



# QUANTIFYING HIDDEN VARIABLES USING BAYESIAN INFERENCE

OMER  
YUKSEL

[omer.yuksel@imc.com](mailto:omer.yuksel@imc.com)

# Introduction

---

- Performance Analyst at IMC
- Technology-driven trading
  - Amsterdam, Chicago and Sydney
- Networks and automation



[www.imc.com](http://www.imc.com)

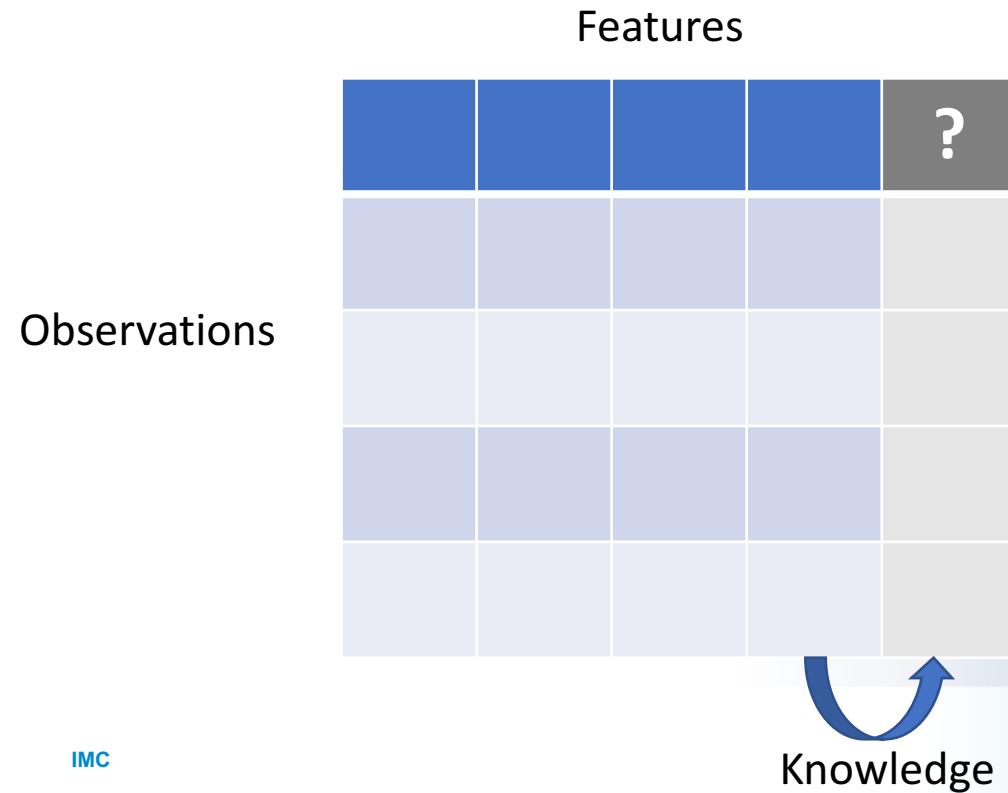
# Problem: estimating hidden variables

---

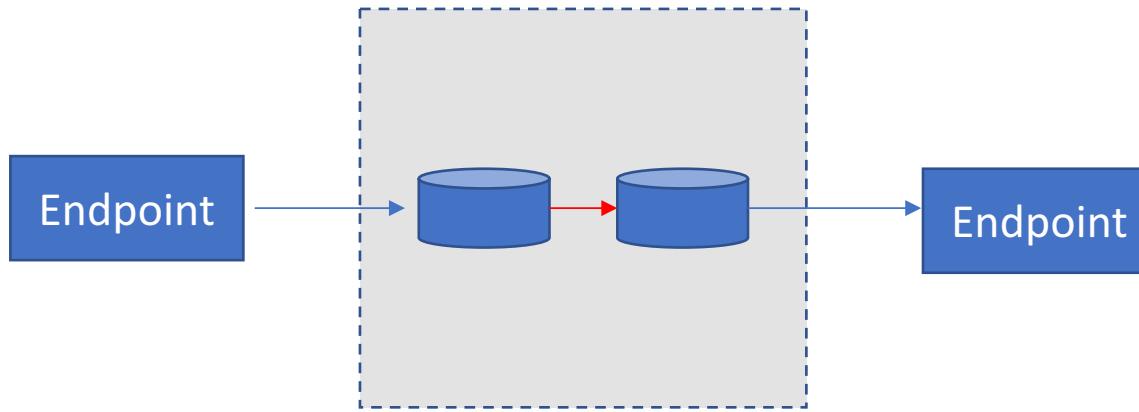
	Features				
Observations					?



# Problem: estimating hidden variables

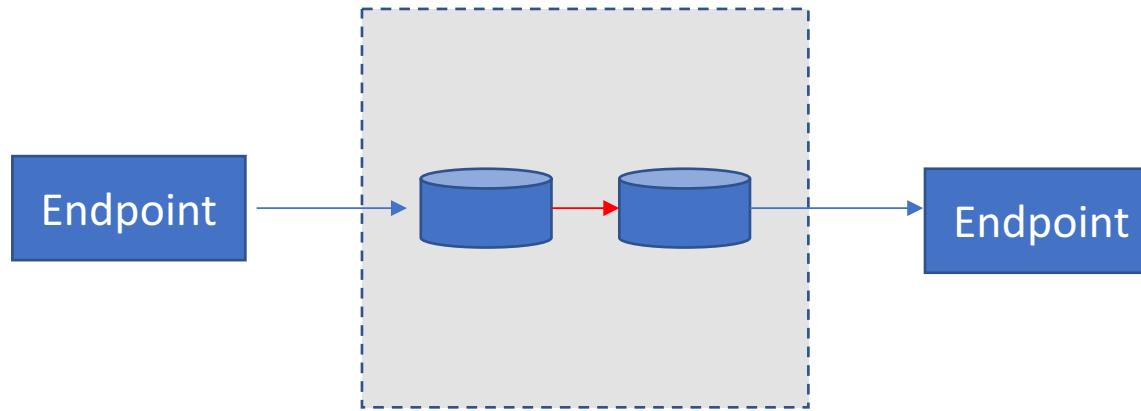


# Communication over unreliable channel



- Lacking end-to-end control
- Possibility of drops
  - Results in retransmission unknown to us

# Communication over unreliable channel

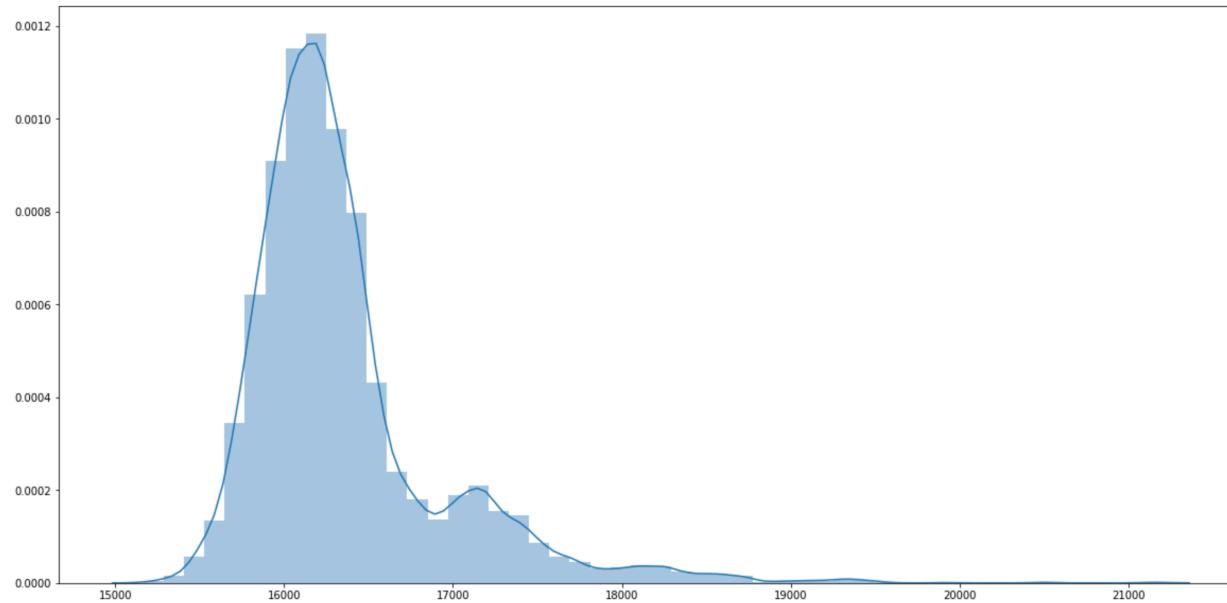


- **Hidden variables:** #drops,  $\text{prob}(\text{drop})$
- **Data:** one-way delays (OWD)
- **Knowledge:** Distribution of one-way delay between

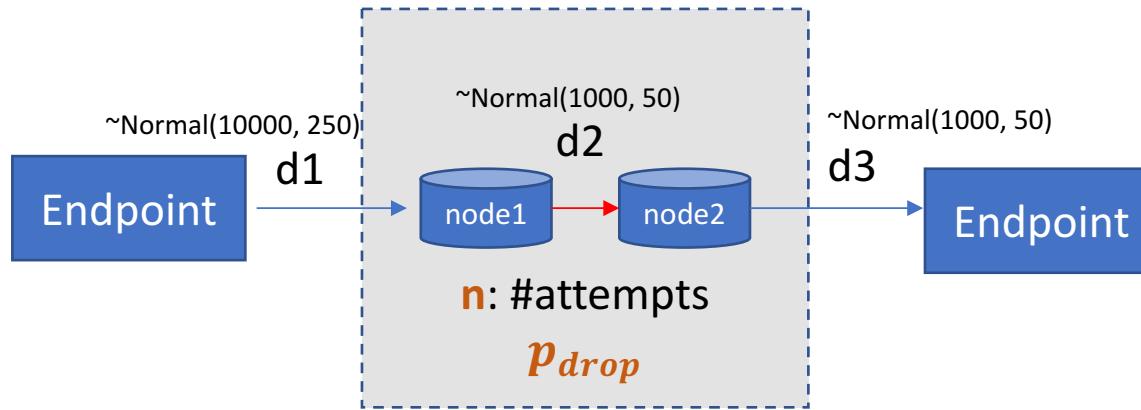
# Data

---

## Distribution of OWD (microseconds)



# Knowledge



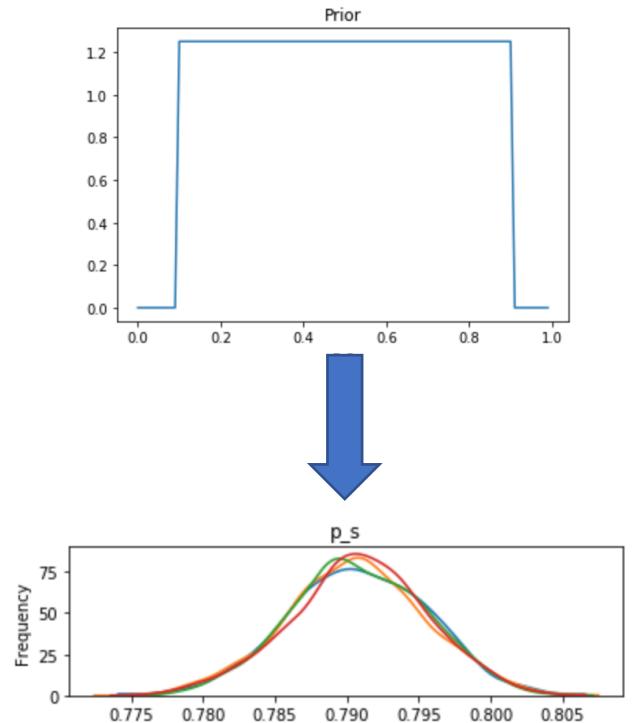
$$OWD = d_1 + n \times d_2 + d_3$$

$p_{drop}$  between 10%-40%

$n \sim \text{Geometric}(P_s = 1 - P_{drop})$

# Bayesian Inference

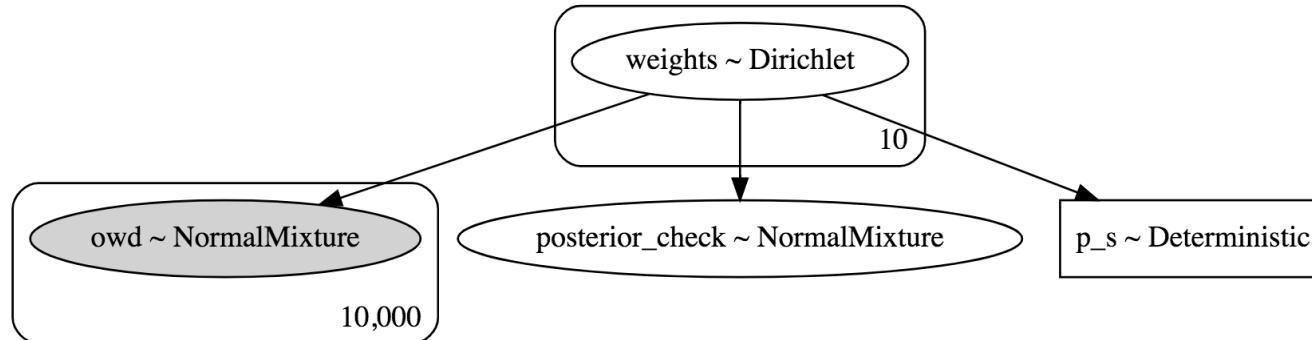
- Updating “belief” with evidence
- Parameters as random variables
  - Constraints in search space
- Requires approximation methods
  - PyMC3, Stan, Edward



Salimans, Tim, Diederik Kingma, and Max Welling. "Markov chain monte carlo and variational inference: Bridging the gap." 2015.

# Model using PyMC3

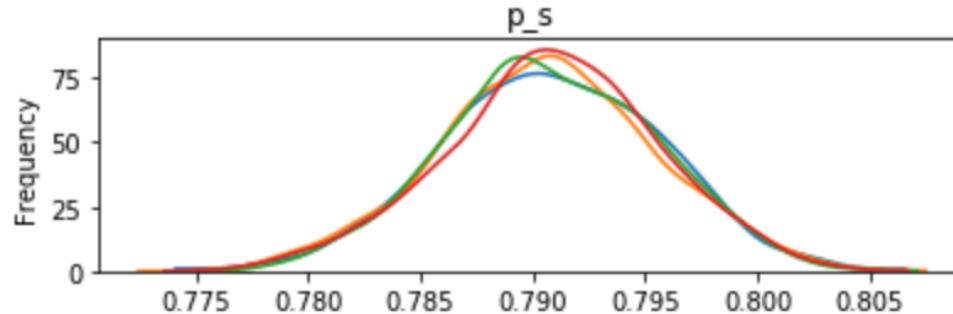
```
weights = pm.Dirichlet('weights', np.ones(max_tries))
owd = pm.NormalMixture(name='owd', mu=MU, sd=STD, w=weights, observed=data)
p_success = pm.Deterministic(name='p_s', var=weights[0])
```



# Results

---

	mean	sd	mc_error	hpdi_2.5	hpdi_97.5	n_eff	Rhat
p_s	0.790558	0.004911	0.000065	0.781044	0.800114	5084.1630 69	1.000097



# Summary

---



- Bridging the gap between knowledge and data
- Challenges
  - Debugging
  - Problem formulation
- PyMC: <https://docs.pymc.io/>

**Slides/notebook/data:** <https://github.com/omersyuksel/PyConDE2018>

**QUESTIONS?**

## Markov-Chain Monte Carlo(MCMC)

### Variables:

**Random**: sampled from a distribution

**Deterministic**: function of other variables

**Observed**: data

### Finding parameters in MCMC:

Trace

Posterior sample

# Posterior samples

---

