{=}$ highly contained in $p^s(Y)$
  $\Rightarrow$ new data should change our oppinion only a little

$$KL\left[\, p(Y|X) \,\|\, p^s(Y) \,\right] \approx 0$$

  [or MMD

  if not $\Rightarrow$ prior was worse than we thought

- if we do not have much prior knowledge, $p^s(Y)$ is "uninformative"
  [should not impose prejudice]
  e.g. Jeffrey's prior

  $\Rightarrow$ the posterior should depend only very little on prior, most information should come from data

$\Rightarrow$ sequential Baysian updating: data arrive in batches $X^{(1)}, X^{(2)}, \dots, X^{(t)}, \dots$

$$p(Y|X^{(1)}) \propto p(X^{(1)}|Y)\, p(Y)$$ iteration 1

$\vdots$

$$p(Y|X^{(1)},\dots,X^{(t)}) \propto p(X^{(t)}|Y) \cdot p(Y|X^{(1)},\dots,X^{(t-1)})$$ iteration t

$\uparrow$ use posterior at $t-1$ as prior for t

$$KL[p(Y|X^{(1)}\dots X^{(t)}) \,\|\, p(Y|X^{(1)},\dots X^{(t-1)})] \to 0 \text{ as } t\to\infty$$

- especially elegant with conjugate priors ($p(Y)$, $p(Y|X^{(1)},\dots)$ are from same distribution family)
- so far, no cheap algorithm to do this with neural networks (iterative alg. is not much cheaper than training from scratch with $X^{(1)},\dots,X^{(t)}$)

---

in amortized SBI: hyper-parameter aware training

- apply parametric variation to training process and tell networks about current parameters

X parameter values of training variation

$$Y \leftarrow \boxed{cNF} \leftarrow t$$

<u>example</u> power-scaling of the prior $p^S(Y) \longrightarrow \dfrac{p^S(Y)^\alpha}{\xi(\alpha)}$ with $\xi(\alpha=1)=1$

for many standard distributions analytic formula exist

$\alpha > 1 \Rightarrow$ prior gets sharper (more informative

$\alpha < 1 \Rightarrow$ prior gets less informative

e.g. $N(0,\sigma^2)^\alpha = N\left(0, \dfrac{\sigma^2}{\alpha}\right)$

[α is inverse temperature in Gibb's distribution $\exp\left(-\dfrac{\text{energy}}{kT}\right)$

$\alpha = \dfrac{1}{T}$

- during training $\alpha = p^S(\alpha)$, $Y \sim \dfrac{p^S(Y)^\alpha}{\xi(\alpha)}$ $X = \phi(Y, \eta)$

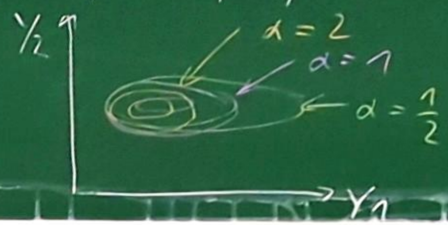
$\alpha \in [\frac{1}{2}, 2]$ additional prior for α (sometimes difficult to choose)

$p^S(\alpha) = \text{uniform}(\frac{1}{2}, 1, 2)$

network is <u>told</u> current value of α $\Rightarrow$ learns posterior $p(Y \mid X, \alpha)$

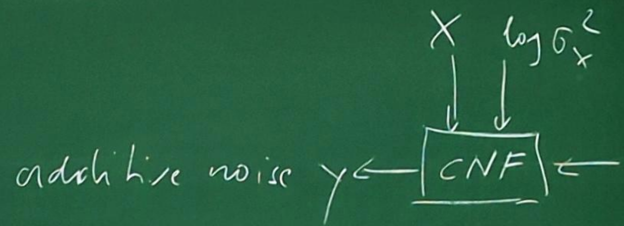
- during inference create $p(Y \mid X^{obs}, \alpha)$ for different values of α and compare

example from epidemiology



Y_1 depends strongly on prior

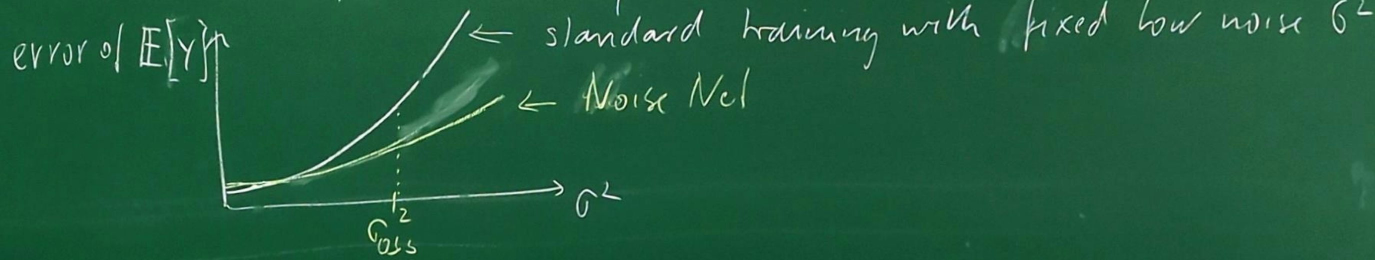
Y_2 is robust (outcome dominated by the data) example 2: NoiseNet [Kan et al. 2023]



X may have variable levels of noise

$\sigma_x^2 \sim p(\sigma_x^2) \rightarrow X = \phi(Y) + N(0, \sigma_x^2)$

- train with different σ_x^2, • inference: estimate actual noise level of X^{obs}



use same principle for all types of perturbations during training

- prior scaling - likelihood variations - data augmentation/perturbation etc. ⇒ hyper-parameter aware network amortizes over permutations

⇒ sensitivity analyses at inference time cheap (does not require additional training)

makes many repetitions (as required for trustworthy sensitivity analysis) tractable

[Elsemüller et al. 2023]

References:

Maximum mean discrepancy (MMD): [Gretton et al. 2012] A Kernel Two-Sample Test, https://jmlr.csail.mit.edu/papers/v13/gretton12a.html
https://www.onurtunali.com/ml/2019/03/08/maximum-mean-discrepancy-in-machine-learning.html

Frechet Inception Distance (FID score): https://en.wikipedia.org/wiki/Fr%C3%A9chet_inception_distance

Uni- and bivariate posterior plots of model vs. MCMC: [Radev et al. 2020] BayesFlow: Learning complex stochastic models with invertible neural networks, https://arxiv.org/abs/2003.06281, (figure 5)

Calibration via uniformity of a histogram: [Radev et al. 2020] BayesFlow (figures 6 and 7)

Calibration via empirical CDF and random walk: [Säilynoja et al. 2022] Graphical Test for Discrete Uniformity and its Applications in Goodness of Fit Evaluation and Multiple Sample Comparison, https://arxiv.org/abs/2103.10522, see also [Elsemüller et al. 2023]

Proper scoring rules: [Gneiting & Rafterty 2007] Strictly proper scoring rules, prediction, and estimation, https://doi.org/10.1198/016214506000001437

Density and coverage metrics: [Naeem et al. 2020] Reliable Fidelity and Diversity Metrics for Generative Models, https://arxiv.org/abs/2002.09797

Vendi score: [Friedman & Dieng 2022] The Vendi Score: A Diversity Evaluation Metric for Machine Learning, https://arxiv.org/abs/2210.02410

Sensitivity analysis: [Elsemüller et al. 2023] Sensitivity-Aware Amortized Bayesian Inference, https://arxiv.org/abs/2310.11122

Sequential Bayesian updating, conjugate priors: [Kessler et al. 2023] On Sequential Bayesian Inference for Continual Learning, https://arxiv.org/abs/2301.01828

NoiseNet: [Kang et al. 2023] Noise-Net: determining physical properties of H_{II} regions reflecting observational uncertainties, https://doi.org/10.1093/mnras/stad072

Self-consistency via Bayes rule: [Schmitt et al. 2023] Leveraging Self-Consistency for Data-Efficient Amortized Bayesian Inference, https://arxiv.org/abs/2310.04395