

$\alpha \in \{ \frac{1}{2}, 1, 2 \}$

$p^s(Y)^n$

## Examples for amortized SBI

① robot arm  $\rightarrow$  see earlier

② Epidemiology: SIR model

[Kermack & McKendrick 1927]

• compartments

{	Susceptible	S	people who are healthy but can get infected
	Infected	I	—  — ill and can transmit disease
	Recovered	R	—  — healthy again and now immune

in the most simple model variant

• simplifying assumptions

- stationary dynamics: behavior of virus/bacteria and people does not change over time (no mutations, no countermeasures)

- only consider averages over all people in each compartment

$\Rightarrow$  all people in same compartment are considered identical

- simplest variant 3 compartments SIR,  $S+I+R=N$  population size

• design model:

- a healthy individual meets on average  $\lambda_1$  people per day  
(assume that  $N$  is so big that  $\lambda_1$  is independent of  $N$ )
- infected people are not isolated, but meet others as usual

$\Rightarrow$  a fraction  $\frac{I}{N}$  of the  $\lambda_1$  meetings is potentially dangerous

- a fraction  $\lambda_2$  of all the dangerous meetings actually leads to transmission

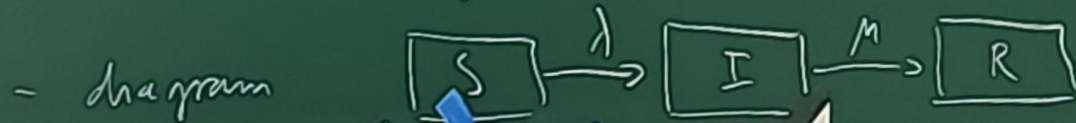
- if we observe a reduction in infection, we cannot distinguish if people became more cautious ( $\lambda_1 \downarrow$ ) or virus is less infectious ( $\lambda_2 \downarrow$ )

$\Rightarrow$  can only recover  $\lambda = \lambda_1 \lambda_2$

$\Rightarrow$  number of new infections per day  $\lambda \frac{I}{N} \cdot S$   $\lambda$  infection rate

- nobody dies, infected people recover after  $\delta$  days on average

$\Rightarrow$  number of people to recover per day  $\frac{1}{\delta} I = \mu I$   $\mu$  recovery rate





- write the dynamics as a system of ordinary differential equations (ODEs)

$$\frac{dS(t)}{dt} = -\lambda \cdot \frac{I(t)}{N} \cdot S(t) \quad \text{minus: infected people leave compartment S}$$

$$\frac{dI(t)}{dt} = \lambda \frac{I(t)}{N} S(t) - \mu I(t) \quad \text{minus: recovered leave comp. I}$$

$$\frac{dR(t)}{dt} = \mu I(t) \quad \text{immunity of R} \hat{=} \text{no one leaves comp. R}$$

- can divide all equations by  $N \Rightarrow [S] = \frac{S}{N}$  etc.

$$\frac{d[S(t)]}{dt} = -\lambda [I(t)] \cdot [S(t)], \quad \frac{d[I(t)]}{dt} = \lambda [I(t)] [S(t)] - \mu [I(t)], \quad \frac{d[R(t)]}{dt} = \mu [I(t)]$$

- adding of three equations proves correct normalization

$$\frac{d[S(t) + I(t) + R(t)]}{dt} = 0 \quad [S(t) + I(t) + R(t)] = \text{const.} \stackrel{!}{=} 1$$

- to solve equations, must define initial conditions, e.g. values at  $t=0$

$I_0 = [I(t=0)] = \text{was 1}$  initial number of infected (when disease is first detected)

$$[R(t=0)] = 0$$

$$[S(t=0)] = 1 - I_0$$

$\Rightarrow$  given  $I_0, \lambda, \mu$  we can solve the ODEs for any time  $t > 0$  by "integration"

• simplest method: Euler forward method. use discrete time steps  $\Delta t$  and approximate

$$[S(t+\Delta t)] = [S(t)] - \lambda [I(t)] [S(t)] \cdot \Delta t$$

$$\left[ \frac{dS}{dt} = \lim_{\Delta t \rightarrow 0} \frac{S(t+\Delta t) - S(t)}{\Delta t}, \text{ now stop lim at finite } \Delta t \right]$$

$$[I(t+\Delta t)] = [I(t)] + (\lambda [I(t)] [S(t)] - \mu [I(t)]) \Delta t$$

$$[R(t+\Delta t)] = [R(t)] + \mu [I(t)] \Delta t$$

• theory say that Euler forward is a good approximation if  $\Delta t$  is small enough

$$\Delta t \leq \min_t \left( \frac{2}{\lambda [I(t)] [S(t)]}, \frac{2}{|\lambda [I(t)] [S(t)] - \mu [I(t)]|}, \frac{2}{\mu [I(t)]} \right) \quad \left( \begin{array}{l} \text{I'm not 100\%} \\ \text{sure} \end{array} \right)$$

• more sophisticated solvers (e.g. Euler backward, Runge-Kutta methods) allow larger time steps



• define observables  $X$

- report on every day number of new infections and newly recovered

- observations are not perfect  $\Rightarrow$  observation model

• reporting delay  $\Delta S^{obs}(t) = f(\Delta S^*(t-L))$   $L$ : delay

• underreporting

$$\Delta S^{obs} \sim \beta \Delta S^*$$

$\beta$ : fraction of detected cases

• noise

- exact values  $\Delta S^*(t) = S^*(t-\Delta t) - S^*(t)$

$$\Delta R^*(t) = R^*(t) - R^*(t-\Delta t)$$

- measured values  $\tilde{\Delta S}(t) = \Delta S^*(t-L) \cdot \epsilon_S$   $\epsilon_S \sim N(\beta, \sigma^2)$

$\epsilon_S$ : relative error, because multiplicative

$$\tilde{\Delta R}(t) = \Delta R^*(t-L) \cdot \epsilon_R \quad \epsilon_R \sim N(\beta, \sigma^2)$$

(for simplicity, we assume that observation parameters are equal for  $\tilde{\Delta S}$  and  $\tilde{\Delta R}$ )

• full simulation:  $Y = [I_0, \lambda, \mu, L, \beta, \sigma^2] \sim p^s(Y)$   
 (for synthetic data) usually, independent prior  $p^s(Y) = p^s(I_0) p^s(\lambda) p^s(\mu) p^s(L) p^s(\beta) p^s(\sigma^2)$

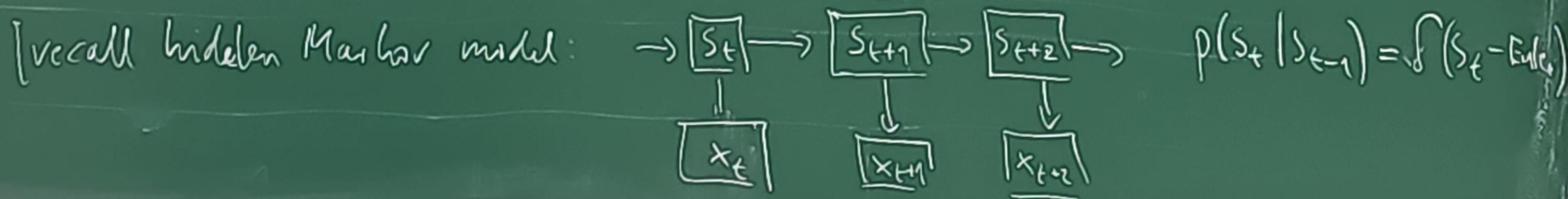
$X = \phi(Y, \eta)$  ODE + reporting noise prior knowledge  
 choose according to epidemiological knowledge

$$\eta = [(\epsilon_S(\Delta t), \epsilon_R(\Delta t)), (\epsilon_S(2\Delta t), \epsilon_R(2\Delta t)), \dots, (\epsilon_S(T), \epsilon_R(T))]$$

$$X = [(\Delta \tilde{S}(\Delta t), \Delta \tilde{R}(\Delta t)), (\Delta \tilde{S}(2\Delta t), \Delta \tilde{R}(2\Delta t)), \dots, (\Delta \tilde{S}(T), \Delta \tilde{R}(T))]$$

$N, T$  are hyperparameters or sampled  $N \sim p^s(N), T \sim p^s(T)$

$\Rightarrow$  use simulation to create a large TS =  $\left\{ (Y_i \sim p^s(Y), X_i = \phi(Y_i, \eta \sim p^s(\eta))) \right\}_{i=1}^M$





• traditional non-probabilistic parameter fitting: least squares

$$\hat{\gamma} = \underset{\gamma}{\operatorname{argmin}} \mathbb{E}_{\eta \sim p^s(\eta)} \left[ \|x^{\text{obs}} - \phi(\gamma, \eta)\|^2 \right]$$

disadvantages: - non-linear least squares might get stuck in a bad local optimum

↑ real data

$\phi(\hat{\gamma}, \eta)$  should reproduce real observations

- disregards the uncertainty and ambiguity in  $\gamma$

at best, we get  $\hat{\gamma} = \underset{\gamma}{\operatorname{argmax}} p^s(\gamma | x^{\text{obs}})$

but not the full  $p^s(\gamma | x^{\text{obs}})$

← true posterior of simulation

• amortized SBI learns a generative neural network for  $p(\gamma | x^{\text{obs}}) \approx p^s(\gamma | x^{\text{obs}})$  using a large TS of synthetic data ( $M = 10\,000 +$ )

• full algorithm:

- ① define the simulation  $\phi(\gamma, \eta)$  and priors  $p^s(\gamma)$  and  $p^s(\eta)$
- ② use simulation to generate synthetic TS
- ③ set up architecture of generative neural network

$X$  (or  $\log X$ )

Summary network

← any feature detector, for time series with varying length  $T$   
⇒ Recurrent network (RNN, LSTM, transformers)

↓  $h(x)$  learned features of  $X$

map variable length  $X$   
to fixed length  $h(x)$

CNF

←  $z \sim N(0, \Pi)$   
 $\hat{p}(Y | h(x))$

(4) train CNF & summary network jointly using NLL loss

$$\hat{p}, \hat{h} = \underset{p, h}{\operatorname{argmin}} \frac{1}{M} \sum_{i=1}^M -\log p(Y_i | h(x_i)) \Rightarrow h(x) \text{ will become optimally informative for } p(Y | h(x))$$

(5) validate  $\hat{p}, \hat{h}$  (check calibration, sensitivity etc.) using a synthetic test set

⇒  $\hat{p}, \hat{h}$  faithfully represent the simulation posterior

(6) infer  $\hat{p}(Y | h(x^{\text{obs}}))$  for real data

(7) check for potential simulation gap (⇒ simulation is unrealistic, "model misspecification")