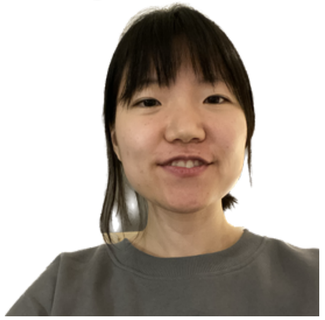
STATISTICS AND DATA SCIENCE: INFORMING POLICY AND COUNTERING MISINFORMATION
PORTLAND OREGON JSM
AUGUST 3–8, 2024

CHICAGO BOOTH
The University of Chicago Booth School of Business

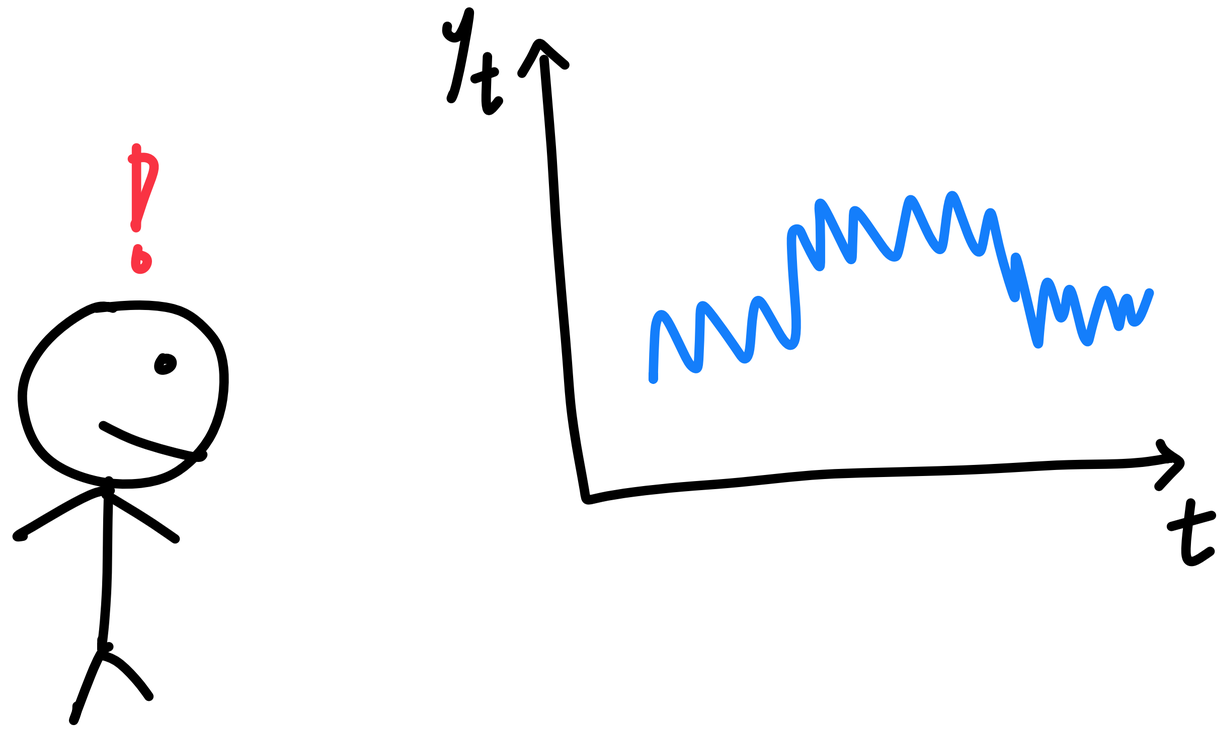
Deep Bayes Factors

Jungeum Kim and Veronika Rockova



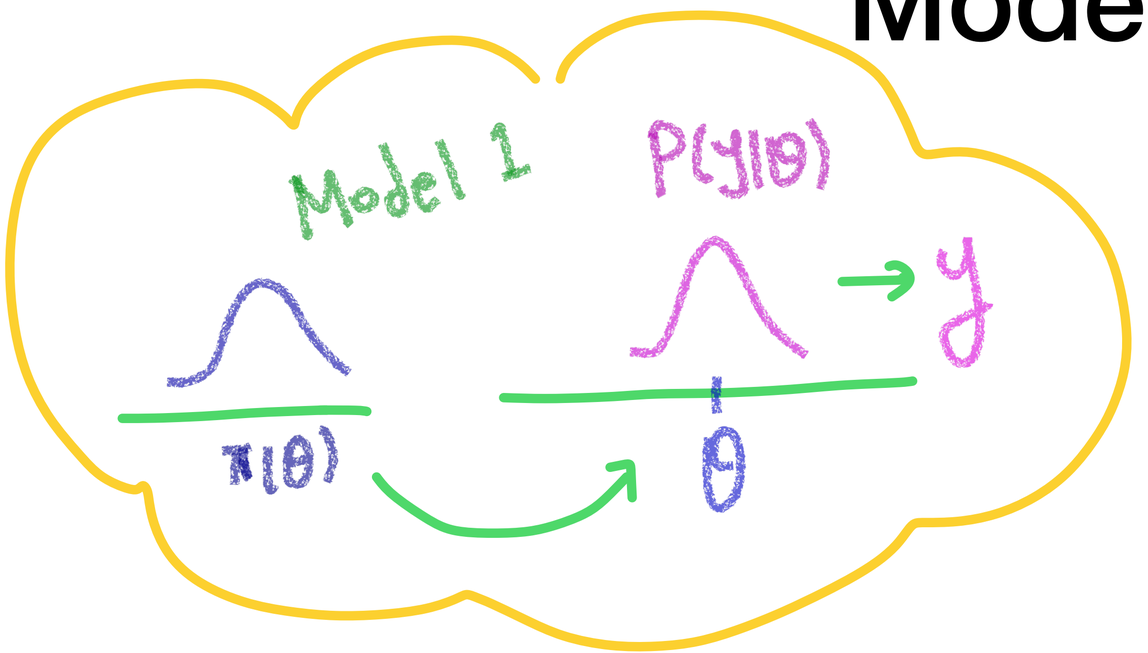
Model Selection

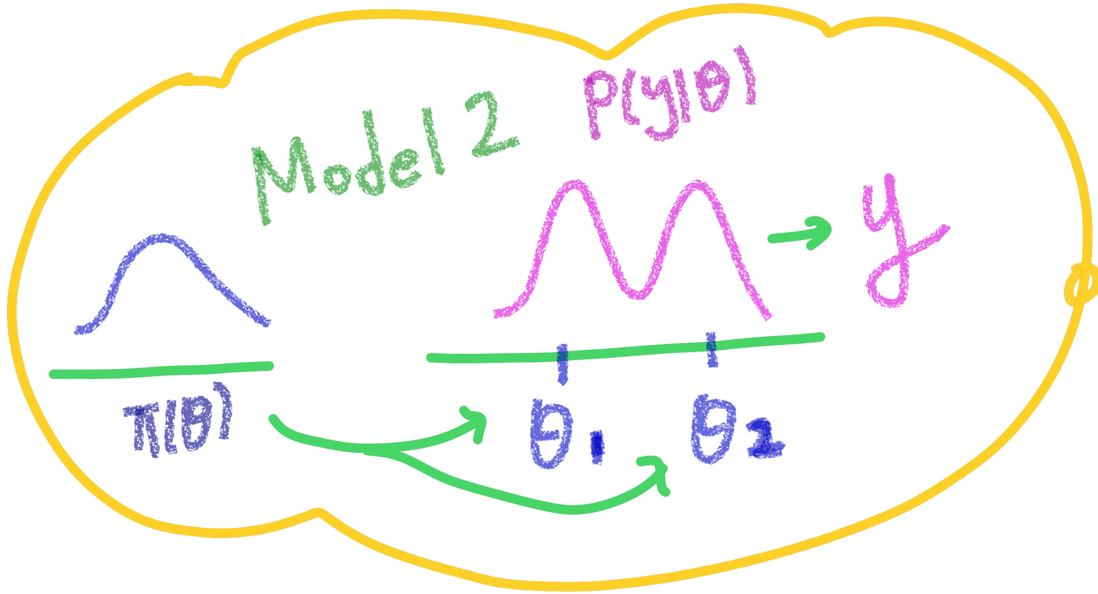
$Y_0^{(n)} = (Y_1, \ldots, Y_n)$: observed data

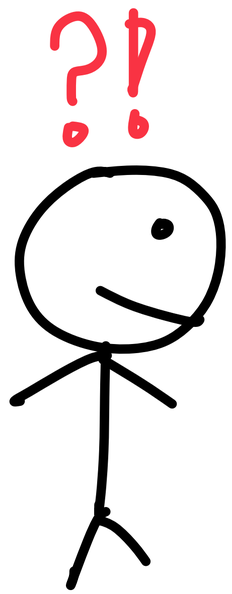


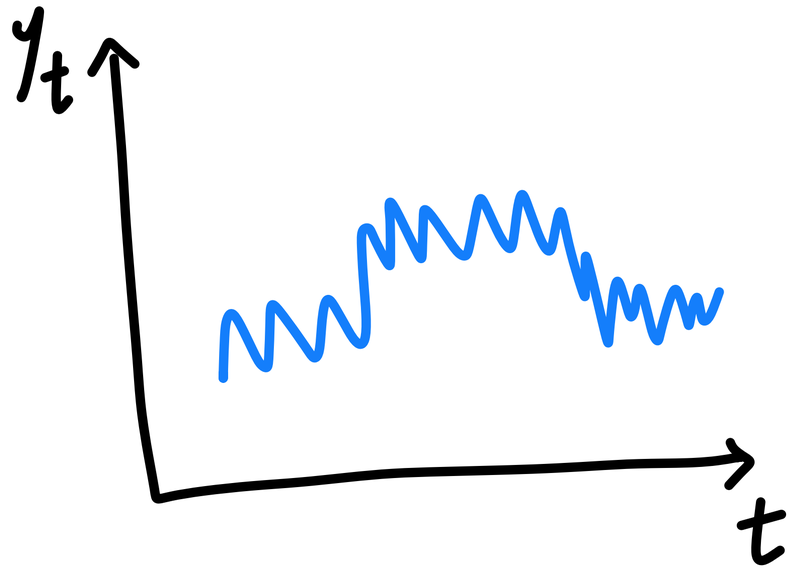
Model Selection

$Y_0^{(n)} = (Y_1, \ldots, Y_n)$: observed data

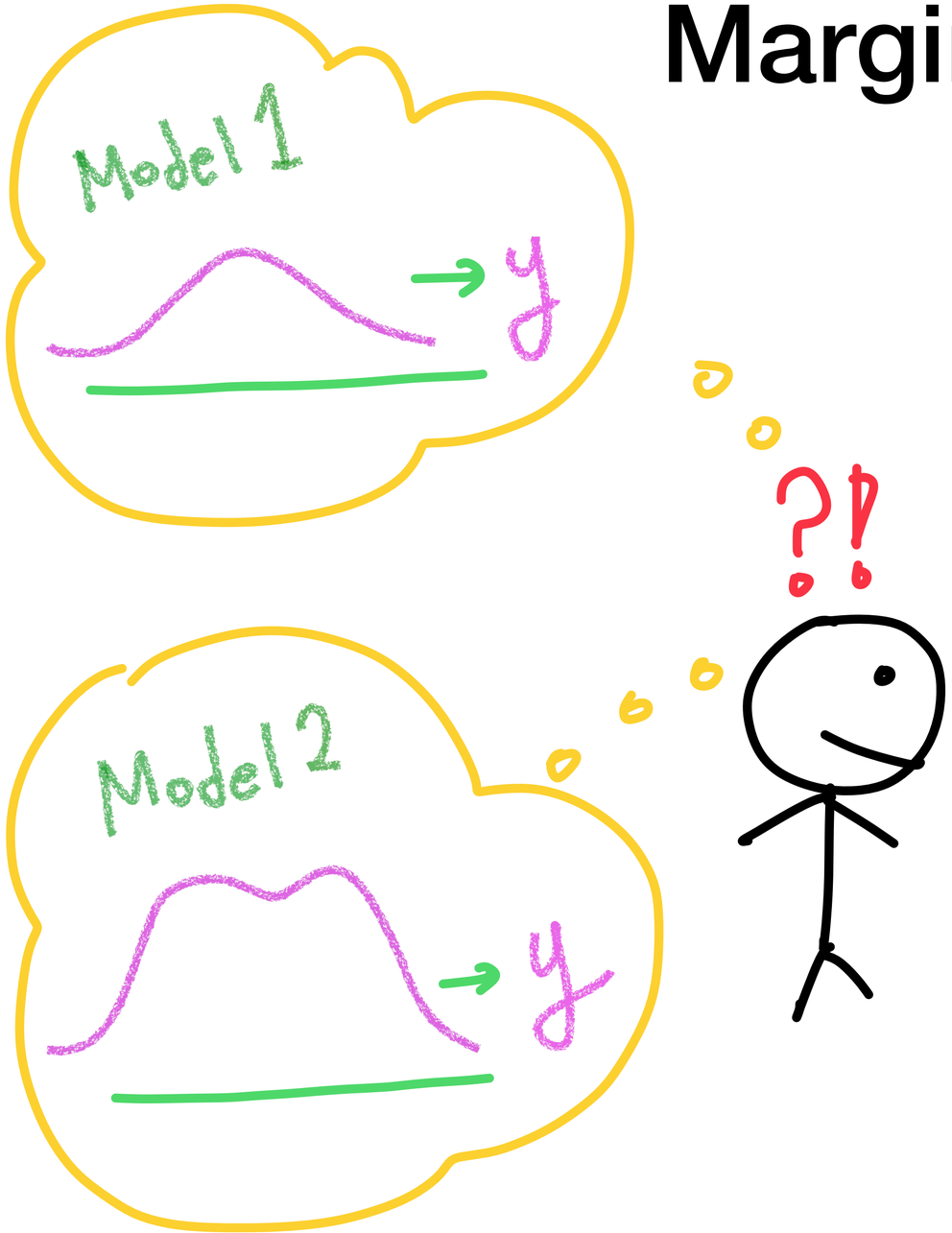








Marginal Likelihood

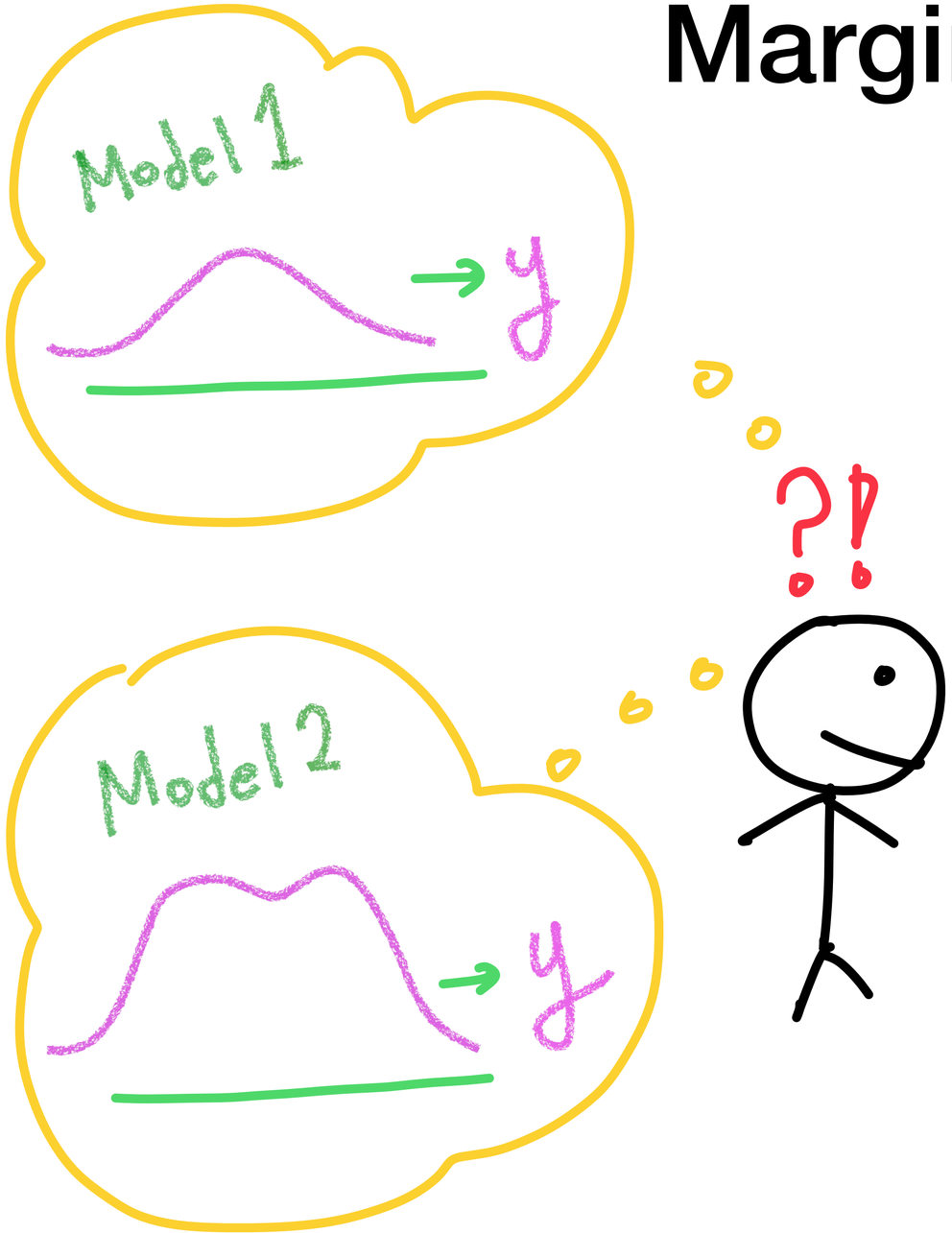


$$Y_0^{(n)} = (Y_1, \ldots, Y_n) : \text{observed data}$$

$$P_m(Y_0^{(n)}) = \int P_m(Y_0^{(n)} \mid \theta_m)\pi_m(\theta_m)d\theta_m,$$

$$m = 1,2$$

Marginal Likelihood Ratio



$$Y_0^{(n)} = (Y_1, \ldots, Y_n) : \text{observed data}$$

$$P_m(Y_0^{(n)}) = \int P_m(Y_0^{(n)} \mid \theta_m)\pi_m(\theta_m)d\theta_m, \quad m = 1,2$$

$$BF_{1,2}(Y_0^{(n)}) = \frac{P_1(Y_0^{(n)})}{P_2(Y_0^{(n)})}$$

Normalization constant problem

$$\pi_i(\theta_i \mid Y_0^{(n)}) \propto P_i(Y_0^{(n)} \mid \theta_i)\pi_i(\theta_i)$$

Marginal Likelihood Ratio

Importance Sampling

-> Require

MCMC : sampling from posterior

Likelihood known, closed form

-> More efficient sampling

Famous: (Warp) Bridge sampling

Bennett (76'), Meng et al (96'), Meng et al (02')

$$Y_0^{(n)} = (Y_1, \ldots, Y_n) : \text{observed data}$$

$$P_m(Y_0^{(n)}) = \int P_m(Y_0^{(n)} \mid \theta_m)\pi_m(\theta_m)d\theta_m, \quad m = 1,2$$

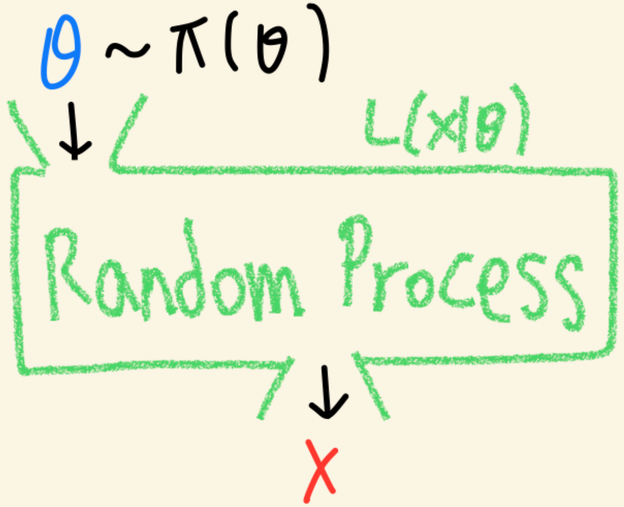
$$BF_{1,2}(Y_0^{(n)}) = \frac{P_1(Y_0^{(n)})}{P_2(Y_0^{(n)})}$$

Normalization constant problem

$$\pi_i(\theta_i \mid Y_0^{(n)}) \propto P_i(Y_0^{(n)} \mid \theta_i)\pi_i(\theta_i)$$

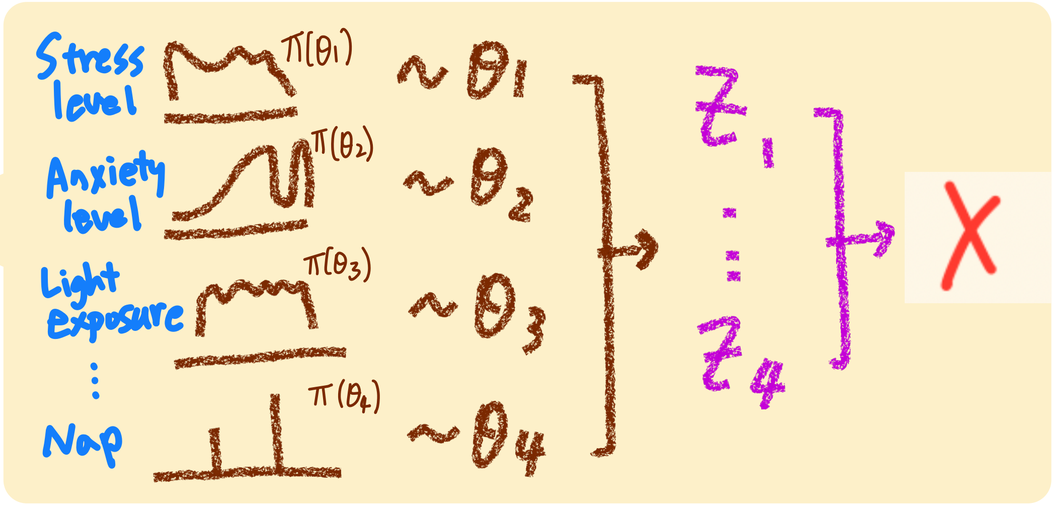
Likelihood Free Settings

Model



Time needed to fall asleep

Model 1



Likelihood?!

DeepBF
(Deep Bayes Factors)

Amortised Bayes Factor estimator

Oxford languages defines **amortisation** as

"The action or process of gradually writing off the initial cost of an asset."

In our context, the initial "cost" is that of training a neural network to make estimation from data.

Cost reduction:

$$\hat{BF}_{1,2}(\cdot) \longrightarrow \hat{BF}_{1,2}(Y^{(n)})$$

Costly Cheap-!

Slide modified from : Zammit-Mangion (Neural estimators for irregular spatial data)

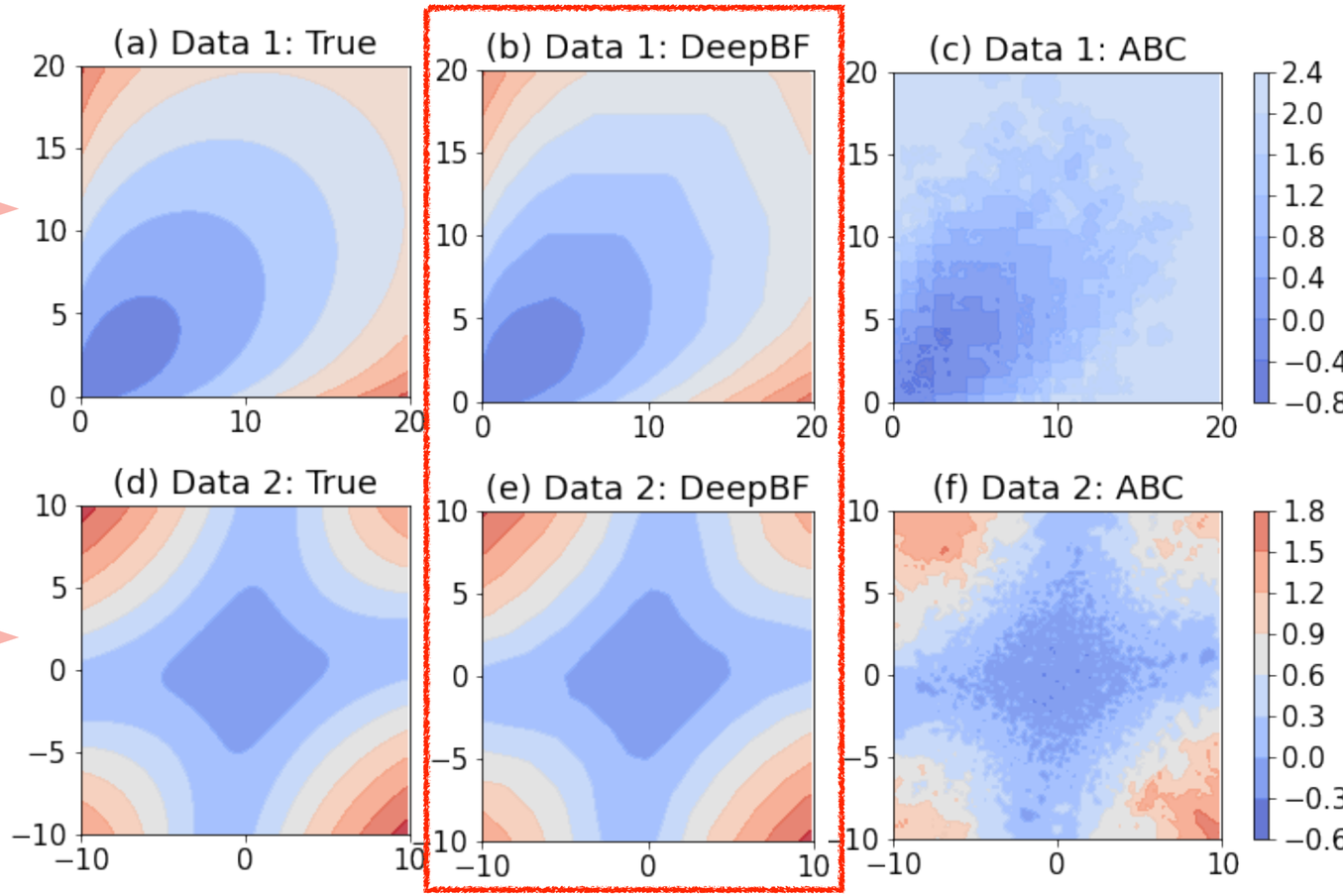
Example

$$M_1 : Y_i | p \sim NB(1,p),\ p \sim Beta(\alpha_1, \beta_1)$$

$$M_2 : Y_i | \lambda \sim Pois(\lambda),\ \lambda \sim Gamma(\alpha_2, \beta_2)$$

$$M_1 : Y_i \sim N(\mu_{11},2^2)/2 + N(\mu_{12},2^2)/2,$$

$$M_2 : Y_i \sim N(\mu_2,2.5^2),\ \mu_* \sim N(0,\sigma_*^2)$$



One training -> many evaluations

Basic Fully Connected Neural Network

(n=2)

Idea: Likelihood Ratio Trick

Hastie et al, 09', Sugiyama 12'

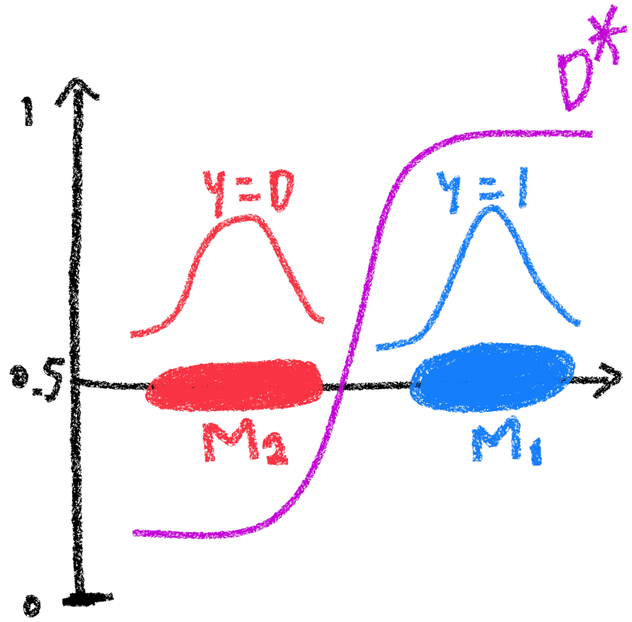
Goal: $BF_{12}(Y_0) = P_1(Y_0)/P_2(Y_0)$ Data: $Y_0 = (Y_1, \ldots, Y_n)$

Define: $y \sim Ber(0.5), p(Y|y=1) = P_1(Y), p(Y|y=0) = P_2(Y)$

$$D^* = \arg\max_{D \in \mathscr{D}_n} [E_{Y \sim P_1} \log D(Y) + E_{Y \sim P_2} \log(1 - D(Y))]$$

$$\Rightarrow \quad \frac{D^*(Y)}{1 - D^*(Y)} = \frac{P_1(Y)}{P_2(Y)} = BF_{1,2}(Y) \text{ for any } Y \in \text{Supp}(P)$$

$\mathscr{D}_n = \{D : \mathscr{Y}^n \to (0,1)\}$



– *generative adversarial networks (Goodfellow et al, 20')*

– *noise-contrastive estimation (Gutmann et al, 10')*

– *Metropolis-Hastings algorithm (Kaji et al, 22')*

Deep BF Training

Hastie et al, 09', Sugiyama 12'

Goal: $BF_{12}(Y_0) = P_1(Y_0)/P_2(Y_0)$ Data: $Y_0 = (Y_1, \ldots, Y_n)$

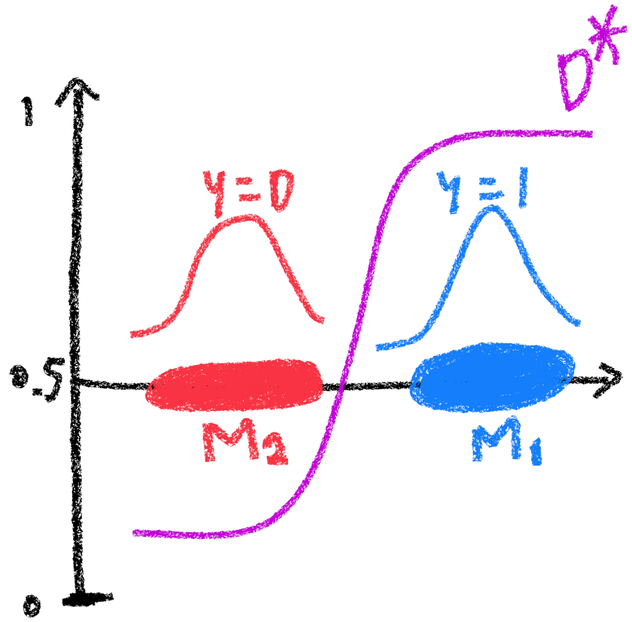
$Y_1, \ldots, Y_T$ from $P_1(Y)$, $\tilde{Y}_1, \ldots, \tilde{Y}_T$ from $P_2(Y)$

$$\mathbb{M}_T(D) = \frac{1}{T}\sum_{i=1}^{T} \log D(Y_i) + \frac{1}{T}\sum_{i=1}^{T} \log(1 - D(\tilde{Y}_i))$$

$$\Rightarrow \quad \hat{BF}_{1,2}(Y_0) = \frac{\hat{D}_T(Y_0)}{1 - \hat{D}_T(Y_0)}$$

Stochastic optimization

On-the-shelve mini-batch sampling



Sampling: $Y \sim P_1(Y)$ by forward sampling

Sample $\theta_{1,j}$ **from** $\pi_1 \quad \Rightarrow \quad Y_j \in \mathcal{Y}^n$ **from** $P_1(\cdot\,|\,\theta_{1j})\; j = 1,\ldots,s$

No posterior sampling but only forward!

Deep BF Training

Hastie et al, 09', Sugiyama 12'

Goal: $BF_{12}(Y_0) = P_1(Y_0)/P_2(Y_0)$ Data: $Y_0 = (Y_1, \ldots, Y_n)$

$Y_1, \ldots, Y_T$ from $P_1(Y)$, $\tilde{Y}_1, \ldots, \tilde{Y}_T$ from $P_2(Y)$

$$\mathbb{M}_T(D) = \frac{1}{T}\sum_{i=1}^{T} \log D(Y_i) + \frac{1}{T}\sum_{i=1}^{T} \log(1 - D(\tilde{Y}_i))$$

$$\Rightarrow \quad \hat{BF}_{1,2}(Y_0) = \frac{\hat{D}_T(Y_0)}{1 - \hat{D}_T(Y_0)}$$

Stochastic optimization

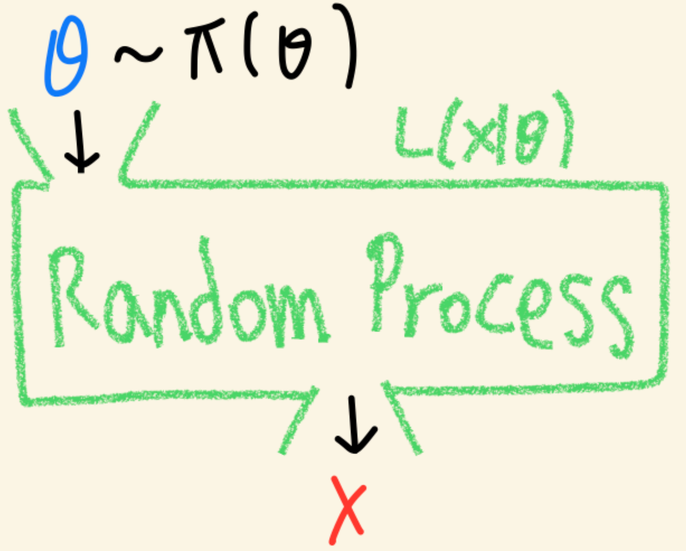
On-the-shelve mini-batch sampling

Sampling: $Y \sim P_1(Y)$ by forward sampling

Sample $\theta_{1,j}$ **from** $\pi_1 \quad \Rightarrow \quad Y_j \in \mathcal{Y}^n$ **from** $P_1(\cdot\,|\theta_{1j})\; j = 1,\ldots,s$

No posterior sampling but only forward!

Model



Theoretical Analysis

$Y_0^{(n)} = (Y_1, \ldots, Y_n)$: observed iid data from P^* (density: p^*)

Hypotheses:

$$H_0 : \ p^* \in \mathcal{F}_0 \quad \text{against} \quad H_1 : \ p^* \in \mathcal{F}_1,$$

$$\mathcal{F}_0 = \{P_1(Y^{(n)} \mid \theta_1) : \theta_1 \in \mathbb{R}^{d_1}\} \qquad \text{and} \qquad \mathcal{F}_1 = \{P_2(Y^{(n)} \mid \theta_2) : \theta_2 \in \mathbb{R}^{d_2}\}$$

Estimation Consistency: $\hat{BF}_{12}(Y_0^{(n)}) \to BF_{12}(Y_0^{(n)})$

Inferential Consistency: $\hat{BF}_{12}(Y_0^{(n)}) \to \infty$ if H_1 is true

Estimation Consistency

Assume that a classifier $\hat{D}$ has been trained on T pairs of training data from $M_1$and M_2. Under the Assumptions 1 and 2 in Kaji et al 22', we have

$$E_1 \left| \log \hat{BF}_{1,2}(Y^{(n)}) - \log BF_{1,2}(Y^{(n)}) \right| = O_{\bar{P}}(\delta_{T,n})$$

$\delta_{T,n}$: non-negative sequence, the quality of the training algorithm

($d(\hat{D}, D^*) = O_P(\delta_{T,n})$, P encompasses all randomness)

E_1: expectation with respect to the marginal $P(Y \mid M_1)$

Consistency: when $\mathscr{D}_n$ is reach enough to approximate D^*

($\delta_{T,n} \to 0$ for any n as $T \to \infty$)

Inferential Consistency

Bayes factor convergence rate (definition)

We say that $BF_{1,2}(Y_0^{(n)})$ is consistent at a rate $1/\nu_n$ for a sequence $\nu_n \to 0$ if

$$\lim_{n\to\infty} P^*(\log BF_{1,2}(Y_0^{(n)}) < 1/\nu_n \,|\, M_1) = 0 \quad \text{and} \quad \lim_{n\to\infty} P^*(\log BF_{1,2}(Y_0^{(n)}) > -1/\nu_n \,|\, M_2) = 0$$

Bayes factor estimation convergence rate (theorem)

Under the same assumption and additional regularity conditions,

$$\lim_{n\to\infty} P^*(\log \hat{BF}_{1,2}(Y_0^{(n)}) < 1/\nu_n \,|\, M_1) = 0 \quad \text{and} \quad \lim_{n\to\infty} P^*(\log \hat{BF}_{1,2}(Y_0^{(n)}) > -1/\nu_n \,|\, M_2) = 0$$

Even if DeepBF is (estimation) inconsistent ($\delta_{T,n} \nrightarrow 0$ as $T \to \infty$),

DeepBF can be consistent in model choice

As long as the true Bayes factors are consistent

And $n^{d_1/2}\delta_{T,n}$ is slower than the speed $1/\nu_n$ of Bayes factor divergence (regularity cond.)

Where to use DeepBF?

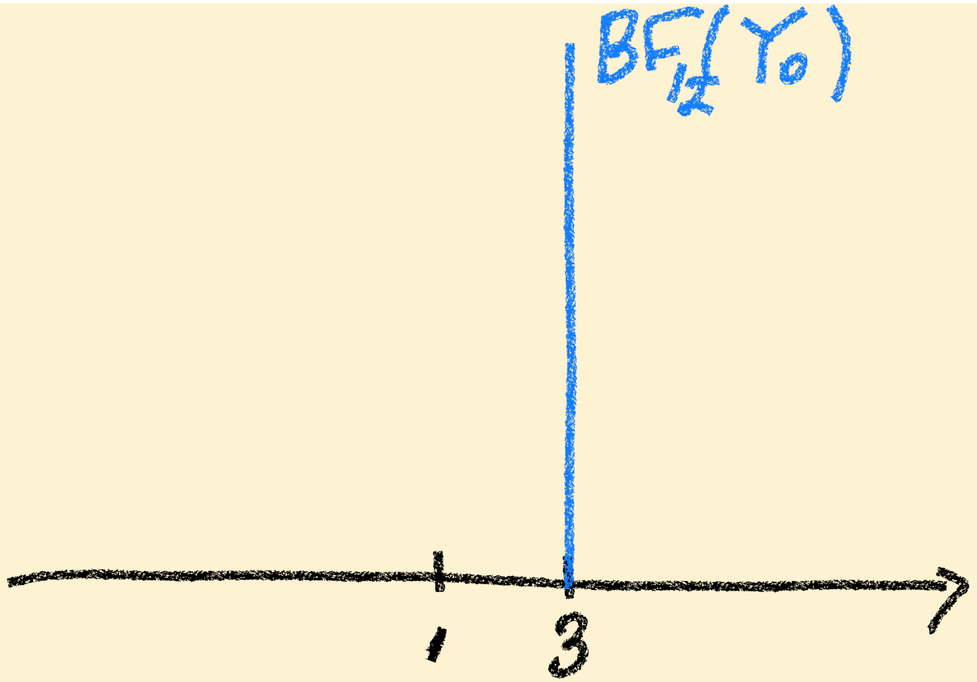
1. Distributional Inference

$BF_{12}(Y)$: random variable

Surprise Measure (Bayarri and Berger, 1998)

$$p_1(Y_0) = P(BF_{12}(Y) > BF_{12}(Y_0) \,|\, M_1)$$

$$p_2(Y_0) = P(BF_{12}(Y) \leq BF_{12}(Y_0) \,|\, M_2)$$



Surprise Measure estimation

$$\hat{p}_1(Y_0) = \sum_{i=1}^{T} (\hat{BF}_{12}(Y_i) > \hat{BF}_{12}(Y_0) \,|\, M_1)$$

DeepBF is useful even when the likelihood is known

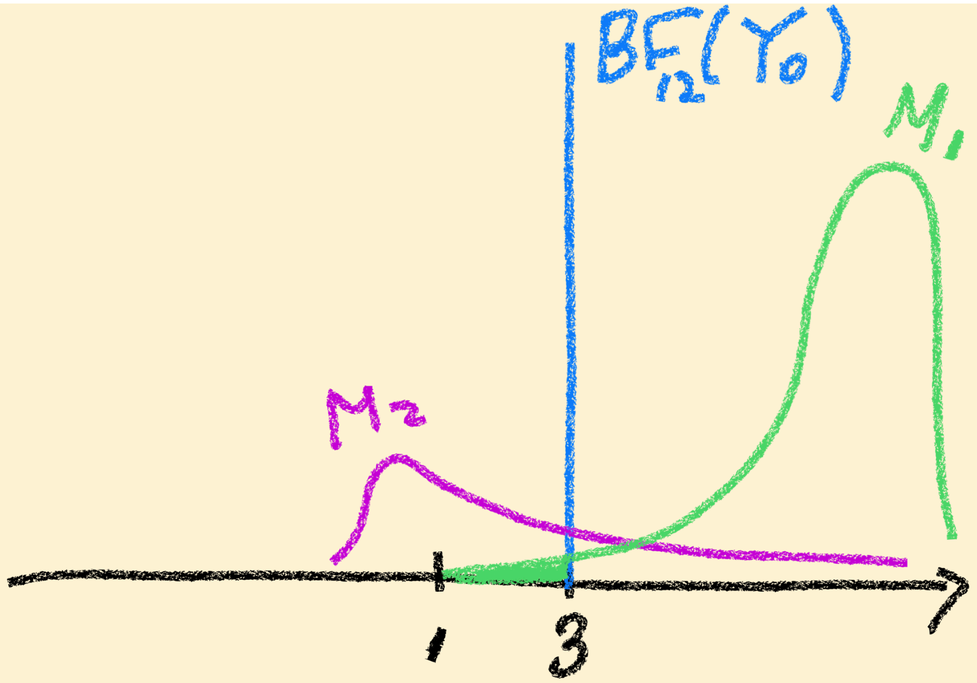
1. Distributional Inference

$BF_{12}(Y)$: random variable

Surprise Measure (Bayarri and Berger, 1998)

$$p_1(Y_0) = P(BF_{12}(Y) > BF_{12}(Y_0) \,|\, M_1)$$

$$p_2(Y_0) = P(BF_{12}(Y) \leq BF_{12}(Y_0) \,|\, M_2)$$



Surprise Measure estimation

$$\hat{p}_1(Y_0) = \sum_{i=1}^{T} (\hat{BF}_{12}(Y_i) > \hat{BF}_{12}(Y_0) \,|\, M_1)$$

DeepBF is useful even when the likelihood is known

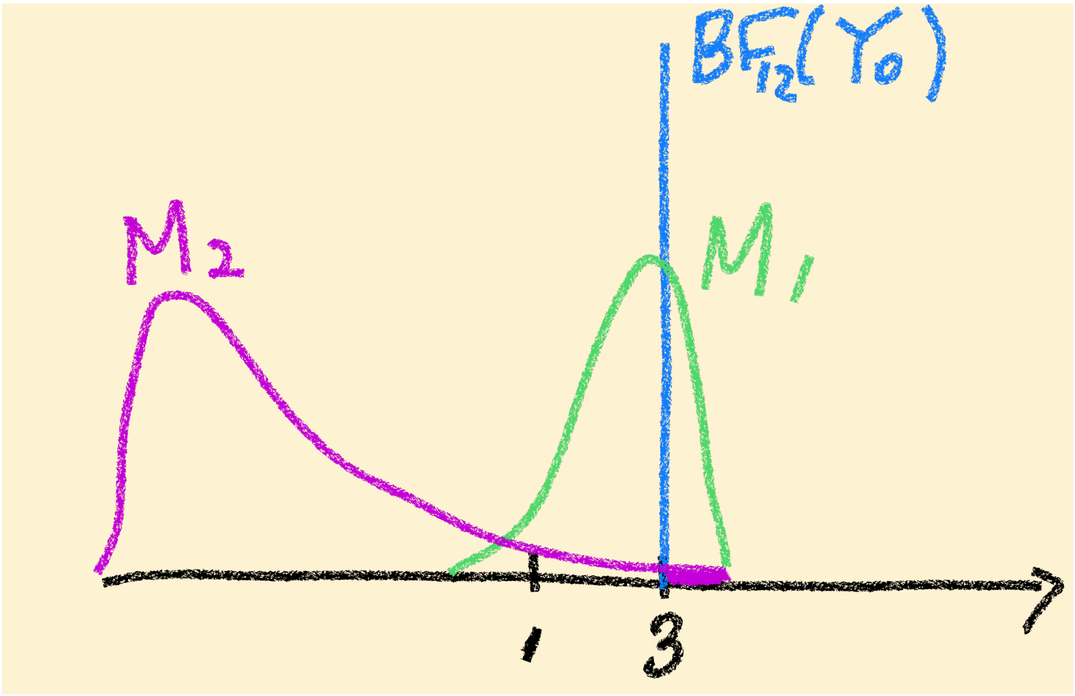
1. Distributional Inference

$BF_{12}(Y)$: random variable

Surprise Measure (Bayarri and Berger, 1998)

$$p_1(Y_0) = P(BF_{12}(Y) > BF_{12}(Y_0) \,|\, M_1)$$

$$p_2(Y_0) = P(BF_{12}(Y) \leq BF_{12}(Y_0) \,|\, M_2)$$



Surprise Measure estimation

$$\hat{p}_1(Y_0) = \sum_{i=1}^{T} \mathbb{I}_{(\hat{BF}_{12}(Y_i) > \hat{BF}_{12}(Y_0) | M_1)}$$

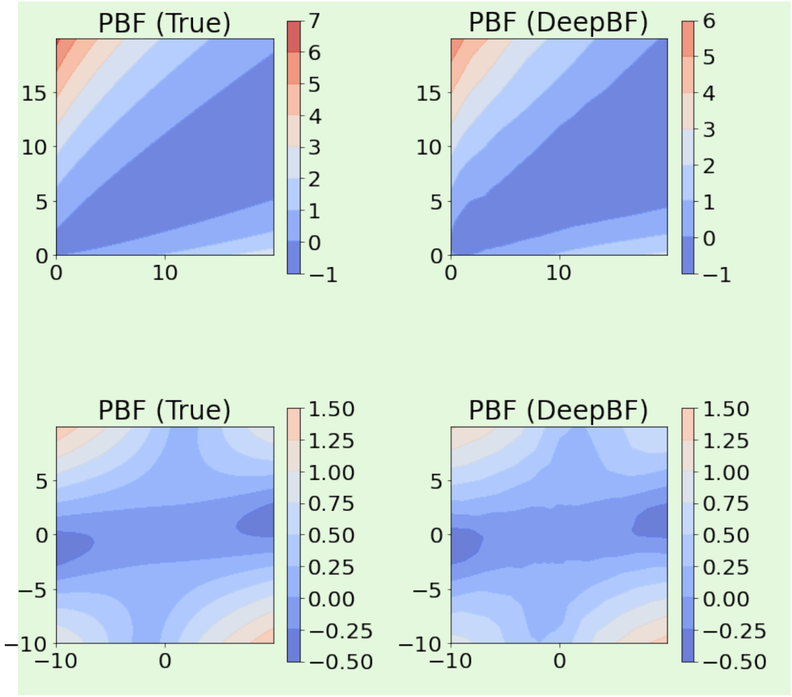
DeepBF is useful even when the likelihood is known

2. Bayes Factor Variants

Partial Bayes Factor

$$Y = (X, Z)$$

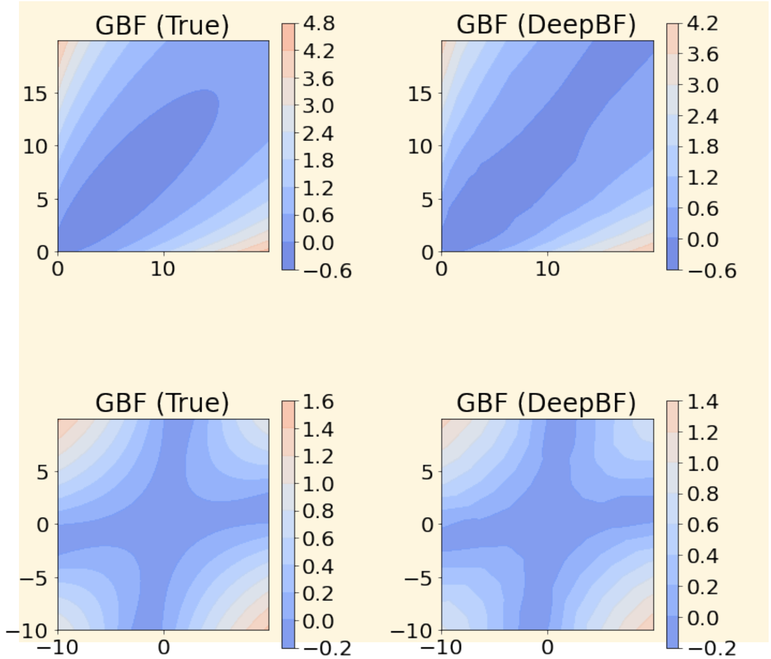
$$PBF_{1,2}(Z|X) = BF_{1,2}(Y)BF_{2,1}(X)$$



Geometric intrinsic Bayes Factor

$$GBF_{1,2}^{n_x}(Y) = \left(\prod_{i=1}^{K} PBF_{2,1}(Y_{\setminus i} \,|\, Y(i)) \right)^{1/K}$$

$$= \left(\prod_{i=1}^{K} B_{1,2}(Y) B_{2,1}(Y(i)) \right)^{1/K}$$



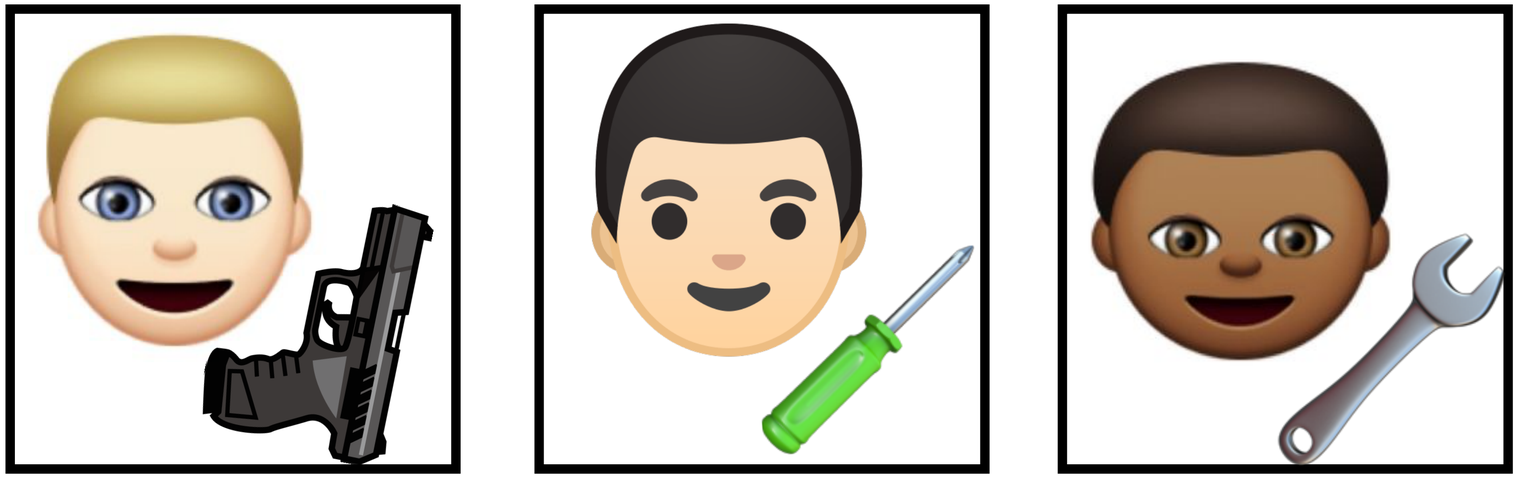
Instrinsic Prior Effect

(Berger and Pericchi, 96')

Only two DeepBF needed!

3. Scenario Based Models

Weapon Identification task



42 participants, 36 trials for each participant

< Cognitive Model >

Model 1: Controlled Recognition Process -> Prejudice -> Random Guess

Model 2: Controlled Recognition Process -> Random Guess -> Prejudice

3. Scenario Based Models

Weapon Identification task

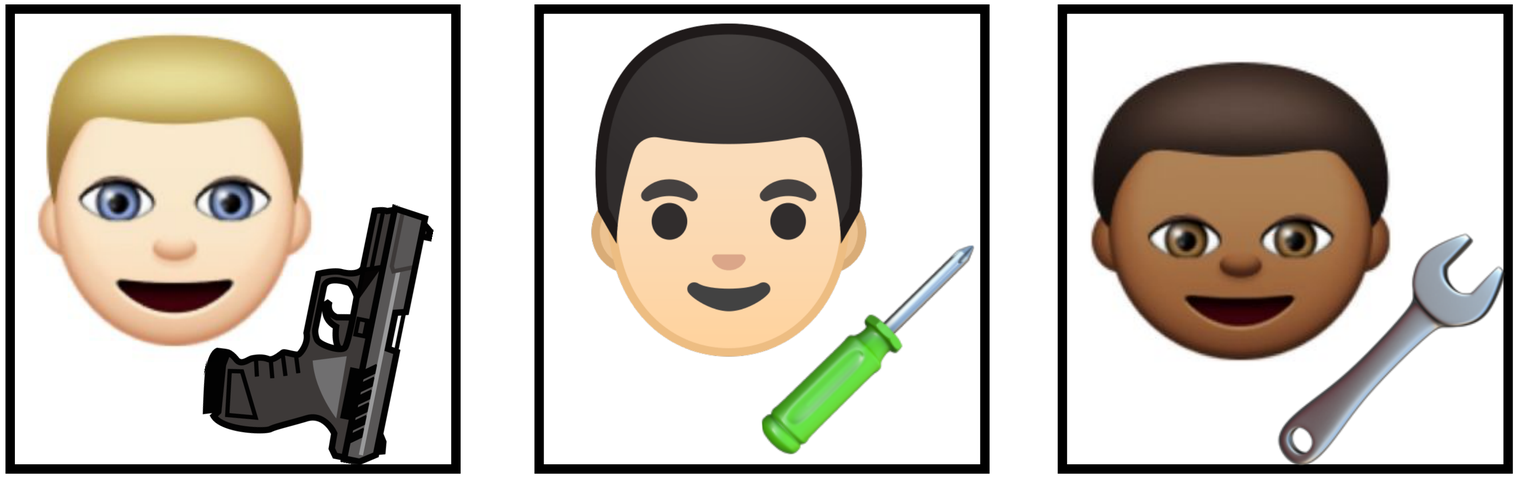
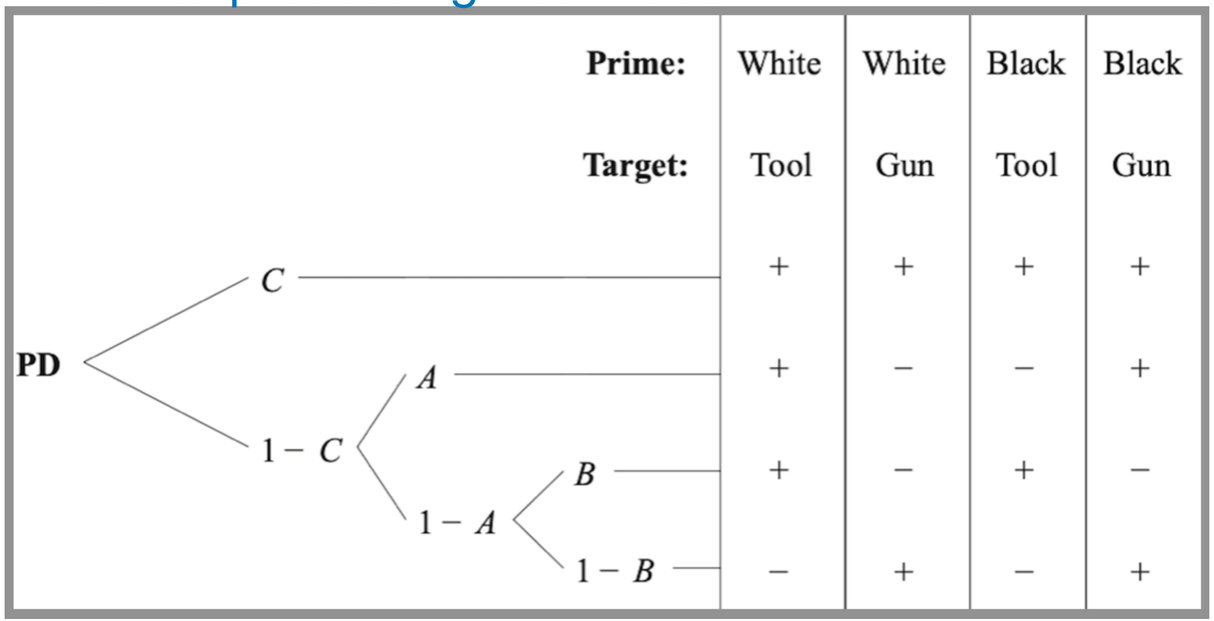
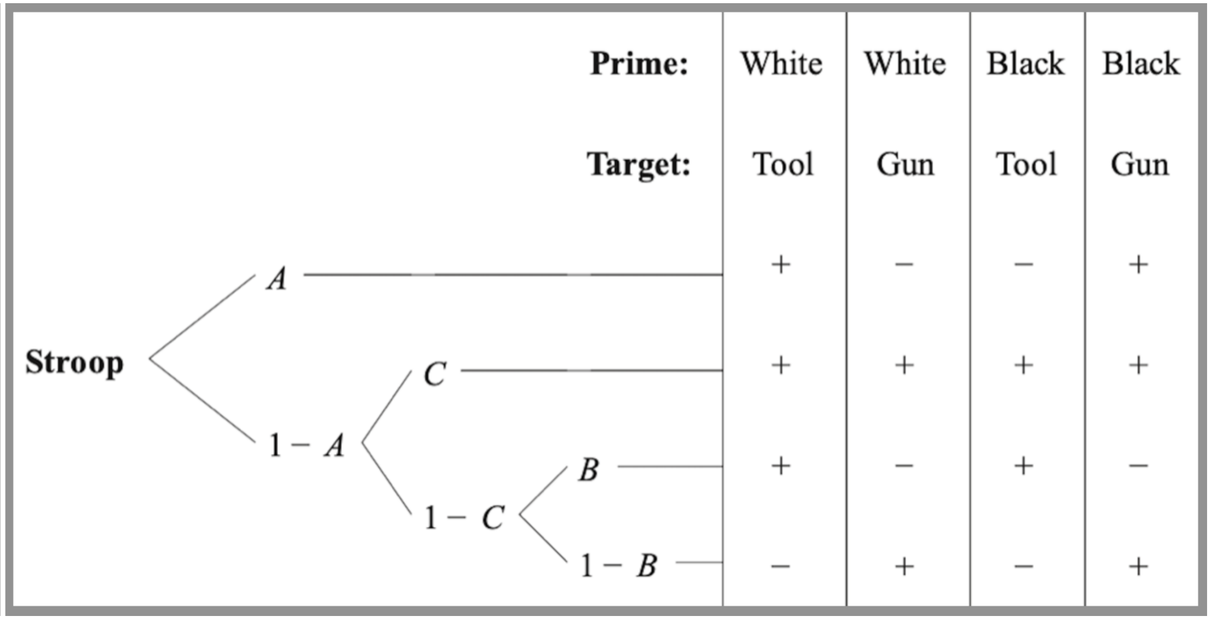


Image copied from Heck et al (23')

multinomial-processing tree model





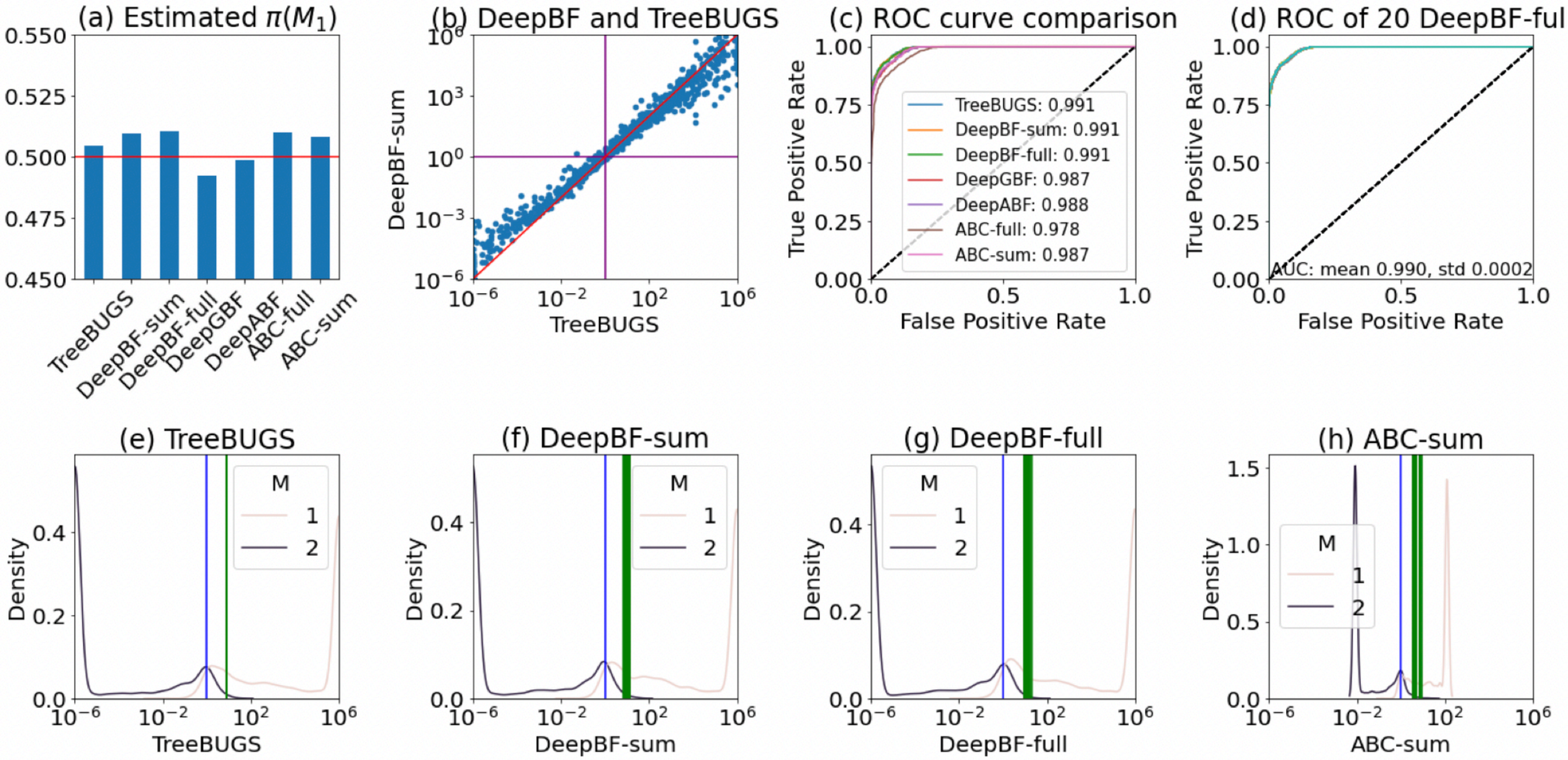


Figure 10: (a) The estimated priors. True is $\pi(M_1) = 0.5$. (b) The scatter plot of the BF estimates by DeepBF and TreeBUGS. (c) The ROC curves of methods in comparison (d) The ROC curves of 20 DeepBF networks. (e-h) The empirical density (KDE) plot of the BF estimates with reference line at 1 (blue) and at the estimated $\widehat{BF}_{1,2}(Y_0^{(n)})$ (green).

Additional Note

What if both models do not make sense?

(Goodness of fit)

Generative Goodness of Fit

Observed data

$$Y_0 = (Y_1, \ldots, Y_n)$$

Training data

$\tilde{Y} = (\tilde{Y}_1, \ldots, \tilde{Y}_n)$: simulated by $\tilde{Y}_i \overset{i.i.d.}{\sim} \pi_M(Y_i \mid Y_0)$

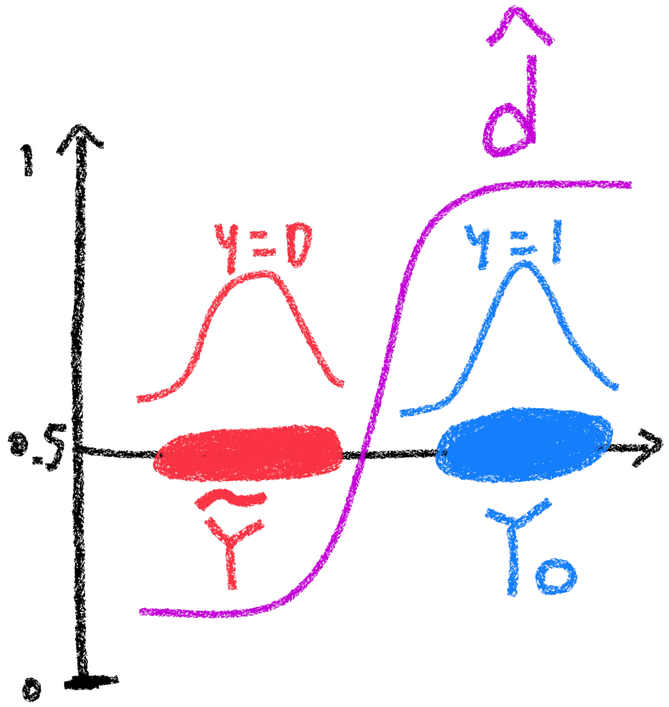
Test data

$\tilde{\tilde{Y}} = (\tilde{Y}_1, \ldots, \tilde{Y}_n)$: simulated by $\tilde{\tilde{Y}}_i \overset{i.i.d.}{\sim} \pi_M(Y_i \mid Y_0)$

Sampling from $\pi_M(Y_i \mid Y_0)$:

θ from $\pi_M(\theta \mid Y_0)$: Bayesian GAN (Wang et al, 22')

$\tilde{Y}_i$ from $P_1(Y \mid \theta)$: Forward sampling

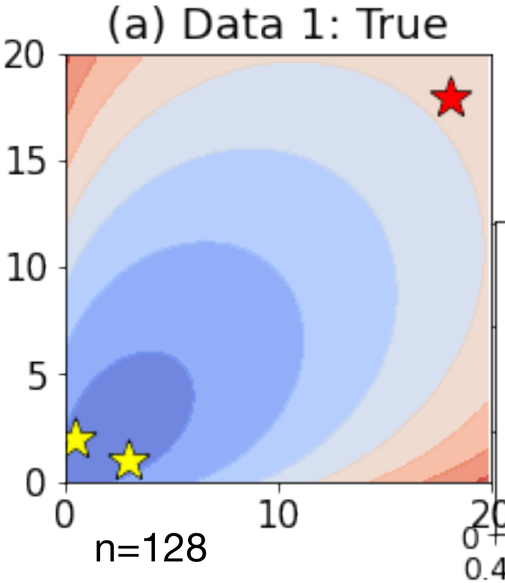


Goodness of fit measure:

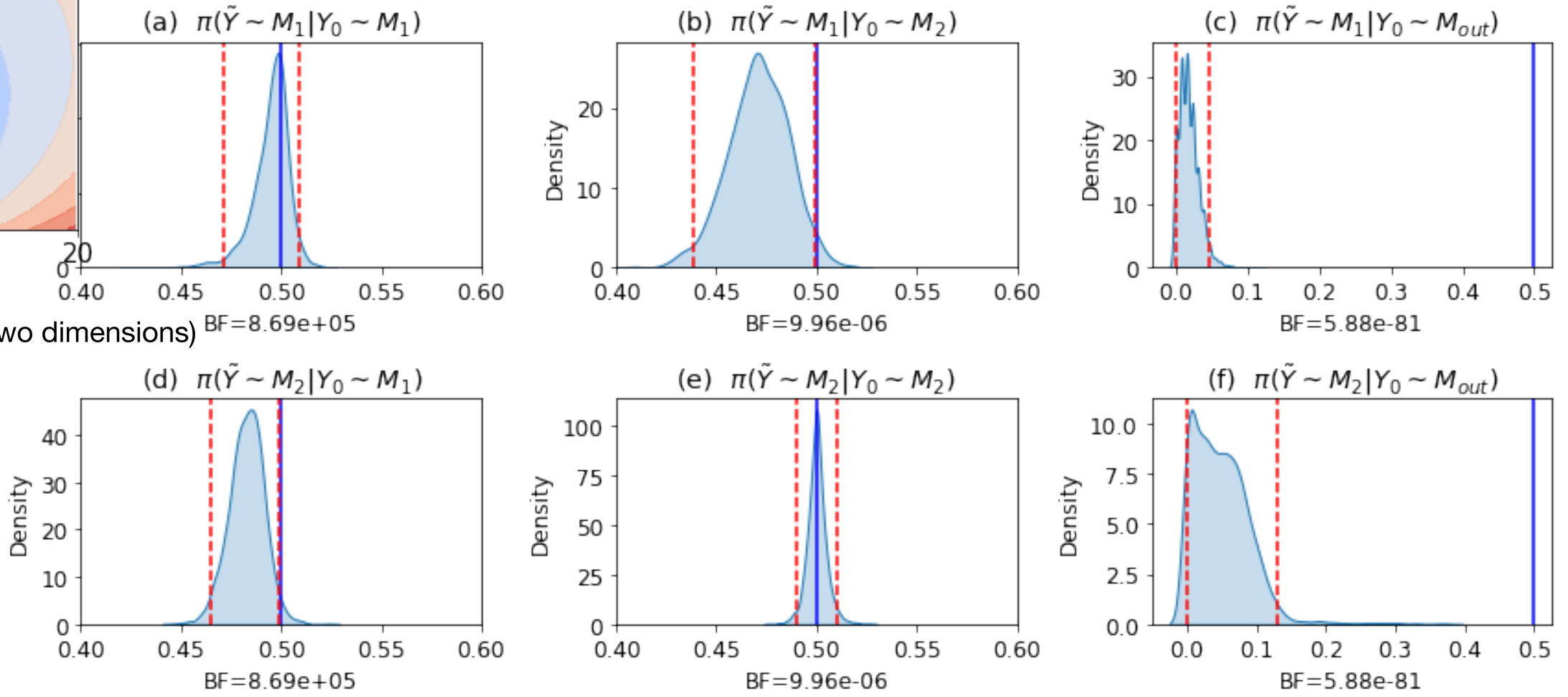
$\hat{d}$: as a one dimensional function

$$Z(\tilde{Y}, \hat{d}) = \frac{1}{n} \sum_{i=1}^{n} \hat{d}(\tilde{\tilde{Y}}_i) \approx \frac{1}{2}$$

Generative Goodness of Fit



(Visualized the first two dimensions)



Concluding Remarks

Summary

DeepBF

One time training, very cheap evaluation on the entire domain

Inferential and estimation consistency

Bayes factor variants

Can be used with the generative goodness of fit

Extension

Multiple competing models

Neural network and hyper parameter tuning $\longrightarrow$ Order invariance

More approachable deep learning: Small N and small computational capacity

Thank you!

For more theoretical/numerical results:
Please check out https://arxiv.org/abs/2312.05411

Background 2

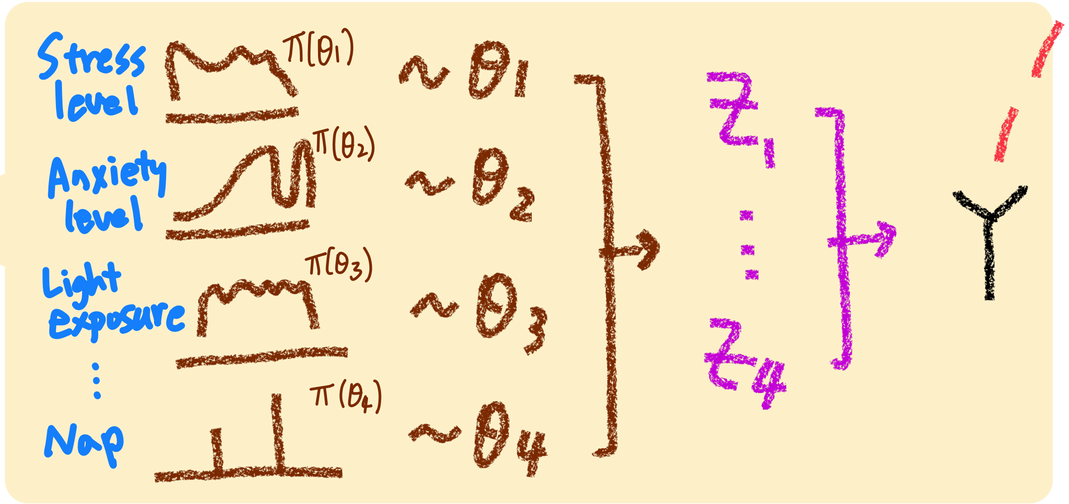
Likelihood Free Inference

Approximate Bayesian Computation (ABC)

Time needed to fall asleep

$d(\eta(Y), \eta(Y_0^{(n)})) \leq \epsilon$?!

Model 1



ABC Accepted $\approx$ Posterior samples.

Posterior sampling!

No sampling from posterior

Only forward sampling

Low acceptance rate

ABC Bayes Factor Estimator

$$\hat{BF}_{1,2}^{\text{ABC}}(Y_0^{(n)}) = \frac{\sum_{i:k^{(i)}=1}^{N} \mathbb{I}\{\rho(\eta(Y^{(i)}), \eta(Y_0^{(n)})) \leq \epsilon\}/\pi(M_1)}{\sum_{i:k^{(i)}=2}^{N} \mathbb{I}\{\rho(\eta(Y^{(i)}), \eta(Y_0^{(n)})) \leq \epsilon\}/\pi(M_2)}$$

because $BF_{1,2}(Y_0^{(n)}) = \dfrac{\pi(M_1 \mid Y_0^{(n)})/\pi(M_1)}{\pi(M_2 \mid Y_0^{(n)})/\pi(M_2)}$

Bias in ABC-based Bayes factor estimators Robert (11')

Summary Statistics

Sufficient for both $\pi_1(Y^{(n)} \mid \theta_1)$ and $\pi_2(Y^{(n)} \mid \theta_2)$

Sufficient for model selection?

i.e., sufficient for $\{(k, \pi_k(Y^{(n)} \mid \theta_k))\}_{k=1,2}$?