

Approximate Bayesian Computation - ABC - Theory and Examples

Antonietta Mira

Data Science Lab, ICS
Università della Svizzera italiana, USI, Lugano, Switzerland
and
Department of Science and High Technology, DISAT
Università dell'Insubria, Como, Italy

joint with Ritabrata Dutta
Warwick University, UK
JP Onnela
Harvard School of Public Health, US

- Generative (network / agent based) models
- Approximate Bayesian Computation
- ABC for spreading processes on networks
- ABCpy

Motivating example

- Systems of scientific and societal interest have large numbers of interacting components
- Representation as networks:
node = component, edge = interaction
- Distinction between models of two things:
 - Models of network structure (e.g, Erdős-Rényi)
 - Models of dynamical processes on networks (e.g., SI model)

Distinction between two types of models:

- **Statistical models** (e.g. ERGM)

DATA DRIVEN

- **Pros**: inference on model parameters; hypothesis testing; model selection
- **Cons**: scalability; hard to incorporate domain knowledge

- **Generative models** (e.g. Price / agent based model)

KNOWLEDGE DRIVEN

assume that microscopic mechanisms that govern network formation and evolution are known, ask what happens if we apply these mechanisms repeatedly

- **Pros**: easy to incorporate domain knowledge, scalability
- **Cons**: no inferential tools; no model comparison

Big picture of statistical inference

GIVEN:

- **Data** = $y = (y_1, \dots, y_n)$
- **Model** which describes data, $p_{y|\theta}(y|\theta)$
indexed by **parameters** = $\theta = (\theta_1, \dots, \theta_d)$
- **Prior probability** density function for θ , p_θ

WANTED:

- **Some probabilistic statement about θ and model**
 - point estimation
 - confidence/credible intervals
 - hypothesis testing
 - prediction
 - model selection

Big picture of statistical inference

GIVEN:

- Data = $y = (y_1, \dots, y_n)$
- **Model** which describes data, $p_{y|\theta}(y|\theta)$
indexed by parameters = $\theta = (\theta_1, \dots, \theta_d)$
- Prior probability density function for θ , p_θ

WANTED:

- Some probabilistic statement about θ and model
 - point estimation
 - confidence/credible intervals
 - hypothesis testing
 - prediction
 - model selection

Two types of models

- **Statistical model**

$$p_{y|\theta}(y_i|\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_i - \mu)^2\right), \quad \boldsymbol{\theta} = (\mu, \sigma)$$

- **Generative model**

- given $\boldsymbol{\theta} = (\mu, \sigma)$

- $z_i \sim \mathcal{N}(0, 1)$

- $y_i = \mu + \sigma z_i$

- $y_i \sim p_{y|\theta}(y_i|\boldsymbol{\theta})$

- In some settings easier to specify a generative / agent based model

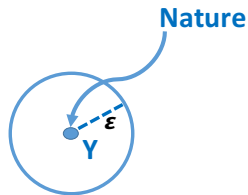
Why generative / agent based models?

- Allow to use **knowledge domain** on how the data were generated without having to make excessive compromises in the modeling
- Neat **interface** with social, medical, biological, physical . . . models of data
- **Scale well** with big data
- Easier to study the **effect of interventions** on generative models rather than statistical models
- Easy to fwd simulate
- BUT no LHD available in closed form

Starting point is Bayes' theorem:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

- $p(\theta|y)$ = posterior
- $p(y|\theta)$ = likelihood
- $p(\theta)$ = prior
- $p(y)$ = evidence

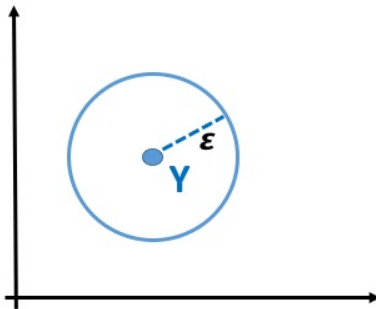


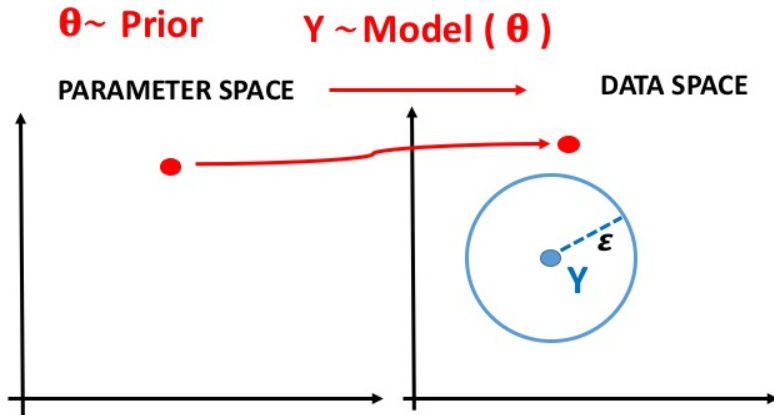
$\theta \sim \text{Prior}$

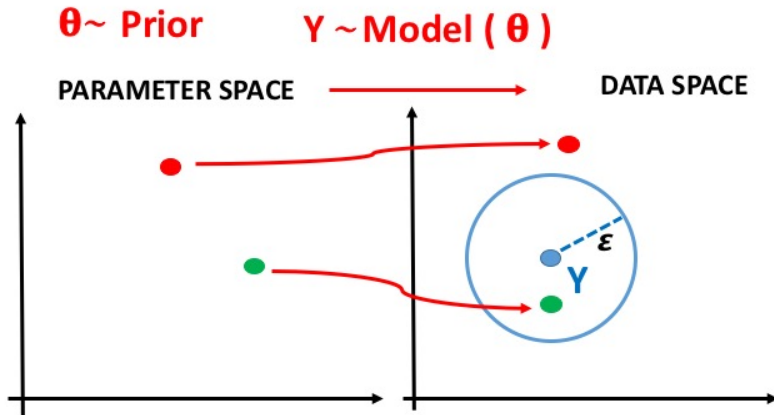
PARAMETER SPACE

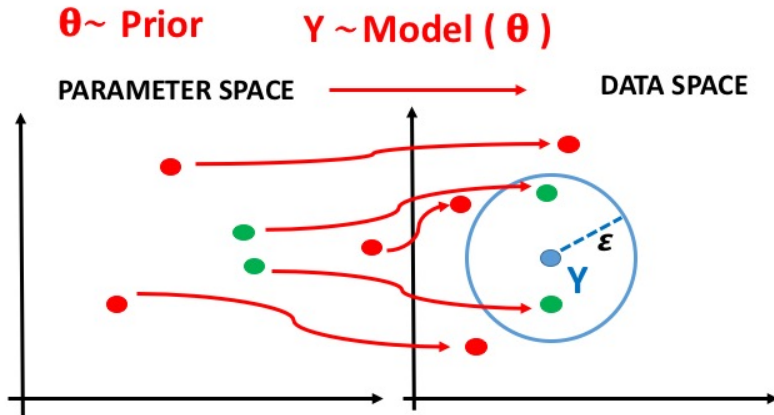


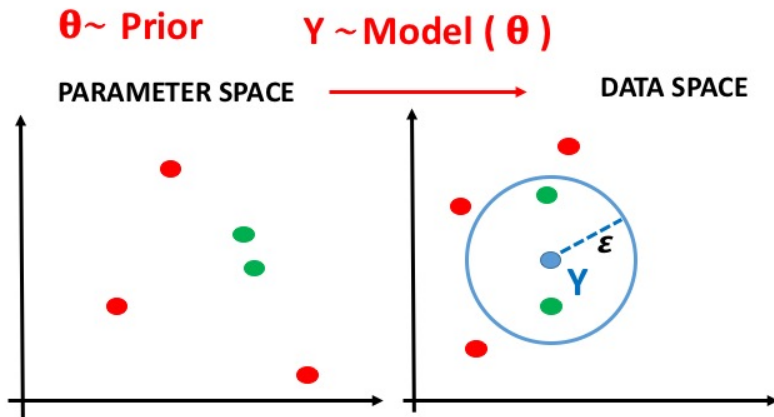
DATA SPACE

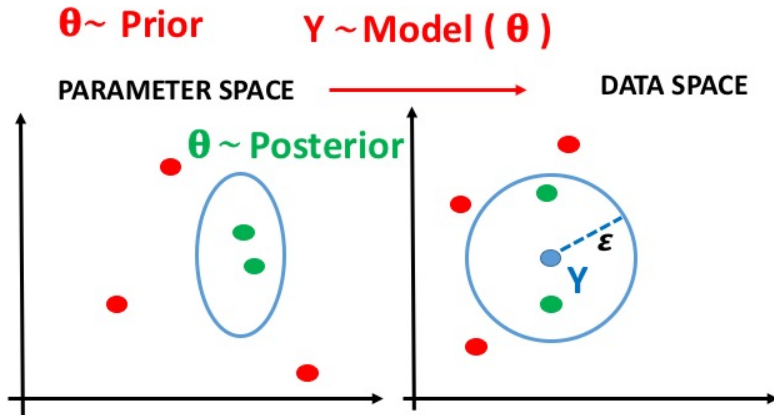












Approximate Bayesian Computation (ABC)

- ABC rejection sampler is the simplest form of ABC

ABC rejection sampler

- **Sample parameter** θ^* from the prior $p(\theta)$
 - **Simulate dataset** y^* under the given model specified by θ^* :
 $y^* \sim p(\cdot | \theta^*)$
 - **Accept** θ^* if $\rho(y^*, y) \leq \epsilon$
-
- Distance measure $\rho(y^*, y)$ determines the level of discrepancy between the **simulated data** y^* and the **observed data** y
 - The **accepted** θ^* are approximately distributed according to the **desired posterior** and, crucially, obtained without the need of explicitly evaluating the LHD

Approximate Bayesian Computation (ABC)

- It may be unfeasible to compute the distance $\rho(y^*, y)$ for **high-dimensional data**
- Lower dimensional **summary statistic** $S(y)$ to capture the relevant information in y
- Comparison is done between $S(y^*)$ and $S(y)$: accept θ^* if $\rho(S(y^*), S(y)) \leq \epsilon$
- If S is **sufficient** wrt θ , then it contains all information in y about θ (by definition), and using $S(y)$ in place of the full dataset does **not introduce any error**
- For most models it may be impossible to find sufficient statistics S , in which case application **relevant** summary statistics need to be used
- Use of **non-sufficient** summary statistics introduces a further level of **approximation**

ABC for Generative Network Models

- ABC + generative network models =
generic + sound inferential framework

ABC rejection sampler for generative network models

- Observe an empirical graph G
 - Set up generative network model M
 - Sample parameter θ^* from the prior $p(\theta)$
 - Simulate graph G^* from the generative network model M using parameter θ^*
 - Accept θ^* if $\rho(S(G^*), S(G)) \leq \epsilon$ using application relevant summaries S
-
- Some simple network summaries: degree sequence, k -stars, subgraph counts, centrality measures (betweenness, eigenvector, etc.)
 - From Rejection-ABC to SABC by Albert et al. (2012)

- **intuitive** and close to standard model calibration practice
- **flexible** and hence well adapted to investigate complex models
- **widens the realm of models** for which inference is performed
- build the **ABC reference table**:
parameter (θ), summary (S), distance (d)
- ϵ reflects the tension bwn **computability** and **accuracy**
- $\epsilon \rightarrow 0$ simulation is from the true **posterior**
- $\epsilon \rightarrow \infty$ simulation is from the **prior**
- ϵ corresponds to a **quantile of the distances**
- ABC is a **kernel smoothing approximation of the LHD**
- rejection ABC is **embarassingly parallel**
- when the number of summary statistics increases \rightarrow **curse of dimensionality**

Exact ML inference for Erdős-Rényi (ER)

For the Erdős-Rényi model the LHD is available:

A_{ij} = dyad (pair of nodes): value 1 (connected) or 0 (not connected)

Y_{ij} = RV coding the state of the dyad

$p = P(Y_{ij} = 1)$

N = number of nodes

n = number of dyads: $N(N-1)/2$ (undirected) and $N(N-1)$ (directed)

L = number of connected dyads (sufficient statistics)

$$\text{LHD} = p(G|p) = \prod_{i \neq j} P(y_{ij} = A_{ij}) = \prod_{i \neq j} p^{A_{ij}} (1-p)^{1-A_{ij}} = p^L (1-p)^{n-L}$$

$\text{MLE} = \hat{p} = L/n$ = proportion of connected dyads to all dyads

$$\text{SE}(\hat{p}) = \sqrt{\hat{p}(1-\hat{p})/n}$$

$$95\% \text{ CI} = \hat{p} \pm 2\text{SE}(\hat{p})$$

Approximate ML inference for ER - (Diggle et al, 1984)

Simulate the **observed** G_0 from an ER model with $N = 100$, $p = 0.05$

Approximate ML inference for ER - (Diggle et al, 1984)

Simulate the **observed** G_0 from an ER model with $N = 100$, $p = 0.05$

Find p that maximizes an approximate LHD obtained by simulation

Approximate ML inference for ER - (Diggle et al, 1984)

Simulate the **observed G_0** from an ER model with $N = 100$, **$p = 0.05$**

Find p that maximizes an approximate LHD obtained by simulation

Set up a **grid of parameter values p** : $0, 0.005, 0.01, \dots, 1$

Approximate ML inference for ER - (Diggle et al, 1984)

Simulate the **observed G_0** from an ER model with $N = 100$, **$p = 0.05$**

Find p that maximizes an approximate LHD obtained by simulation

Set up a **grid of parameter values p** : $0, 0.005, 0.01, \dots, 1$

For each grid point p_i , $i = 1, 2, \dots, n$ simulate $K = 1000$ pseudo-networks

Approximate ML inference for ER - (Diggle et al, 1984)

Simulate the **observed** G_0 from an ER model with $N = 100$, $p = 0.05$

Find p that maximizes an approximate LHD obtained by simulation

Set up a **grid of parameter values** p : $0, 0.005, 0.01, \dots, 1$

For each grid point p_i , $i = 1, 2, \dots, n$ simulate $K = 1000$ pseudo-networks

For each p_i , this gives rise to

a sequence of **networks** G_i^1, \dots, G_i^K ,

corresponding **summary statistics** $S(G_i^1), \dots, S(G_i^K)$,

distances $d_i^1 = \rho(S(G_i^1), S(G_0)), \dots, d_i^K = \rho(S(G_i^K), S(G_0))$

Pool the distances for all values of i and $k =$ a collection of nK values

The smaller the distance, the closer the network G_i^k is to the observed one G_0 in the sense of the chosen summary statistic

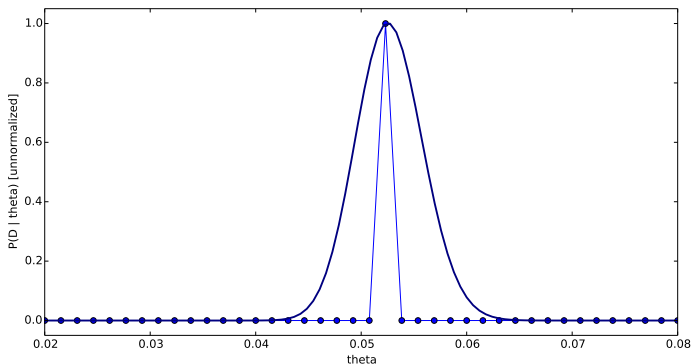
Choose a **cutoff value** $d^* =$ 10th percentile

For each grid point p_i count the number of networks G_i^k for which the corresponding distance $d_i^k < d^*$

Obtain an unnormalized **step function approximation**

$$f_i = \frac{1}{K} \sum_{k=1}^K \mathbf{1}_{d_i^k < d^*} \text{ to the LHD}$$

\hat{p}_{AML} = the value p_i for which f_i obtains its maximum ≈ 0.049



True $p = 0.05$ - estimated $\hat{p}_{\text{AML}} \approx 0.049$

The **solid** (smooth) line is the **exact LHD**

We set the threshold at the 10th percentile

CI are obtained as in the ML case but using \hat{p}_{AML} in place of the exact estimate \hat{p}

Exact Bayesian inference for ER

Beta **prior** $p(\theta) = \text{Beta}(\alpha = 5, \beta = 50)$

Beta **posterior** $p(\theta|D) = \text{Beta}(\alpha + L, \beta + [N(N - 1)/2] - L)$

Bayesian Estimator = **posterior mean**

Approximate Bayesian inference for Erdős-Rényi

Simulate the **observed** ER graph **G** from a model with $N = 100$ and $p = 0.05$

Approximate Bayesian inference for Erdős-Rényi

Simulate the **observed** ER graph G from a model with $N = 100$ and $p = 0.05$

Number of edges = summary statistic = $S(G) = L$

Approximate Bayesian inference for Erdős-Rényi

Simulate the **observed** ER graph G from a model with $N = 100$ and $p = 0.05$

Number of edges = summary statistic = $S(G) = L$

We perform 100,000 draws from the prior p_i

For each p_i , we generate

a sequence of **graphs** G_i^1, \dots, G_i^K ,

compute corresponding **summary statistics** $S(G_i^1), \dots, S(G_i^K)$,

retain p_i if $S(G_i) = L_i = L = S(G)$

We retained only 162 values of p_i

Approximate Bayesian inference for Erdős-Rényi

Simulate the **observed** ER graph G from a model with $N = 100$ and $p = 0.05$

Number of edges = summary statistic = $S(G) = L$

We perform 100,000 draws from the prior p_i

For each p_i , we generate

a sequence of **graphs** G_i^1, \dots, G_i^K ,

compute corresponding **summary statistics** $S(G_i^1), \dots, S(G_i^K)$,

retain p_i if $S(G_i) = L_i = L = S(G)$

We retained only 162 values of p_i

The **mean and median** of the **prior** are **0.091** and **0.086**
the **95% credible interval** is $[0.031, 0.178]$

The **mean and median** of the **posterior** are **0.050** and **0.050**
the **95% credible interval** is $[0.043, 0.055]$

Degree distribution = summary statistic = $S(G)$

KS = distance measure

We performed 100,000 draws from the prior

For each p_i , we generate

a sequence of **graphs** G_i^1, \dots, G_i^K ,

compute corresponding **summaries** $S(G_i^1), \dots, S(G_i^K)$,

obtain a sequence of **distances**

$$y_i^1 = \rho(S(G_i^1), S(G)), \dots, y_i^K = \rho(S(G_i^K), S(G))$$

Pool the distances y_i^k

for all values of i and s : $100,000 \cdot K$ total

Choose a **cutoff value** $d^* = 10\text{th percentile}$

Higher acceptance rate: 761

Pool the distances y_i^k

for all values of i and s : 100,000 · K total

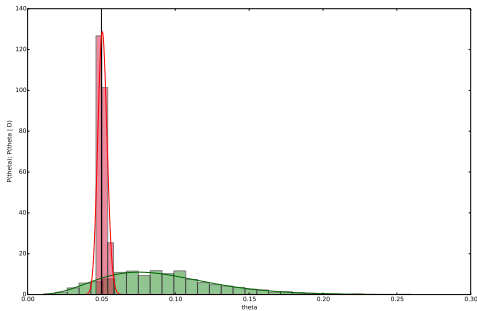
Choose a cutoff value $d^* = 10\text{th percentile}$

Higher acceptance rate: 761

The mean and median of the prior are 0.09 and 0.09
the 95% CI is [0.03, 0.18]

The mean and median of the posterior are 0.05, 0.05
the 95% CI is [0.04, 0.06]

Inference for Erdős-Rényi



TRUE:

density functions for the **prior** (solid green) and the **posterior** (solid red)

Here the prior distribution is the beta distribution $B(\alpha = 5, \beta = 50)$

True value of the parameter is $\theta = p = 0.05$ = black vertical line

ESTIMATED:

Prior $p(\theta)$ (green histogram) and **posterior** $p(\theta|y)$ (red histogram)

Approximate Bayesian inference for Easley-Kleinberg (EK)

The EK model is a simple model of **directed networks**

Unlike ER, the EK model is a **growing network** model

new vertices attach themselves to existing ones using a *preferential attachment* rule: each new node is not equally likely to attach itself to any of the existing nodes but, instead, has a preference for some nodes over others *based on their degree*

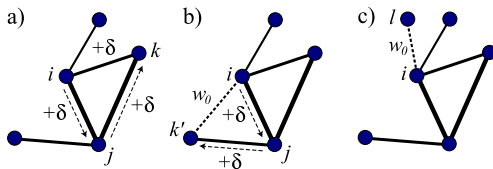
More specifically, *linear preferential attachment* specifies that an incoming node will attach to an existing vertex v_i with probability proportional to its degree k_i

More complicated functions of k_i are possible, and in general one can incorporate nodal attributes, or covariates, in the expression for the attachment function

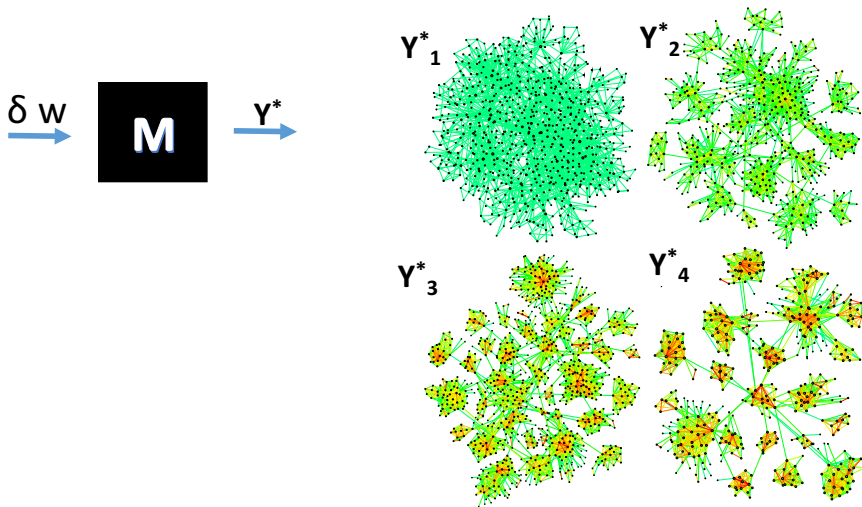
RESULTS: Similar performance to the Erdős-Rényi model

Mechanistic Model of Social and Contact Networks

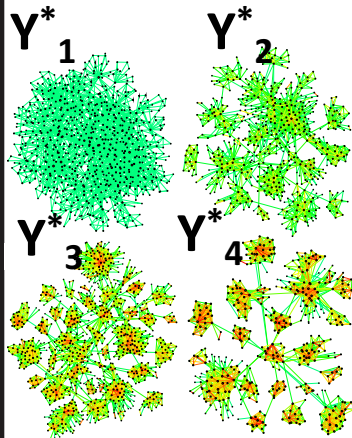
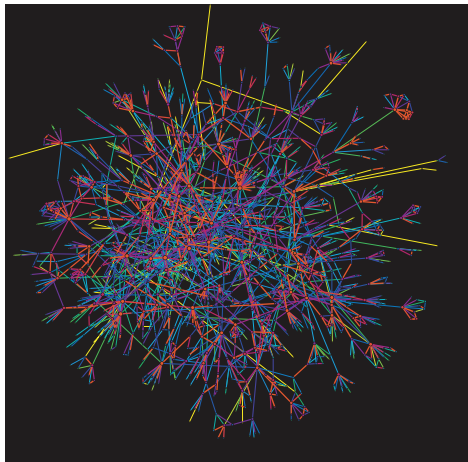
- From the perspective of time expenditure of subject i :
 - spend time with **existing friends** (a)
 - become **friend of a friend** (b)
 - make **totally new friends** (c)



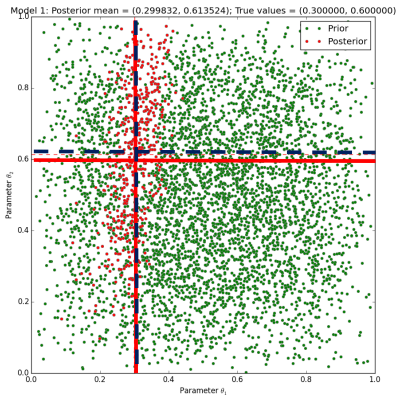
Mechanistic Model of Social and Contact Networks



Mechanistic Model of Social and Contact Networks



Inference for social network generative model



ABC + social network generative model

Prior and posterior draws

True parameter values: 0.3 and 0.6 (solid lines)

Posterior means: 0.299 and 0.613 (dashed lines)

Hypothesis Testing

$H_0 : \theta > \theta^*$ VS $H_1 : \theta \leq \theta^*$ for some arbitrary θ^*

Bayesians compute $P(H_0|y) = \int_{\theta^*}^{\infty} p(\theta|y) d\theta$

The integral can be estimated by summing over a finite set of samples θ_t from the posterior resulting in the estimator

$$\hat{P}(H_0|y) = \frac{1}{T} \sum_{t=1}^T \mathbf{1}_{\theta_t > \theta^*}$$

For the ER model with $N = 1000$ and $p = 0.25$ test

$H_0 : p > 0.3$ VS $H_1 : p \leq 0.3$

10,000 draws from the prior of which 759 draws were accepted

The 95% posterior CI was $[0.194, 0.302]$

And $P(H_0|y) = P(p > 0.3|G) \approx 0.032$

The **posterior odds** are defined as

$$\frac{P(H_0|y)}{P(H_1|y)} = \frac{P(H_0|G)}{P(H_1|G)} = \frac{P(H_0|G)}{1 - P(H_0|G)} \approx \frac{0.032}{0.968} \approx 0.033,$$

suggesting that H_1 is $1/0.033 = 30.25$ i.e. **over 30 times more likely than H_0**

We can confidently reject the null HP

- Hierarchical model: prior on the model, prior on the parameter within the model, LHD
- Models do not need to be nested
- Standard Bayesian approach to model comparison involves **Bayes factors** and **posterior probabilities** of model index

ABC for model comparison (Part I)

- Observe an **empirical graph** G
- Identify alternative possible generative network models M_1 and M_2
- Draw model index from the model prior:
 $\tau_1 = P(\mathcal{M} = 1) = P(\mathcal{M} = 2) = \tau_2$
- Draw parameter θ^* from the prior $p(\theta|\mathcal{M})$
- Simulate graph G^* from the given generative network model using parameter θ^*
- Accept θ^* if $\rho(S(G^*), S(G)) \leq \epsilon$ using any summaries S

Majority rule among the smallest distances index set

The problem is viewed as a classification question and is solved using a **Nearest-Neighbors classifier**

To ensure reliability of the method, the number of simulations should be **large** and the number of summaries statistics **small**

ABC for model comparison (Part II)

- Use the reference table to train a RF to infer the model index from the summary statistics
- Apply the RF classifier to the observed data
- Use a second RF in regression to estimate the posterior probability of the selected model

Approximate Bayesian Computation (ABC) references

- ABC in population genetics, MA Beaumont, W Zhang, DJ Balding - [Genetics](#), 2002
- Comparative evaluation of a new effective population size estimator based on approximate Bayesian computation DA Tallmon, G Luikart, MA Beaumont - [Genetics](#), 2004
- Inferring population history with DIY ABC: a user-friendly approach to ABC, JM Cornuet, F Santos, MA Beaumont, CP Robert, JM Marin, . . . - [Bioinformatics](#), 2008
- COMPUTER PROGRAMS: onesamp: a program to estimate effective population size using ABC, DA Tallmon, A Koyuk, G Luikart, MA Beaumont - [Molecular Ecology Resources](#), 2008
- Adaptive ABC, MA Beaumont, JM Cornuet, JM Marin, CP Robert - [Biometrika](#), 2009
- Approximate Bayesian computation without summary statistics: the case of admixture, VC Sousa, M Fritz, MA Beaumont, L Chikhi - [Genetics](#), 2009
- Review: Marin, [Statistics and Computing](#), 2012

- We study the spread of **epidemics** / **fake news** on a network:
Can be considered as a spreading process on a network
- **Data** = a fixed and known human **contact** interaction network
= a fixed and known **social** network
= \mathbb{N}
- **Data** = at some time points over a period of observation
we know which nodes in the network are **infected**
= we know which websites **report the fake news**
- We consider 2 spreading processes: **simple** and **complex**

Simple spreading process: Susceptible-Infected (SI)

- **Rate of diffusion, θ :**
How fast a disease or a fake news spreads on a network
- **Seed-node, n_{sn} :**
Which node was the source of the infection / the fake news

Algorithm 1

SI process on network \mathbb{N} , starting at n_{SN}^0 for time-points $t = 0, \dots, T$

```
1: for  $t = 0$  to  $T$  do
2:    $\mathbb{I}_t =$  Infected nodes at time  $t$ 
3:   for each  $i \in \mathbb{I}_t$  do
4:     Select  $j$  from the neighbors of  $i$ 
       with equal probability
5:     If neighbor  $j$  is already infected, do nothing
       If not, infect it with probability  $\theta$ 
6:   end for
7: end for
```

ABC for spreading process

- Observe temporal data, y , on a fixed network structure
- Set up a model M described by SI process
- Sample parameter θ^* , n_{SN}^* from the prior $p(\theta, n_{SN})$
- Simulate temporal data y^* from model M using θ^* , n_{SN}^*
- Accept θ^* , n_{SN}^* if $d(S(y^*), S(y)) \leq \epsilon$
using application relevant summaries S
- For inference, we use simulated annealing ABC
SABC, *Statistics and Computing*, 2015

Need to define:

- 1 Summary statistics, $S(y)$
- 2 Discrepancy measure, $d(S(y), S(y^*))$
- 3 Priors on parameters, $p(\theta)$

- **Summary statistics:**

- $\mathbf{s} = (s_1, s_2, \dots, s_T)$, s_i = proportion of infected nodes at t_i
If whole network is infected at time τ , then $s_t = 1$ for $t > \tau$
- $\mathbf{G} = (G_1, G_2, \dots, G_T)$, G_i = sub-graph of infected nodes at t_i

- **Distance:** Two distances combined

① $d_1(\mathbf{s}^1, \mathbf{s}^2) = \sqrt{\frac{1}{T} \sum_{t=1}^T (s_t^1 - s_t^2)^2}$

② $d_2(\mathbf{G}^1, \mathbf{G}^2) = \frac{1}{T} \sum_{t=1}^T \sum_{i \in G_t^1} \sum_{j \in G_t^2} \frac{\rho(i,j)}{\rho_{max}}$, where
 $\rho(i,j)$ = the shortest path between nodes i and j and
 ρ_{max} = maximum shortest path length on the network

③ Final distance:

$$\mathbf{d}((\mathbf{s}^1, \mathbf{G}^1), (\mathbf{s}^2, \mathbf{G}^2)) = \frac{1}{2} (d_1(\mathbf{s}^1, \mathbf{s}^2) + d_2(\mathbf{G}^1, \mathbf{G}^2))$$

- **Prior on the seed node:** Uniform on the infected nodes at t_1
- **Prior on the infectivity parameter:** Uniform on $[0, 1]$
- **Perturbation Kernel:** A perturbation kernel used to explore the parameter space is defined as a distribution on (θ, n_{SN}) given the present parameter values (θ^*, n_{SN}^*) ,
 $K((\theta, n_{SN}) | (\theta^*, n_{SN}^*)) = K_1(\theta | \theta^*, \hat{\sigma}) K_2(n_{SN} | n_{SN}^*)$ where

 K_1 = Gaussian with $\hat{\sigma}$ being the variance of the θ sampled in the previous step of SABC multiplied by η

 K_2 = discrete distribution on the neighboring nodes of n_{SN}^* with each node having a probability inversely proportional to its degree

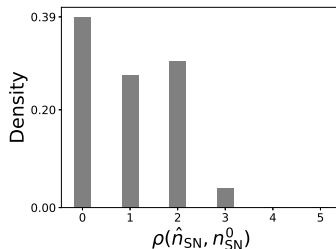
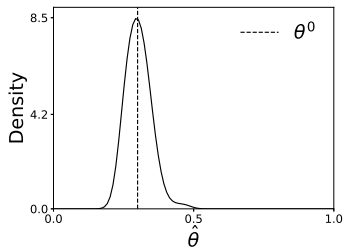
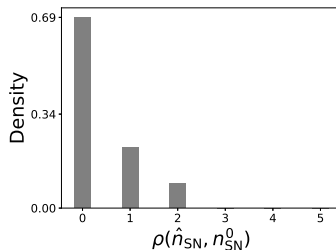
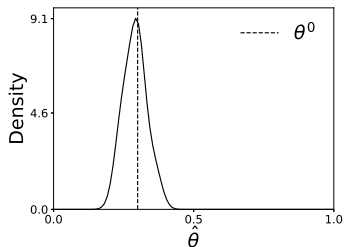
Network structure and its connectivity affect the disease spread

Two different network topologies

- 1 Barabasi-Albert network (BA):
scale-free degree distribution
- 2 Erdős-Rényi network (ER):
edges are independent and equally likely
- 3 BA($m=4$) and ER($p=0.05$) network with 100 nodes
- 4 First observed time slice: t_0
- 5 Length of the simulated SI process: T

Process = SI on network = BA (top) & ER (bottom)

- Average over 100 simulated contagion, same true parameters

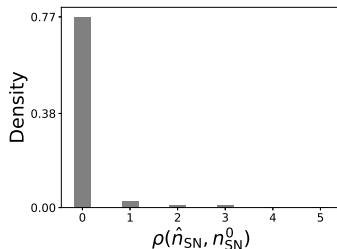
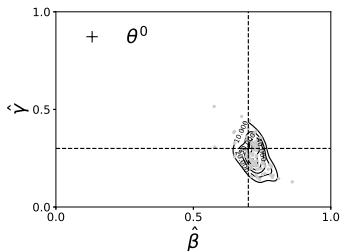
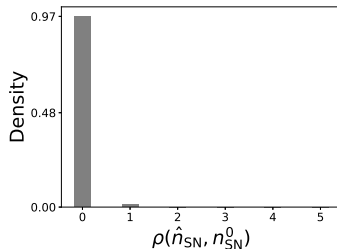
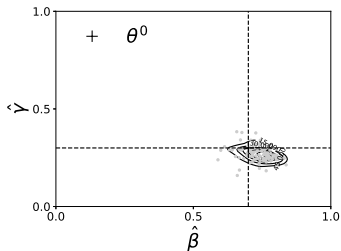


Complex contagion process (CC): Modeling fake news on social network

- **Rate of exposure (β):**
Can we infer the rate of exposure of a node to infected nodes?
- **Threshold (γ):**
Can we infer the percentage of infected neighboring nodes needed to infect a new node or to make it believe the fake news?
- **Seed-node (n_{sn}):**
Can we infer the source of the infection or of the fake news?
- **Algorithmic details** are given in **Dutta et. al. 2017d**

Process = CC on network = BA (top) & ER (bottom)

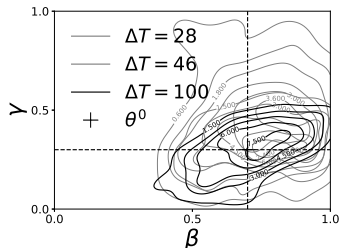
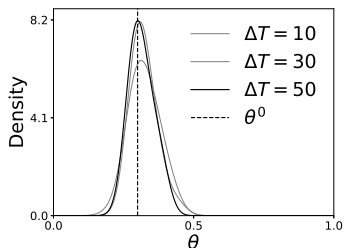
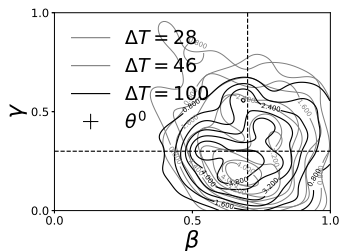
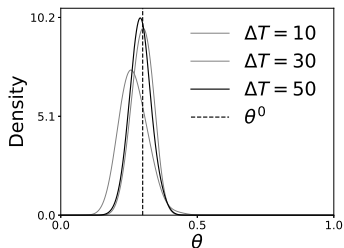
- Average over 100 simulated contagion, same true parameters



Sensitivity of inference to spreading time $\Delta t = T - t_0$

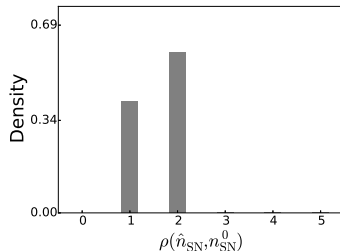
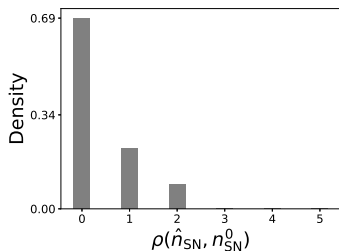
BA (top) & ER (bottom)

SC (left) & CC (right)



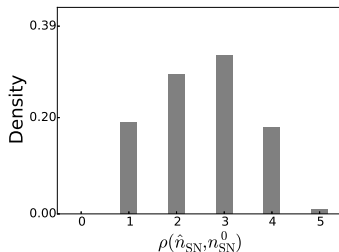
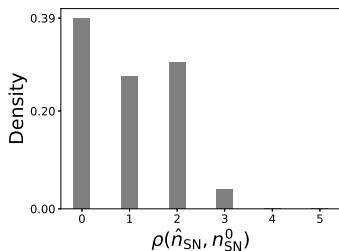
Literature comparison: Seed node detection for Simple Contagion on Barabasi-Albert network

- **Netsleuth algorithm:** Prakash et. al. *Efficiently spotting the starting points of an epidemic in a large graph* Knowledge and Information Systems, 2014
- Bayes estimate (left) vs Netsleuth algorithm (right)



Literature comparison: Seed node detection for Simple Contagion on Erdős-Rényi network

- **Netsleuth algorithm:** Prakash et. al. *Efficiently spotting the starting points of an epidemic in a large graph.* Knowledge and Information Systems, 2014
- Bayes estimate (left) vs Netsleuth algorithm (right)



Real Network: SI on Indian village contact network

- Human-human interaction network by considering a financial network between 354 villagers living in South Indian state of Karnataka
- 354 nodes and 1541 edges

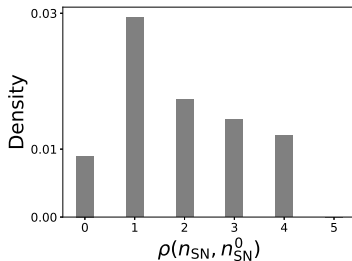
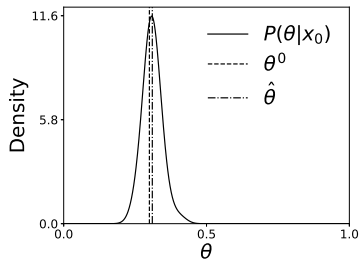


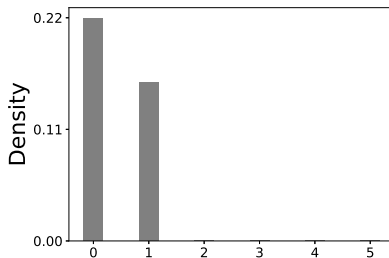
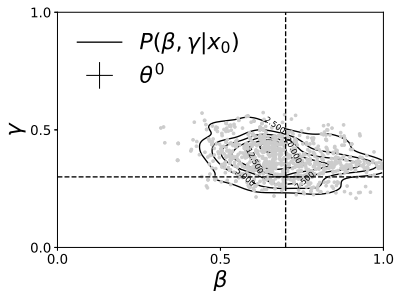
Figure 1: *

Marginal posterior distribution

The shortest path length between $n_{SN}^0 = 70$ and $\hat{n}_{SN} = 59$ is 1

Real Network: CC on Facebook social network

- 4039 nodes and 88234 edges



- **ABCpy**: A python suite of ABC, user friendly and modular [Dutta et. al. 2017a]
- **Super-computers**: Developed in collaboration with Swiss Super Computing Center (CSCS)
- **Reproducibility**
- **Usability**: In collaboration with CSCS, we offer to infer model/parameter of your problem using the most powerful super computer of Europe (**CRAY**)
- **Map-Reduce**: For parallelization we use *Map-reduce* scheme of Spark, MPI and dynamic allocation MPI (implemented by us to mitigate imbalance in ABC)

Implemented ABC algorithms

- For inference:

- 1 Rejection ABC [Tavaré et. al. 1997]
- 2 Population Monte Carlo ABC PMC-ABC [Beaumont 2010]
- 3 Sequential Monte Carlo ABC SMC-ABC [Del Moral et al 2012]
- 4 Replenishment SMC ABC RSMC-ABC [Drovandi et al 2011]
- 5 Adaptive Population Monte Carlo ABC APMC-ABC [Lenormand et al 2013]
- 6 ABC with subset simulation ABCsubsim [Chiachio et al 2014]
- 7 Simulated Annealing ABC SABC [Albert et al 2015]
- 8 (Coming soon) Surrogate modeling based ABC

- **For summary selection:** Semi-automatic summary selection [Fearnhead and Prangle, 2012]
- **Specialized distances:** Classifier-ABC [Gutmann et. al. 2018], with automatic summary selection
- **Model selection:** Random forest ensemble model selection [Pudlo et. al., 2015]
- **Additional:** Population Monte Carlo to perform pseudo-marginal approach using approximate likelihoods:
 - 1 Synthetic Likelihood [Woods 2010]
 - 2 Penalised Logistic Regression [Dutta et. al. 2017c]

- **ABC**: very powerful methodology for sound statistical inference in generative network models + processes
- **ABCpy**: python framework for ABC
- **Reference**: Dutta, Mira, Onnela
Proceedings of the Royal Society A, 2019