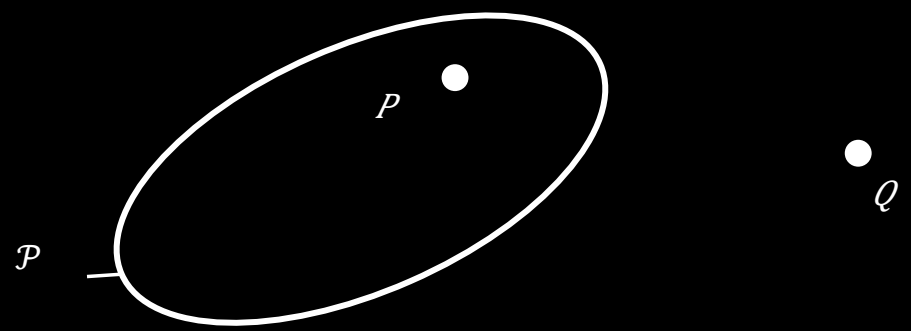
The ML Invasion of ABC

Richard Wilkinson

University of Sheffield

Learning $P_\theta(x)$

Learning Probabilistic Models



Assumptions on P:

- tractable sampling
- tractable parameter gradient with respect to sample
- tractable likelihood function

Approximate Bayesian Computation (ABC)

ABC algorithms are a collection of Monte Carlo methods used for calibrating simulators (ML: 'learning generative models') in a Bayesian manner.

- they do not require explicit knowledge of the likelihood function $P_\theta(x)$
- inference is done using simulation from the model (they are 'likelihood-free').

Approximate Bayesian Computation (ABC)

ABC algorithms are a collection of Monte Carlo methods used for calibrating simulators (ML: 'learning generative models') in a Bayesian manner.

- they do not require explicit knowledge of the likelihood function $P_\theta(x)$
- inference is done using simulation from the model (they are 'likelihood-free').

They are (were)

- Simple to implement
- Intuitive
- Embarrassingly parallelizable
- and can usually be applied

Originated in Genetics 1990s, studied by statistics in 2000s, and more recently in ML.

Learning $P_\theta(x)$

Learning Probabilistic Models

Integral Probability Metrics [Müller, 1997] [Sriperumbudur et al., 2010] $\gamma_{\mathcal{F}}(P,Q)=\sup_{f\in\mathcal{F}}\left\lvert\int f\,\mathrm{d}P-\int f\,\mathrm{d}Q\right\rvert$	Proper scoring rules [Gneiting and Raftery, 2007] $S(P,Q)=\int S(P,x)\,\mathrm{d}Q(x)$	f-divergences [Ali and Silvey, 1966] $D_f(P\|Q)=\int q(x)f(p(x)/q(x))\,\mathrm{d}x$

P: Expectation
Q: Expectation
Structure in $\mathcal{F}$
Examples:
- Energy statistic [Szekely, 1997]
- Kernel MMD [Gretton et al., 2012], [Smola et al., 2007]
- Wasserstein distance [Cuturi, 2013]
- DISCO Nets [Bouchacourt et al., 2016]
	• P: Distribution
	• Q: Expectation
	• Examples:
	  • Log-likelihood [Fisher, 1922], [Good, 1952]
	  • Quadratic score [Bernardo, 1979]
		• P: Distribution
		• Q: Distribution
		• Examples:
		  • Kullback-Leibler divergence [Kullback and Leibler, 1952]
		  • Jensen-Shannon divergence
		  • Total variation
		  • Pearson χ^2

Learning $P_\theta(x)$

Learning Probabilistic Models

Integral Probability Metrics [Müller, 1997] [Sriperumbudur et al., 2010]	Proper scoring rules [Gneiting and Raftery, 2007]	f-divergences [Ali and Silvey, 1966]
$\gamma_{\mathcal{F}}(P,Q)=\sup_{f\in\mathcal{F}}\left\lvert\int f\,\mathrm{d}P-\int f\,\mathrm{d}Q\right\rvert$	$S(P,Q)=\int S(P,x)\,\mathrm{d}Q(x)$	$D_f(P\|Q)=\int q(x)f(p(x)/q(x))\,\mathrm{d}x$

P: Expectation
Q: Expectation
Structure in $\mathcal{F}$
Examples:
- Energy statistic [Szekely, 1997]
- Kernel MMD [Gretton et al., 2012], [Smola et al., 2007]
- Wasserstein distance [Cuturi, 2013]
- DISCO Nets [Bouchacourt et al., 2016]
	• *P*: Distribution
	• *Q*: Expectation
	• Examples:
	  • Log-likelihood [Fisher, 1922], [Good, 1952]
	  • Quadratic score [Bernardo, 1979]
		• P: Distribution
		• Q: Distribution
		• Examples:
		  • Kullback-Leibler divergence [Kullback and Leibler, 1952]
		  • Jensen-Shannon divergence
		  • Total variation
		  • Pearson χ^2

These approaches generally require tractable likelihoods, and focus is not necessarily on uncertainty.

Uncertainty

- Proper score $S(P_\theta, x)$ e.g.

$$S(P_\theta, x) = \log P_\theta(x) = \ell(\theta)$$

  - $\hat{\theta} = \arg\max S(P_\theta, x)$ is consistent etc
  - We can construct CIs

$$\{\theta : 2(\ell(\hat{\theta}) - \ell(\theta)) \leq \chi^2_{p,1-\alpha}\}$$

  - Dawid *et al.* 2014 and others derive CIs for general proper scores.
  - Why use other scores? Tractability, robustness, etc.
  - Any kernel k leads to a proper scoring rule (Zawadzki and Lahaie 2015)

Uncertainty

- Proper score $S(P_\theta, x)$ e.g.

$$S(P_\theta, x) = \log P_\theta(x) = \ell(\theta)$$

  - $\hat{\theta} = \arg\max S(P_\theta, x)$ is consistent etc
  - We can construct CIs

$$\{\theta : 2(\ell(\hat{\theta}) - \ell(\theta)) \leq \chi^2_{p,1-\alpha}\}$$

  - Dawid *et al.* 2014 and others derive CIs for general proper scores.
  - Why use other scores? Tractability, robustness, etc.
  - Any kernel k leads to a proper scoring rule (Zawadzki and Lahaie 2015)

- If we're fitting using integral probability metrics, e.g.

$$\hat{\theta} = \arg\min_\theta \widehat{MMD}(P_\theta, Q)$$

  or generative adversarial networks, can we calculate uncertainty about θ?

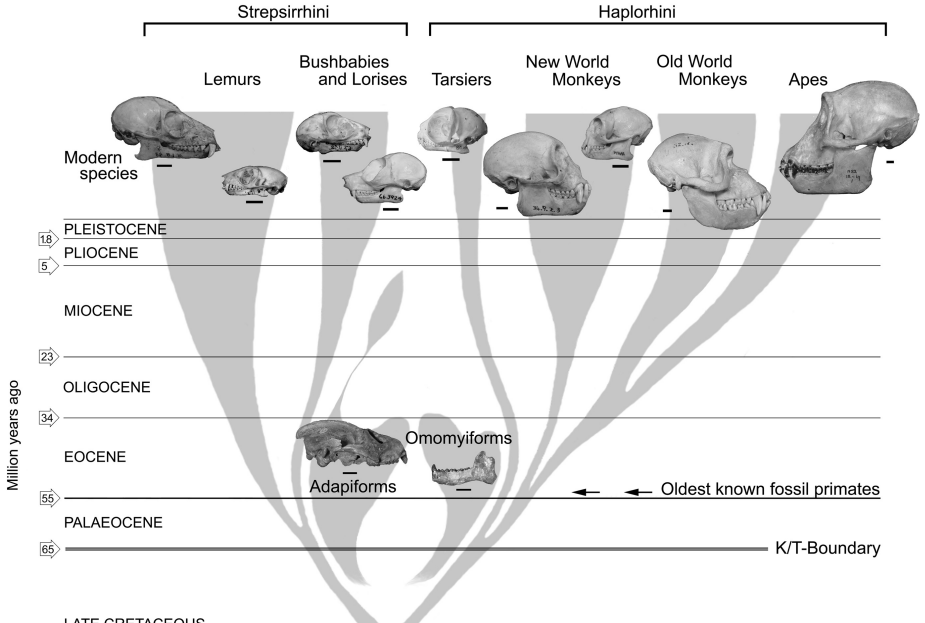
Bayesian type approaches

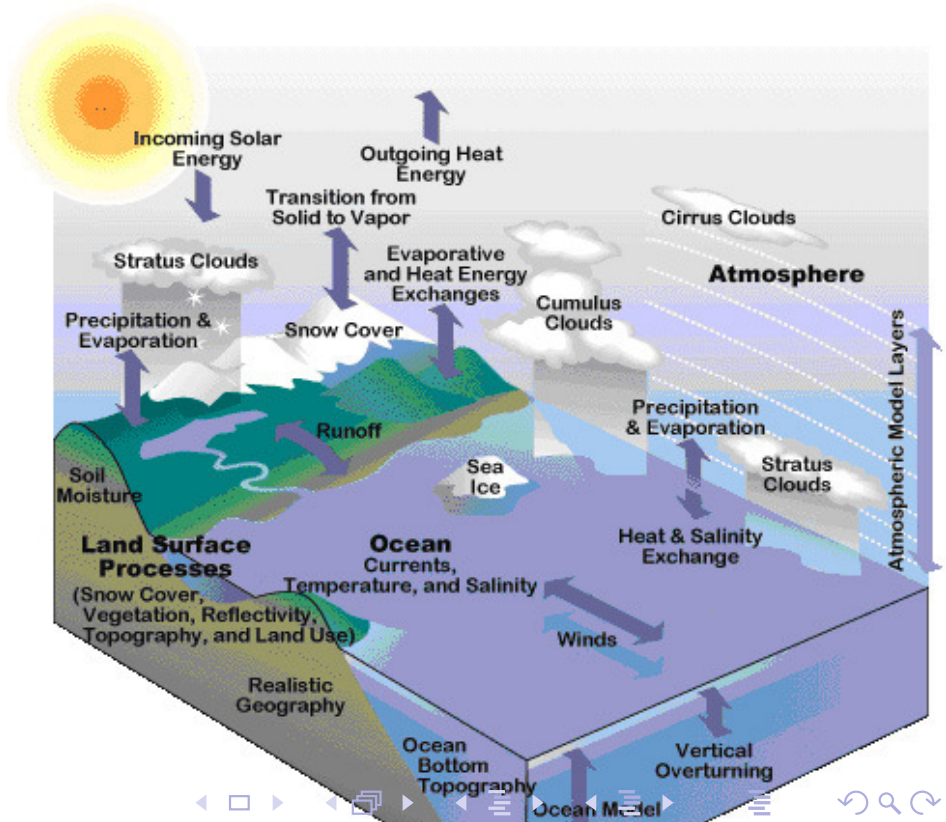
$$\pi(\theta|D) \propto \pi(D|\theta)\pi(\theta)$$

for some 'likelihood' function $\pi(D|\theta)$ (not necessarily $P_\theta(D)$)

Why the different focus?

- Working with probabilities is preferable
- Small n, large uncertainty problems are interesting! The uncertainty can matter.





Rejection ABC

Uniform Rejection Algorithm

- Draw θ from $\pi(\theta)$
- Simulate $X \sim f(\theta)$
- Accept θ if $\rho(D, X) \leq \epsilon$

Rejection ABC

Uniform Rejection Algorithm

- Draw θ from $\pi(\theta)$
- Simulate $X \sim f(\theta)$
- Accept θ if $\rho(D, X) \leq \epsilon$

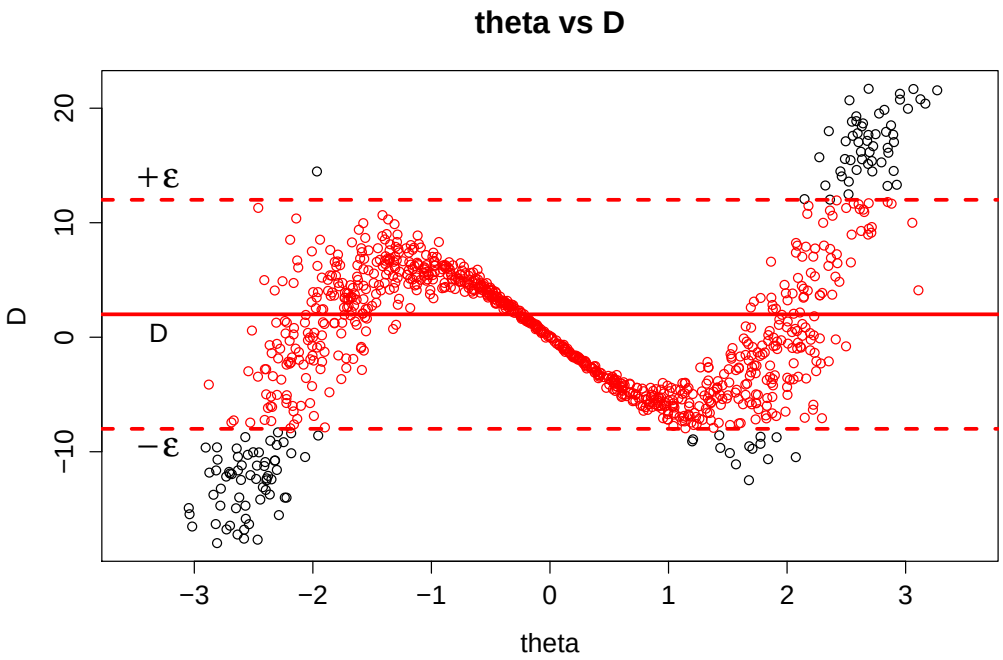
ϵ reflects the tension between computability and accuracy.

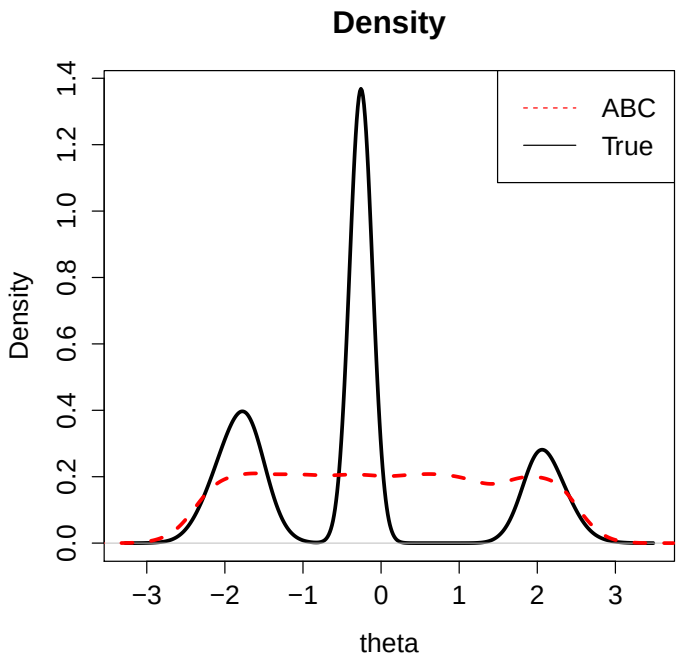
- As $\epsilon \to \infty$, we get observations from the prior, $\pi(\theta)$.
- If $\epsilon = 0$, we generate observations from $\pi(\theta \mid D)$.

Rejection sampling is inefficient, but we can adapt other MC samplers such as MCMC and SMC.

Simple $\to$ Popular with non-statisticians

$\epsilon = 10$

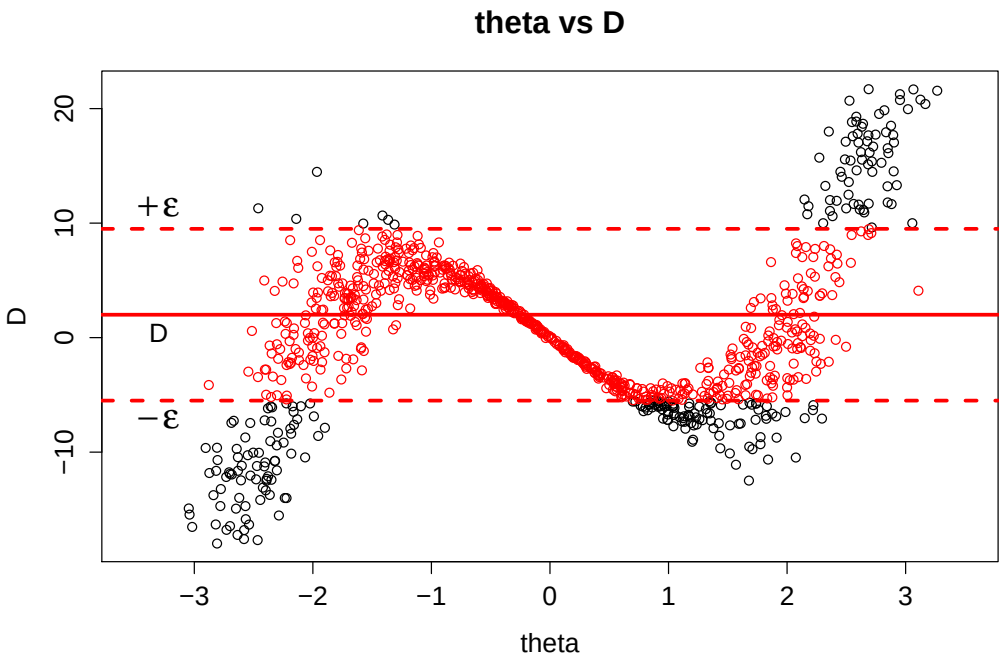


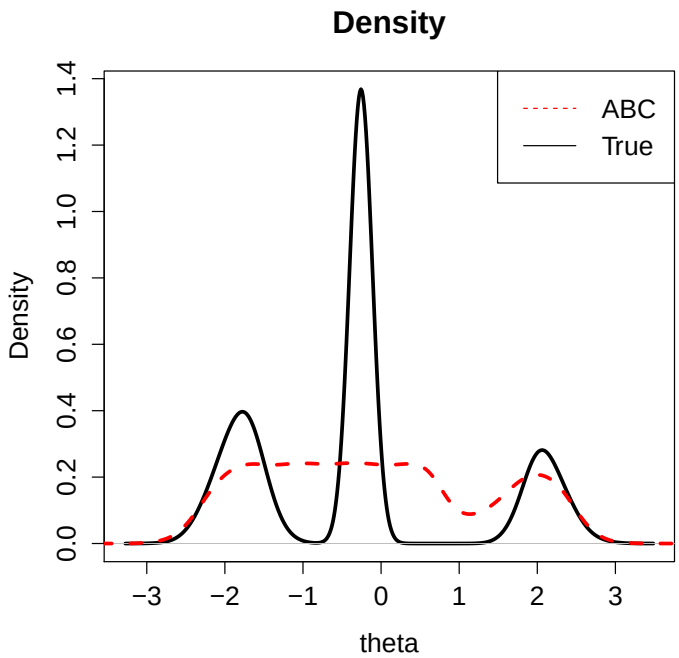


$$\theta \sim U[-10, 10], \qquad X \sim N(2(\theta + 2)\theta(\theta - 2), 0.1 + \theta^2)$$

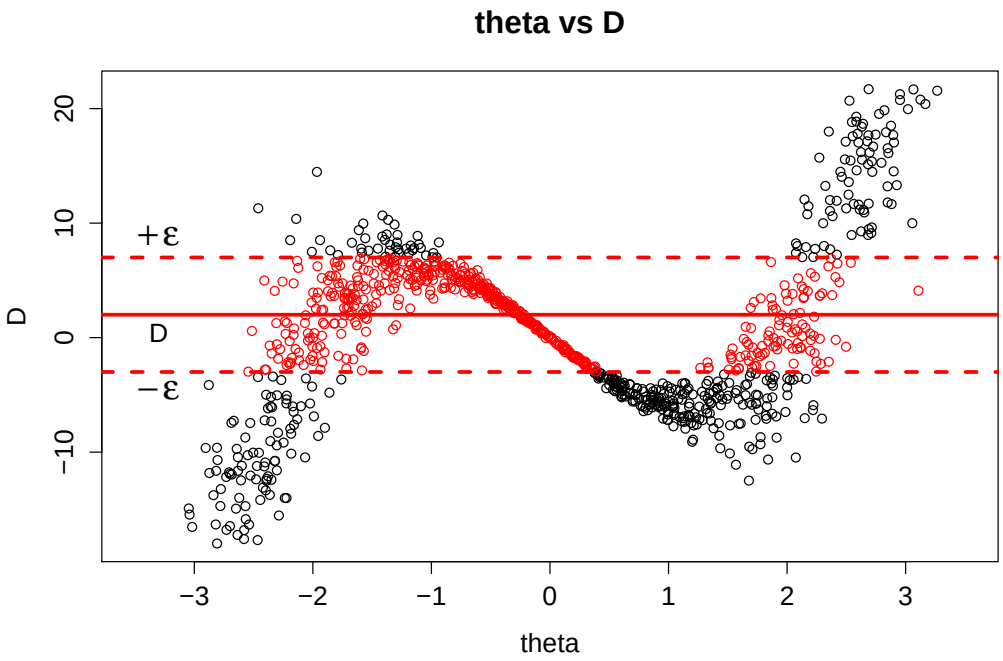
$$\rho(D, X) = |D - X|, \qquad D = 2$$

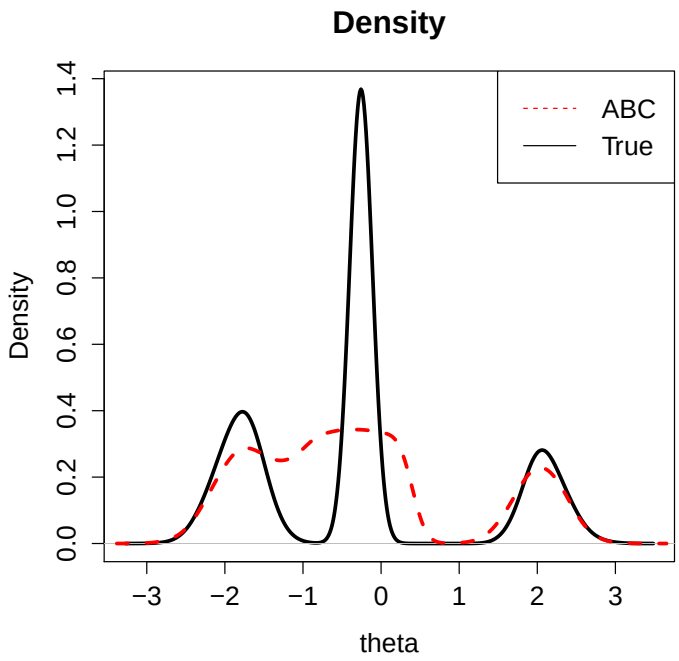
$\epsilon = 7.5$


theta vs D
D
theta
+ε
D
−ε
20
10
0
−10
−3
−2
−1
0
1
2
3

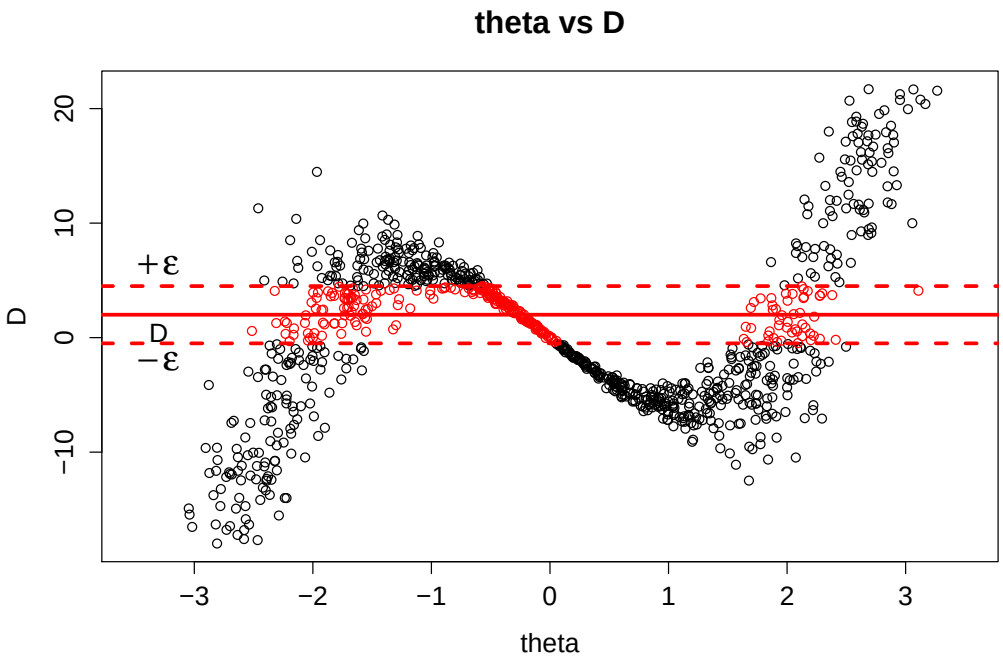

Density
Density
theta
ABC
True
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3

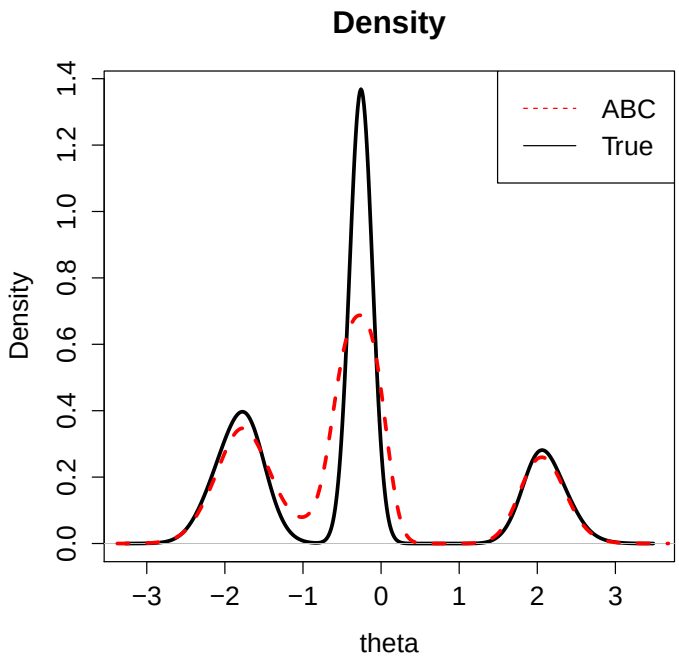
$\epsilon = 5$


theta vs D
D
theta
+ε
D
−ε
20
10
0
−10
−3
−2
−1
0
1
2
3

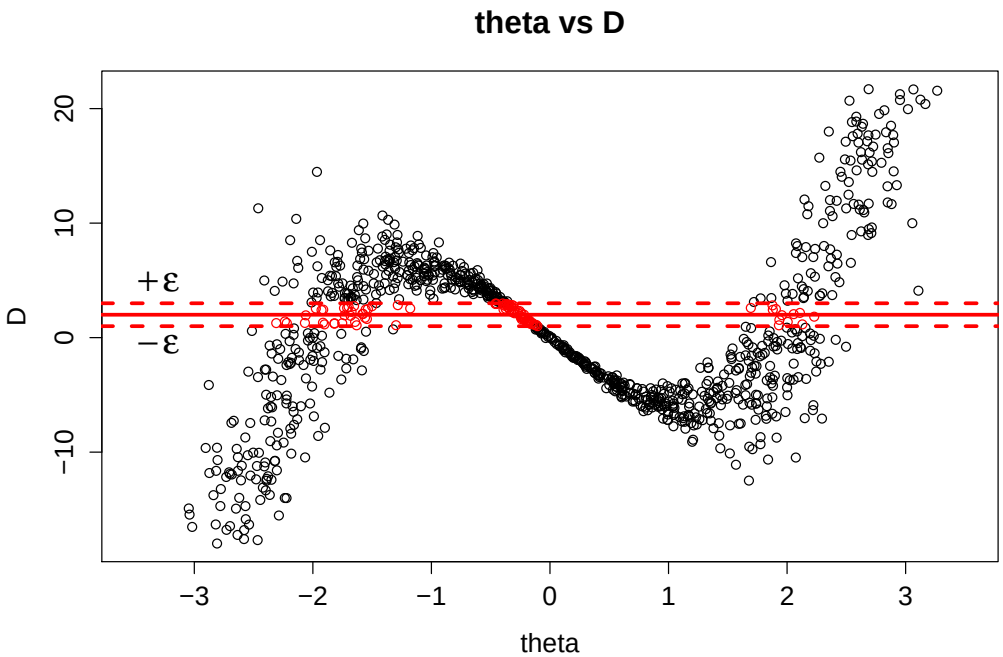

Density
Density
theta
ABC
True
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3

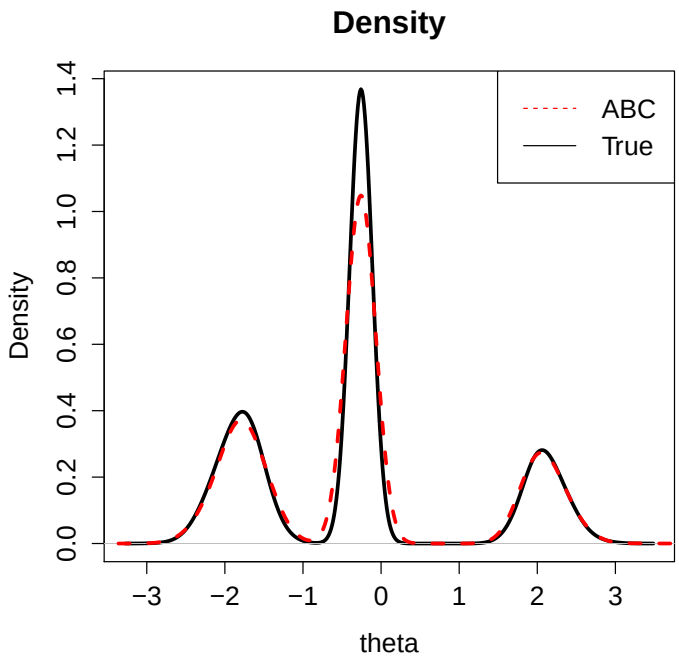
$\epsilon = 2.5$


theta vs D
D
theta
+ε
D
−ε
20
10
0
−10
−3
−2
−1
0
1
2
3


Density
Density
theta
ABC
True
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3

$\epsilon = 1$


theta vs D
D
theta
+ε
−ε
20
10
0
−10
−3
−2
−1
0
1
2
3


Density
Density
theta
ABC
True
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−3
−2
−1
0
1
2
3

The ABC target

Uniform ABC is doing 'exact' inference for the posterior

$$\pi(\theta|D) \propto \int \mathbb{I}_{\rho(D,X)\leq\epsilon} P_\theta(X)\pi(\theta)\mathrm{d}X$$

which is the likelihood for $D = X + e$ where $e \sim U[-\epsilon, \epsilon]$ if $\rho(D, X) = |D - X|$

The ABC target

Uniform ABC is doing 'exact' inference for the posterior

$$\pi(\theta|D) \propto \int \mathbb{I}_{\rho(D,X)\leq\epsilon} P_\theta(X)\pi(\theta)\mathrm{d}X$$

which is the likelihood for $D = X + e$ where $e \sim U[-\epsilon, \epsilon]$ if $\rho(D, X) = |D - X|$

- For $\dim(X)$ large, often use $\rho(T(D), T(X))$
- Or use any general distribution $\pi(D|X)$
- Or a scoring rule $S(P_\theta, D)$ and assume, e.g.,

$$\pi(D|X) \propto \exp(-S(\widehat{P_\theta}, D))$$

- KDEs, ...

The ABC target

Uniform ABC is doing 'exact' inference for the posterior

$$\pi(\theta|D) \propto \int \mathbb{I}_{\rho(D,X)\leq\epsilon} P_\theta(X)\pi(\theta)\mathrm{d}X$$

which is the likelihood for $D = X + e$ where $e \sim U[-\epsilon, \epsilon]$ if $\rho(D, X) = |D - X|$

- For $\dim(X)$ large, often use $\rho(T(D), T(X))$
- Or use any general distribution $\pi(D|X)$
- Or a scoring rule $S(P_\theta, D)$ and assume, e.g.,

$$\pi(D|X) \propto \exp(-S(\widehat{P_\theta}, D))$$

- KDEs, ...

Some scores are more robust to model discrepancy than log-likelihood. Some approaches, such as history matching, are explicitly conservative methods that seek to rule out implausible θ rather than find good θ.

Approaches from ML: Choice of summaries

Approaches from ML: Choice of summaries

- Many attempts (following Beaumont *et al.* 2003, Fearnhead and Prangle 2012) to build models to predict θ from X, and then use this in the acceptance kernel, e.g.

$$g : X \to \theta \qquad \rho(D, X) = \|g(D) - g(X)\|$$

  using linear regression, RFs, (C)NN, etc

Approaches from ML: Choice of summaries

- Many attempts (following Beaumont *et al.* 2003, Fearnhead and Prangle 2012) to build models to predict θ from X, and then use this in the acceptance kernel, e.g.

$$g : X \to \theta \qquad \rho(D, X) = \|g(D) - g(X)\|$$

  using linear regression, RFs, (C)NN, etc
- Park *et al.* 2016 use the MMD in place of choosing a vector of summaries $T(X)$.
  - Some evidence it can work for non-exchangeable samples.

Approaches from ML: Choice of summaries

- Many attempts (following Beaumont *et al.* 2003, Fearnhead and Prangle 2012) to build models to predict θ from X, and then use this in the acceptance kernel, e.g.

$$g : X \to \theta \qquad \rho(D, X) = \|g(D) - g(X)\|$$

  using linear regression, RFs, (C)NN, etc
- Park *et al.* 2016 use the MMD in place of choosing a vector of summaries $T(X)$.
  - Some evidence it can work for non-exchangeable samples.

Note that these approaches are used to define a 'likelihood' for use within a Bayesian analysis.

Approaches developed by ML: surrogate modelling

Wood 2010 introduced a synthetic likelihood

$$\pi(D|\theta) = \mathcal{N}(\theta|\mu_\theta, \Sigma_\theta)$$

where μ_θ and Σ_θ are the mean and covariance of the simulator output when run at θ, and plugged this into an MCMC sampler.

Approaches developed by ML: surrogate modelling

Wood 2010 introduced a synthetic likelihood

$$\pi(D|\theta) = \mathcal{N}(\theta|\mu_\theta, \Sigma_\theta)$$

where μ_θ and Σ_θ are the mean and covariance of the simulator output when run at θ, and plugged this into an MCMC sampler.

- This suggested modelling dependence on θ to mitigate the cost

> *[...] the forward model may exhibit regularity in its dependence on the parameters of interest[...]. Replacing the forward model with an approximation or "surrogate"* **decouples** *the required number of forward model evaluations from the length of the MCMC chain, and thus can vastly reduce the overall cost of interence. Conrad et al. 2015*

Approaches developed by ML: surrogate modelling

Wood 2010 introduced a synthetic likelihood

$$\pi(D|\theta) = \mathcal{N}(\theta|\mu_\theta, \Sigma_\theta)$$

where μ_θ and Σ_θ are the mean and covariance of the simulator output when run at θ, and plugged this into an MCMC sampler.

- This suggested modelling dependence on θ to mitigate the cost

  > *[...] the forward model may exhibit regularity in its dependence on the parameters of interest[...]. Replacing the forward model with an approximation or "surrogate"* **decouples** *the required number of forward model evaluations from the length of the MCMC chain, and thus can vastly reduce the overall cost of interence. Conrad et al. 2015*

- Monte Carlo is dumb (which is its strength).
  - We have to learn continuity, and smoothness of the likelihood function.
- Instead fit a GP (to something) and use this in the inference
  - develop a single good MCMC sampler.

Surrogate ABC

- Wilkinson 2014
- Meeds and Welling 2014
- Gutmann and Corander 2015
- Strathmann, Sejdinovic, Livingstone, Szabo, Gretton 2015
- $\vdots$

With obvious influence from emulator community (e.g. Sacks, Welch, Mitchell, and Wynn 1989, Kennedy and O'Hagan 2001)

Constituent elements:

- Target of approximation
- Aim of inference and inference scheme
- Choice of surrogate/emulator
- Training/acquisition rule

$\exists$ a relationship to probabilistic numerics

Target of approximation for the surrogate

- Simulator output within synthetic likelihood (Meeds et al 2014) e.g.

$$\mu_\theta = \mathbb{E}f(\theta) \qquad \text{and} \qquad \Sigma_\theta = \mathbb{V}\text{ar}f(\theta)$$

- (ABC) Likelihood type function (Wilkinson 2014)

$$L_{ABC}(\theta) = \mathbb{E}_{X|\theta} K_\epsilon[\rho(T(D), T(X))] \equiv \mathbb{E}_{X|\theta}\pi_\epsilon(D|X)$$

- Discrepancy function (Gutmann and Corander, 2015), for example

$$J(\theta) = \mathbb{E}\rho(S(D), S(X))$$

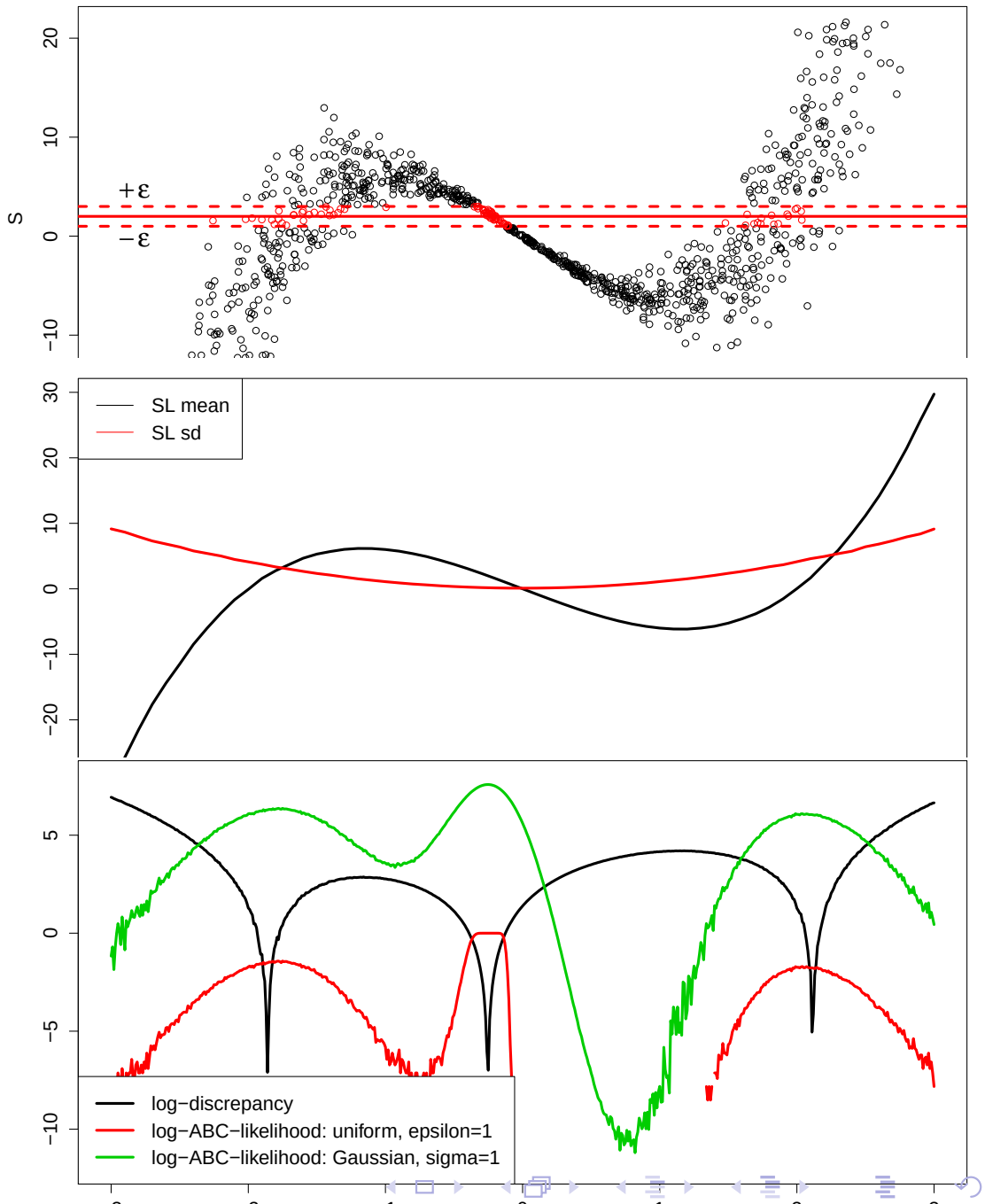
- Gradients (Strathmann et al 2015)

The difficulty of each approach depends on smoothness, dimension, focus etc.

$$S \sim N(2(\theta+2)\theta(\theta-2), 0.1+\theta^2)$$

Synthetic likelihood:

ABC likelihood and discrepancy:



Inference

- Directly use the surrogate to calculate the posterior (Kennedy and O'Hagan 2001 etc) - over-utilizes the surrogate, sacrificing exact sampling.
- Correct for the use of a surrogate, e.g., using a Metropolis step (Rasmussen 2003, Sherlock *et al.* 2015, etc), which requires simulator evaluations at every stage - under-utilizes the surrogate, sacrificing speed-up.

Inference

- Directly use the surrogate to calculate the posterior (Kennedy and O'Hagan 2001 etc) - over-utilizes the surrogate, sacrificing exact sampling.
- Correct for the use of a surrogate, e.g., using a Metropolis step (Rasmussen 2003, Sherlock *et al.* 2015, etc), which requires simulator evaluations at every stage - under-utilizes the surrogate, sacrificing speed-up.

Instead, Conrad *et al.* 2015 developed an intermediate approach that asymptotically samples from the exact posterior.

- proposes new θ - if uncertainty in surrogate prediction is such that it is unclear whether to accept or reject, then rerun simulator, else trust surrogate.

Inference

- Directly use the surrogate to calculate the posterior (Kennedy and O'Hagan 2001 etc) - over-utilizes the surrogate, sacrificing exact sampling.
- Correct for the use of a surrogate, e.g., using a Metropolis step (Rasmussen 2003, Sherlock *et al.* 2015, etc), which requires simulator evaluations at every stage - under-utilizes the surrogate, sacrificing speed-up.

Instead, Conrad *et al.* 2015 developed an intermediate approach that asymptotically samples from the exact posterior.

- proposes new θ - if uncertainty in surrogate prediction is such that it is unclear whether to accept or reject, then rerun simulator, else trust surrogate.

> *It is inappropriate to be concerned about mice when there are tigers abroad (Box 1976)*

Model discrepancy, ABC approximations, sampling errors etc may mean it is not worth worrying...

Acquisition rules

The key determinant of emulator accuracy is the design used to train the GP

$$D_n = \{\theta_i, f(\theta_i)\}_{i=1}^{N}$$

Usual design choices are space-filling designs

- Maximin latin hypercubes, Sobol sequences

Acquisition rules

The key determinant of emulator accuracy is the design used to train the GP

$$D_n = \{\theta_i, f(\theta_i)\}_{i=1}^{N}$$

Usual design choices are space-filling designs

- Maximin latin hypercubes, Sobol sequences

Calibration doesn't need a global approximation to the simulator - this is wasteful.

Instead build a sequential design $\theta_1, \theta_2, \ldots$ using our current surrogate model to guide the choice of design points according to some acquisition rule.

History matching waves

The ABC log-likelihood $l(\theta) = \log L(\theta)$ typical ranges across a wide range of values, consequently, most models struggle to accurately approximate the log-likelihood across the entire parameter space.

History matching waves

The ABC log-likelihood $l(\theta) = \log L(\theta)$ typical ranges across a wide range of values, consequently, most models struggle to accurately approximate the log-likelihood across the entire parameter space.

- But we only need to make good predictions near $\hat{\theta}$
- Introduce waves of history matching.
- In each wave, build a GP model that can rule out regions of space as implausible.

History matching waves

The ABC log-likelihood $l(\theta) = \log L(\theta)$ typical ranges across a wide range of values, consequently, most models struggle to accurately approximate the log-likelihood across the entire parameter space.

- But we only need to make good predictions near $\hat{\theta}$
- Introduce waves of history matching.
- In each wave, build a GP model that can rule out regions of space as implausible.

We decide that θ is implausible if

$$\mathbb{P}(\tilde{l}(\theta) > \max_{\theta_i} l(\theta_i) - T) \leq 0.001$$

where $\tilde{l}(\theta)$ is the GP model of $\log \pi(D|\theta)$

Choose T so that if $l(\hat{\theta}) - l(\theta) > T$ then $\pi(\theta|y) \approx 0$.

- Ruling θ to be implausible is to set $\pi(\theta|y) = 0$
- Equivalent to doing inference with log-likelihood $L(\theta)\mathbb{I}_{l(\hat{\theta})-l(\theta)<T}$

Choice of T is problem specific; start conservatively with T large and decrease

Conclusion

ML can improve existing ABC approaches

- Finding low dimensional representations of complex objects
- Optimization of inference/search for good parameters
  - Can we combine these ideas?
- ⋮

ABC should also be part of ML toolbox

- Intractable likelihood models
- Probabilistic programming
- Computer vision problems.