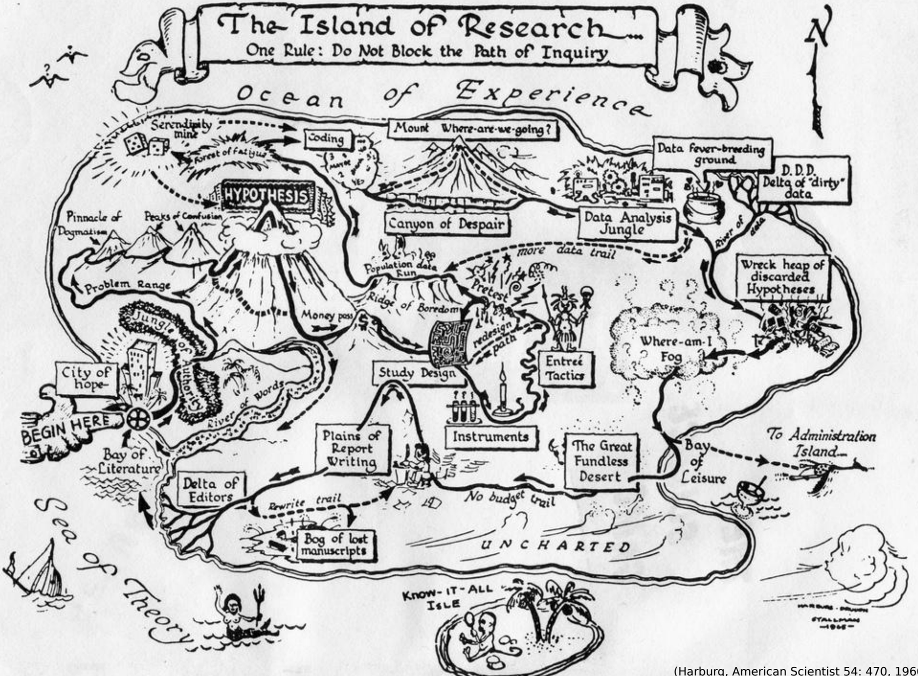
Efficient Statistical Inference for Intractable Models

Michael Gutmann

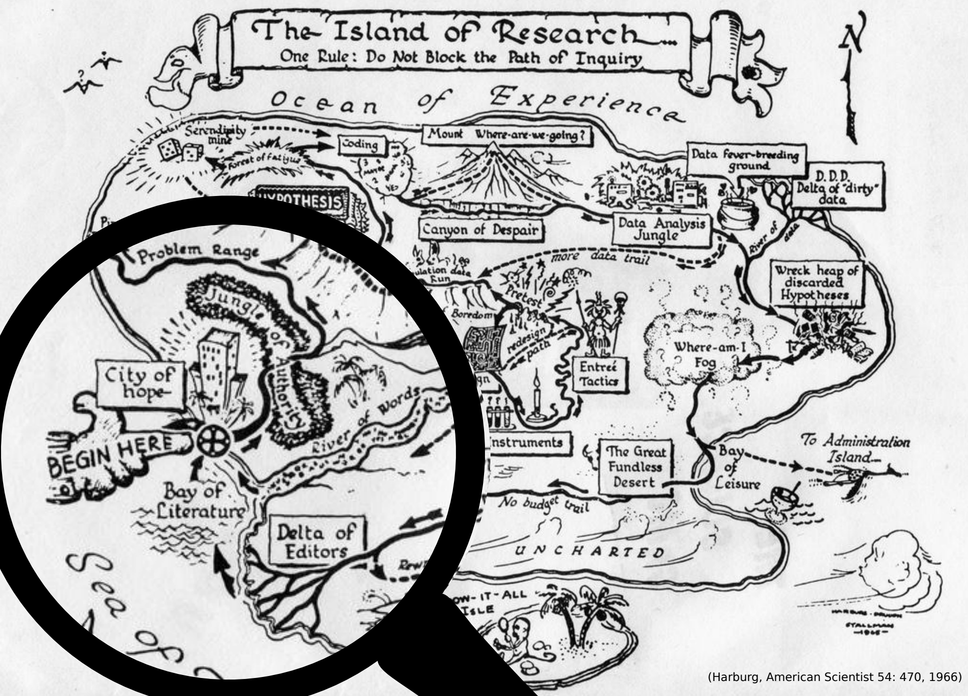
http://homepages.inf.ed.ac.uk/mgutmann

Institute for Adaptive and Neural Computation
School of Informatics, University of Edinburgh

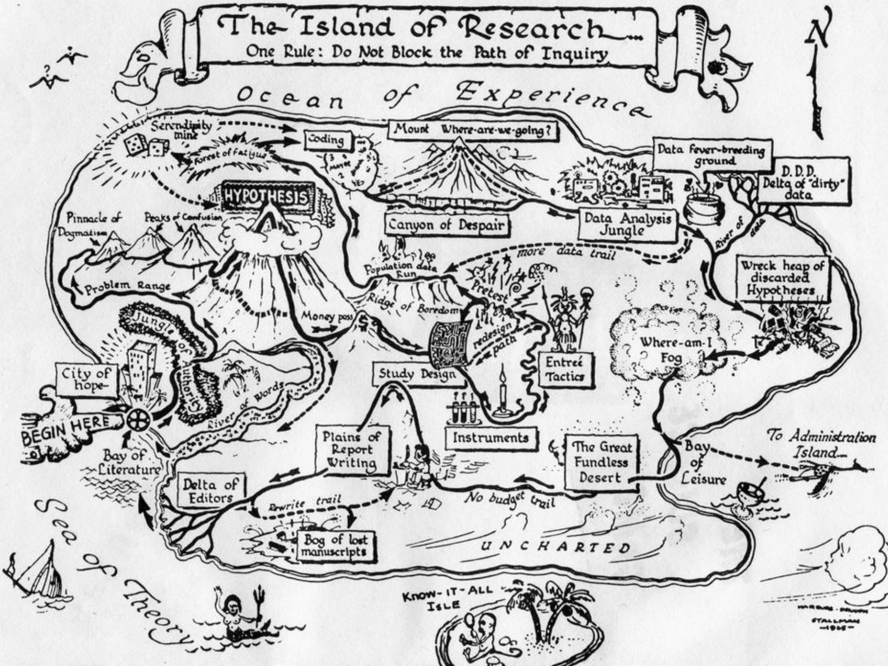
31st August 2017



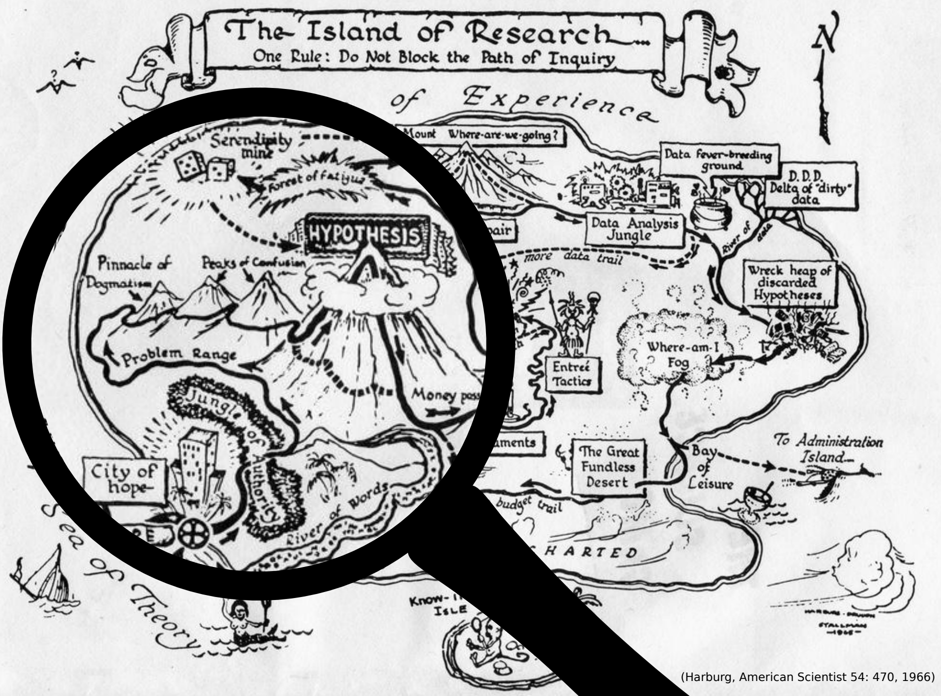
(Harburg, American Scientist 54: 470, 1966)



(Harburg, American Scientist 54: 470, 1966)

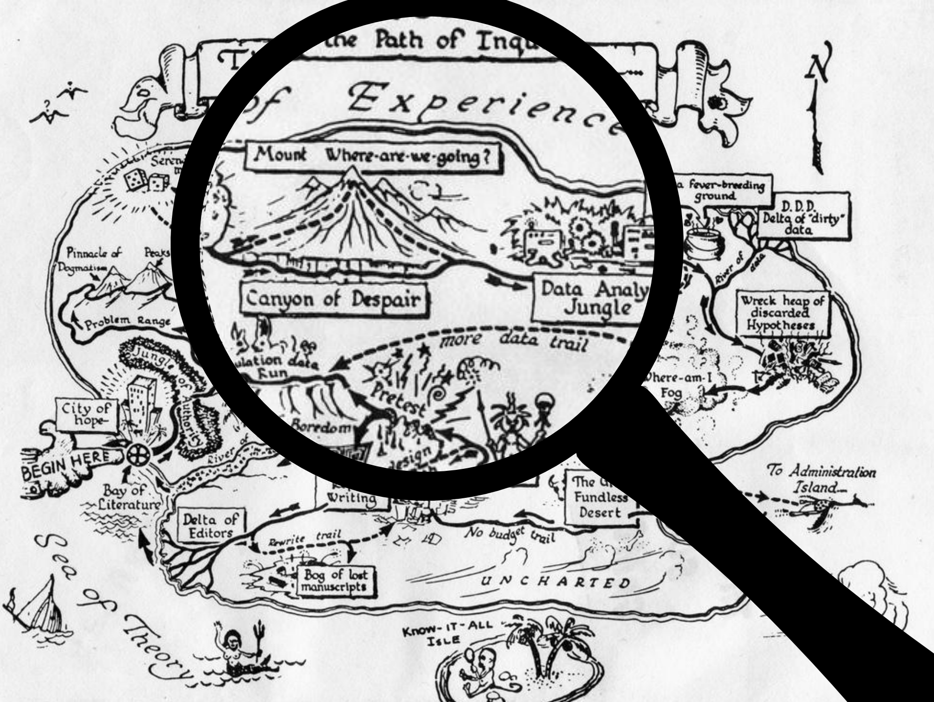


(Harburg, American Scientist 54: 470, 1966)

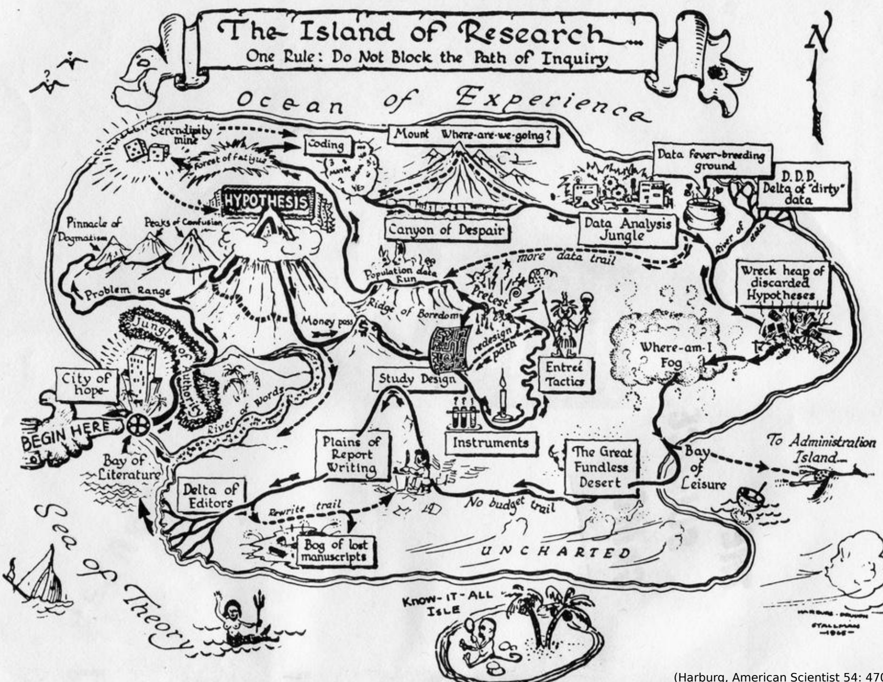


(Harburg, American Scientist 54: 470, 1966)

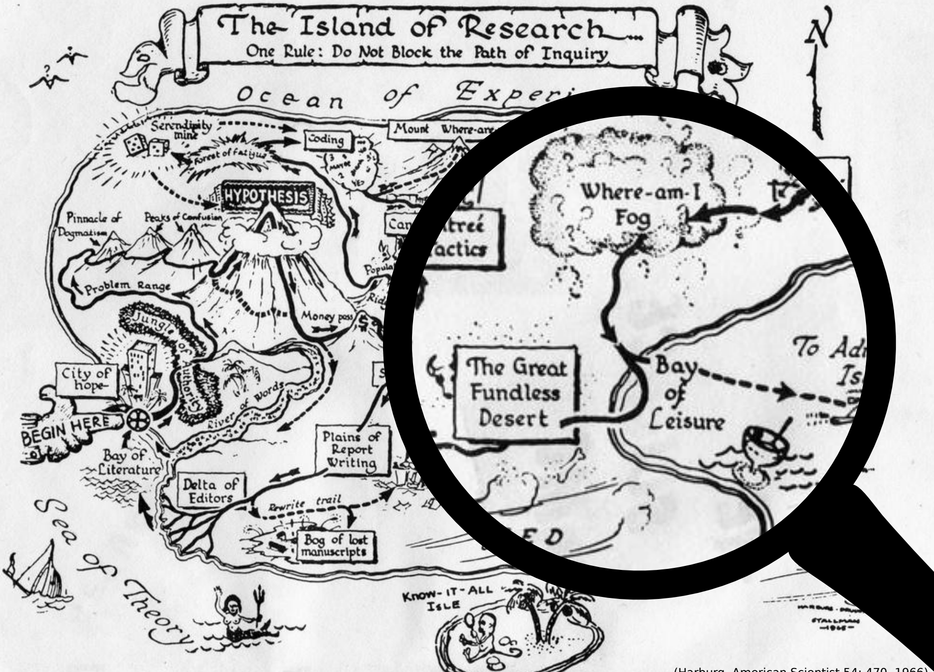
(Harburg, American Scientist 54: 470, 1966)



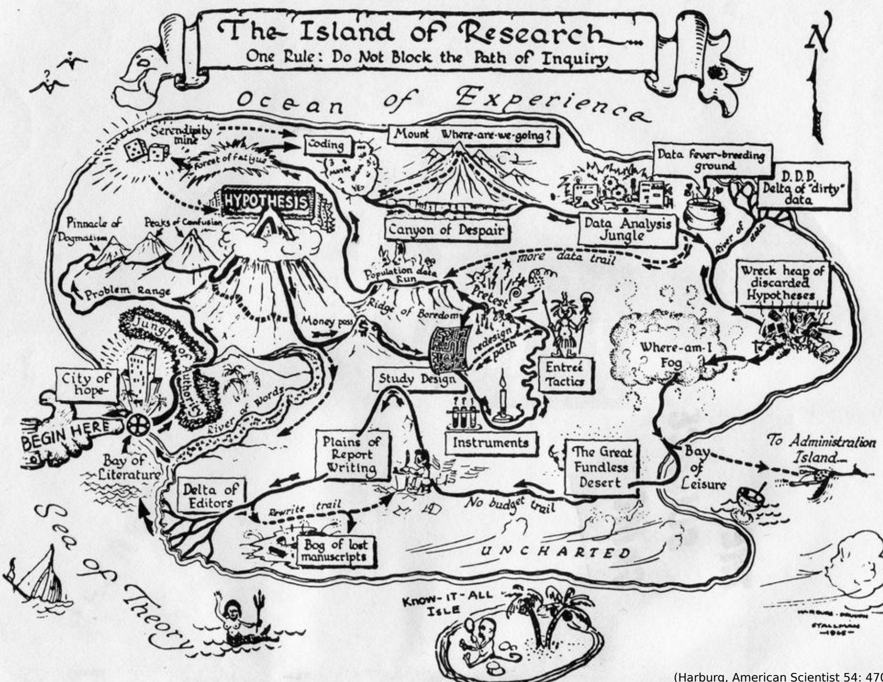
(Harburg, American Scientist 5



(Harburg, American Scientist 54: 470, 1966)



(Harburg, American Scientist 54: 470, 1966)



(Harburg, American Scientist 54: 470, 1966)

The Island of Research
One Rule: Do Not Block the Path of Inquiry
Ocean of Experience
N
Serendipity mine
Forest of Fatigue
Coding
Mount Where-are-we-going?
Data fever-breeding ground
D.D.D. Delta of "dirty" data
HYPOTHESIS
Pinnacle of Dogmatism
Peaks of Confusion
Canyon of Despair
Data Analysis Jungle
River of data
more data trail
Population data Run
Problem Range
Pretest
Wreck heap of discarded Hypotheses
Ridge of Boredom
Money pass
redesign path
Jungle of Authority
Where-am-I Fog
City of hope
Entreé Tactics
River of Words
BEGIN HERE
Rewrite trail
Bay of Literature
Bog of lost manuscripts
Bay of Leisure
To Administration Island
KNOW-IT-ALL ISLE
Sea of The

(Harburg, American Scientist 54: 470, 1966)

Progress in data science

- In the 60's, data science was very difficult.
- Today it's easier.

  We have

  - databases to store and access large amounts of data
  - clusters to parallelise the computing
  - the framework of statistical modelling and inference to provide the basic principles for analysing data.
- Challenge to further progress:
  - The basic principles do not take computational cost into account.
  - For complex data and models, exact inference is computationally impossible.
  - Good approximate solutions are needed.

Message of the talk

We can use machine learning to perform highly efficient approximate inference for intractable models.

Program

Introduction to statistical inference
- Likelihood function
- Case of exact inference

Models where exact inference is intractable
- Unnormalised models
- Generative models

Inference for unnormalised models
- Solution via logistic regression
- Application in unsupervised deep learning

Inference for generative models
- General overview
- Solution via logistic regression

Program

Introduction to statistical inference
- Likelihood function
- Case of exact inference

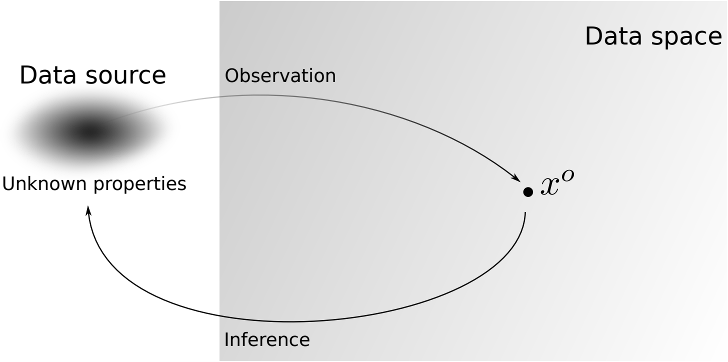
Models where exact inference is intractable
- Unnormalised models
- Generative models

Inference for unnormalised models
- Solution via logistic regression
- Application in unsupervised deep learning

Inference for generative models
- General overview
- Solution via logistic regression

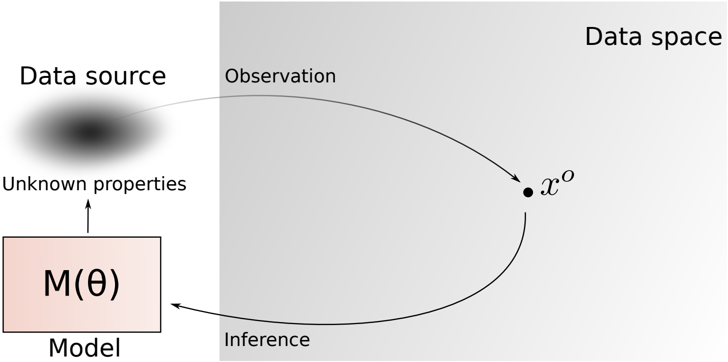
Goal of statistical inference

- Goal: Given data $\mathbf{x}^o$, learn about properties of its source
- Enables decision making, predictions, ...



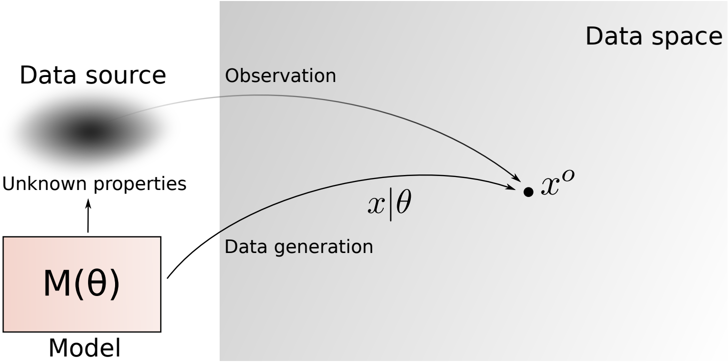
General approach

- Set up a model with potential properties $\boldsymbol{\theta}$ (hypotheses)
- See which $\boldsymbol{\theta}$ are in line with the observed data $\mathbf{x}^o$



The likelihood function $L(\boldsymbol{\theta})$

- Measures agreement between $\boldsymbol{\theta}$ and the observed data $\mathbf{x}^o$
- Probability to generate data like $\mathbf{x}^o$ if hypothesis $\boldsymbol{\theta}$ holds



Performing statistical inference

- If $L(\boldsymbol{\theta})$ is known, inference is straightforward
- Maximum likelihood estimation

$$\hat{\boldsymbol{\theta}} = \operatorname{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) \tag{1}$$

- Bayesian inference

$$p(\boldsymbol{\theta}|\mathbf{x}^o) \propto p(\boldsymbol{\theta}) \times L(\boldsymbol{\theta}) \tag{2}$$
$$\text{posterior} \propto \text{prior} \times \text{likelihood}$$

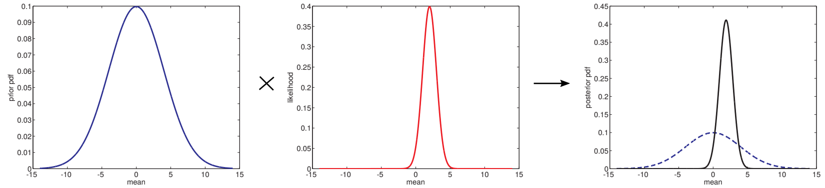
Allows us to learn from data by updating probabilities

Model specification

- Textbook: model $\equiv$ family of probability density functions
- Probability density functions (pdfs) $p(\mathbf{x}|\boldsymbol{\theta})$ satisfy

$$\underbrace{p(\mathbf{x}|\boldsymbol{\theta}) \geq 0}_{\text{non-negativity}} \qquad \underbrace{\int p(\mathbf{x}|\boldsymbol{\theta})\mathrm{d}\mathbf{x} = 1}_{\text{normalisation}} \tag{3}$$

- Likelihood function $L(\boldsymbol{\theta}) \propto p(\mathbf{x}^o|\boldsymbol{\theta})$
- Closed form solutions are possible



prior likelihood posterior

Program

Intractable models I worked on

- Not all models are specified as family of pdfs $p(\mathbf{x}|\boldsymbol{\theta})$.
- I worked on

  1. Unnormalised models
  2. Generative models with unobserved variables

- The models are rather different, common point:

  Multiple integrals needed to be computed to represent the models in terms of pdfs $p(\mathbf{x}|\boldsymbol{\theta})$.

- Solving the integrals exactly is computationally impossible. (curse of dimensionality)

  $\Rightarrow$ No model pdfs $p(\mathbf{x}|\boldsymbol{\theta})$
  $\Rightarrow$ No likelihood function $L(\boldsymbol{\theta}) \propto p(\mathbf{x}^o|\boldsymbol{\theta})$
  $\Rightarrow$ No exact inference

Unnormalised models

- Used for modelling
  - images (Markov random fields)
  - text (neural probabilistic language models)
  - social networks (exponential random graphs)
  - ...
- Specified via a non-negative function $\phi(\mathbf{x}|\boldsymbol{\theta}) \propto p(\mathbf{x}|\boldsymbol{\theta})$,

$$\int \cdots \int \phi(\mathbf{x}|\boldsymbol{\theta}) \mathrm{d}\mathbf{x} = Z(\boldsymbol{\theta}) \neq 1 \qquad p(\mathbf{x}|\boldsymbol{\theta}) = \frac{\phi(\mathbf{x}|\boldsymbol{\theta})}{Z(\boldsymbol{\theta})} \tag{4}$$

- Advantage: Specifying unnormalised models is often easier than specifying normalised models
- Disadvantage: Integral defining $Z(\boldsymbol{\theta})$, called the partition function, can generally not be computed.

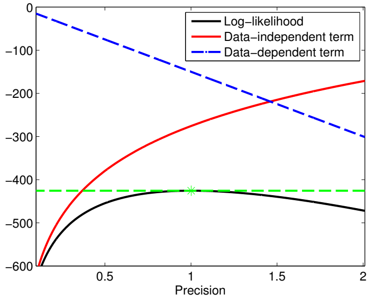
  $\Rightarrow$ Likelihood function is intractable.

Intractable partition function implies intractable likelihood

- Consider $p(x;\theta) = \frac{\phi(x;\theta)}{Z(\theta)} = \frac{\exp\left(-\theta\frac{x^2}{2}\right)}{\sqrt{2\pi/\theta}}$
- Log-likelihood function for precision $\theta \geq 0$

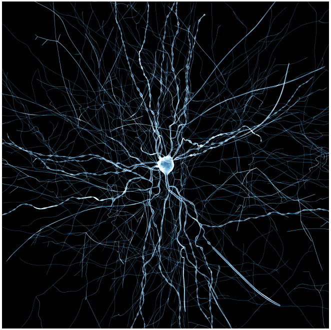
$$\ell(\theta) = -n\log\sqrt{\frac{2\pi}{\theta}} - \theta\sum_{i=1}^{n}\frac{x_i^2}{2} \tag{5}$$

- Data-dependent (blue) and independent part (red) balance each other.
- If $Z(\theta)$ is intractable, $\ell(\theta)$ is intractable.



Generative models

- Models which specify a mechanism for generating data $\mathbf{x}^o$
  - e.g. stochastic dynamical systems
  - computer models / simulators of some complex biological process
  - aka: simulator-based models, implicit models, probabilistic programs
- Widely used
  - Evolutionary biology: Simulating evolution
  - Neuroscience: Simulating neural circuits
  - Health science: Simulating the spread of an infectious disease



Generative models

- Advantage: detailed and realistic modelling
- Disadvantage: likelihood function is generally intractable due to unobserved variables.
- To compute $p(\mathbf{x}|\boldsymbol{\theta})$ one has to take into account all possible states of the unobserved variables (marginalisation)

$$p(\mathbf{x}|\boldsymbol{\theta}) = \int \cdots \int p(\mathbf{x}, \mathbf{z}|\theta)\mathrm{d}\mathbf{z} \tag{6}$$

- This is generally computationally impossible.

Program

Problem statement

- Task: Estimate the parameters $\boldsymbol{\theta}$ of a parametric model $p(.|\boldsymbol{\theta})$ of a d dimensional random vector $\mathbf{x}$
- Given: Data $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ (iid)
- Given: Unnormalised model $\phi(.|\boldsymbol{\theta})$

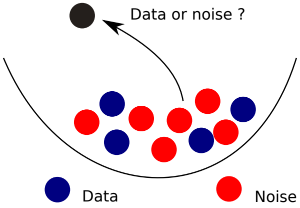
$$\int_{\boldsymbol{\xi}} \phi(\boldsymbol{\xi}|\boldsymbol{\theta}) \, d\boldsymbol{\xi} = Z(\boldsymbol{\theta}) \neq 1 \qquad p(\mathbf{x}|\boldsymbol{\theta}) = \frac{\phi(\mathbf{x}|\boldsymbol{\theta})}{Z(\boldsymbol{\theta})} \tag{7}$$

Normalising partition function $Z(\boldsymbol{\theta})$ not known / computable.

Basic idea

- Formulate the estimation problem as a classification problem: observed data vs. auxiliary "noise" (with known properties)
- Successful classification $\equiv$ learn the differences between the data and the noise
- differences $+$ known noise properties $\Rightarrow$ properties of the data

- Unsupervised learning by supervised learning
- We used (nonlinear) logistic regression for classification

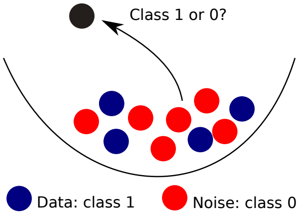


Logistic regression (1/2)

- Let $\mathbf{Y} = (\mathbf{y}_1, \ldots \mathbf{y}_m)$ be a sample from a random variable $\mathbf{y}$ with known (auxiliary) distribution p_{noise}.
- Introduce labels and form regression function:

$$P(C = 1|\mathbf{u}; \boldsymbol{\theta}) = \frac{1}{1 + G(\mathbf{u}; \boldsymbol{\theta})} \qquad G(\mathbf{u}; \boldsymbol{\theta}) \geq 0 \tag{8}$$

- Determine the parameters $\boldsymbol{\theta}$ such that $P(C = 1|\mathbf{u}; \boldsymbol{\theta})$ is
  - large for most $\mathbf{x}_i$
  - small for most $\mathbf{y}_i$.



Logistic regression (2/2)

- Maximise (rescaled) conditional log-likelihood using the labelled data $\{(\mathbf{x}_1, 1), \ldots, (\mathbf{x}_n, 1), (\mathbf{y}_1, 0), \ldots, (\mathbf{y}_m, 0)\}$,

$$J_n^{\mathrm{NCE}}(\boldsymbol{\theta}) = \frac{1}{n}\left(\sum_{i=1}^{n} \log P(C = 1|\mathbf{x}_i; \boldsymbol{\theta}) + \sum_{i=1}^{m} \log \left[P(C = 0|\mathbf{y}_i; \boldsymbol{\theta})\right]\right)$$

- For large sample sizes n and m, $\hat{\boldsymbol{\theta}}$ satisfying

$$G(\mathbf{u}; \hat{\boldsymbol{\theta}}) = \frac{m}{n}\frac{p_{\text{noise}}(\mathbf{u})}{p_{\text{data}}(\mathbf{u})} \tag{9}$$

is maximising $J_n^{\mathrm{NCE}}(\boldsymbol{\theta})$. Without any normalisation constraints.

proof

Noise-contrastive estimation

(Gutmann and Hyvärinen, 2010; 2012)

(Gutmann and Hirayama, 2011)

- Assume unnormalised model $\phi(.|\boldsymbol{\theta})$ is parametrised such that its scale can vary freely.

$$\boldsymbol{\theta} \to (\boldsymbol{\theta}; c) \qquad\qquad \phi(\mathbf{u}|\boldsymbol{\theta}) \to \exp(c)\phi(\mathbf{u}|\boldsymbol{\theta}) \tag{10}$$

- Noise-contrastive estimation:
  1. Choose p_{noise}
  2. Generate auxiliary data $\mathbf{Y}$
  3. Estimate $\boldsymbol{\theta}$ via logistic regression with

$$G(\mathbf{u}; \boldsymbol{\theta}) = \frac{m}{n} \frac{p_{\text{noise}}(\mathbf{u})}{\phi(\mathbf{u}|\boldsymbol{\theta})}. \tag{11}$$

Noise-contrastive estimation

(Gutmann and Hyvärinen, 2010; 2012)

(Gutmann and Hirayama, 2011)

- Assume unnormalised model $\phi(.|\boldsymbol{\theta})$ is parametrised such that its scale can vary freely.

$$\boldsymbol{\theta} \to (\boldsymbol{\theta}; c) \qquad \phi(\mathbf{u}|\boldsymbol{\theta}) \to \exp(c)\phi(\mathbf{u}|\boldsymbol{\theta}) \tag{10}$$

- Noise-contrastive estimation:
  1. Choose p_{noise}
  2. Generate auxiliary data $\mathbf{Y}$
  3. Estimate $\boldsymbol{\theta}$ via logistic regression with

$$G(\mathbf{u}; \boldsymbol{\theta}) = \frac{m}{n} \frac{p_{\text{noise}}(\mathbf{u})}{\phi(\mathbf{u}|\boldsymbol{\theta})}. \tag{11}$$

- $G(\mathbf{u}; \boldsymbol{\theta}) \to \frac{m}{n} \frac{p_{\text{noise}}(\mathbf{u})}{p_{\text{data}}(\mathbf{u})} \quad \Rightarrow \quad \phi(\mathbf{u}|\boldsymbol{\theta}) \to p_{\text{data}}(\mathbf{u})$

Example

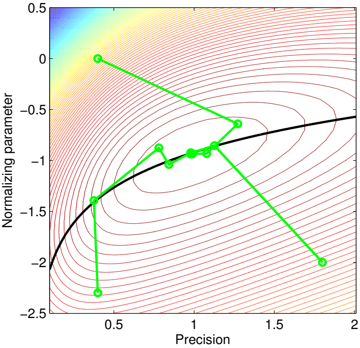
- Unnormalised Gaussian:

$$\phi(x; \boldsymbol{\theta}) = \exp(\theta_2) \exp\left(-\theta_1 \frac{x^2}{2}\right), \quad \theta_1 > 0,\ \theta_2 \in \mathbb{R}, \qquad (12)$$

- Parameters: θ_1 (precision), $\theta_2 \equiv c$ (scaling parameter)

Contour plot of $J_n^{\mathrm{NCE}}(\boldsymbol{\theta})$:

- Gaussian noise with $\nu = m/n = 10$
- True precision $\theta_1^\star = 1$
- Black: normalised models
  Green: optimisation paths



Statistical properties

(Gutmann and Hyvärinen, 2012)

- Assume $p_{\text{data}} = p(.|\boldsymbol{\theta}^\star)$
- Consistency: As n increases,

$$\hat{\boldsymbol{\theta}}_n = \text{argmax}_{\boldsymbol{\theta}} J_n^{\text{NCE}}(\boldsymbol{\theta}), \tag{13}$$

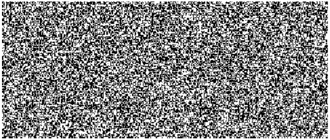
  converges in probability to $\boldsymbol{\theta}^\star$.
- Efficiency: As $\nu = m/n$ increases, for any valid choice of p_{noise}, noise-contrastive estimation tends to "perform as well" as MLE (it is asymptotically Fisher efficient).

Application examples

- Models of text: e.g. Mnih and Teh, 2012, *A fast and simple algorithm for training neural probabilistic language models*
- Models of images: e.g. Gutmann and Hyvärinen, 2013, *A three-layer model of natural image statistics*
- Machine translation: e.g. Zoph et al, 2016, *Simple, fast noise-contrastive estimation for large RNN vocabularies*
- Product recommendation: e.g. Tschiatschek et al, 2016, *Learning probabilistic submodular diversity models via noise contrastive estimation*

Unsupervised deep learning on natural images

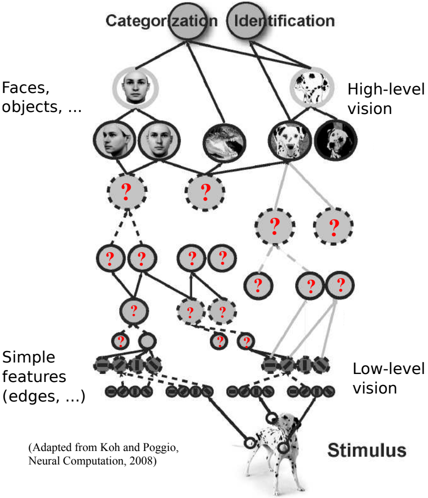
- Natural images $\equiv$ images which we see in our environment
- Understanding their properties is important
  - for modern image processing
  - for understanding biological visual systems



Unsupervised deep learning on natural images

- Rapid object recognition by feedforward processing
- Computations in middle layers poorly understood
- Our approach: learn the computations from data
- Idea: the units indicate how probable an input image is.
  (up to normalisation)

(Gutmann and Hyvärinen, 2013)

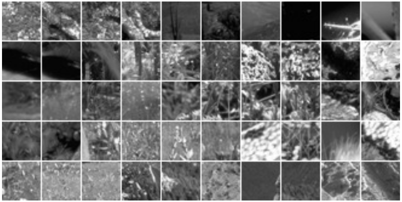


(Adapted from Koh and Poggio, Neural Computation, 2008)

Image data

Consider two kinds of image data:

1. Image patches of size 32 by 32, extracted from larger images (left).
2. "Tiny images" dataset, converted to grey scale: complete scenes downsampled to 32 by 32 images (right)
   (Torralba et al, TPAMI 2008)



Multi-layer model

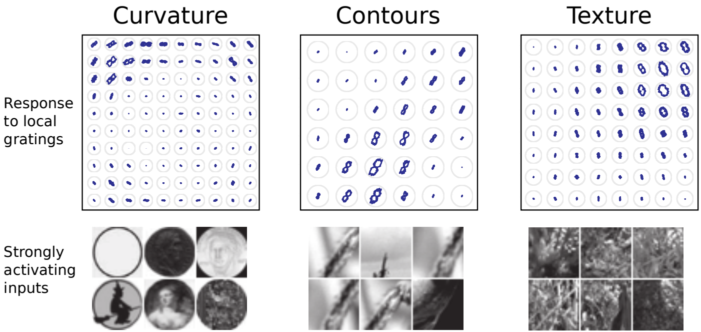
- Let $\mathbf{I}$ be a vectorised image. Processing layers:

$$
\begin{aligned}
\mathbf{x} &= \text{gain control}\,(\mathbf{I}) \\
y_i^{(1)} &= \max\left(\mathbf{w}_i^{(1)} \cdot \mathbf{x}, 0\right), && i = 1 \ldots 600 \\
y_i^{(2)} &= \log\left(\mathbf{w}_i^{(2)} \cdot (\mathbf{y}^{(1)})^2 + 1\right), && i = 1 \ldots 100 \\
\mathbf{z}^{(2)} &= \text{gain control}\left(\mathbf{y}^{(2)}\right) \\
y_i^{(3)} &= \max\left(\mathbf{w}_i^{(3)} \cdot \mathbf{z}^{(2)}, 0\right), && i = 1 \ldots 50
\end{aligned}
$$

  Gain control: centring, normalising the norm after whitening, possibly dimension reduction

- The outputs $y_i^{(3)}$ define how probable an input image is. (up to normalisation $\Rightarrow$ unnormalised model)
- The weights are the parameters to be learned ($> 2 \cdot 10^5$ parameters)
- Only constraint: $w_{ki}^{(2)} \geq 0$.

- 1st layer: $\approx$ local Fourier transform (Gabor filters)
- 2nd layer: local max-pooling
- 3rd layer: emergence of units sensitive to curvature, longer contours, and texture
- Close link to neural processing in the visual cortex



(Gutmann and Hyvärinen, 2013)

Program

- Introduction to statistical inference
  - Likelihood function
  - Case of exact inference
- Models where exact inference is intractable
  - Unnormalised models
  - Generative models
- Inference for unnormalised models
  - Solution via logistic regression
  - Application in unsupervised deep learning
- Inference for generative models
  - General overview
  - Solution via logistic regression

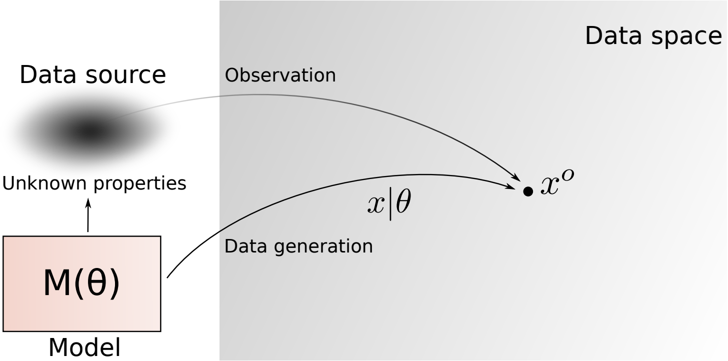
Problem statement

Perform Bayesian inference for models where

1. the likelihood function is too costly to compute
2. sampling – simulating data – from the model is possible

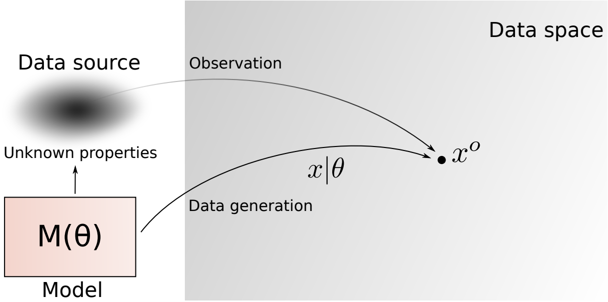
The likelihood function $L(\boldsymbol{\theta})$

- Probability that the model generates data like $\mathbf{x}^o$ when using parameter value $\boldsymbol{\theta}$
- Generally well defined but intractable for simulator-based models



Three foundational issues

1. How should we assess whether $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
2. How should we compute the probability of the event $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
3. For which values of $\boldsymbol{\theta}$ should we compute it?



Likelihood: Probability that the model generates data like $\mathbf{x}^o$ for parameter value $\boldsymbol{\theta}$

Approximate Bayesian computation

Recent review: Lintusaari et al (2017) "Fundamentals and recent developments in approximate Bayesian computation", Systematic Biology

1. How should we assess whether $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
   $\Rightarrow$ Check whether $||T(\mathbf{x}_{\boldsymbol{\theta}}) - T(\mathbf{x}^o)|| \leq \epsilon$
2. How should we compute the proba of the event $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
   $\Rightarrow$ By counting
3. For which values of $\boldsymbol{\theta}$ should we compute it?
   $\Rightarrow$ Sample from the prior (or other proposal distributions)

Approximate Bayesian computation

Recent review: Lintusaari et al (2017) "Fundamentals and recent developments in approximate Bayesian computation", Systematic Biology

1. How should we assess whether $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
   $\Rightarrow$ Check whether $||T(\mathbf{x}_{\boldsymbol{\theta}}) - T(\mathbf{x}^o)|| \leq \epsilon$
2. How should we compute the proba of the event $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
   $\Rightarrow$ By counting
3. For which values of $\boldsymbol{\theta}$ should we compute it?
   $\Rightarrow$ Sample from the prior (or other proposal distributions)

Difficulties:

- Choice of summary statistics $T()$ and threshold ϵ
- Typically high computational cost

Synthetic likelihood

(Simon Wood, Nature, 2010)

1. How should we assess whether $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
2. How should we compute the proba of the event $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
   - $\Rightarrow$ Compute summary statistics $\mathbf{t}_{\boldsymbol{\theta}} = T(\mathbf{x}_{\boldsymbol{\theta}})$
   - $\Rightarrow$ Model their distribution as a Gaussian
   - $\Rightarrow$ Compute likelihood function with $T(\mathbf{x}^o)$ as observed data
3. For which values of $\boldsymbol{\theta}$ should we compute it?
   - $\Rightarrow$ Use obtained "synthetic" likelihood function as part of a Monte Carlo method

Synthetic likelihood

(Simon Wood, Nature, 2010)

1. How should we assess whether $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
2. How should we compute the proba of the event $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
   - $\Rightarrow$ Compute summary statistics $\mathbf{t}_{\boldsymbol{\theta}} = T(\mathbf{x}_{\boldsymbol{\theta}})$
   - $\Rightarrow$ Model their distribution as a Gaussian
   - $\Rightarrow$ Compute likelihood function with $T(\mathbf{x}^o)$ as observed data
3. For which values of $\boldsymbol{\theta}$ should we compute it?
   - $\Rightarrow$ Use obtained "synthetic" likelihood function as part of a Monte Carlo method

Difficulties:

- Choice of summary statistics $T()$
- Gaussianity assumption may not hold
- Typically high computational cost

Overview of some of my work

1. How should we assess whether $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
   - $\Rightarrow$ Use classification (Gutmann et al, 2014, 2017)

2. How should we compute the proba of the event $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
3. For which values of $\boldsymbol{\theta}$ should we compute it?
   - $\Rightarrow$ Use Bayesian optimisation (Gutmann and Corander, 2013-2016)
     Compared to standard approaches: speed-up by a factor of 1000 more

1. How should we assess whether $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
2. How should we compute the proba of the event $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
   - $\Rightarrow$ Use density ratio estimation / logistic regression
     (Dutta et al, 2016, arXiv:1611.10242)

Overview of some of my work

1. How should we assess whether $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?

   $\Rightarrow$ Use classification (Gutmann et al, 2014, 2017)

2. How should we compute the proba of the event $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
3. For which values of $\boldsymbol{\theta}$ should we compute it?

   $\Rightarrow$ Use Bayesian optimisation (Gutmann and Corander, 2013-2016)
   Compared to standard approaches: speed-up by a factor of 1000 more

1. How should we assess whether $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?
2. How should we compute the proba of the event $\mathbf{x}_{\boldsymbol{\theta}} \equiv \mathbf{x}^o$?

   $\Rightarrow$ Use density ratio estimation / logistic regression
   (Dutta et al, 2016, arXiv:1611.10242)

Basic idea

(Dutta et al, 2016, arXiv:1611.10242)

- Frame posterior estimation as ratio estimation problem

$$p(\boldsymbol{\theta}|\mathbf{x}) = \frac{p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})} = p(\boldsymbol{\theta})r(\mathbf{x},\boldsymbol{\theta}), \qquad r(\mathbf{x},\boldsymbol{\theta}) = \frac{p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})}$$

- Estimating $r(\mathbf{x},\boldsymbol{\theta})$ is the difficult part since $p(\mathbf{x}|\boldsymbol{\theta})$ unknown.
- Estimate $\hat{r}(\mathbf{x},\boldsymbol{\theta})$ yields estimate of the likelihood function and posterior

$$\hat{L}(\boldsymbol{\theta}) \propto \hat{r}(\mathbf{x}^o,\boldsymbol{\theta}), \qquad \hat{p}(\boldsymbol{\theta}|\mathbf{x}^o) = p(\boldsymbol{\theta})\hat{r}(\mathbf{x}^o,\boldsymbol{\theta}). \tag{14}$$

Basic idea

(Dutta et al, 2016, arXiv:1611.10242)

- Frame posterior estimation as ratio estimation problem

$$p(\boldsymbol{\theta}|\mathbf{x}) = \frac{p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})} = p(\boldsymbol{\theta})r(\mathbf{x},\boldsymbol{\theta}), \qquad r(\mathbf{x},\boldsymbol{\theta}) = \frac{p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})}$$

- Estimating $r(\mathbf{x},\boldsymbol{\theta})$ is the difficult part since $p(\mathbf{x}|\boldsymbol{\theta})$ unknown.
- Estimate $\hat{r}(\mathbf{x},\boldsymbol{\theta})$ yields estimate of the likelihood function and posterior

$$\hat{L}(\boldsymbol{\theta}) \propto \hat{r}(\mathbf{x}^o,\boldsymbol{\theta}), \qquad \hat{p}(\boldsymbol{\theta}|\mathbf{x}^o) = p(\boldsymbol{\theta})\hat{r}(\mathbf{x}^o,\boldsymbol{\theta}). \tag{14}$$

- Often more practical to estimate log-ratio $h(\mathbf{x},\boldsymbol{\theta}) = \log r(\mathbf{x},\boldsymbol{\theta})$

$$\hat{L}(\boldsymbol{\theta}) \propto \exp(\hat{h}(\mathbf{x}^o,\boldsymbol{\theta})), \quad \hat{p}(\boldsymbol{\theta}|\mathbf{x}^o) = p(\boldsymbol{\theta})\exp(\hat{h}(\mathbf{x}^o,\boldsymbol{\theta})) \tag{15}$$

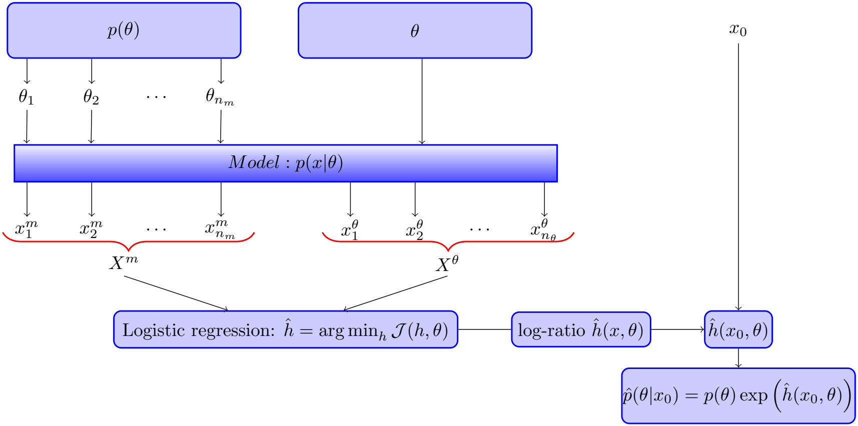
Estimating the posterior

- From theory of noise-contrastive estimation: ratio $r(\mathbf{x}, \boldsymbol{\theta})$, or log-ratio $h(\mathbf{x}, \boldsymbol{\theta})$ can be estimated by logistic regression
- Formulate classification problem with
  - one class: data sampled from $p(\mathbf{x}|\boldsymbol{\theta})$
  - other class: data sampled from marginal $p(\mathbf{x})$
- Logistic regression gives (point-wise in $\boldsymbol{\theta}$)

$$\hat{h}(\mathbf{x}, \boldsymbol{\theta}) \to \log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{p(\mathbf{x})} = \log r(\mathbf{x}, \boldsymbol{\theta}) \qquad (16)$$

- We operate on synthetic data only; can generate as much data as we wish

Estimating the posterior



(Dutta et al, 2016, arXiv:1611.10242)

Auxiliary model

- We need to specify a model for h.
- For simplicity: linear model

$$h(\mathbf{x}) = \sum_{i=1}^{b} \beta_i \psi_i(\mathbf{x}) = \beta^\top \psi(\mathbf{x}) \tag{17}$$

where $\psi_i(\mathbf{x})$ are summary statistics
- More complex models possible
- Simple linear model leads to a generalisation of synthetic likelihood (Dutta et al, 2016, arXiv:1611.10242)
- L_1 penalty on β for weighing and selecting summary statistics

Application to ARCH model

- Model:

$$x^{(t)} = \theta_1 x^{(t-1)} + e^{(t)} \tag{18}$$

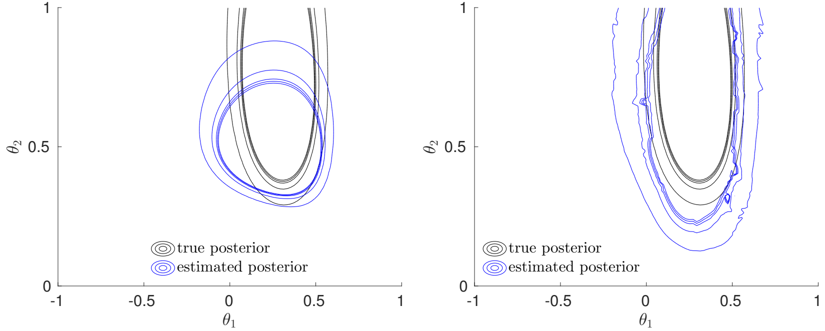
$$e^{(t)} = \xi^{(t)} \sqrt{0.2 + \theta_2 (e^{(t-1)})^2} \tag{19}$$

  $\xi^{(t)}$ and $e^{(0)}$ independent standard normal r.v., $x^{(0)} = 0$
- 100 time points
- Parameters: $\theta_1 \in (-1, 1), \quad \theta_2 \in (0, 1)$
- Uniform prior on θ_1, θ_2

Application to ARCH model

- Summary statistics $\psi_i(\mathbf{x})$:
  - auto-correlations with lag one to five
  - all (unique) pairwise combinations of them
  - a constant
- To check robustness: 50% irrelevant summary statistics (drawn from standard normal)
- Comparison with synthetic likelihood with equivalent set of summary statistics (relevant sum. stats. only)

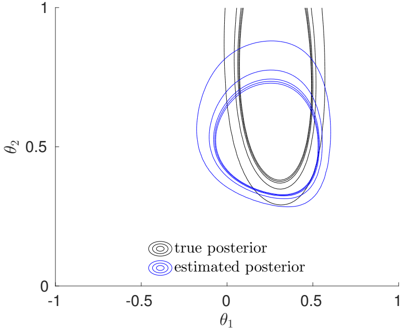
Example posterior



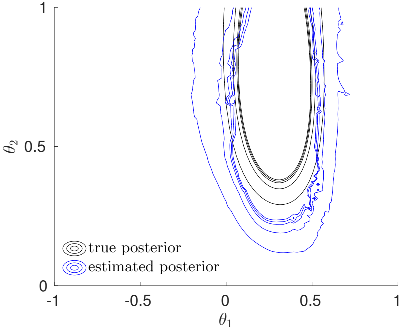
(a) synthetic likelihood

(b) proposed method

Example posterior



(c) synthetic likelihood

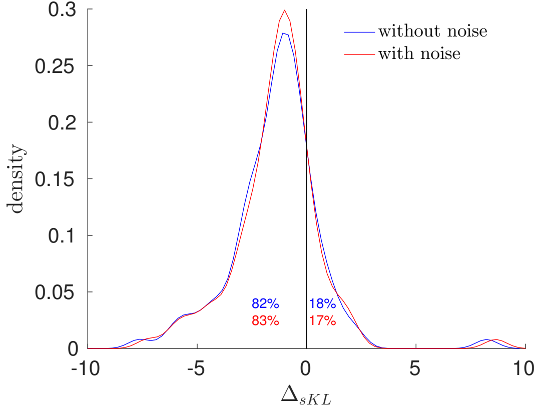


(d) proposed method subject to noise

Systematic analysis

- Symmetrised Kullback-Leibler divergence between estimated and true posterior
- Point-wise comparison with synthetic likelihood (100 data sets)

Δ_{sKL} = SKL for proposed method − SKL for synthetic likelihood



Key results

For details, see arXiv:1611.10242v1

- Frame the problem of Bayesian inference with intractable generative models as ratio estimation problem
- Use logistic regression to solve the problem
- Approach includes synthetic likelihood as special case
- For same summary statistics, typically more accurate inferences than the synthetic likelihood
- Robustness to irrelevant summary statistics thanks to regularisation
- Enables selection of relevant summary statistics
- No threshold to choose (unlike in ABC)

Conclusions

- Statistical modelling and inference are part of the foundations of data science.
  - They are not concerned with computational cost.
  - Exact inference is impossible for complex models.
- Unnormalised models
  - Noise-contrastive estimation
  - Formulated the inference problem as a classification problem
- Generative models
  - General overview
  - Formulated the inference problem as a classification problem

Conclusions

- Statistical modelling and inference are part of the foundations of data science.
  - They are not concerned with computational cost.
  - Exact inference is impossible for complex models.
- Unnormalised models
  - Noise-contrastive estimation
  - Formulated the inference problem as a classification problem
- Generative models
  - General overview
  - Formulated the inference problem as a classification problem

By re-framing inference problems,
we can use machine learning to perform highly efficient
approximate inference for intractable models.

Maximiser of the NCE objective function

Appendix

Maximiser of the NCE objective function

Proof of Equation (9)

For large sample sizes n and m, $\hat{\boldsymbol{\theta}}$ satisfying

$$G(\mathbf{u}; \hat{\boldsymbol{\theta}}) = \frac{m}{n} \frac{p_{\text{noise}}(\mathbf{u})}{p_{\text{data}}(\mathbf{u})}$$

is maximising $J_n^{\text{NCE}}(\boldsymbol{\theta})$,

$$J_n^{\text{NCE}}(\boldsymbol{\theta}) = \frac{1}{n}\left(\sum_{i=1}^{n} \log P(C=1|\mathbf{x}_i; \boldsymbol{\theta}) + \sum_{i=1}^{m} \log\left[P(C=0|\mathbf{y}_i; \boldsymbol{\theta})\right]\right)$$

without any normalisation constraints.

Proof of Equation (9)

$$
\begin{aligned}
J_n^{\mathrm{NCE}}(\boldsymbol{\theta}) &= \frac{1}{n}\left(\sum_{i=1}^{n} \log P(C=1|\mathbf{x}_i;\boldsymbol{\theta}) + \sum_{i=1}^{m} \log\left[P(C=0|\mathbf{y}_i;\boldsymbol{\theta})\right]\right) \\
&= \frac{1}{n}\sum_{t=1}^{n} \log P(C=1|\mathbf{x}_i;\boldsymbol{\theta}) + \frac{m}{n}\frac{1}{m}\sum_{t=1}^{m} \log\left[P(C=0|\mathbf{y}_i;\boldsymbol{\theta})\right]
\end{aligned}
$$

Fix the ratio $m/n = \nu$ and let $n \to \infty$ and $m \to \infty$. By law of large numbers, J_n^{NCE} converges to J^{NCE},

$$
J^{\mathrm{NCE}}(\boldsymbol{\theta}) = \mathrm{E}_{\mathbf{x}}\left(\log P(C=1|\mathbf{x};\boldsymbol{\theta})\right) + \nu\, \mathrm{E}_{\mathbf{y}}\left(\log P(C=0|\mathbf{y};\boldsymbol{\theta})\right) \quad (20)
$$

With $P(C=1|\mathbf{x};\boldsymbol{\theta}) = \frac{1}{1+G(\mathbf{x};\boldsymbol{\theta})}$ and $P(C=0|\mathbf{y};\boldsymbol{\theta}) = \frac{G(\mathbf{y};\boldsymbol{\theta})}{1+G(\mathbf{y};\boldsymbol{\theta})}$ we have

$$
\begin{aligned}
J^{\mathrm{NCE}}(\boldsymbol{\theta}) = -\,\mathrm{E}_{\mathbf{x}} \log(1+G(\mathbf{x};\boldsymbol{\theta})) + \nu\, \mathrm{E}_{\mathbf{y}} \log G(\mathbf{y};\boldsymbol{\theta}) - \\
\nu\, \mathrm{E}_{\mathbf{y}} \log\left(1+G(\mathbf{y};\boldsymbol{\theta})\right) \quad (21)
\end{aligned}
$$

Consider the objective $J^{\mathrm{NCE}}(\boldsymbol{\theta})$ as a function of $H = \log G$ rather than $\boldsymbol{\theta}$,

$$
\begin{aligned}
\mathcal{J}^{\mathrm{NCE}}(H) =& -\mathsf{E}_{\mathbf{x}} \log(1 + \exp H(\mathbf{x})) + \nu\, \mathsf{E}_{\mathbf{y}}\, H(\mathbf{y}) - \nu\, \mathsf{E}_{\mathbf{y}} \log\left(1 + \exp H(\mathbf{y})\right) \\
=& -\int p_{\mathrm{data}}(\boldsymbol{\xi}) \log(1 + \exp H(\boldsymbol{\xi})) d\boldsymbol{\xi} + \nu \int p_{\mathrm{noise}}(\boldsymbol{\xi}) H(\boldsymbol{\xi}) d\boldsymbol{\xi} \\
& - \nu \int p_{\mathrm{noise}}(\boldsymbol{\xi}) \log(1 + \exp H(\boldsymbol{\xi})) d\boldsymbol{\xi} \\
=& -\int \left(p_{\mathrm{data}}(\boldsymbol{\xi}) + \nu p_{\mathrm{noise}}(\boldsymbol{\xi})\right) \log(1 + \exp H(\boldsymbol{\xi})) d\boldsymbol{\xi} + \\
& \nu \int p_{\mathrm{noise}}(\boldsymbol{\xi}) H(\boldsymbol{\xi}) d\boldsymbol{\xi}
\end{aligned}
$$

We now expand $\mathcal{J}^{\mathrm{NCE}}(H + \epsilon q)$ around H for an arbitrary function q and a small scalar ϵ.

With

$$
\begin{aligned}
\log(1+\exp\left[H(\boldsymbol{\xi})+\epsilon q(\boldsymbol{\xi})\right]) = & \log(1+\exp H(\boldsymbol{\xi})) + \frac{\epsilon q(\boldsymbol{\xi})}{1+\exp(-H(\boldsymbol{\xi}))} \\
& + \frac{\epsilon^2}{2}\frac{q(\boldsymbol{\xi})}{1+\exp(-H(\boldsymbol{\xi}))}\frac{q(\boldsymbol{\xi})}{1+\exp(H(\boldsymbol{\xi}))} \\
& + O(\epsilon^3)
\end{aligned}
$$

we have

$$
\begin{aligned}
\mathcal{J}^{\mathrm{NCE}}(H+\epsilon q) = & -\int (p_{\mathsf{data}}(\boldsymbol{\xi}) + \nu p_{\mathsf{noise}}(\boldsymbol{\xi}))\log(1+\exp H(\boldsymbol{\xi}))d\boldsymbol{\xi} \\
& -\epsilon\int \frac{p_{\mathsf{data}}(\boldsymbol{\xi})+\nu p_{\mathsf{noise}}(\boldsymbol{\xi})}{1+\exp(-H(\boldsymbol{\xi}))} q(\boldsymbol{\xi})d\boldsymbol{\xi} \\
& -\frac{\epsilon^2}{2}\int \frac{p_{\mathsf{data}}(\boldsymbol{\xi})+\nu p_{\mathsf{noise}}(\boldsymbol{\xi})}{1+\exp(-H(\boldsymbol{\xi}))}\frac{q(\boldsymbol{\xi})^2}{1+\exp(H(\boldsymbol{\xi}))}d\boldsymbol{\xi} \\
& +\nu\int p_{\mathsf{noise}}(\boldsymbol{\xi})H(\boldsymbol{\xi})d\boldsymbol{\xi} + \epsilon\nu\int p_{\mathsf{noise}}(\boldsymbol{\xi})q(\boldsymbol{\xi})d\boldsymbol{\xi} + O(\epsilon^3)
\end{aligned}
$$

Collecting terms gives:

$$
\begin{aligned}
\mathcal{J}^{\mathrm{NCE}}(H+\epsilon q) = &\mathcal{J}^{\mathrm{NCE}}(H) + \\
&\epsilon \int \left(\nu p_{\text{noise}}(\boldsymbol{\xi}) - \frac{p_{\text{data}}(\boldsymbol{\xi}) + \nu p_{\text{noise}}(\boldsymbol{\xi})}{1+\exp(-H(\boldsymbol{\xi}))} \right) q(\boldsymbol{\xi}) d\boldsymbol{\xi} \\
&- \frac{\epsilon^2}{2} \int \frac{p_{\text{data}}(\boldsymbol{\xi}) + \nu p_{\text{noise}}(\boldsymbol{\xi})}{1+\exp(-H(\boldsymbol{\xi}))} \frac{q(\boldsymbol{\xi})^2}{1+\exp(H(\boldsymbol{\xi}))} d\boldsymbol{\xi} + O(\epsilon^3)
\end{aligned}
$$

The second-order term is negative for all (non-trivial) q and H. The first-order term is zero for all q if and only if

$$
\begin{aligned}
\nu p_{\text{noise}}(\boldsymbol{\xi}) &= \frac{p_{\text{data}}(\boldsymbol{\xi}) + \nu p_{\text{noise}}(\boldsymbol{\xi})}{1+\exp(-H^*(\boldsymbol{\xi}))} \\
\nu p_{\text{noise}}(\boldsymbol{\xi}) + \nu p_{\text{noise}}(\boldsymbol{\xi}) \exp(-H^*(\boldsymbol{\xi})) &= p_{\text{data}}(\boldsymbol{\xi}) + \nu p_{\text{noise}}(\boldsymbol{\xi}) \\
\exp(-H^*(\boldsymbol{\xi})) &= \frac{p_{\text{data}}(\boldsymbol{\xi})}{\nu p_{\text{noise}}(\boldsymbol{\xi})}
\end{aligned}
$$

which shows that $\hat{\boldsymbol{\theta}}$ such that $G(\boldsymbol{\xi}; \hat{\boldsymbol{\theta}}) = \exp(H^*(\boldsymbol{\xi})) = \nu \frac{p_{\text{noise}}}{p_{\text{data}}}$ is maximising $J^{\mathrm{NCE}}(\boldsymbol{\theta})$.

back