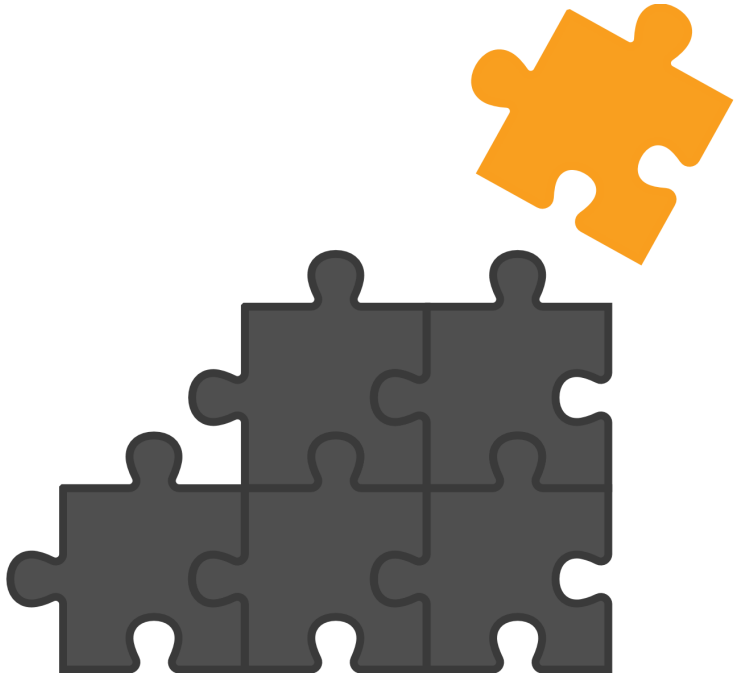


solarwinds

2020 Tech Summit – Grow Together II

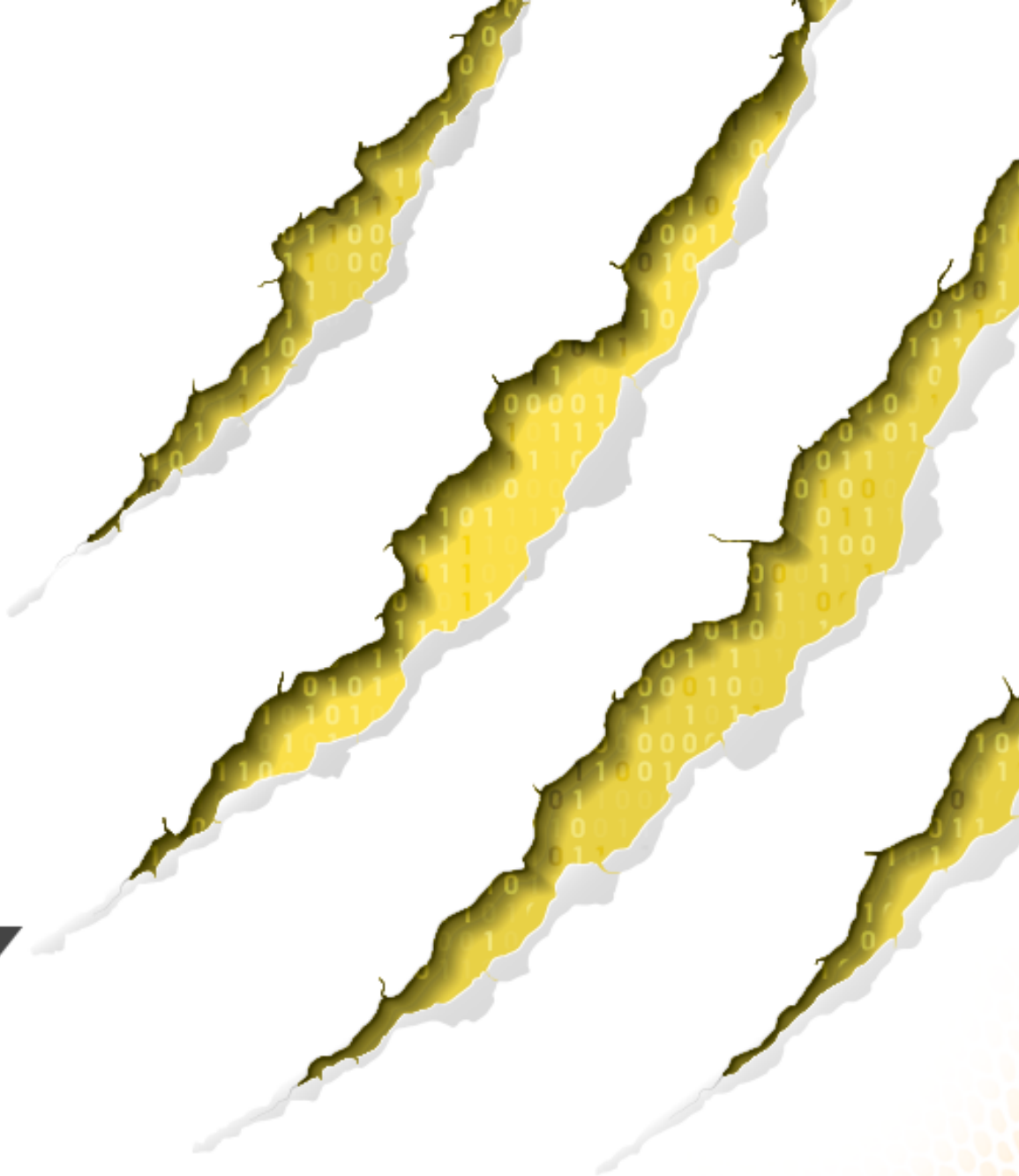
Dec-2019 Brno, CZ

UNCAGE

your inner data scientist animal
with probabilistic programming
and **SolarWinds®**

TechSummit Innovate Day

12 December | Brno



Introduction to probabilistic programming

- The probabilistic perspective
- Bayesian inference
- Markov-chain Monte Carlo (MCMC)
- Probabilistic programming
- Probabilistic time series demo

Alex Chan

Data Scientist

SolarWinds MSP @ Edinburgh

linkedin.com/in/aybchan

github.com/aybchan

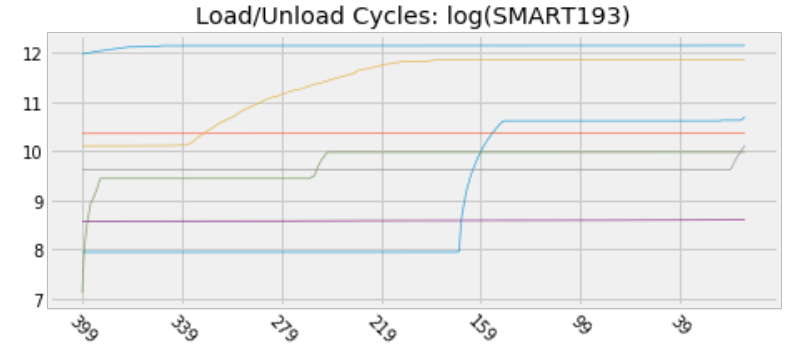
About me

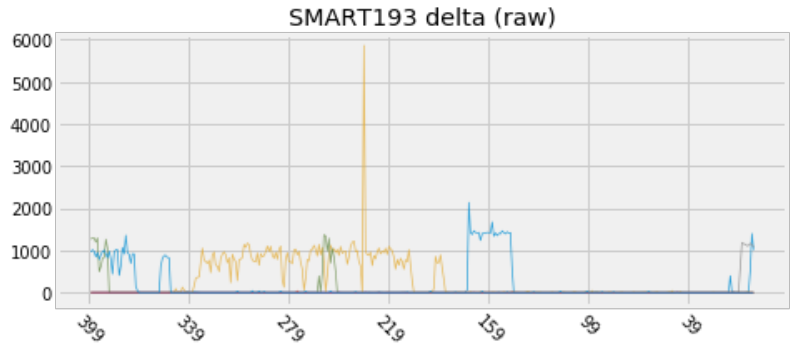
- Background in CS and AI
- Data Scientist in Edinburgh
- SolarWinds data science:
- Hard drive failure prediction

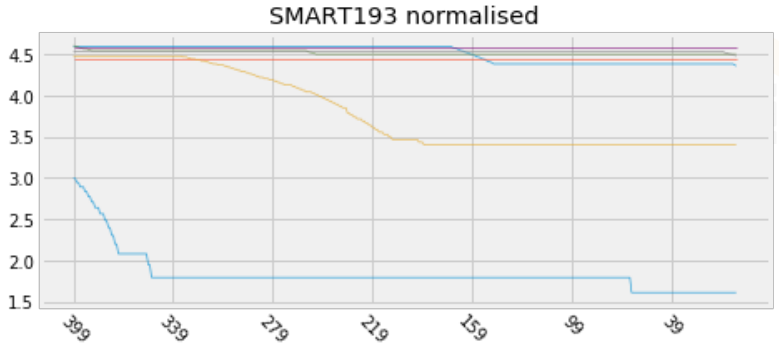
Alex Chan

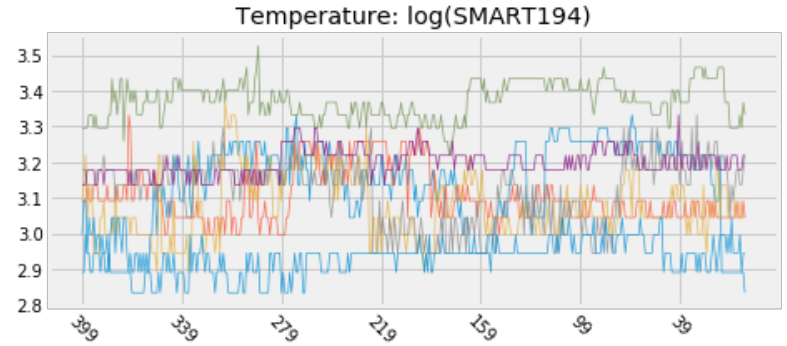
Data Scientist
SolarWinds MSP @ Edinburgh

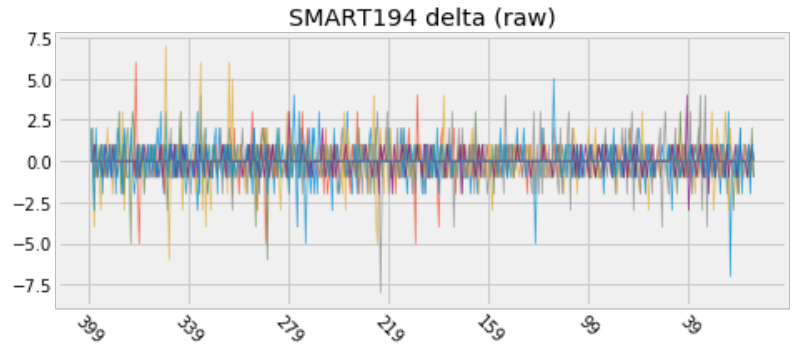
linkedin.com/in/aybchan
github.com/aybchan

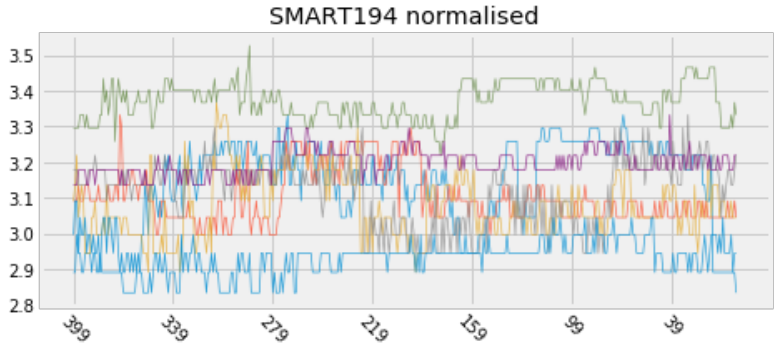


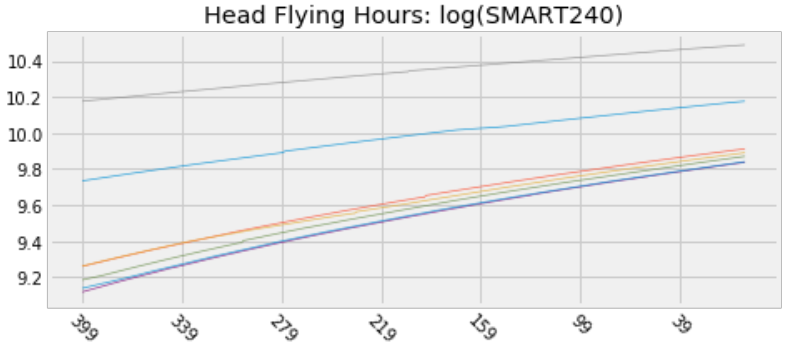


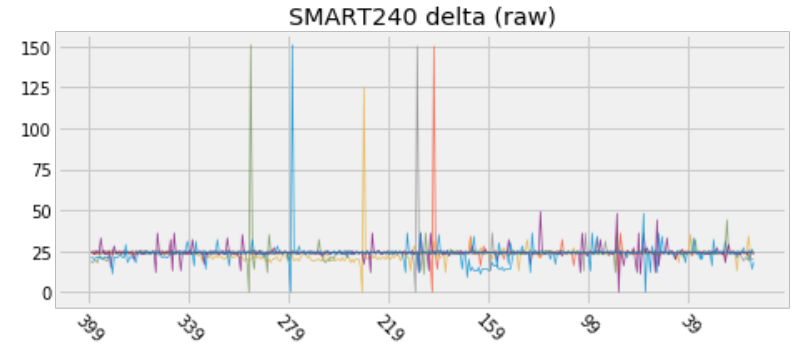


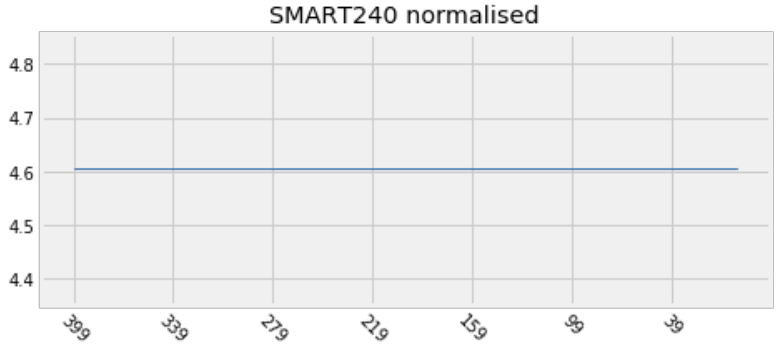


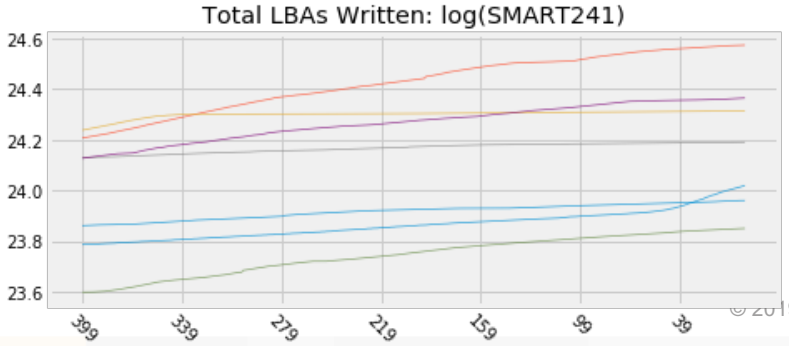


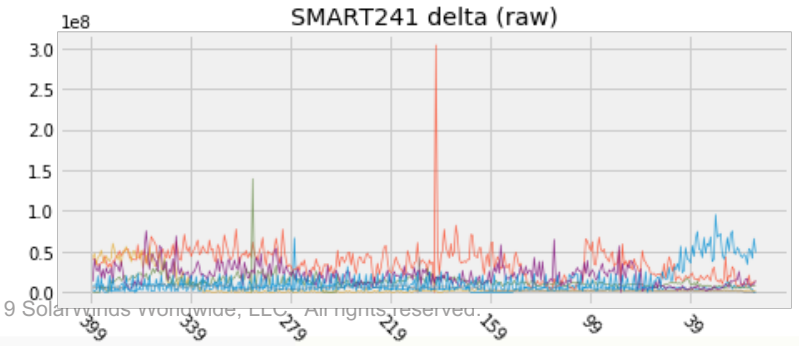


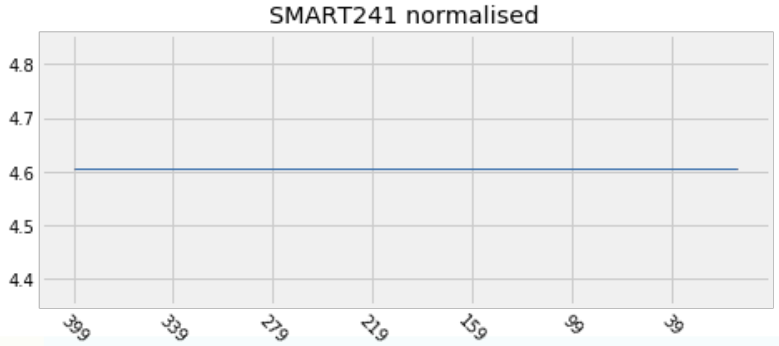












About me

- Background in CS and AI
- Data Scientist in Edinburgh
- SolarWinds data science:
  - Hard drive failure prediction
  - Spam detection
  - NSFW image recognition

Alex Chan

Data Scientist
SolarWinds MSP @ Edinburgh

linkedin.com/in/aybchan
github.com/aybchan

Workshop materials

- github.com/solarwinds/probprog-workshop
- Follow README to build and run the Docker image
- We will use Jupyter notebooks in the browser for demos

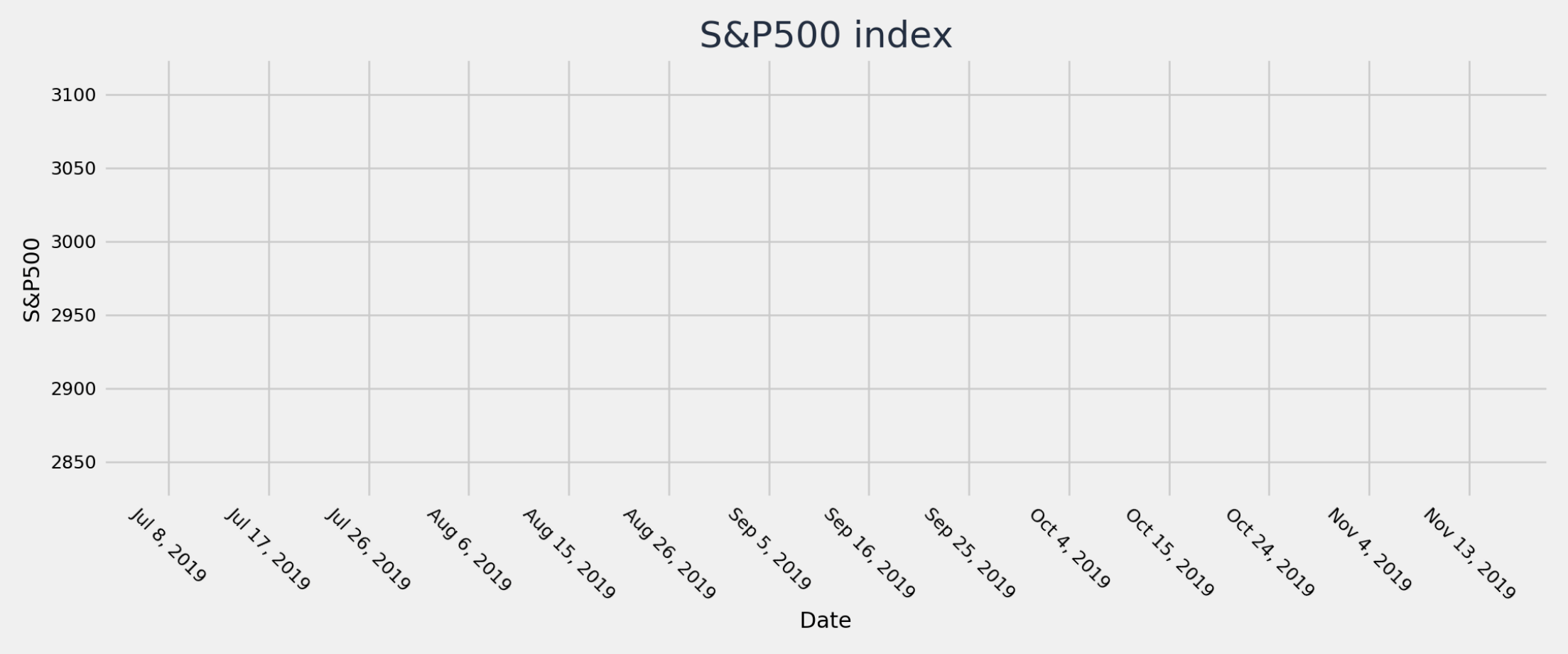
Alex Chan

Data Scientist
SolarWinds MSP @ Edinburgh

linkedin.com/in/aybchan
github.com/aybchan

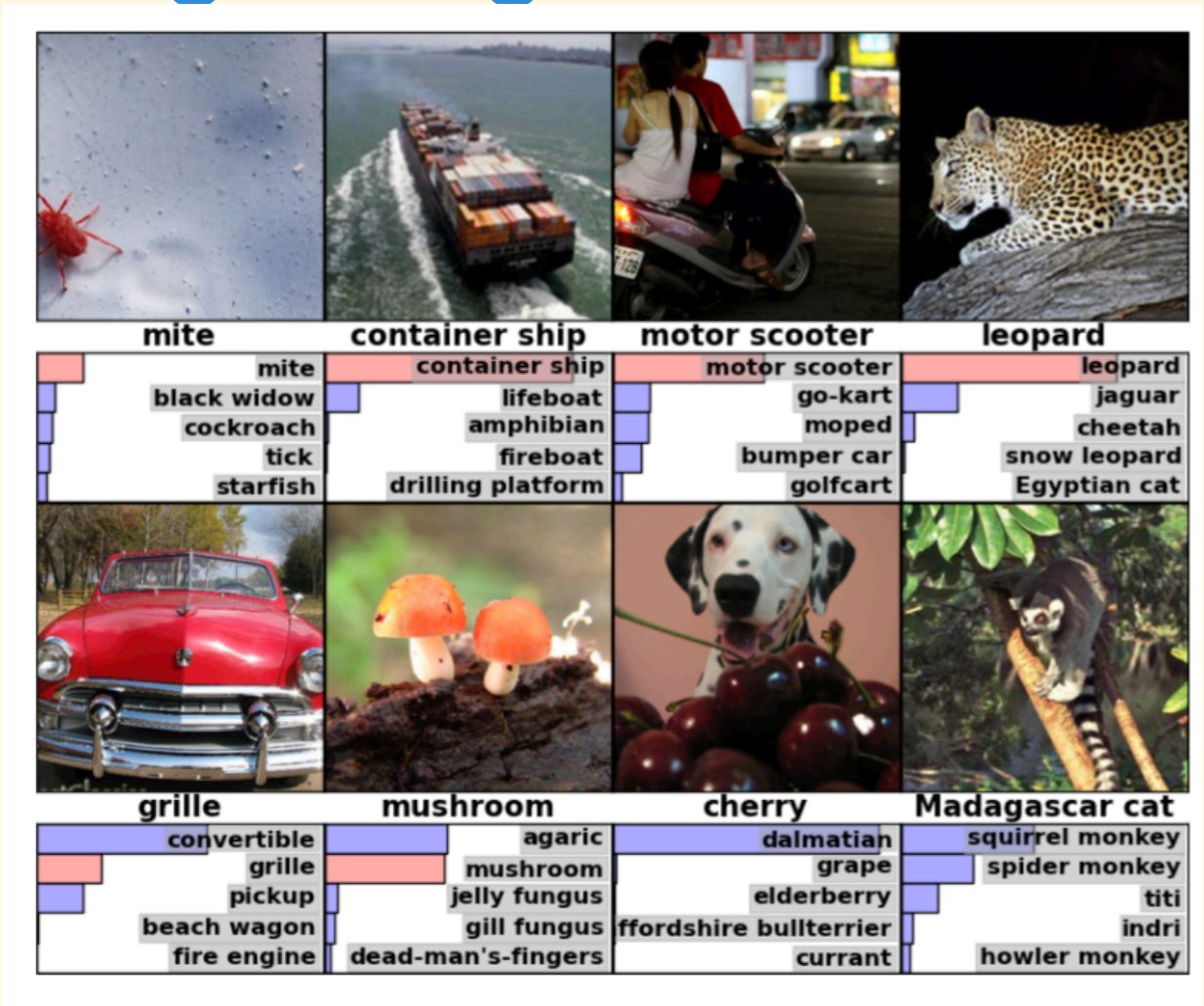
The machine learning problem

Example 1: Time series forecasting



Given the past 4 months of data, predict the value at the next time steps

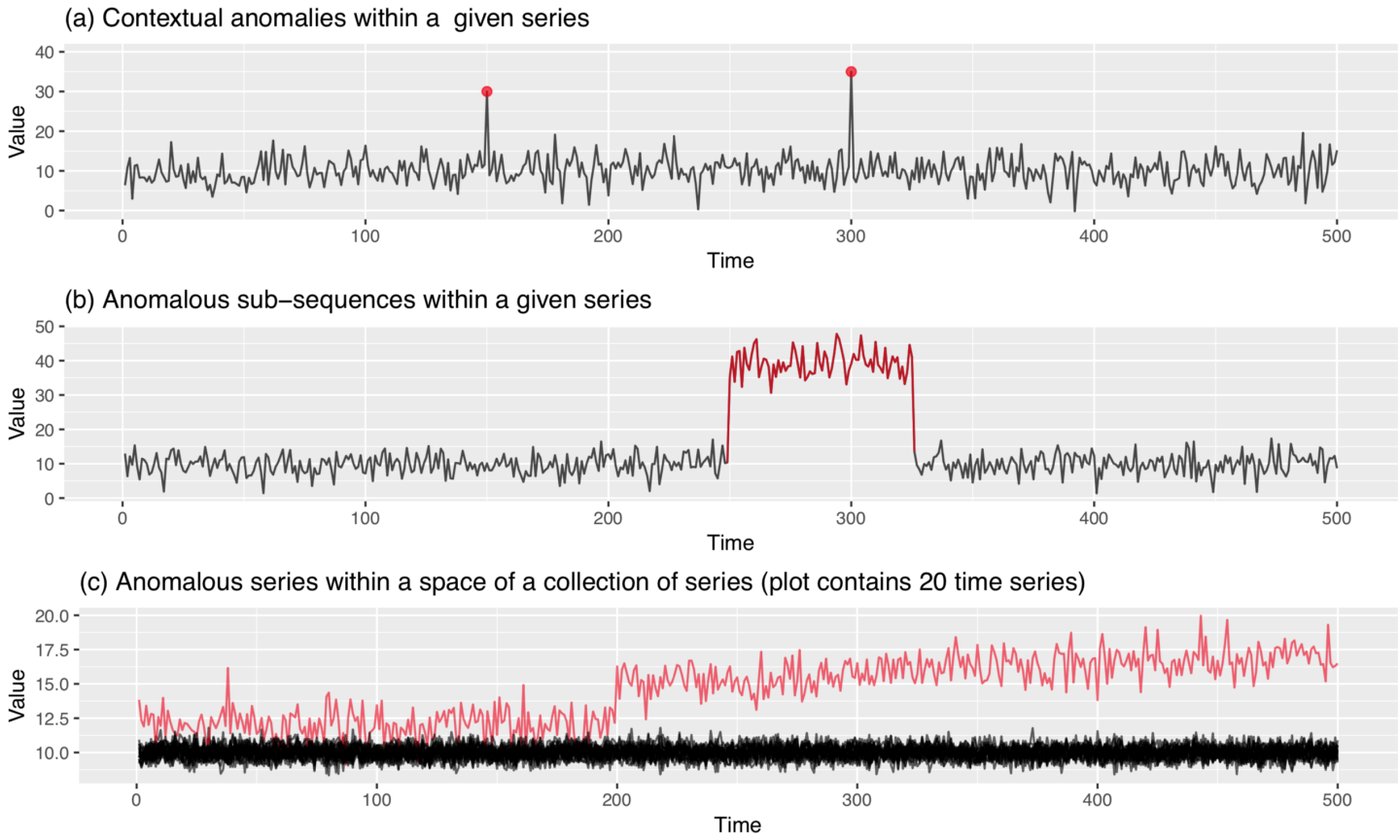
Example 2: Image recognition



Source: ImageNet Classification with Deep Convolutional Neural Networks (2012)
https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Given an image, classify its content with a label

Example 3: Anomaly detection

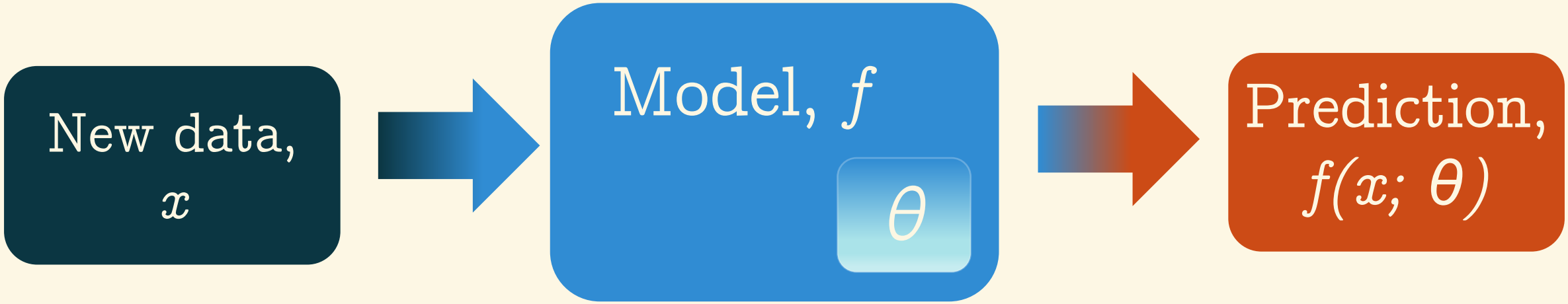


Source: Anomaly detection in streaming nonstationary temporal data (2018)
https://robjhyndman.com/papers/oddstream.pdf

Given a stream of data (e.g. network traffic), detect any unusual behavior

What is machine learning?

- We want a function that can give predictions given new **data**

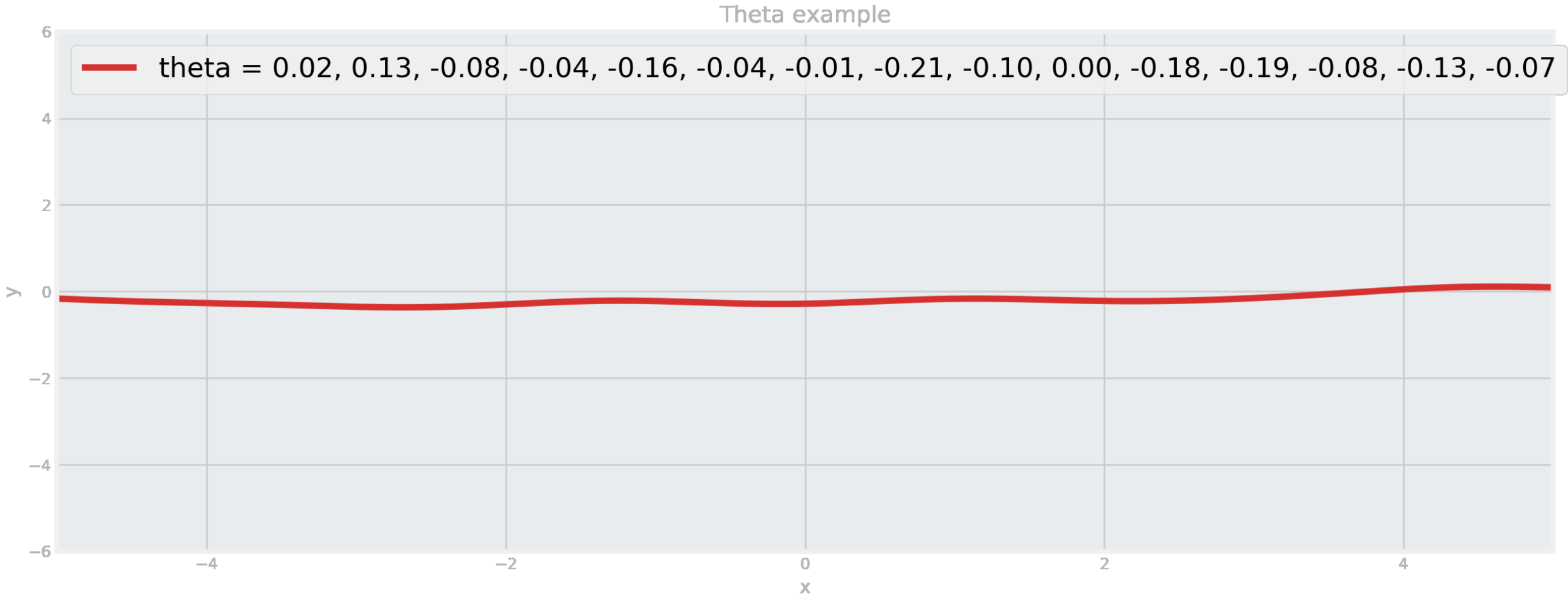


- Machine learning is about finding a good fit for the parameters $\boldsymbol{\theta}$ from the data

Parameter tuning

What is θ?

What is θ?

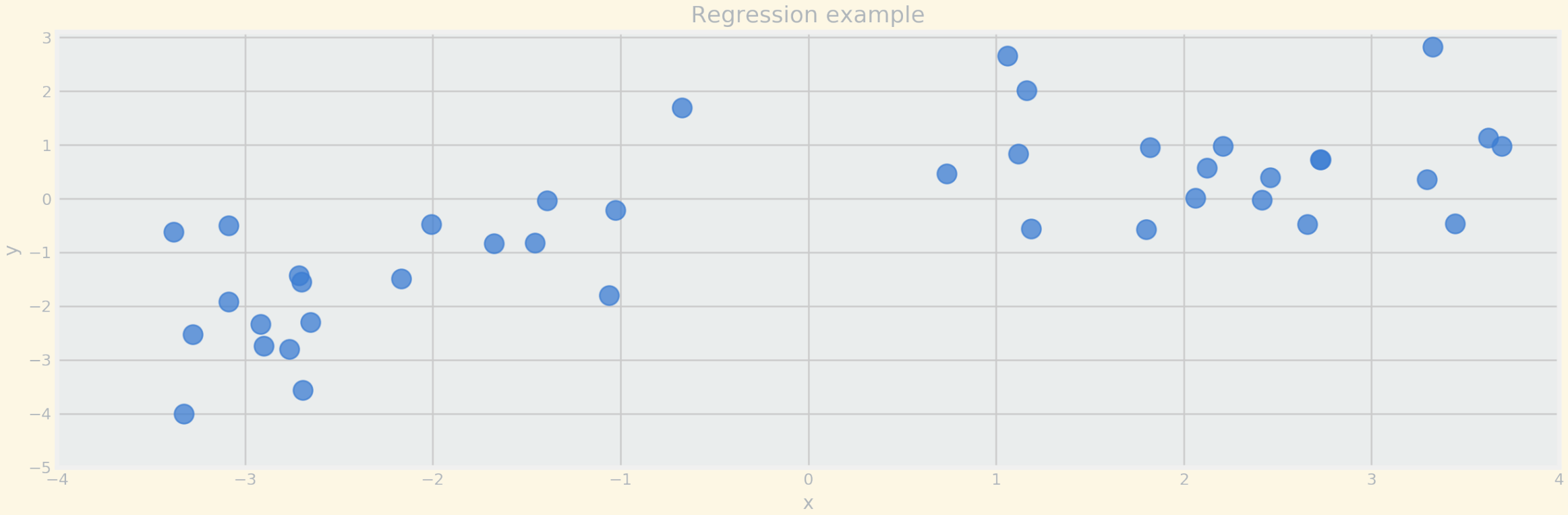


The probabilistic approach

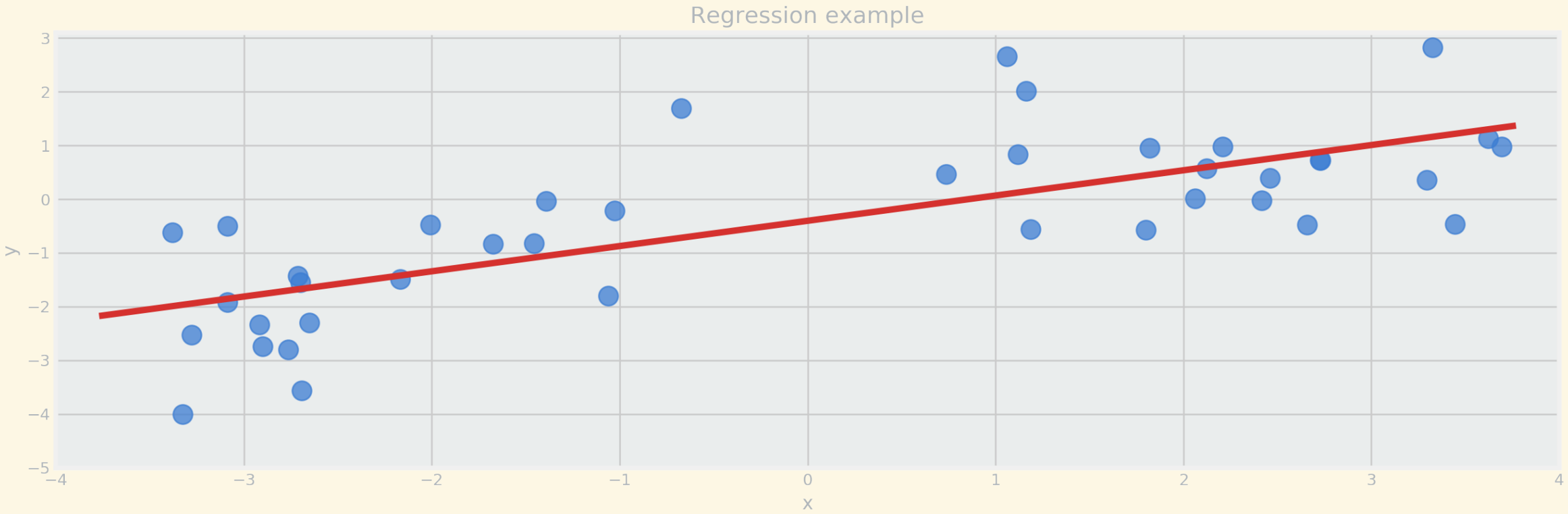
The probabilistic approach

- Use probability theory and reasoning throughout
- Probability distributions express beliefs about quantities
- Principled handling of uncertainty
- *Let data speak for itself*

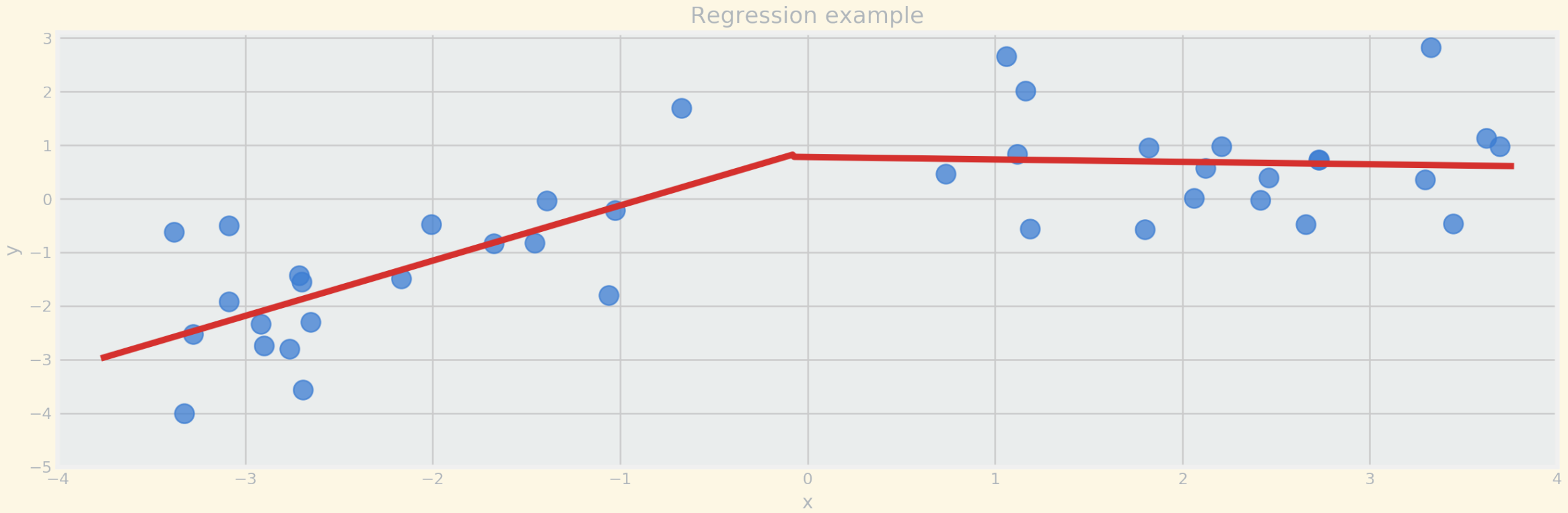
Modelling some data



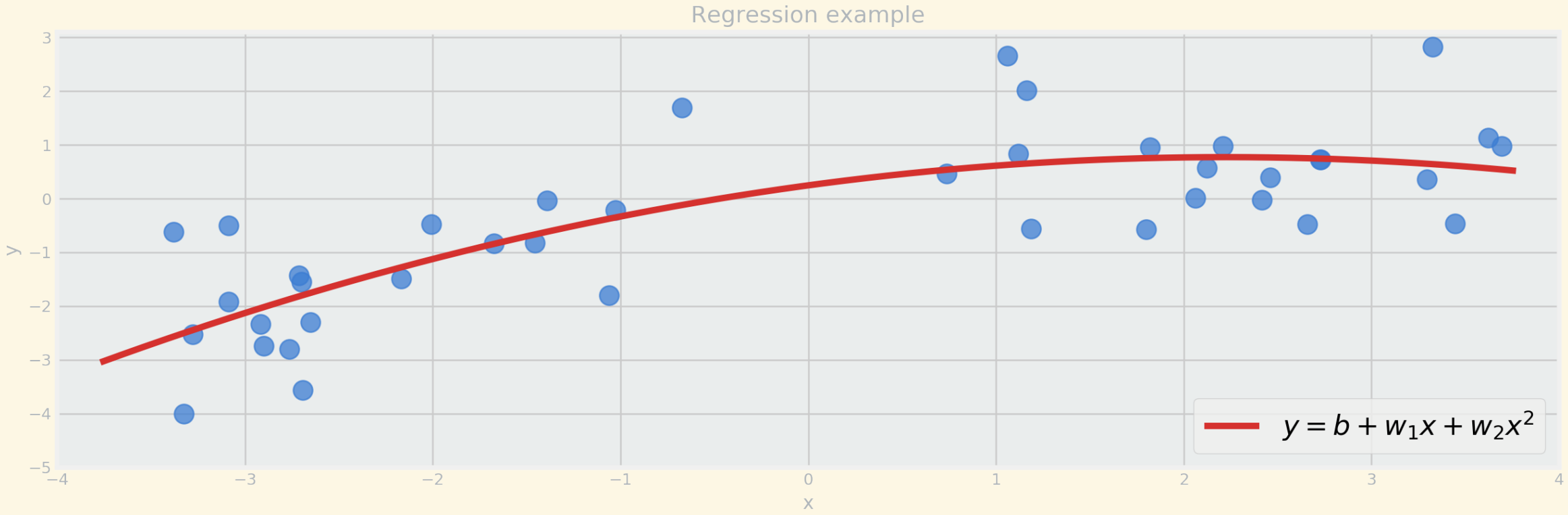
Example: Straight line



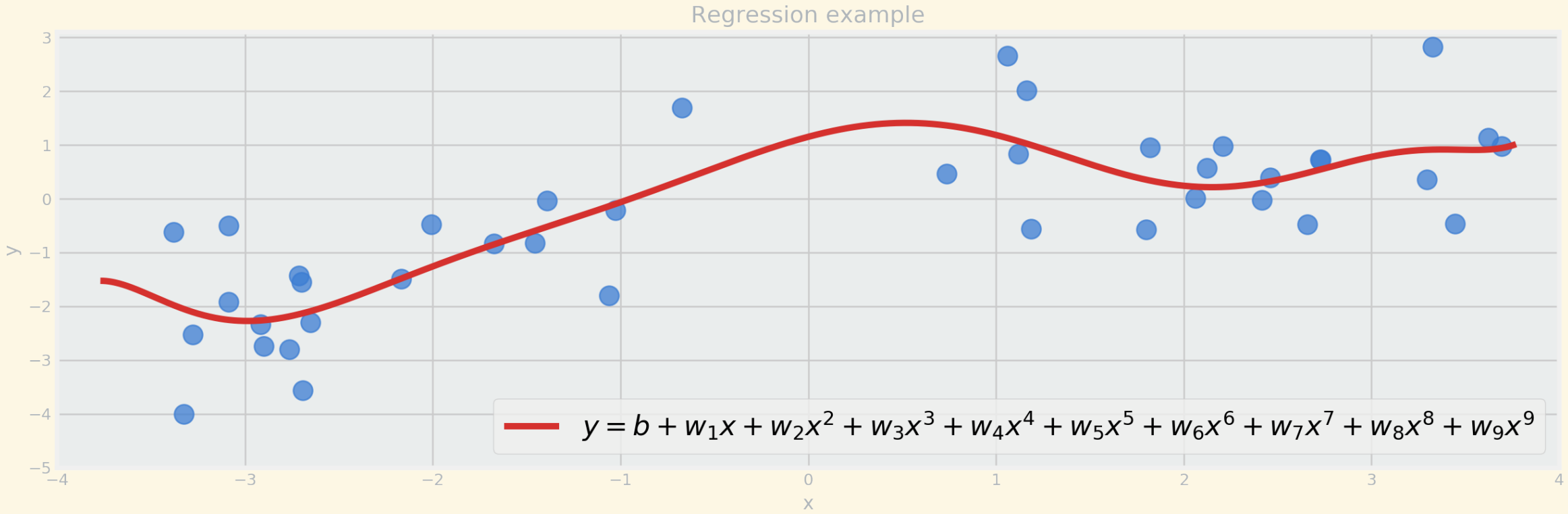
Example: Piecewise linear



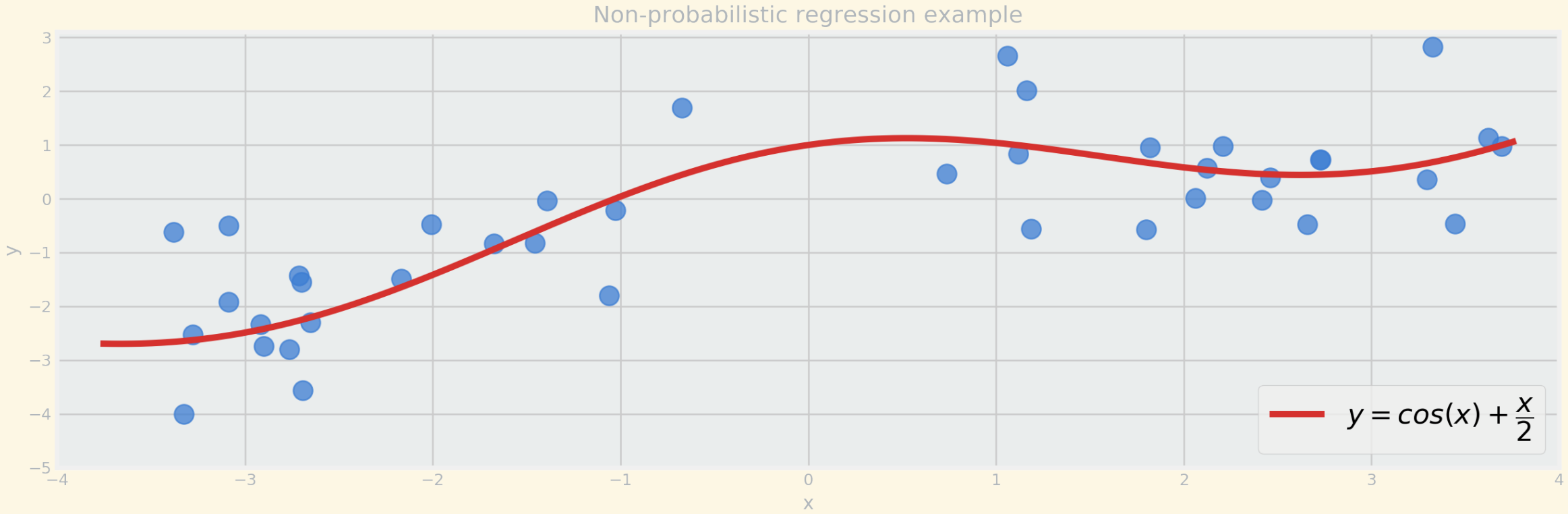
Example: Degree 2 polynomial (quadratic)



Example: Degree 9 polynomial



Example: The true function



- Many plausible models to explain this data
- Is it reasonable to pick just one of them?

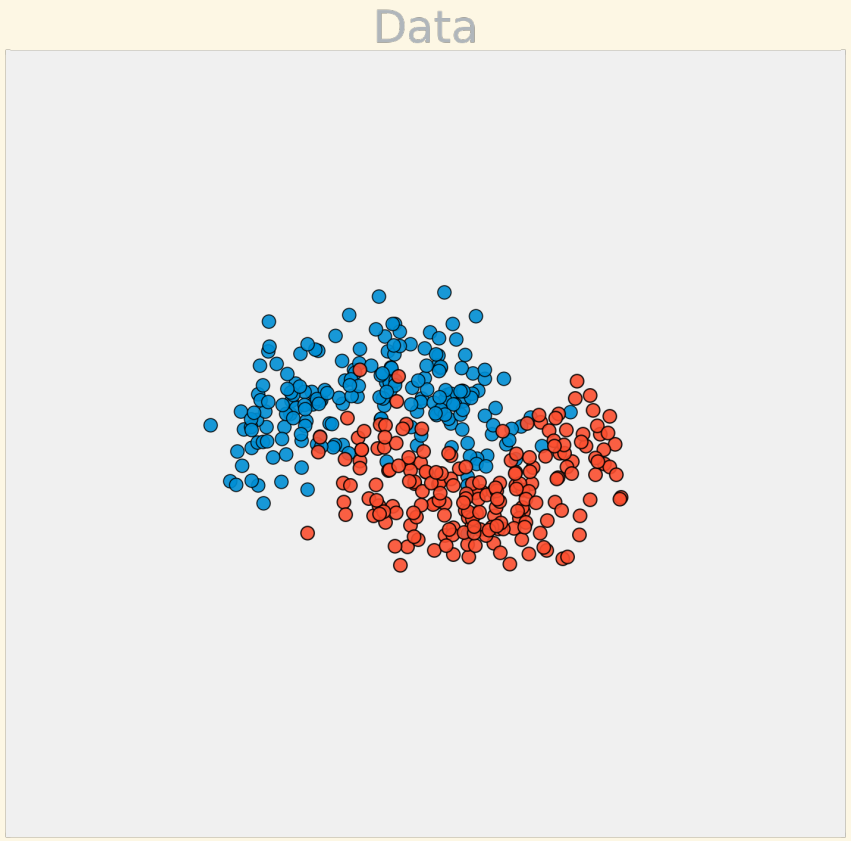
The probabilistic perspective on ML

- Usually many ways to explain the same data

  Many plausible models, M

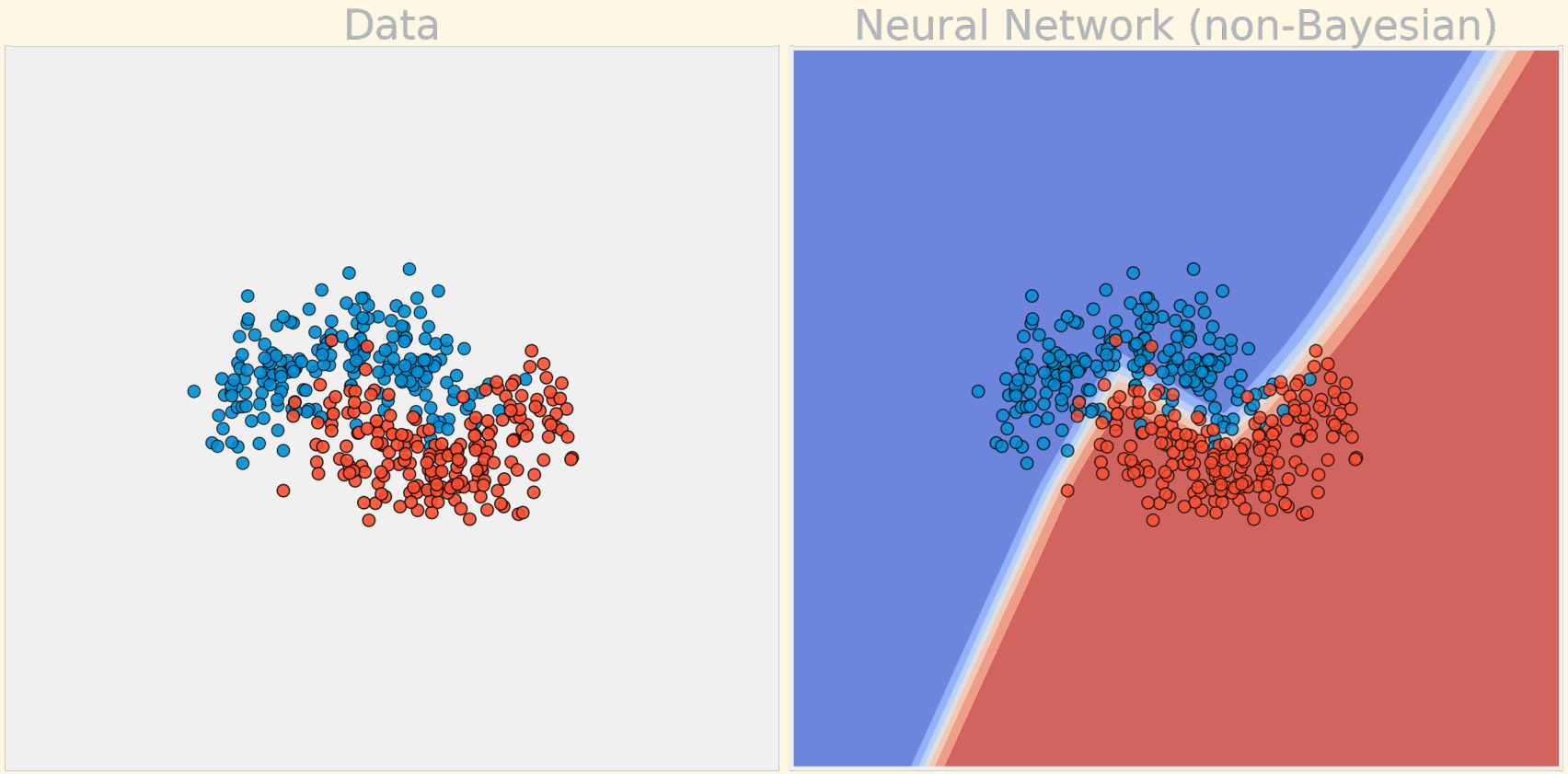
  Many plausible settings of the parameters θ
- Uncertainty about any given hypothesis M , θ
- Why pick just one?
- Instead, form a probability distribution over possible M and θ

What's the point of uncertainty?



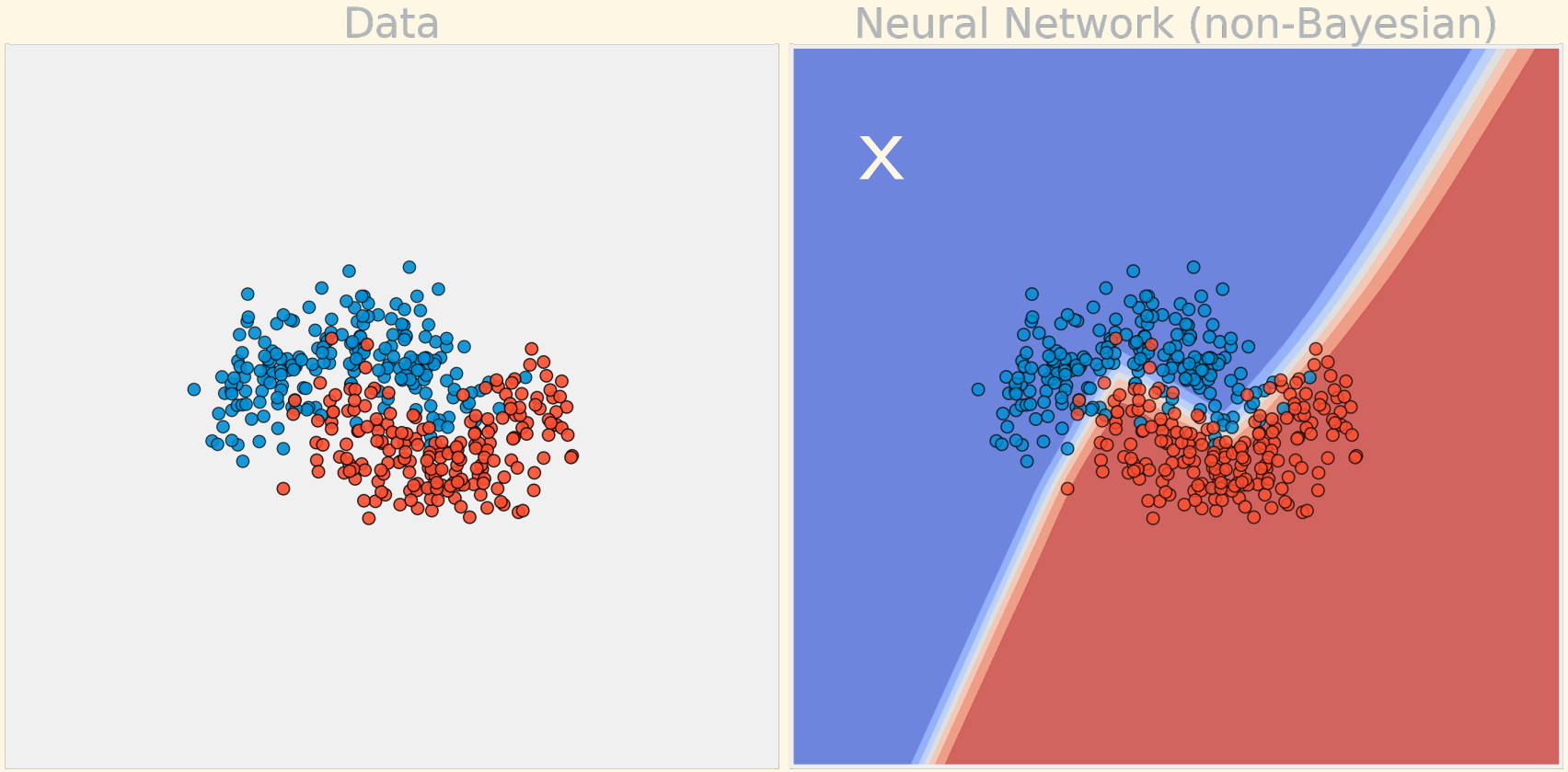
- What does modeling with uncertainty look like?

What's the point of uncertainty?

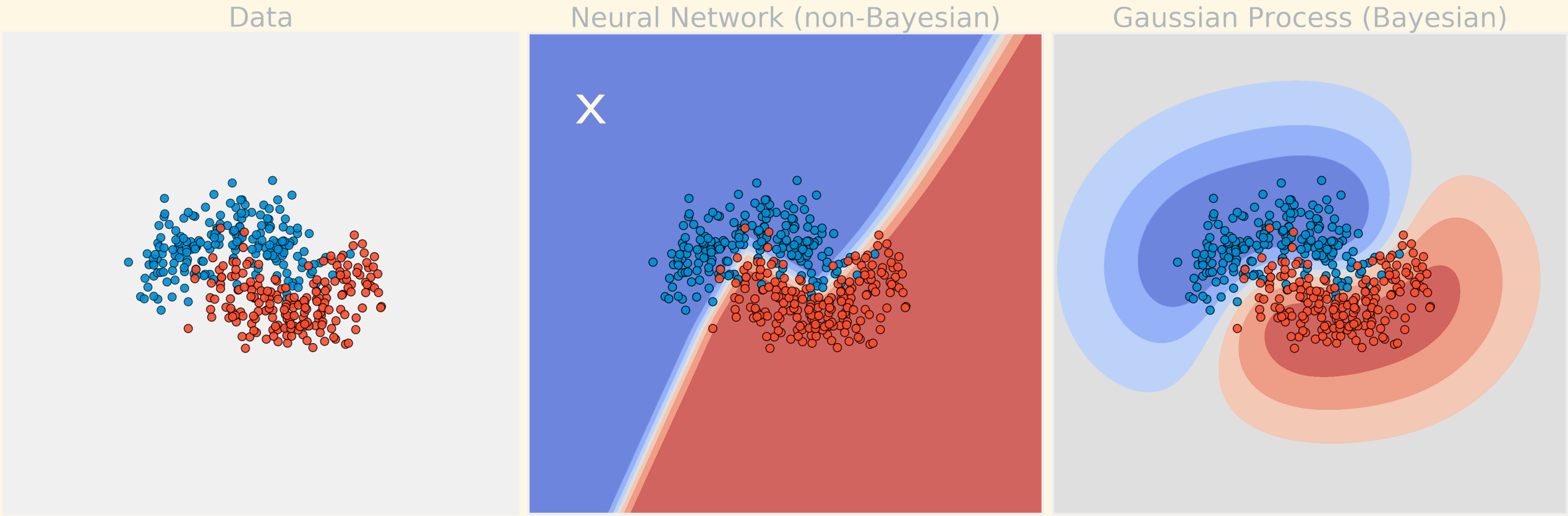


- Non-probabilistic neural network in **sklearn**: not bad

What's the point of uncertainty?

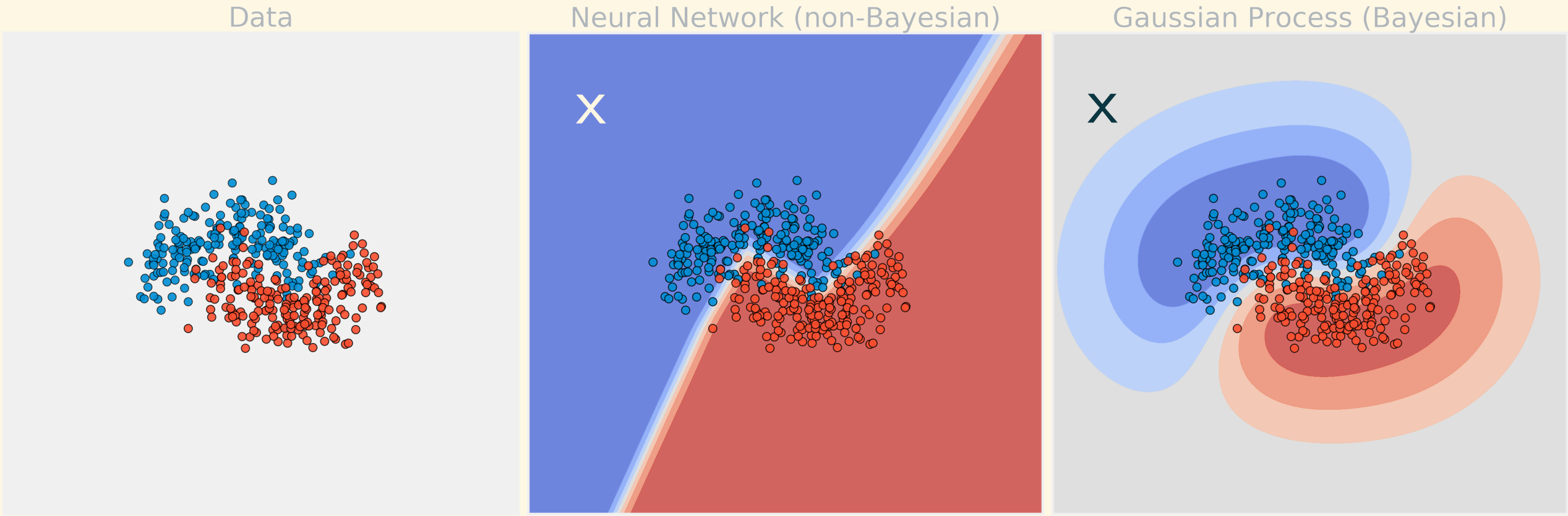


What's the point of uncertainty?



- We can be overly confident where data has not been observed

What's the point of uncertainty?



- The Bayesian model is uncertain where we have not observed data

Adversarial patches in image classification

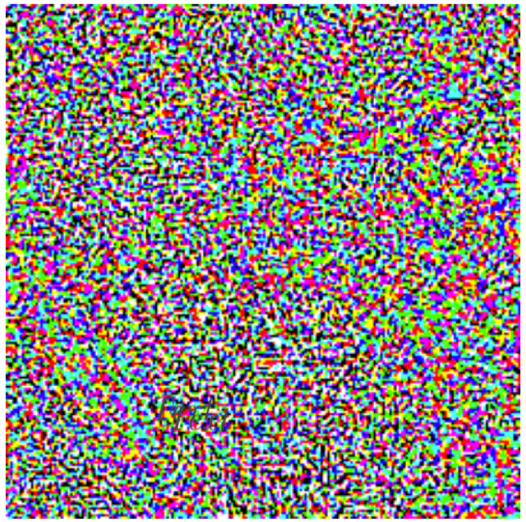
“panda”
57.7% confidence

Source: Explaining and Harnessing Adversarial Examples (2014)
https://arxiv.org/abs/1412.6572

Adversarial patches in image classification

“panda”
57.7% confidence

$+ .007 \times$

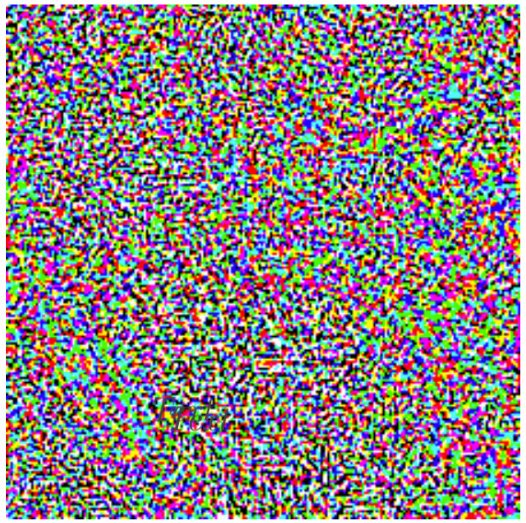


$=$

Source: Explaining and Harnessing Adversarial Examples (2014)
https://arxiv.org/abs/1412.6572

Adversarial patches in image classification

$+ .007 \times$



$=$

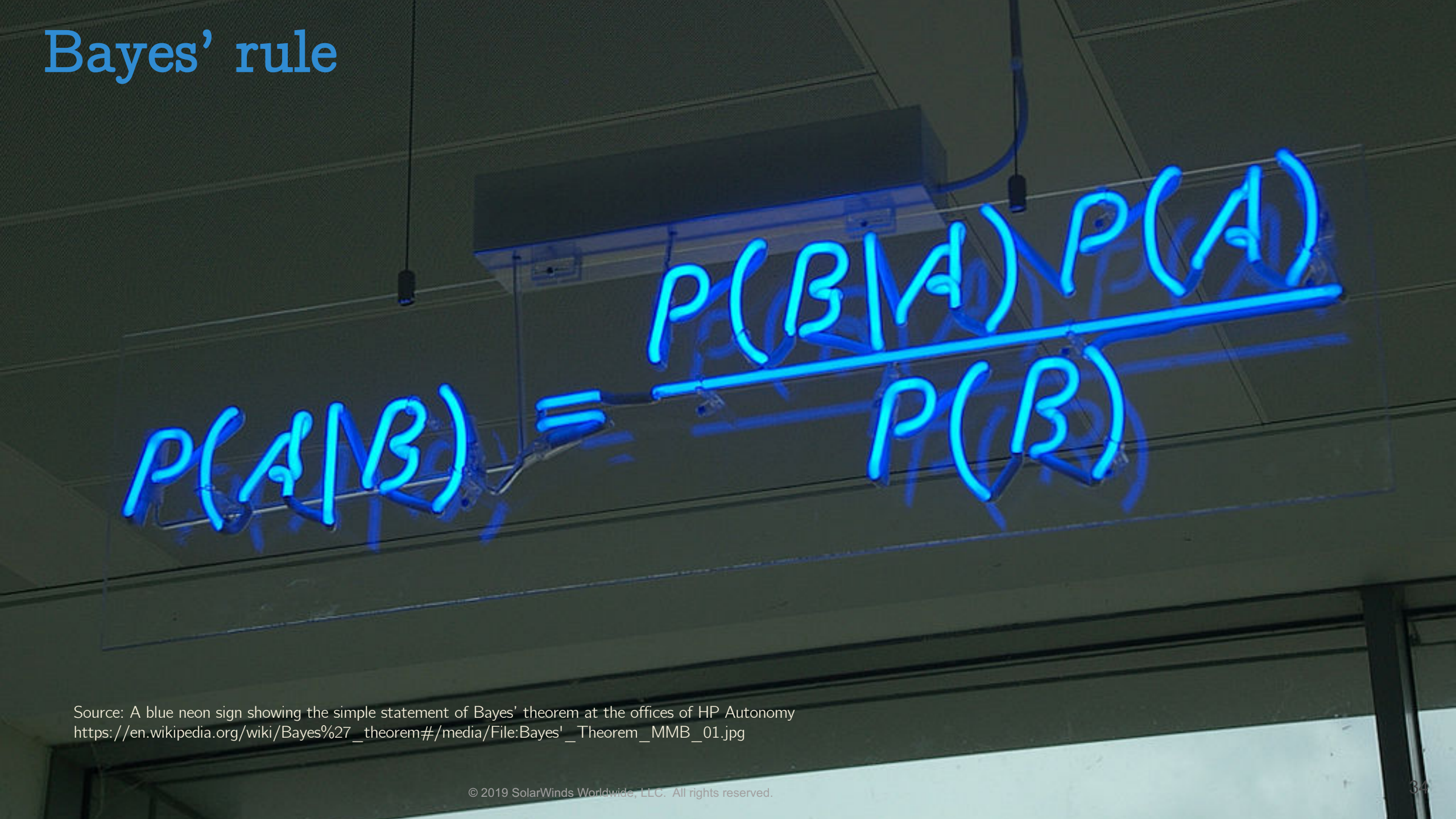
“panda”
57.7% confidence

“monkey”
99.3 % confidence

Source: Explaining and Harnessing Adversarial Examples (2014)
https://arxiv.org/abs/1412.6572

Reasoning with uncertainty

Bayes' rule



Source: A blue neon sign showing the simple statement of Bayes' theorem at the offices of HP Autonomy
https://en.wikipedia.org/wiki/Bayes%27_theorem#/media/File:Bayes'_Theorem_MMB_01.jpg

Bayes' rule

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

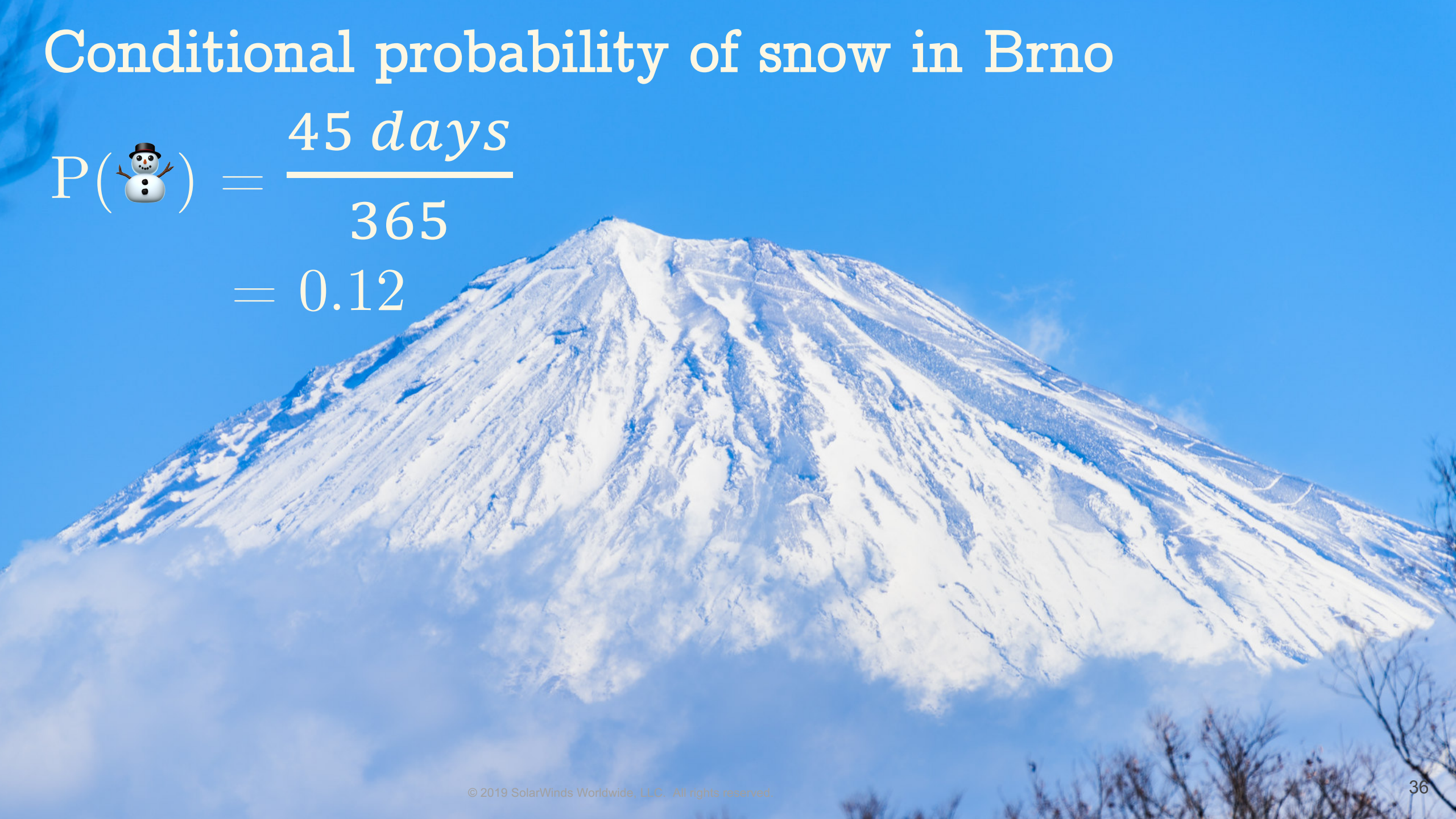
Posterior distribution

- Describes probability of event A, given event B
- After observing an event B, I update my beliefs about event A

Conditional probability of snow in Brno

$$\mathrm{P}(\text{☃}) = \frac{45\ days}{365}$$

$$= 0.12$$



Conditional probability of snow in Brno

$$\mathrm{P}(\text{☃}) = \frac{45\ days}{365}$$

$$= 0.12$$

$$\mathrm{P}(\text{☃} \mid \mathrm{December}) = \frac{11\ days}{31}$$

$$= 0.36$$

Bayes' rule: Posterior distribution

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Posterior distribution

- Describes probability of event A, given event B
- After observing an event B, I update my beliefs about event A

Bayes' rule: Posterior distribution for ML

$$\underbrace{P(\theta|Data)}_{Posterior\ distribution} = \frac{P(Data|\theta)P(\theta)}{P(Data)}$$

- Describes probability of a hypothesis θ, given observed data
- Having observed ***Data***, I update my beliefs about the hypothesis $\boldsymbol{\theta}$

Bayesian coin flip

- What's the probability of heads?
- Flip the coin to get some data

Bayesian coin flip: Flip

- I flip three heads: HHH
- Q: What is the coin's probability of heads?
- A: 1.00? 0.80? 0.50?

Bayesian coin flip: What is p?

- Let p=*probability of heads*
- Suppose $p = \frac{1}{2}$:

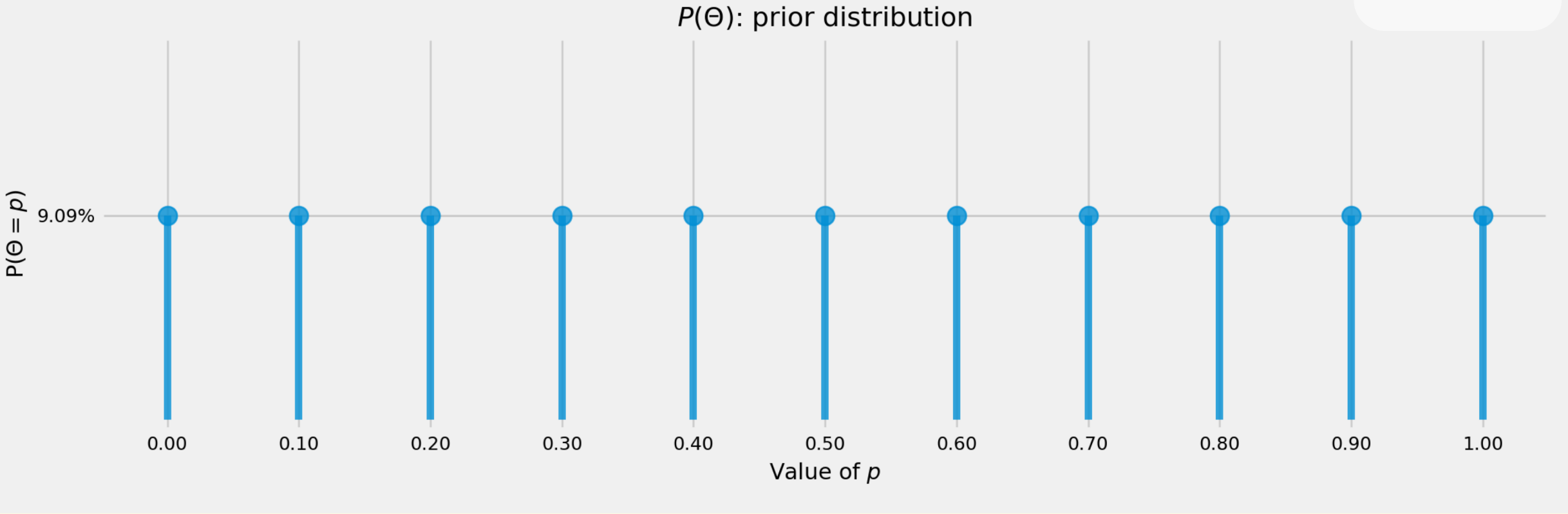
$$\mathrm{P}(\mathrm{HHH} \mid \mathrm{p}{=}0.5) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8}$$

- Do we have enough data to know p?
- Small data is a common problem in ML

Bayesian coin flip: Uninformative prior

Prior

$P(\theta)$

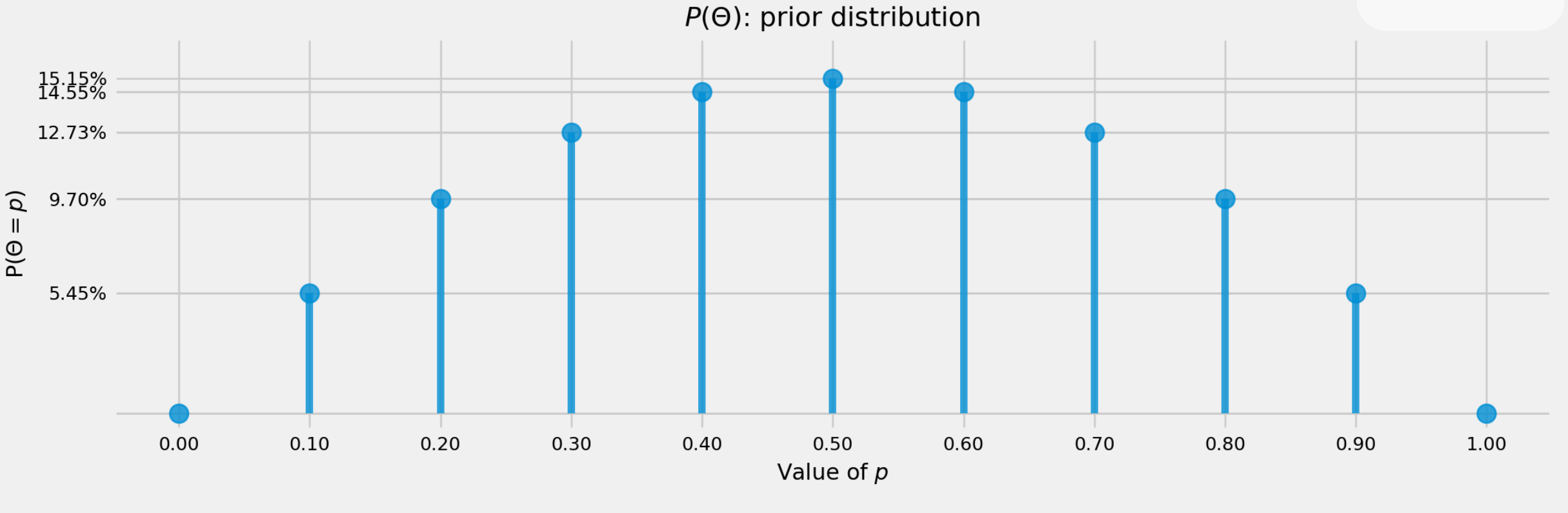


- A uniform distribution expresses no prior beliefs (uninformative)
- This says: 'I don't know how coins work'

Bayesian coin flip: Our prior

Prior

$P(\theta)$



- Define our prior beliefs in a probability distribution
- This prior expresses our intuitions about coin-shaped objects

Bayes example: Prior

$$P(\theta|Data) = \frac{P(Data|\theta)P(\theta)}{P(Data)}$$

Posterior distribution

Prior

Bayes example: Likelihood

$$\underbrace{P(\theta|Data)}_{Posterior\ distribution} = \frac{\overbrace{P(Data|\theta)}^{Likelihood}\overbrace{P(\theta)}^{Prior}}{P(Data)}$$

Bayes example: Likelihood

Likelihood

$$P(Data|\theta)$$

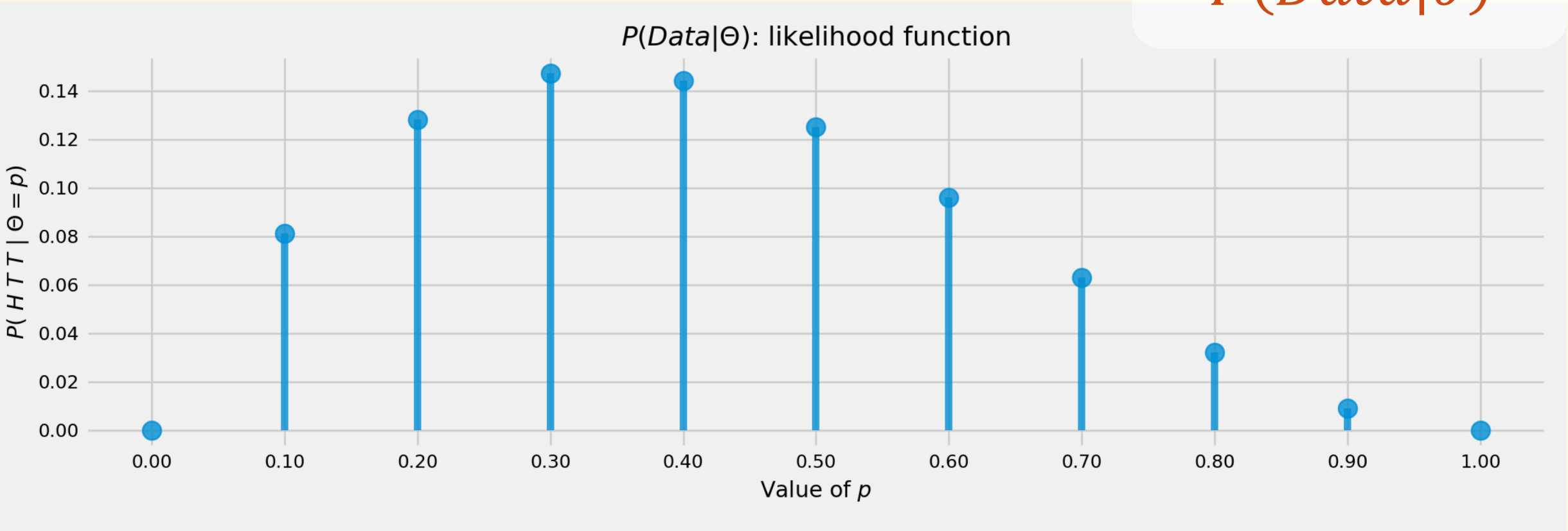
- Likelihood of the data given the parameters
- We've already seen this:

$$\text{P(Data='HHH' | p=0.5)} = \left(\frac{1}{2}\right)^3 = \frac{1}{8}$$

Bayes example: Likelihood

Likelihood

$P(Data|\theta)$



- P(Data='HTT' | θ)

Bayes example: Marginal likelihood

$$P(\theta|Data) = \frac{P(Data|\theta)P(\theta)}{P(Data)}$$

Posterior distribution — *Likelihood* — *Prior* — *Marginal likelihood*

- Normalising constant
- Makes sure the posterior probability sums to 1

Bayes example: Marginal likelihood

$$P(\theta|Data) = \frac{P(Data|\theta)P(\theta)}{\sum_{\theta'} P(Data|\theta')\,P(\theta')}$$

Posterior distribution = $P(\theta|Data)$

Likelihood = $P(Data|\theta)$

Prior = $P(\theta)$

Marginal likelihood = $\sum_{\theta'} P(Data|\theta')\,P(\theta')$

- Sum the likelihood x prior for all possible settings for θ

Bayesian coin flipping

- Notebook 1 👨‍💻👩‍💻

Bayesian coin flipping

- Notebook 1 👨‍💻👩‍💻
- I have some prior beliefs
- I observe some data
- I update my posterior beliefs
- Great!

There's a massive caveat...

Normalising constant

$$\underbrace{P(\theta|Data)}_{Posterior\ distribution} = \frac{\overbrace{P(Data|\theta)}^{Likelihood}\overbrace{P(\theta)}^{Prior}}{\underbrace{\int P(Data|\theta)P(\theta)\,d\theta}_{Marginal\ likelihood}}$$

The bottom term contains the crucial normalising constant...

... all probability distributions must sum to 1...

... but it's really difficult to calculate!

Normalising constant

$$P(\theta|Data) = \frac{P(Data|\theta)P(\theta)}{\text{😟}}$$

Likelihood *Prior*

Posterior distribution

Marginal likelihood

The bottom term contains the crucial normalising constant…

… all probability distributions must sum to 1…

… but it's really difficult to calculate!

Is Bayesian inference realistic?

Yes!

Doing Bayesian inference

Is Bayesian inference realistic?

- Conjugate priors

Conjugate priors

$$P(\theta|Data) = \frac{P(Data|\theta)P(\theta)}{P(Data)}$$

Posterior distribution — *Likelihood* — *Prior* — *Evidence*

- The posterior $P(\theta|X)$ is intractable for most choices of the likelihood $P(X|\theta)$ and the prior $P(\theta)$, but...
- There is a clever way around it!
- If we choose a 'conjugate' prior $P(\theta)$, we can get a clean analytic solution to $P(\theta|X)$
- Don't need to worry about $\int P(X|\theta)P(\theta)\,d\theta$ 😌

Limitations of conjugate priors

What conjugate priors can do

What we'd like to do

Conjugate prior examples

When likelihood function is a continuous distribution [edit]

Likelihood	Model parameters	Conjugate prior distribution	Prior hyperparameters	Posterior hyperparameters	Interpretation of hyperparameters
Normal with known variance σ^2	μ (mean)	Normal	$\mu_0,\ \sigma_0^2$	$\frac{1}{\frac{1}{\sigma_0^2}+\frac{n}{\sigma^2}}\left(\frac{\mu_0}{\sigma_0^2}+\frac{\sum_{i=1}^n x_i}{\sigma^2}\right),\left(\frac{1}{\sigma_0^2}+\frac{n}{\sigma^2}\right)^{-1}$	mean was estimated from observations with total precision (sum of all individual precisions) $1/\sigma_0^2$ and with sample mean μ_0
Normal with known precision τ	μ (mean)	Normal	$\mu_0,\ \tau_0$	$\frac{\tau_0\mu_0+\tau\sum_{i=1}^n x_i}{\tau_0+n\tau},\ \tau_0+n\tau$	mean was estimated from observations with total precision (sum of all individual precisions) τ_0 and with sample mean μ_0
Normal with known mean μ	σ^2 (variance)	Inverse gamma	$\alpha,\ \beta$ [note 5]	$\alpha+\frac{n}{2},\ \beta+\frac{\sum_{i=1}^n (x_i-\mu)^2}{2}$	variance was estimated from 2α observations with sample variance β/α (i.e. with sum of squared deviations 2β, where deviations are from known mean μ)
Normal with known mean μ	σ^2 (variance)	Scaled inverse chi-squared	$\nu,\ \sigma_0^2$	$\nu+n,\ \frac{\nu\sigma_0^2+\sum_{i=1}^n (x_i-\mu)^2}{\nu+n}$	variance was estimated from ν observations with sample variance σ_0^2

Limitations of conjugate priors

- Limits us to a table of mutually compatible likelihood and prior forms
- Trade-off between tractability and expressiveness

Is Bayesian inference realistic?
(still)

Yes!

The holy grail

Posterior distribution

$$P(\theta|Data) = \frac{P(Data|\theta)P(\theta)}{\int P(Data|\theta)P(\theta)\,d\theta}$$

Is Bayesian inference still realistic?

- Modern inference algorithms:
- Markov chain Monte Carlo ('MCMC')
- Approximate Bayesian computation ('ABC')
- Variational inference
- ...

Is Bayesian inference still realistic?

- Modern inference algorithms:
- Markov chain Monte Carlo ('MCMC')
- Approximate Bayesian computation ('ABC')
- Variational inference
- ...

MCMC is the work horse of probabilistic programming languages!

Markov chain Monte Carlo

Markov chain Monte Carlo

- Class of methods for sampling from distributions

Markov chain Monte Carlo

- Class of methods for sampling from distributions

MC I: Monte Carlo

- Monte Carlo method:

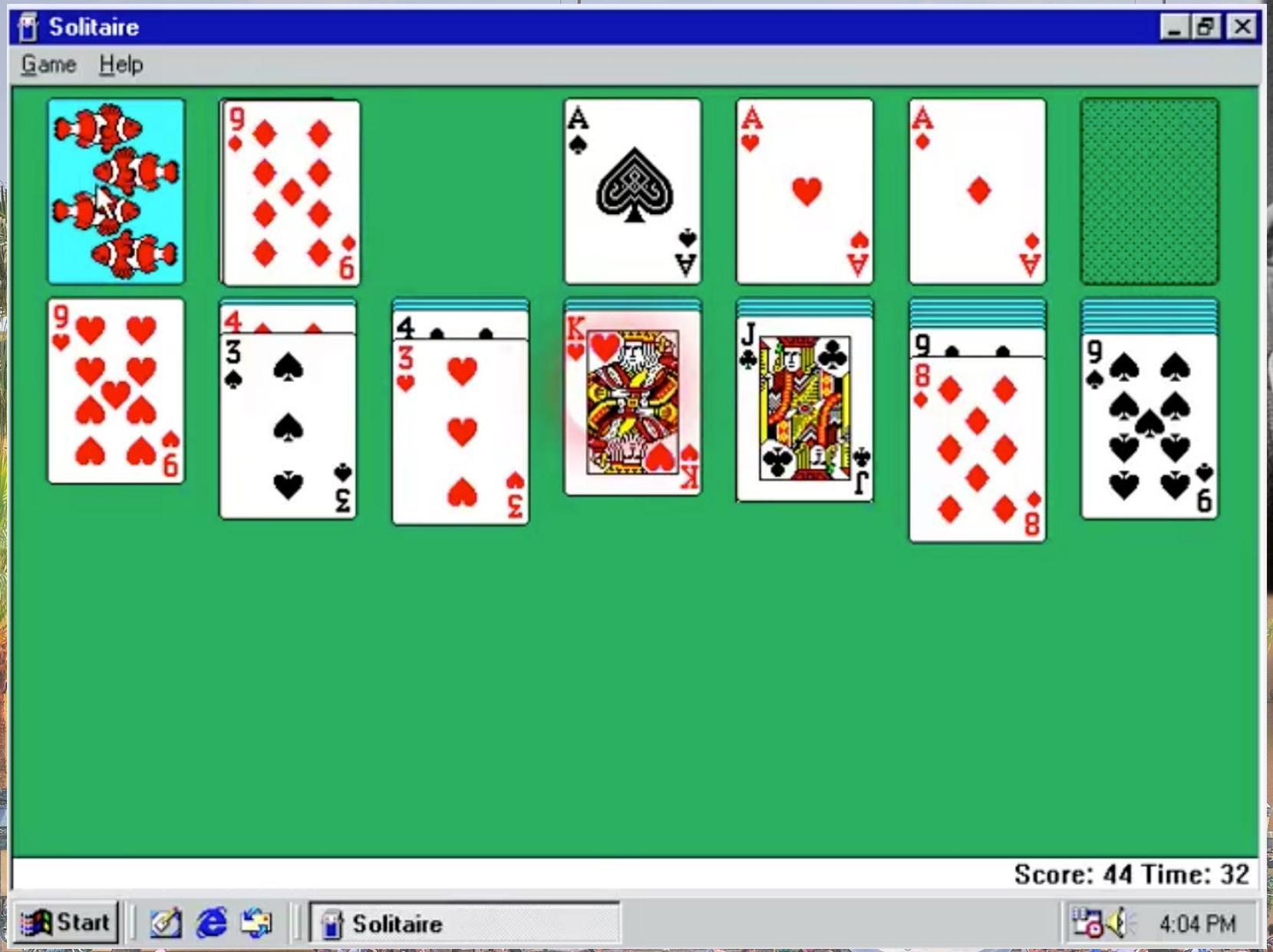
  Approximate quantities by sampling

MC I: Monte Carlo

MC I: Monte Carlo

Stanislaw Ulam

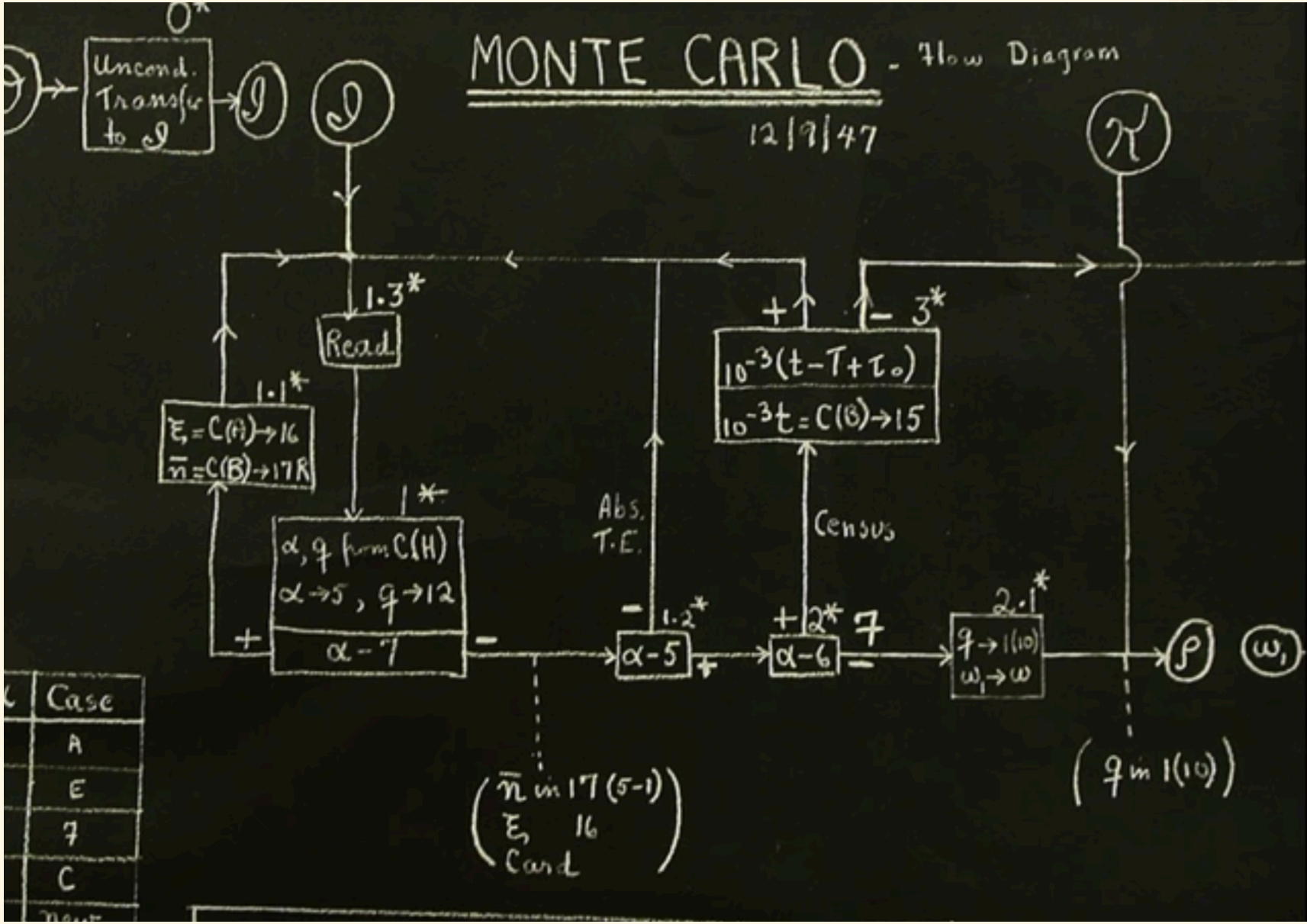
MC I: Monte Carlo



Stanislaw Ulam

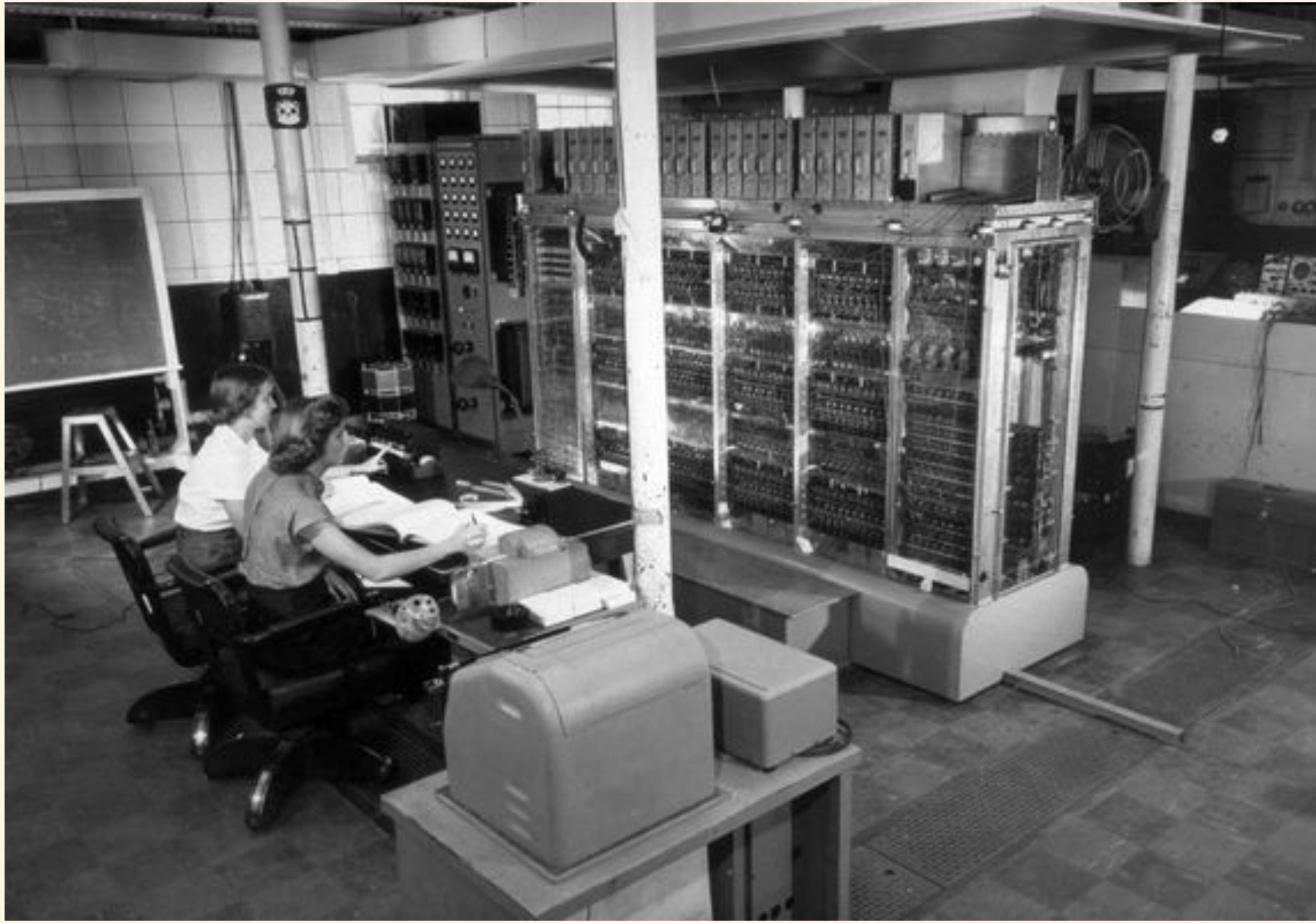
Early history of Markov chain Monte Carlo

- 1946: Manhattan Project, ENIAC used for Monte Carlo

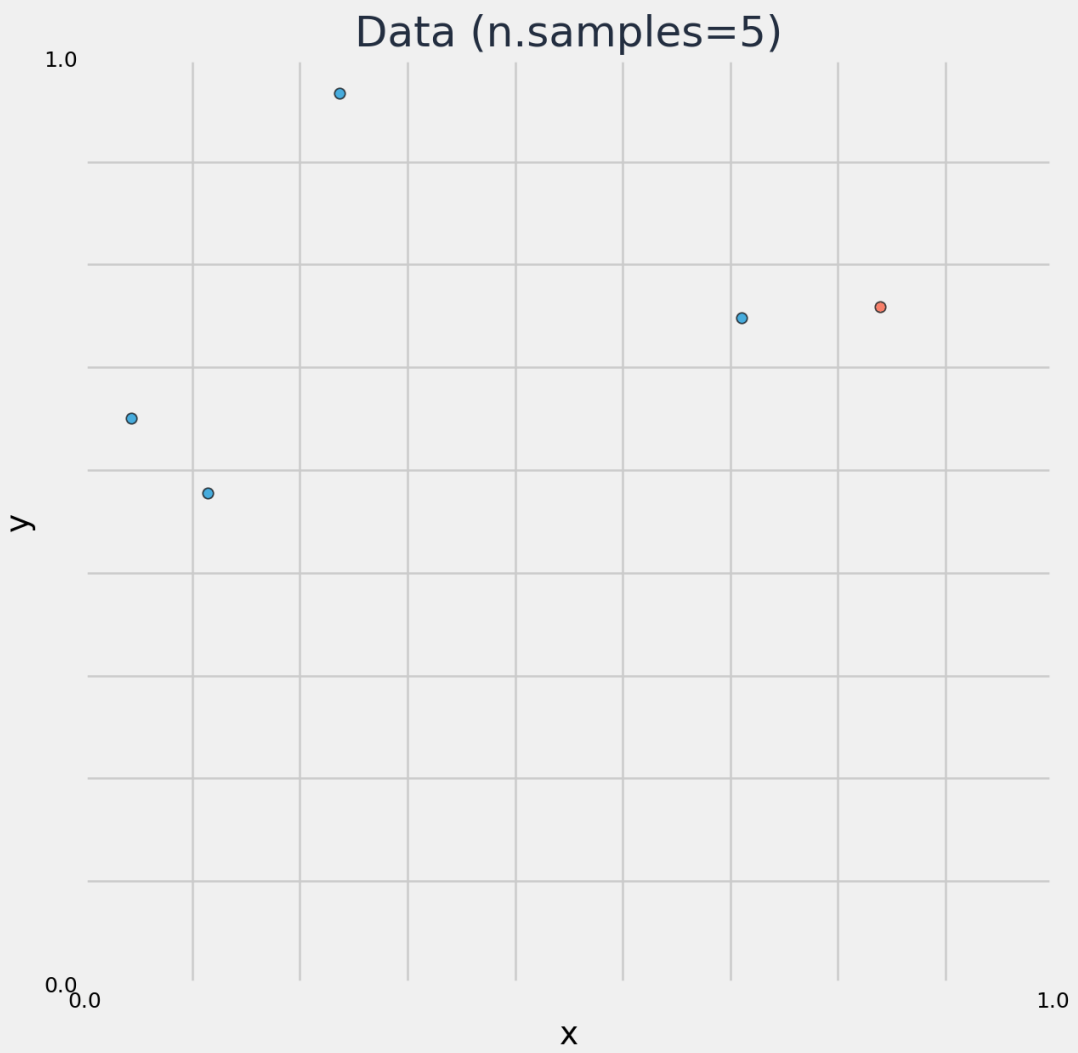


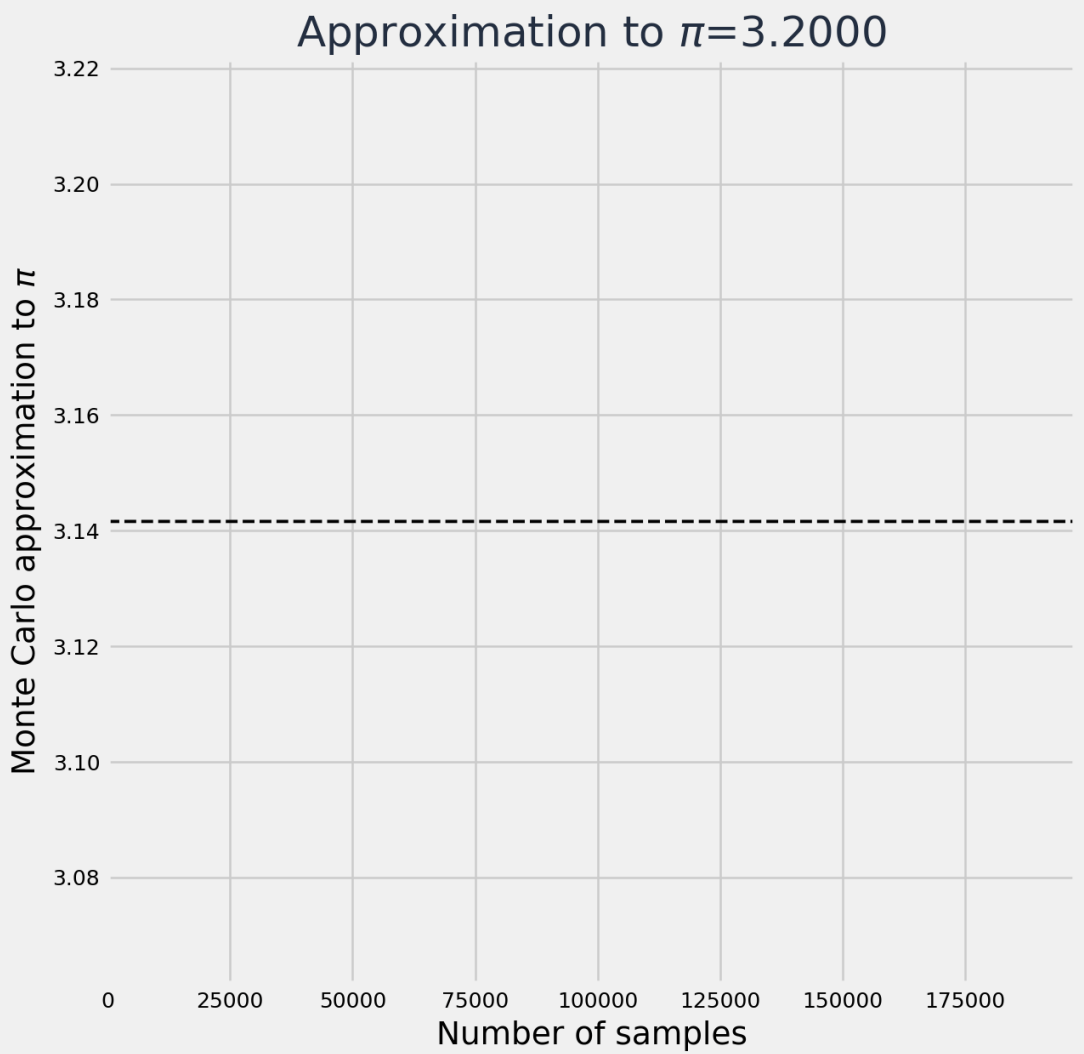
Early history of Markov chain Monte Carlo

- 1946: Manhattan Project, ENIAC used for Monte Carlo
- 1952: Manhattan Project, MANIAC I used for MCMC



Monte Carlo: π example





Markov chain Monte Carlo

- Class of methods for sampling from distributions

MC I: Monte Carlo

- Monte Carlo method:

  Approximate quantities by sampling

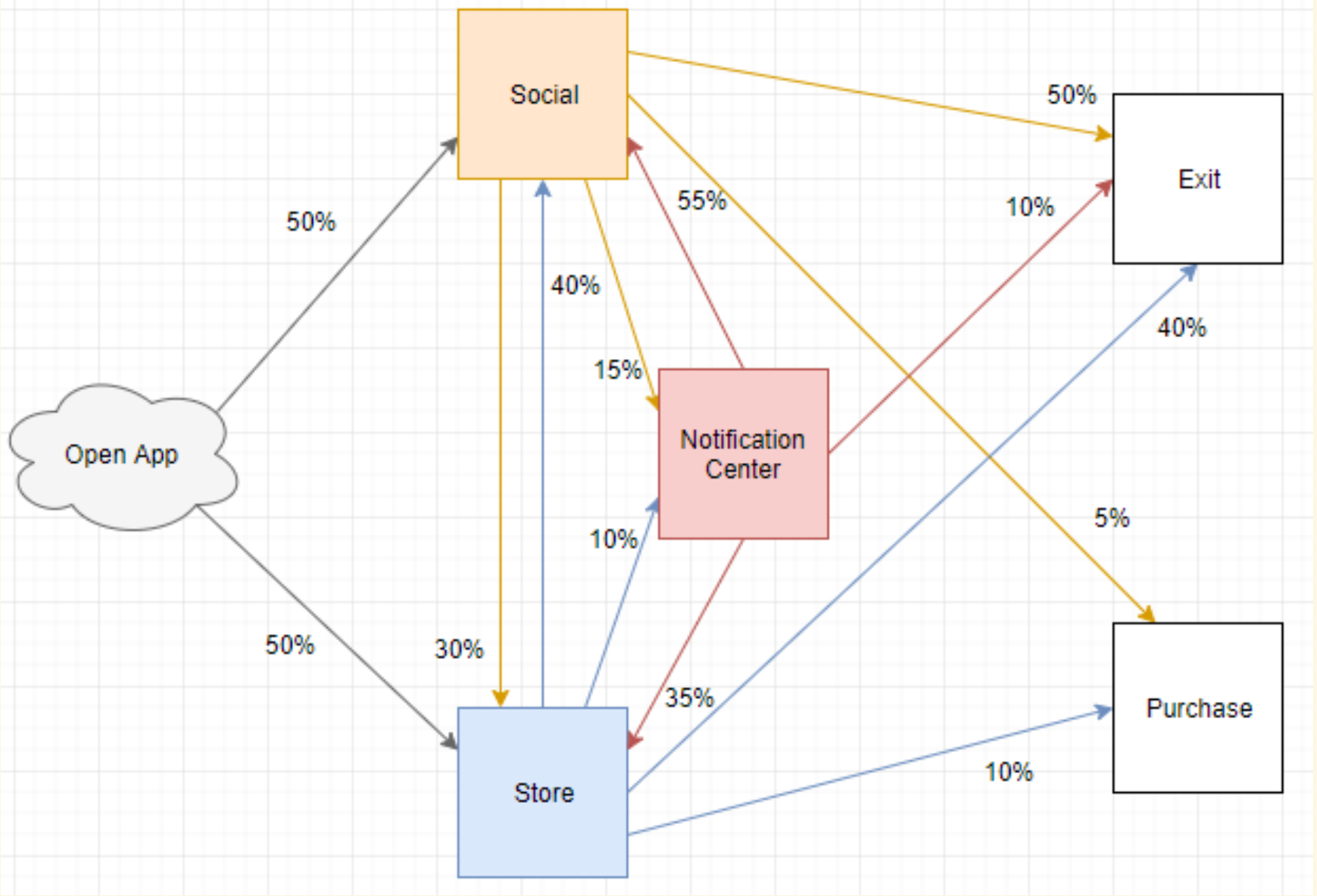
MC II: Markov chain

- Markov chain:

  Describes a stochastic sequence of events

MC II: Markov chain, used in PageRank

MC II: Markov chain, app user example



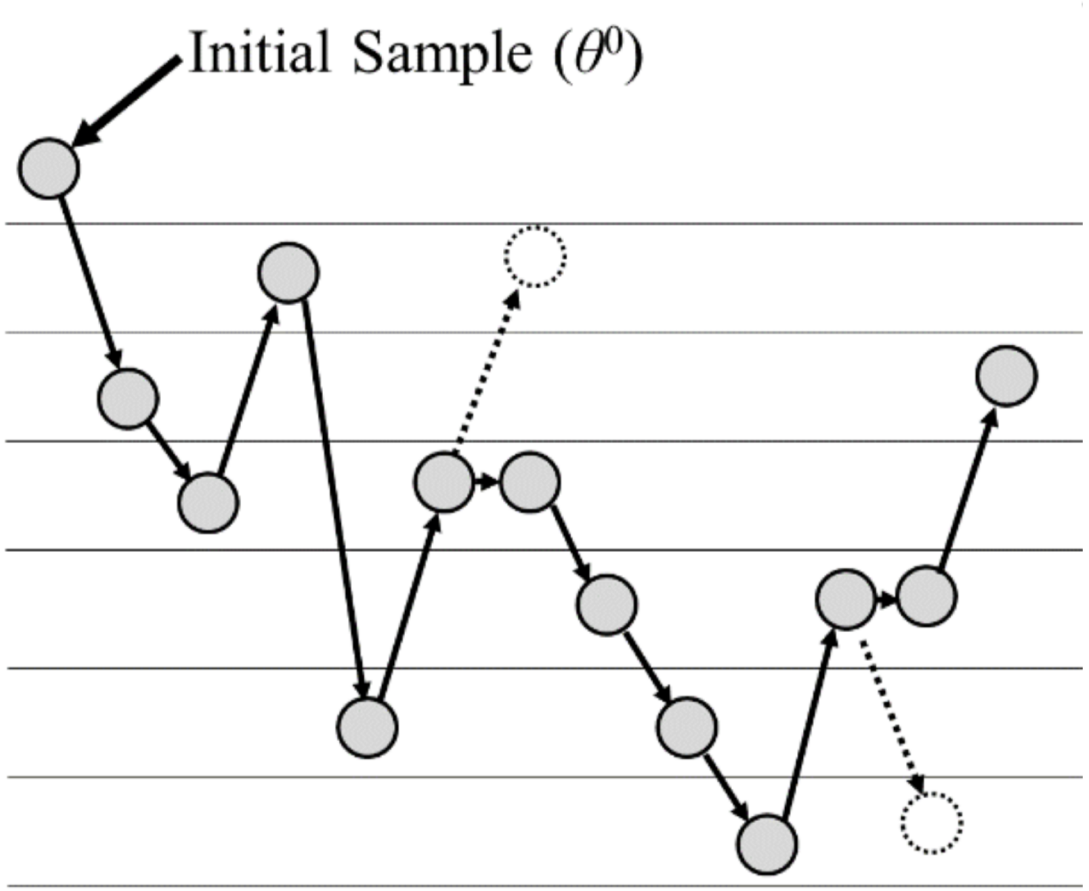
MC II: Markov chain demo

http://setosa.io/markov/

MC II: Stationary distribution of a Markov chain

- Notebook 2 👨‍💻👩‍💻
- Some Markov chains have a stationary distribution
- No matter which state we start at
- If we transition in the chain long enough
- The proportion of time spent in each state converges (to the stationary distribution)

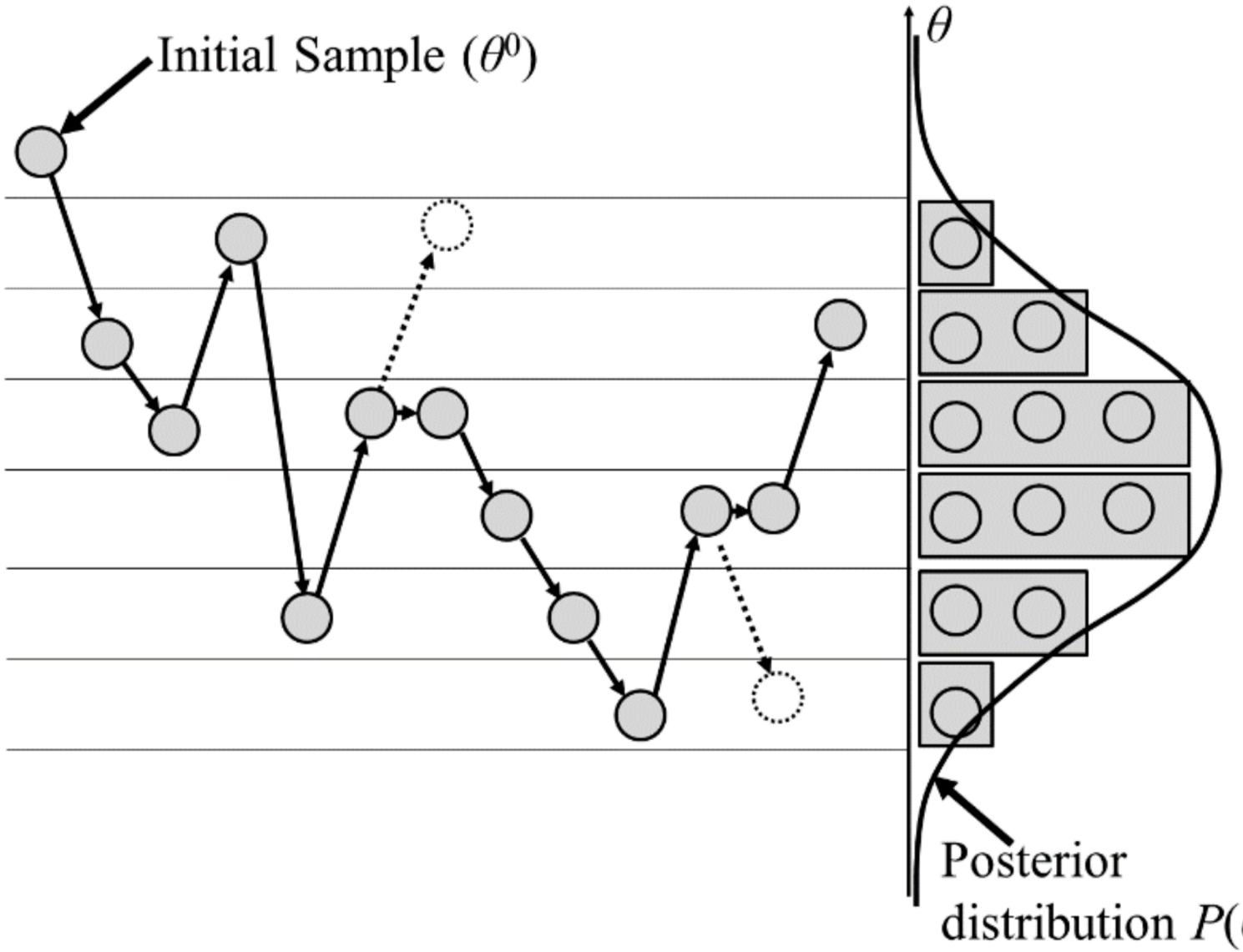
Towards MCMC



- Start at some initial state
- Take a random walk, this is the proposed next state
- **Transition** to the proposed state with some **magic probability** (we come back to this later)
- Keep doing this

Source: Metamodel for Efficient Estimation of Capacity-Fade Uncertainty in Li-Ion Batteries for Electric Vehicles (2015) https://ideas.repec.org/a/gam/jeners/v8y2015i6p5538-5554d50871.html

Towards MCMC



- The chain of steps form the samples
- The bag of samples is a Monte Carlo estimate of the posterior!
- (The chain's stationary distribution happens to be the posterior)

Source: Metamodel for Efficient Estimation of Capacity-Fade Uncertainty in Li-Ion Batteries for Electric Vehicles (2015)
https://ideas.repec.org/a/gam/jeners/v8y2015i6p5538-5554d50871.html

MCMC demo

- Notebook 3 👨‍💻👩‍💻

https://chi-feng.github.io/mcmc-demo/app.html

Why did that work?

- The accept rule is the ratio of the posterior probability of the the proposed sample θ_{i+1} to that of the current sample θ_i

- Accept probability $= \frac{Posterior\ prob.\ of\ proposed\ sample}{Posterior\ prob.\ of\ current\ sample}$

Ratio of posterior probabilities

Why did that work?

- Written out in mathematical notation:

- Accept probability = $\frac{P(\theta{=}\theta_{i+1}|Data)}{P(\theta{=}\theta_i|Data)}$

Ratio of posterior probabilities

Why did that work?

- Written out in mathematical notation:
- Accept rule elegantly removes the difficult integral altogether
- Accept probability = $$\frac{\frac{P(Data|\theta=\theta_{i+1})P(\theta=\theta_{i+1})}{\cancel{\int_\theta P(Data|\theta)P(\theta)d\theta}}}{\frac{P(Data|\theta=\theta_i)P(\theta=\theta_i)}{\cancel{\int_\theta P(Data|\theta)P(\theta)d\theta}}}$$

Ratio of posterior probabilities

Why did that work?

- Written out in mathematical notation:
- Accept rule elegantly removes the difficult integral altogether

This is easy to calculate!

- Accept probability = $$\frac{P(Data|\theta{=}\theta_{i+1})P(\theta{=}\theta_{i+1})}{P(Data|\theta{=}\theta_{i})P(\theta{=}\theta_{i})}$$

Ratio of unnormalized posteriors

The holy grail recovered

Posterior distribution

$$P(\theta|Data) = \frac{P(Data|\theta)P(\theta)}{\int P(Data|\theta)P(\theta)\,d\theta}$$

- MCMC gives us an empirical (Monte Carlo) approximation to the posterior distribution

One more thing...

Later history of Markov chain Monte Carlo

- 1970s: Vanilla MCMC: Metropolis-Hastings

  Not suited to some posterior distributions

- 1987: Hamiltonian Monte Carlo

  Better convergence speed

Hamiltonian Monte Carlo

- Notebook 4 👨‍💻👩‍💻
- Metropolis-Hastings can take a long time to converge for certain distributions
- e.g getting stuck in local maxima in multi-modal distributions
- Hamiltonian Monte Carlo uses intuitions from physics, gradient information:
- Propose next sample by rolling a ball around the probability surface
- Faster convergence to posterior distribution

Hamiltonian Monte Carlo demo

- Notebook 4 👨‍💻👩‍💻

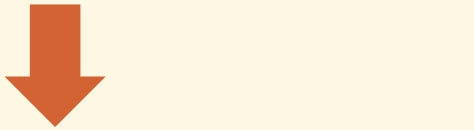
https://chi-feng.github.io/mcmc-demo/app.html

The story so far

Machine learning

The story so far

Machine learning



Probabilistic perspective

The story so far

Machine learning

Probabilistic perspective

Bayes' theorem

The story so far

Machine learning

Probabilistic perspective

Bayes' theorem

Tractability of Bayesian inference

The story so far

Machine learning

Probabilistic perspective

Bayes' theorem

Tractability of Bayesian inference

Markov chain Monte Carlo

The story so far

Machine learning

Probabilistic perspective

Bayes' theorem

Tractability of Bayesian inference

Markov chain Monte Carlo

Probabilistic programming

Probabilistic programming

Why is probabilistic programming emerging now?

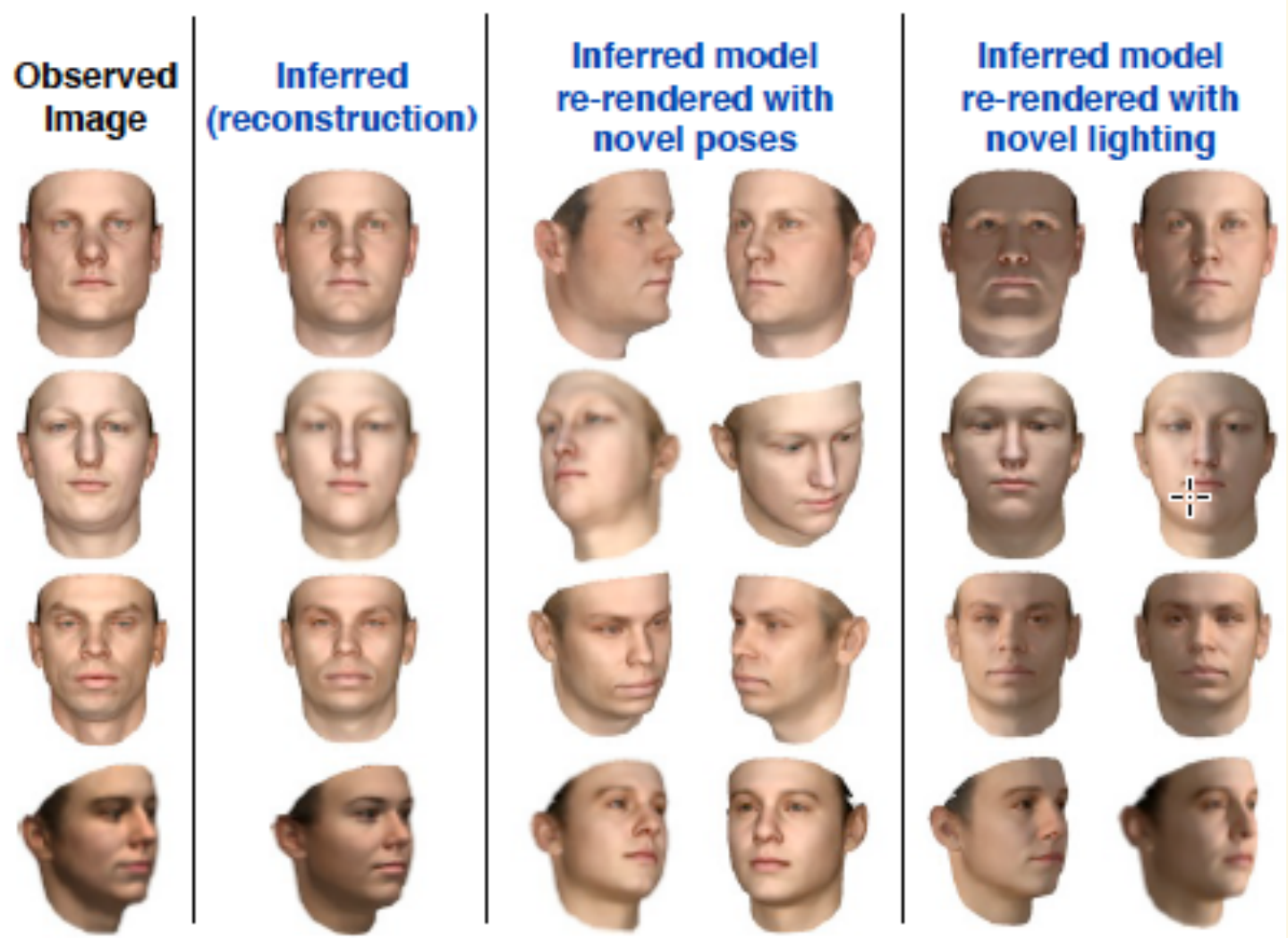
- 1970s: Metropolis-Hastings algorithm
- 1987: Hamiltonian Monte Carlo
- 1990s: Flood of interest in MCMC
- 2011: No-U-Turn Sampler (NUTS)
- 2013: DARPA begins funding prob. prog. research
- Today: Bayesian revolution in the Sciences

Probabilistic programming languages

- High-level packages for doing Bayesian inference
- They solve our intractability problem by using MCMC
- Just specify a probabilistic model and press the button 🚨
- Infer quantities from empirical samples

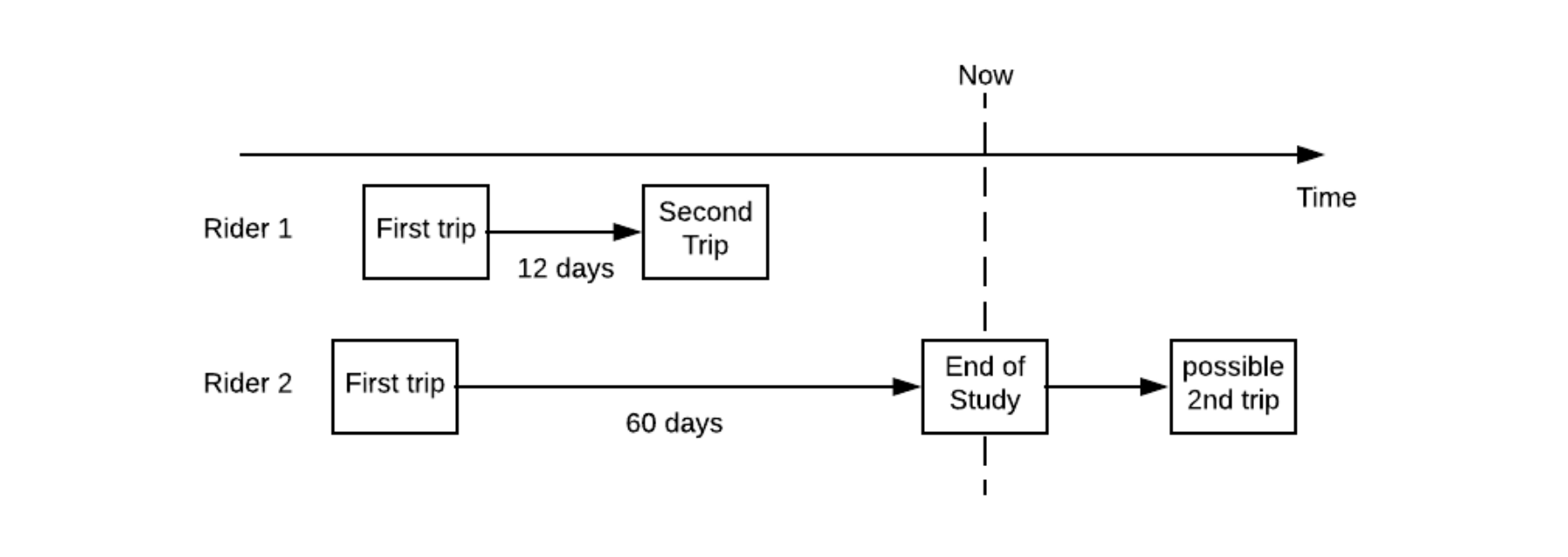
Prob. prog. in practice: Inverse graphics

- Probabilistic programming for scene perception (2015)
- Deduce 3D graphic model of face from 2D image
- 50 lines of code



Source: Picture: A probabilistic programming language for scene perception (2015) https://mrkulk.github.io/www_cvpr15/1999.pdf

Prob. prog. in practice: Uber



Source: Modeling Censored Time-to-Event Data Using Pyro, an Open Source Probabilistic Programming Language (2019)
https://eng.uber.com/modeling-censored-time-to-event-data-using-pyro/

- Pyro built on PyTorch for probabilistic deep learning
- Model time-to-event (time to second trip)

Probabilistic programming languages

Prob. prog. language	Interface	Backend
Stan	C++, R, Python, Julia, others	STAN Math Library
TensorFlow Probability	Python	TensorFlow
Pyro	Python	PyTorch
PyMC3	Python	Theano
PyMC4	Python	TensorFlow

About PyMC3

- Popular prob. prog. package for **Python**
- Models **compile to C** for speed
- MCMC sampling **(NUTS)**
- Uses Theano for **automatic differentiation**
- **PyMC4** in currently in development, move to **TensorFlow**

PyMC3: Coin flipping

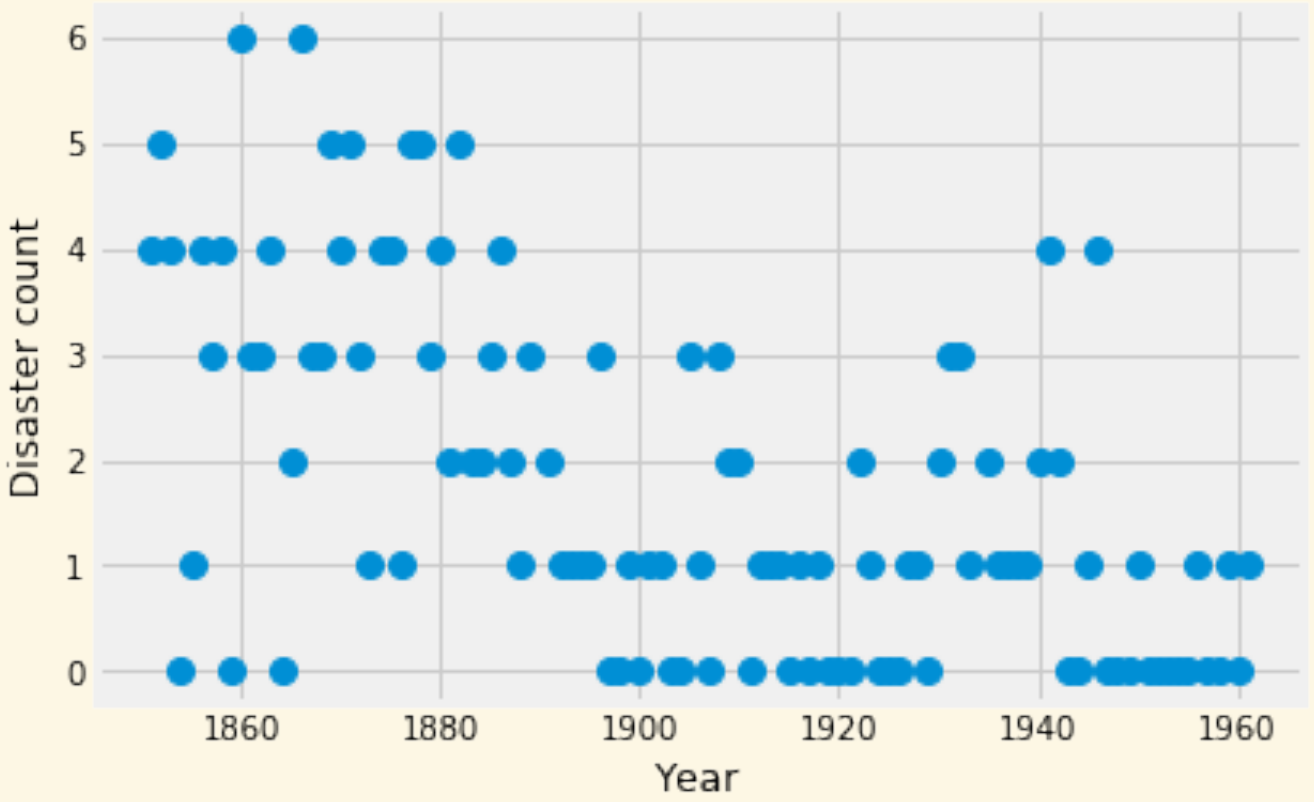
- Notebook 5 👨‍💻👩‍💻
- Pick prior and likelihood for problem
- Write out the model in high-level (math) syntax
- Press the Magic Inference Button 🚨
- Plot results

Posterior distribution

$$P(\theta|Data) = \frac{P(Data|\theta)P(\theta)}{\int P(Data|\theta)P(\theta)\,d\theta}$$

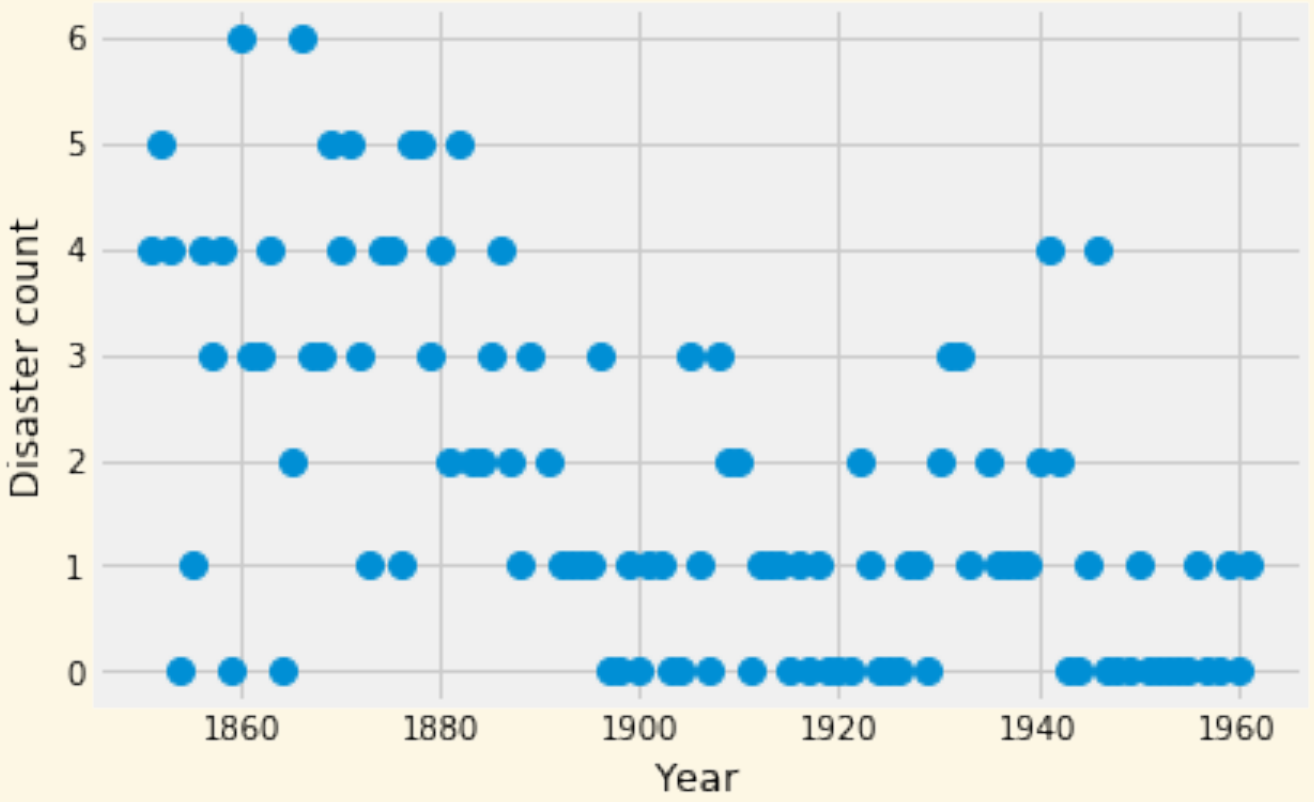
PyMC3: Coal disasters

- Notebook 6 👨‍💻👩‍💻
- UK coal mining disasters, 1851-1962
- Missing data



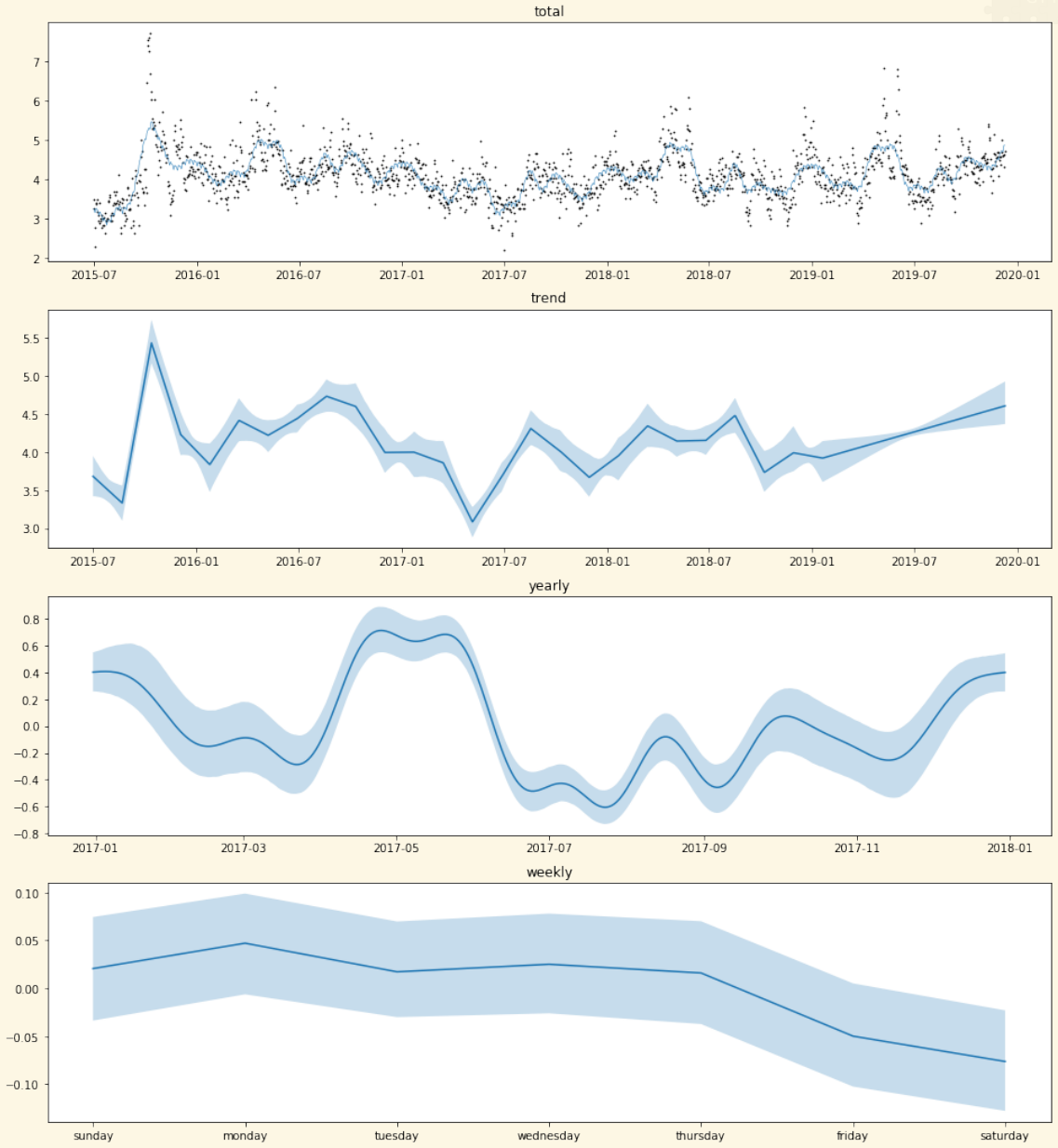
PyMC3: Coal disasters

- Notebook 6 👨‍💻👩‍💻
- Changepoint detection
- Missing data imputation
- Uncertainty measures



PyMC3: Bayesian Time Series

- Notebook 7 👩‍💻👩‍💻
- (takes a long time to run, open solutions/S7 for compiled version)
- Modelling Wikipedia page views of Jurgen Klopp
- Changepoint detection
- Specify seasonality, trend
- Decomposition of effects
- Uncertainty measures in forecasting



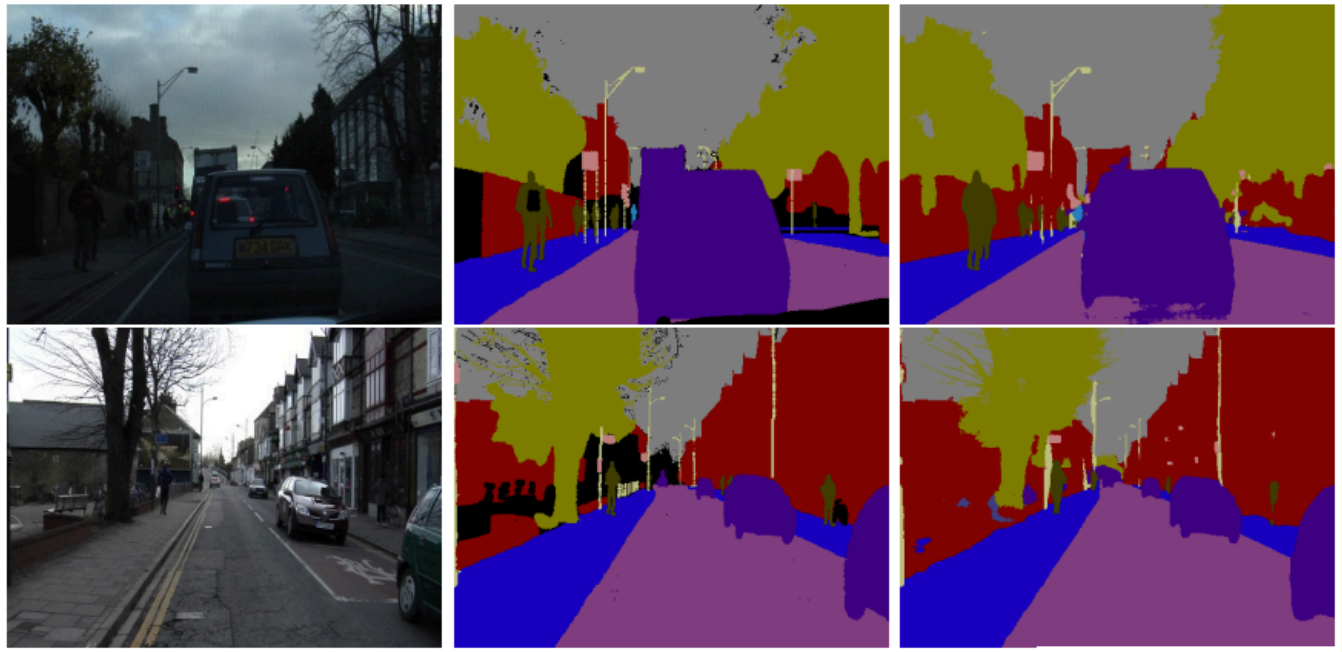
Probabilistic deep learning

- In deep neural networks, weights initialised by randomisation
- We have no priors on weights
- Deep learning models not robust to:

  Perturbation in weights: i.e. training for one more batch

  Perturbation in input data: i.e. adversarial example
- c.f.: Dropout as a Bayesian Approximation (2015)

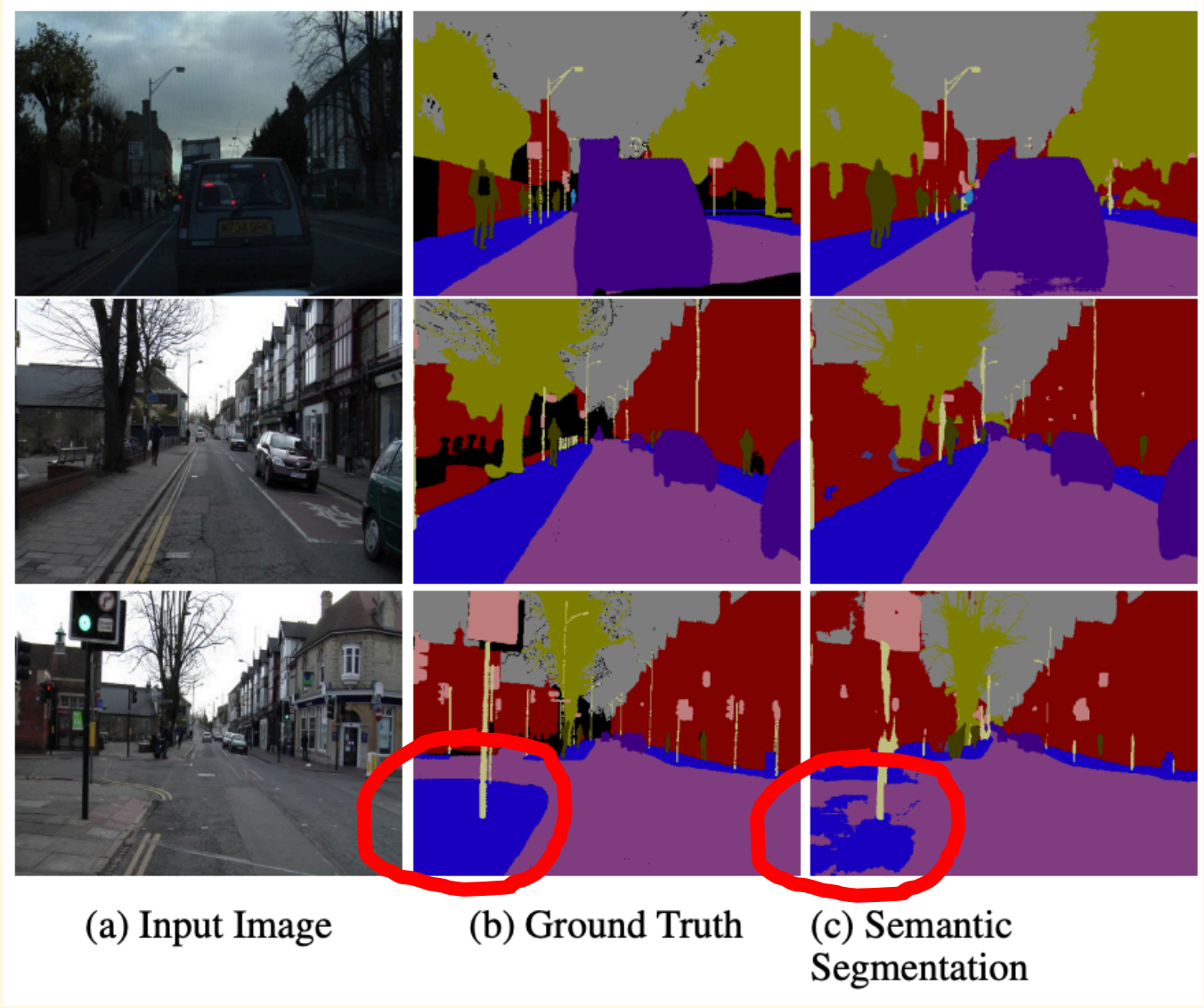
Image segmentation



(a) Input Image (b) Ground Truth (c) Semantic Segmentation

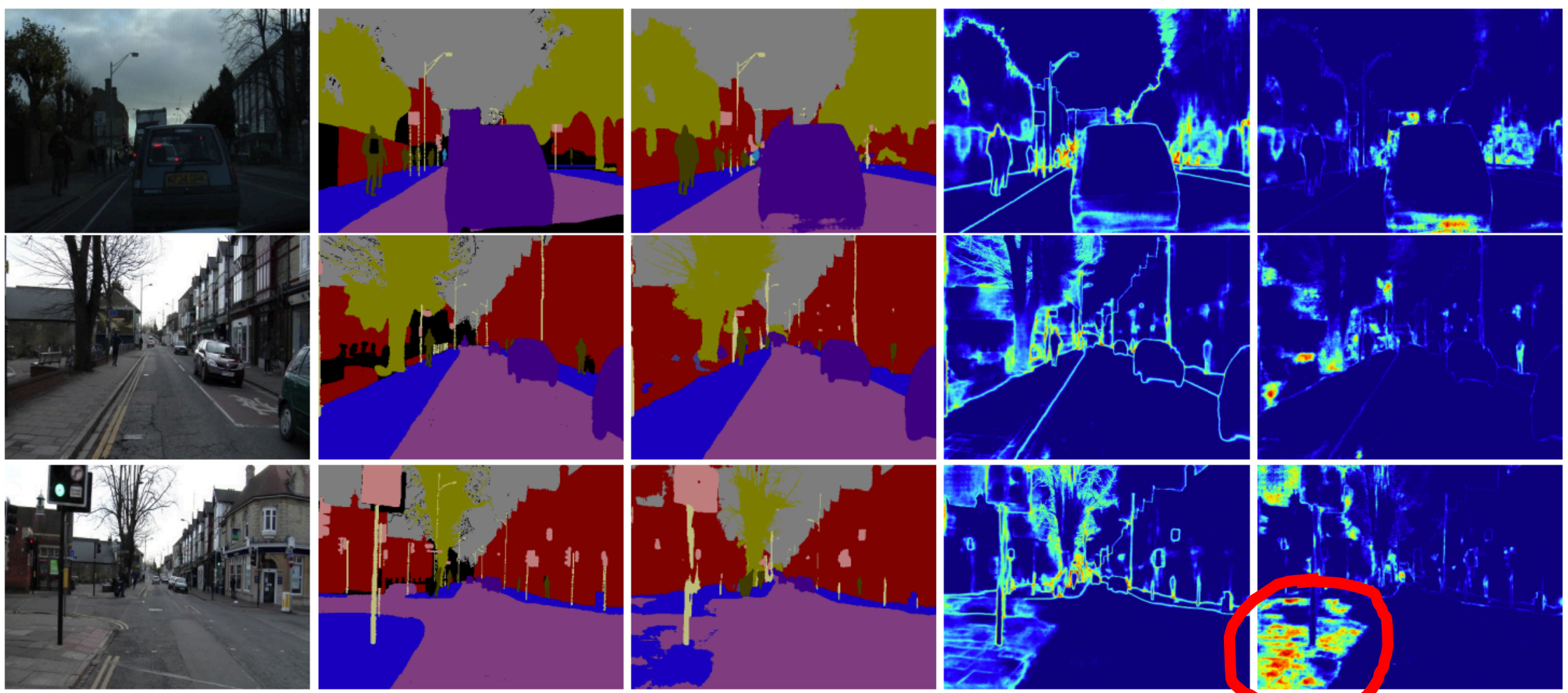
Source: What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? (2017)
https://arxiv.org/pdf/1703.04977.pdf

Image segmentation



Source: What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? (2017)
https://arxiv.org/pdf/1703.04977.pdf

Image segmentation



Source: What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? (2017)
https://arxiv.org/pdf/1703.04977.pdf

Probabilistic programming drawbacks

- Slow
- Not always best choice for low latency ML requirements
- Need to know statistics
- Specifying a prior distribution can be tricky

Why build probabilistic models?

- Principled handling of uncertainty
- Explainable models
- Generate data from model
- Modularity and composability
- Works well with small and medium data
- Bayesian models are online (update with new data)
- Robust, intuitive, scientific
- Fun

Overview

- Machine learning
- The probabilistic perspective
- Bayes' rule
- Bayesian inference, challenges
- Markov chain Monte Carlo
- Probabilistic programming and PyMC3

References

Textbooks

Machine Learning: A Probabilistic Perspective

Kevin P. Murphy. MIT Press (2012)

Pattern Recognition and Machine Learning

Christopher Bishop. Springer (2006)

Information Theory, Inference and Learning Algorithms

David J.C. Mackay. Cambridge University Press (2012)

Probabilistic Graphical Models

Daphne Koller. MIT Press (2009)

Probability Theory: The Logic of Science

E.T. Jaynes. Cambridge University Press (2003)

Videos

CPSC540: Markov-chain Monte Carlo

Nando de Freitas, 2013

Variational Inference in Python

Austin Rochford, 2016

Deep Probabilistic Methods with PyTorch

Chris Ormandy, 2018

Links

MCMC Demo

Chi Feng, 2019

References

Papers

MCMC using Hamiltonian dynamics

R. M. Neal (2010)

A Conceptual Introduction to Hamiltonian Monte Carlo

M. Betancourt (2017)

Probabilistic programming in Python using PyMC3

Salvatier, J., Wiecki T.V., Fonnesbeck C. (2016)

Forecasting at Scale

Taylor, S.J., Letham, B. (2017)

Are our brains Bayesian?

Robert Bain (2016)

Thank you!

Questions?

This presentation contains forward-looking statements regarding future product plans and development efforts. SolarWinds considers various features and functionality prior to any final generally available release. Information in this presentation regarding future features and functionality is not and should not be interpreted as a commitment from SolarWinds that it will deliver any specific feature or functionality in the future or, if it delivers such feature or functionality, any time frame when that feature or functionality will be delivered. All information is based upon current product interests, and product plans and priorities can change at any time. SolarWinds undertakes no obligation to update any forward-looking statements regarding future product plans and development efforts if product plans or priorities change.