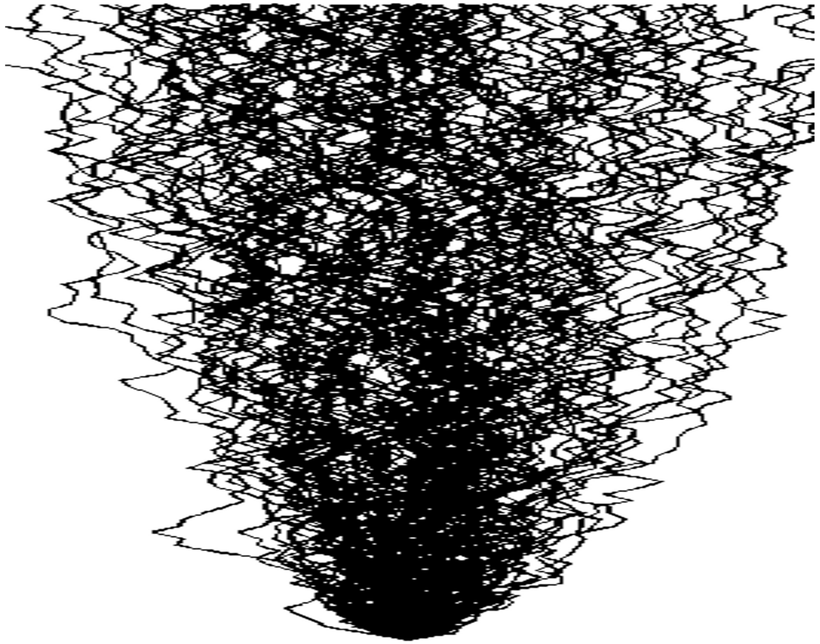
BROWN

Laboratory of Neural Computation and Cognition

TUTORIAL ON ABC EXTENSION TO HDDM

Alexander Fengler, An (Ian) Vo



ROBERT J. & NANCY D. CARNEY
INSTITUTE FOR BRAIN SCIENCE
BROWN UNIVERSITY

COGNITIVE LINGUISTIC & PSYCHOLOGICAL SCIENCES
BROWN UNIVERSITY

Outline:

1.(Minimal) ***Theory*** Part: Motivation and approach

2. Analyzing variations of the *Drift Diffusion Model* with ***HDDMnn***

We will talk a little bit about *likelihood-free* methods first…

Likelihood free ?

I have a model, don’t I automatically have a likelihood function?

It depends…

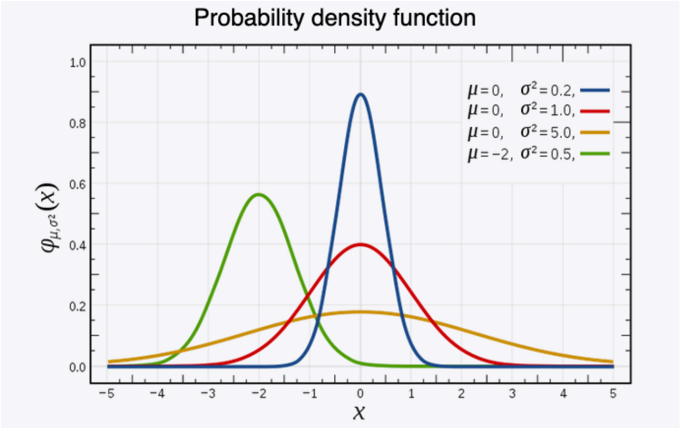
Likelihood

What is a ***likelihood function***?

x is our data point

The likelihood function (which is a probability density function / pdf / given the parameters) gives us the likelihood of observing **x** given some parameters $\theta = (\mu, \sigma^2)$ under a probability model (Here the **normal distribution**).

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Data Generating Process (Simulation Model)

What is a ***data generating process***?

A **data generating process** is an **algorithm / procedure** that allows us to generate data in accordance with a generative model.

Let’s try to generate data from a **normal distribution**,

$$\xcancel{f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}} \qquad ?$$

Data Generating Process (Simulation Model)

What is a ***data generating process***?

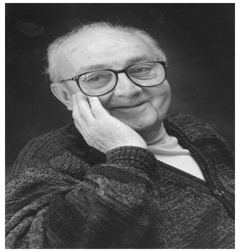
A **data generating process** is an **algorithm / procedure** that allows us to generate data in accordance with a model.

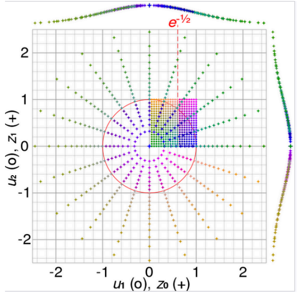
Let's try to generate data from a **normal distribution**,

Generate uniform random numbers → **Perform fancy transformations** → $x_1, x_2, x_3, \ldots, x_n \sim N(\mu, \sigma)$

Whole field of study

Famous algorithm





Finally some random numbers from a normal distribution

Likelihood vs. Data Generating Process

Not so apparent from using **R / Matlab / Python…**

```
> dnorm(x = 0, mean = 0, sd = 1)

> dpois(x = 1, lambda = 10)

> dgamma(x = 1, shape = 1, rate = 1)

> dbeta(x = 0.5, shape1 = 10, shape2 = 10)
        .
        .
        .
```

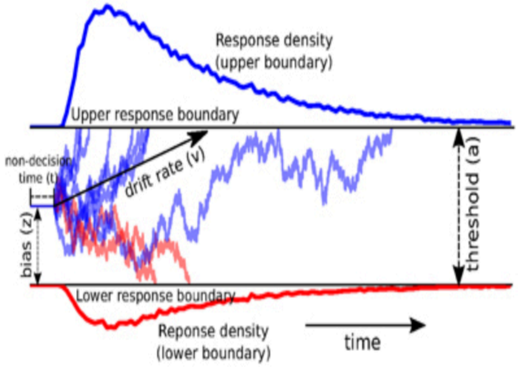
Give me the likelihood of x ….

```
> rnorm(n = 10, mean = 0, sd = 0)

> rpois(x = 1, lambda = 10)

> rgamma(x = 1, shape = 1, rate = 1)

> rbeta(x = 0.5, shape1 = 10, shape2 = 10)
        .
        .
        .
```

Give me a simulated datapoint x ….

Q ?

So how about DDMs?



Gives us ***samples*** from the generative process of the model

Likelihood

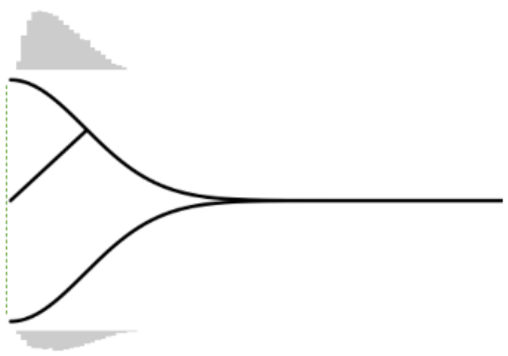
$$f(t|\mu, a, w) = \frac{\pi}{a^2} \exp\left(-\mu a w - \frac{\mu^2 t}{2}\right) \sum_{k=1}^{\infty} k \exp\left(-\frac{k^2 \pi^2 t}{2a^2}\right) sin(k\pi w)$$

Phew… nasty but get's the job done!

DGP

```
run_ddm = function(mu = 1, a = 1, w = 0.5, dt = 0.01){
  particle = a * w
  t = 0
  while (particle < a && particle > 0){
    particle = particle + (v * dt) + sqrt(dt) * rnorm(n = 1,
                                                       mu = 0,
                                                       sd = 1)
    t = t + dt
  }
  return(t)
}
```

So how about slight variations of DDMs?



Still easy to get **samples** ….

Likelihood

$$f(t|\mu, a, w) = \frac{\pi}{a^2} \exp\left(-\mu a w - \frac{\mu^2 t}{2}\right) \sum_{k=1}^{\infty} k \exp\left(-\frac{k^2 \pi^2 t}{2a^2}\right) sin(k\pi w)$$

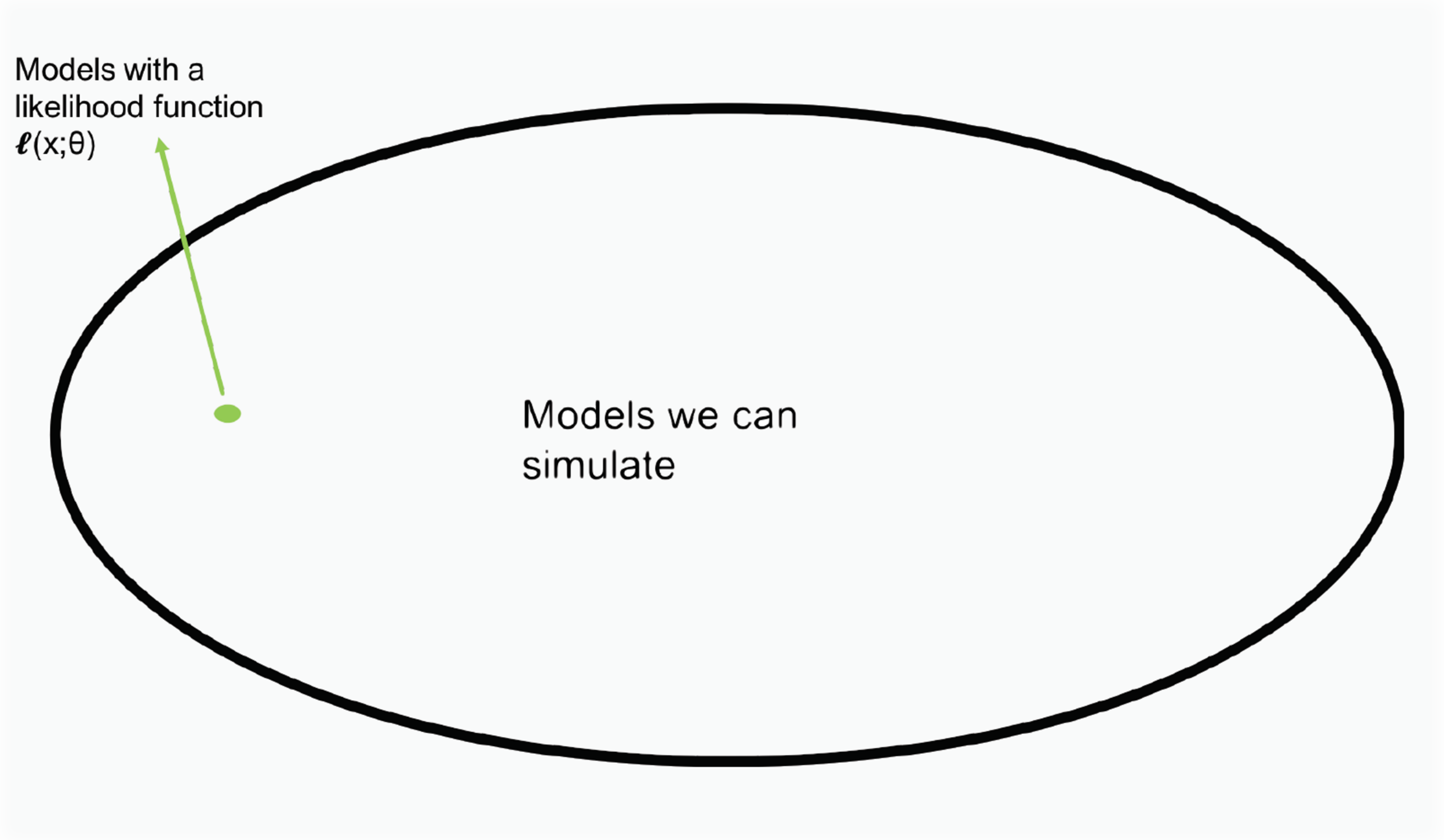
?

Likelihood of crossing the ***lower boundary*** *at time* **t**, given parameters $(\alpha, \beta, \mu, a, w)$

DGP

```
run_ddm = function(mu = 1, a = 1, w = 0.5, dt = 0.01){
  particle = a * w
  t = 0
  while (particle < a && particle > 0){
    particle = particle + (v * dt) + sqrt(dt) * rnorm(n = 1,
                                                        mu = 0,
                                                        sd = 1)
    t = t + dt
  }
  return(t)
}
```

An more accurate view of the world of models…



Ok lets say I have no Likelihood function… Big deal?

$$\pi(\theta|X) \propto P(X|\theta)\pi(\theta)$$

Well… at first sight it's pretty much the ***end of your career as a Bayesian***.

No Likelihood, no Bayes rule!

Domain of: 'Likelihood free inference methods’

Active Area of Research with many potential approaches.

We will show and use **one particular method** to **extend our ability** to perform Bayesian Inference on such *likelihood-free* **extensions** of the DDM model framework.

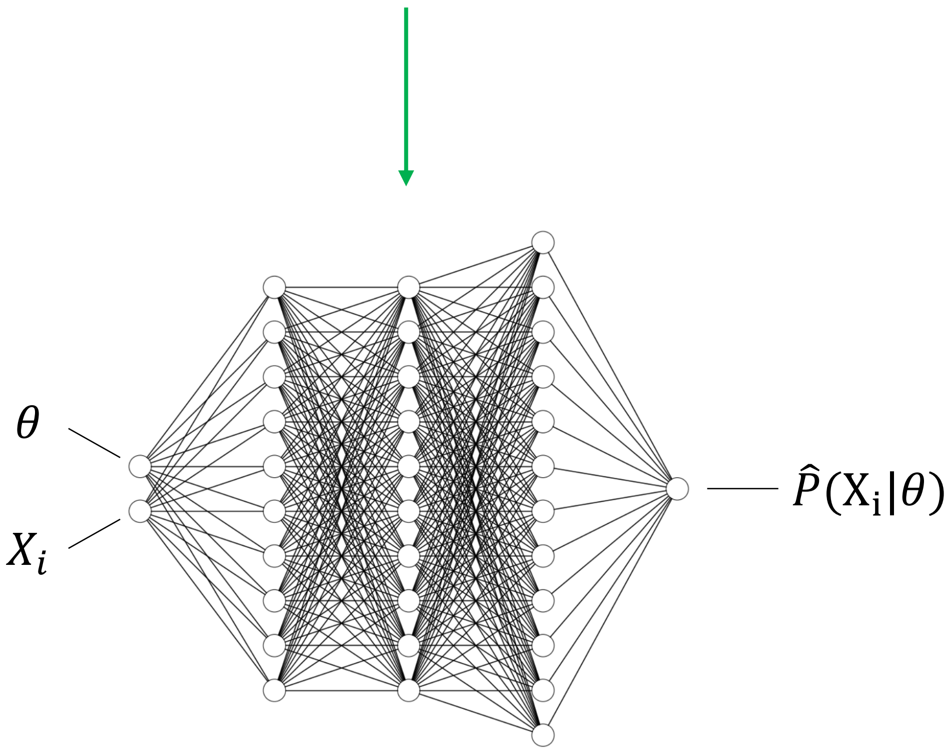
How are we going about it?

We go from: $$\pi(\theta|X) \propto P(X|\theta)\pi(\theta)$$

To: $$\pi(\theta|X) \propto \hat{P}(X|\theta)\pi(\theta)$$

Approximate Likelihood

What is $\hat{P}(X|\theta)$ here ?

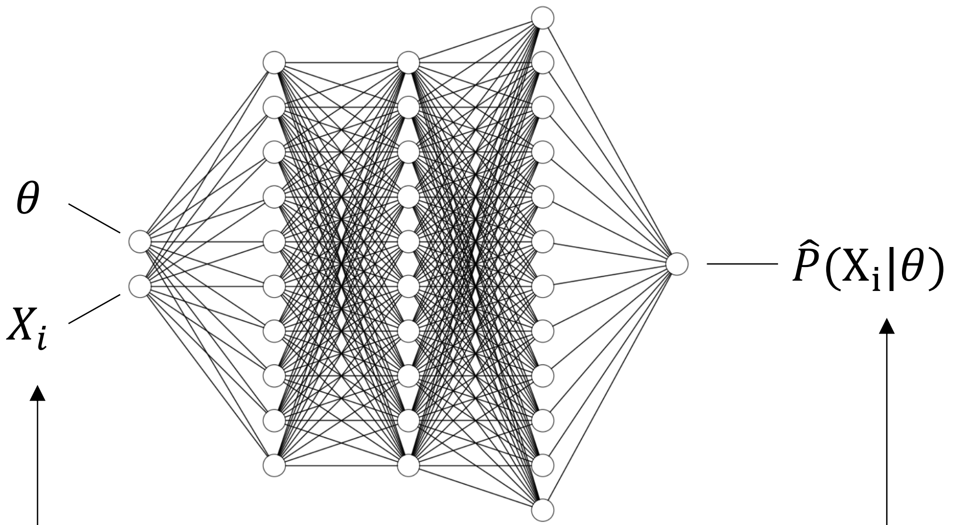


It's a simple neural network!

(Multilayered Perceptron)

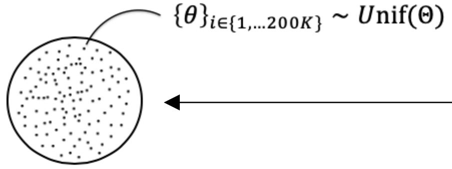
This network approximates the likelihood function that was unknown a priori!

How do we get $\hat{P}(X|\theta)$?

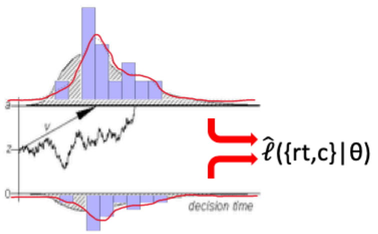


TRAINING:

1. Sample parameters from relevant parameter space



2. For each θ_i , generate 10000 simulations $\{\mathrm{rt,c}\} \sim DGP(\theta)$ and compute a kde based *synthetic likelihood function* $\hat{\ell}(\{\mathrm{rt,c}\}|\theta)$.

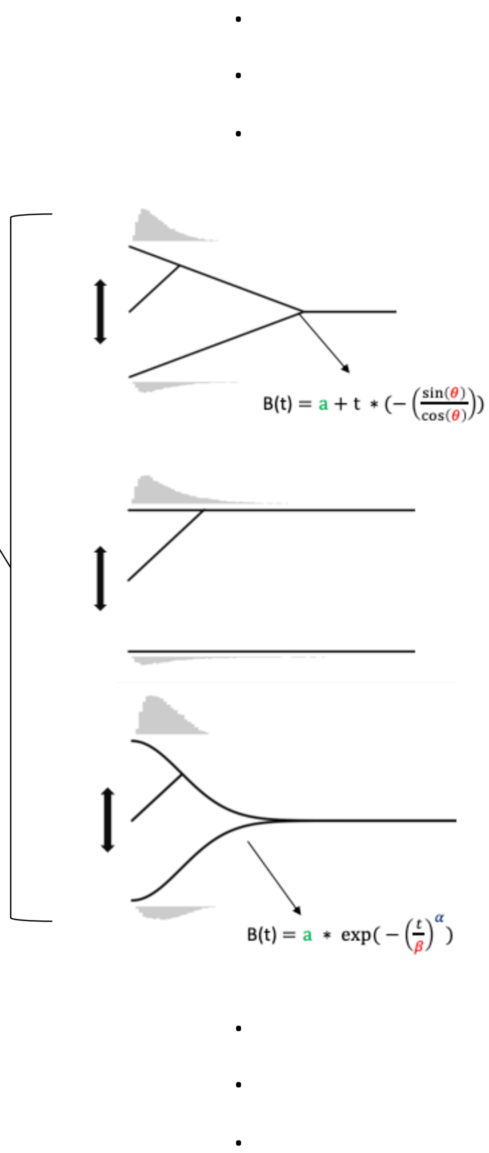


3. Generate training data from *syntetic likelihood functions.*

$\theta \sim Unif(\{\theta\}_i)$

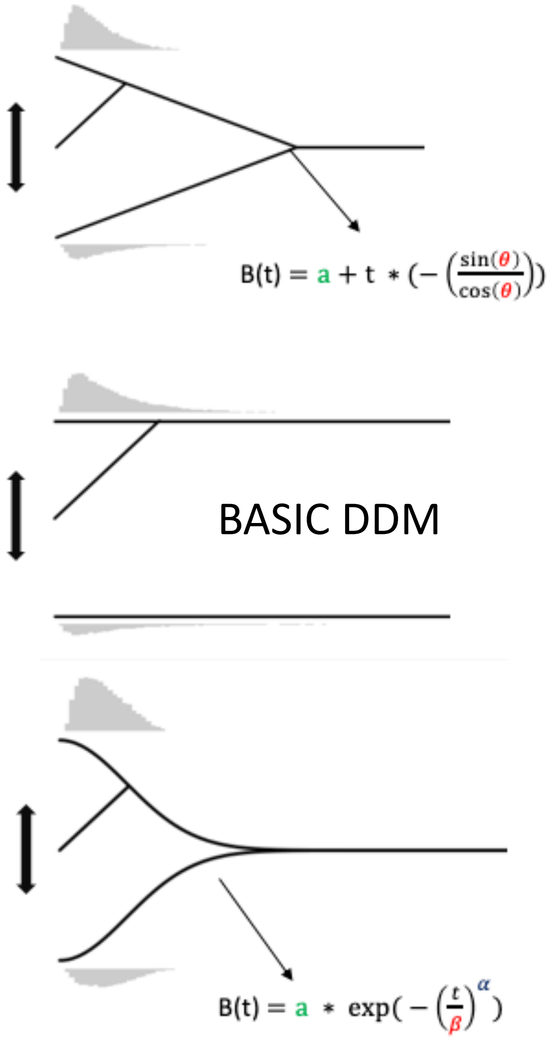
- p = 0.8 $\{\mathrm{rt,c}\} \sim \hat{\ell}(\{\mathrm{rt,c}\}|\theta)$
- p = 0.1 $\{\mathrm{rt,c}\} \sim Unif([-1,0]) \times Unif(\{-1,1\})$
- p = 0.1 $\{\mathrm{rt,c}\} \sim Unif([5,20]) \times Unif(\{-1,1\})$

id	θ_1	...	θ_n	rt	c
1	0.5	...	-0.75	1.76	-1
⋮	⋮	⋮	⋮	⋮	⋮
200m	0.6	...	-1.0	1.89	1



We will focus on exactly these models later ….

ANGLE MODEL



WEIBULL MODEL

Q ?

Now on to using it!

If your tutorial notebook is not ready, follow the **instructions** here:

https://github.com/AlexanderFengler/hddmnn_tutorial

We will give it 5 minutes to let everyone catch up. Post your questions in the chat or raise your hand.