Robust Optimisation Monte Carlo

Michael U. Gutmann

michael.gutmann@ed.ac.uk

School of Informatics, University of Edinburgh

2 March 2021

References

Paper:
Borislav Ikonomov and Michael U. Gutmann
Robust Optimisation Monte Carlo
AISTATS 2020

Software:
Vasileios Gkolemis, Michael Gutmann
Extending the statistical software package Engine for Likelihood-Free Inference
`https://arxiv.org/abs/2011.03977`

Key messages

1. Optimisation Monte Carlo (OMC) is an existing method for efficient Bayesian inference with implicit models.
2. While efficient OMC under-estimates uncertainty by collapsing regions of near-constant likelihood into a single point.
3. A robust generalisation, robust OMC, explains and corrects this failure mode while maintaining OMC's benefits.

Contents

Background: Optimisation Monte Carlo

Contribution 1: A failure mode of Optimisation Monte Carlo

Contribution 2: Robust Optimisation Monte Carlo (ROMC)

Contents

Background: Optimisation Monte Carlo

Contribution 1: A failure mode of Optimisation Monte Carlo

Contribution 2: Robust Optimisation Monte Carlo (ROMC)

Problem considered

Given

- a parametric model $p(\mathbf{x}|\boldsymbol{\theta})$ whose likelihood function is intractable but from which we can generate samples

$$\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta})$$

- a prior distribution $p(\boldsymbol{\theta})$ on $\boldsymbol{\theta}$
- observed data $\mathbf{x}_o$

estimate $p(\boldsymbol{\theta}|\mathbf{x}_o)$ / obtain approximate samples from it.

Assumptions

- The parametric model $p(\mathbf{x}|\boldsymbol{\theta})$ is implicitly defined by a simulator/generative process $g(\boldsymbol{\theta}, \mathbf{u})$

$$\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta}) \qquad \Longleftrightarrow \qquad \mathbf{x} = g(\boldsymbol{\theta}, \mathbf{u}), \quad \mathbf{u} \sim p(\mathbf{u})$$

- $g(\boldsymbol{\theta}, \mathbf{u})$ is a black-box computer programme taking $\boldsymbol{\theta}$ as input. Randomness is represented by $\mathbf{u} \sim p(\mathbf{u})$.
- We are provided with a distance (discrepancy) function $d(\mathbf{x}, \mathbf{x}_o)$ between simulated data $\mathbf{x}$ and observed data $\mathbf{x}_o$.

The distance function

- There are methods for likelihood-free inference that do not require a distance function, e.g. by
  - modelling the likelihood (e.g. Wood, 2010; Price et al, 2017; Papamakarios 2019)
  - framing posterior estimation as a ratio estimation problem

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

  "Likelihood-free inference by ratio estimation" (LFIRE)

  (Thomas et al, 2016, 2020; Hermans et al 2020)

The distance function

- There are methods for likelihood-free inference that do not require a distance function, e.g. by
  - modelling the likelihood (e.g. Wood, 2010; Price et al, 2017; Papamakarios 2019)
  - framing posterior estimation as a ratio estimation problem

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

    "Likelihood-free inference by ratio estimation" (LFIRE)

    (Thomas et al, 2016, 2020; Hermans et al 2020)

- Optimisation Monte Carlo requires a distance function

$$d(\mathbf{x}, \mathbf{x}_o) = ||\Phi(\mathbf{x}) - \Phi(\mathbf{x}_o)||_2$$

  where $\Phi(.)$ are known summary statistics.

Optimisation Monte Carlo (Meeds and Welling, NIPS 2015)

- Main property: uses optimisation to increase efficiency.

Optimisation Monte Carlo (Meeds and Welling, NIPS 2015)

- Main property: uses optimisation to increase efficiency.
- Assumptions:
  - (approximate) derivative of $\Phi(\mathbf{x}) = \Phi(g(\boldsymbol{\theta}, \mathbf{u})) = \mathbf{f}(\boldsymbol{\theta}, \mathbf{u})$ wrt $\boldsymbol{\theta}$ is available
  - $\dim(\boldsymbol{\theta}) \leq \dim(\Phi(\mathbf{x}))$

Optimisation Monte Carlo (Meeds and Welling, NIPS 2015)

- Main property: uses optimisation to increase efficiency.
- Assumptions:
  - (approximate) derivative of $\Phi(\mathbf{x}) = \Phi(g(\boldsymbol{\theta}, \mathbf{u})) = \mathbf{f}(\boldsymbol{\theta}, \mathbf{u})$ wrt $\boldsymbol{\theta}$ is available
  - $\dim(\boldsymbol{\theta}) \leq \dim(\Phi(\mathbf{x}))$
- Algorithm to generate n weighted samples $\boldsymbol{\theta}_i^*$:

1: **for** $i \leftarrow 1$ to n **do**
2: $\quad \mathbf{u}_i \sim p(\mathbf{u})$ ▷ Set seed
3: $\quad \boldsymbol{\theta}_i^* = \arg\min_{\boldsymbol{\theta}} ||\mathbf{f}(\boldsymbol{\theta}, \mathbf{u}_i) - \Phi(\mathbf{x}_o)||_2$ ▷ Optimisation
4: $\quad$ Compute $\mathbf{J}_i$ with columns $\partial \mathbf{f}(\boldsymbol{\theta}_i^*, \mathbf{u}_i)/\partial\theta_k$
5: $\quad$ Compute $w_i = p(\boldsymbol{\theta}_i^*) * (\det(\mathbf{J}_i^\top \mathbf{J}_i))^{-1/2}$
6: $\quad$ Accept $\boldsymbol{\theta}_i^*$ as posterior sample with weight w_i

(Note that samples with too large final distances may be omitted.)

Intuition

Intuition for weight formula $w_i = p(\boldsymbol{\theta}_i^*) * (\det(\mathbf{J}_i^\top \mathbf{J}_i))^{-1/2}$

- Volume of the region around a posterior sample $\boldsymbol{\theta}_i^*$ containing points that should also be considered posterior samples.
- $(\det(\mathbf{J}_i^\top \mathbf{J}_i))^{-1/2}$ is proportional to the volume of an ellipse defined by $\mathbf{J}_i^\top \mathbf{J}_i$.

Contents

Background: Optimisation Monte Carlo

Contribution 1: A failure mode of Optimisation Monte Carlo

Contribution 2: Robust Optimisation Monte Carlo (ROMC)

Application to inverse graphics

- Implicit model/simulator given by a graphics renderer
- We used Open Differential Renderer (Loper and Black, 2014)

20 parameters:
- Shape
- Rotation/Pose
- Illumination
- Colour

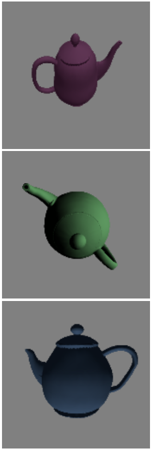
Renderer

(forward problem)

$\rightleftharpoons$

Inference

(inverse problem)



Application to inverse graphics

- Example considered: Infer colour of the object when external lighting conditions are unknown.
- The inverse problem may have multiple solutions (multi-modal posterior)

(a) Gray teapot under red light.

(b) Red teapot under white light.

Results for colour inference task

- We used OMC and the (simpler) rejection ABC algorithm.

Results for colour inference task

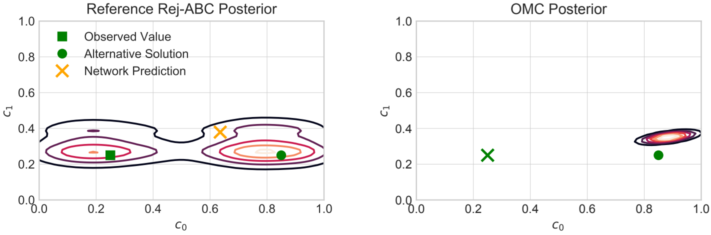
- We used OMC and the (simpler) rejection ABC algorithm.
- Rejection ABC relies on trial and error instead of optimisation to determine the posterior samples. Slow but reliable.

Results for colour inference task

- We used OMC and the (simpler) rejection ABC algorithm.
- Rejection ABC relies on trial and error instead of optimisation to determine the posterior samples. Slow but reliable.
- Same distance function $d(\mathbf{x}, \mathbf{x}_o)$.

Results for colour inference task

- We used OMC and the (simpler) rejection ABC algorithm.
- Rejection ABC relies on trial and error instead of optimisation to determine the posterior samples. Slow but reliable.
- Same distance function $d(\mathbf{x}, \mathbf{x}_o)$.
- Posteriors for two colours c_0 and c_1 (red and green):

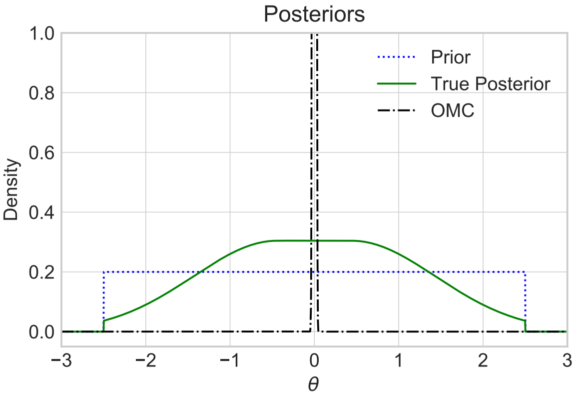


Why did OMC fail?

- The OMC weights $w_i = p(\boldsymbol{\theta}_i^*) * (\det(\mathbf{J}_i^\top \mathbf{J}_i))^{-1/2}$ are unstable (ESS was 1.2!)
- This happens when the (approximate) likelihood function has nearly flat regions so that $\det(\mathbf{J}_i^\top \mathbf{J}_i) \approx 0$.

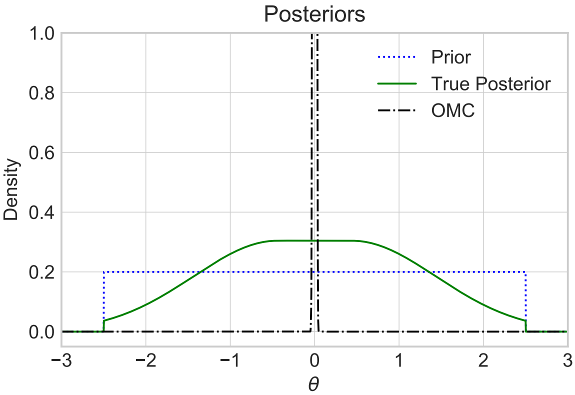
Why did OMC fail?

- The OMC weights $w_i = p(\boldsymbol{\theta}_i^*) * (\det(\mathbf{J}_i^\top \mathbf{J}_i))^{-1/2}$ are unstable (ESS was 1.2!)
- This happens when the (approximate) likelihood function has nearly flat regions so that $\det(\mathbf{J}_i^\top \mathbf{J}_i) \approx 0$.



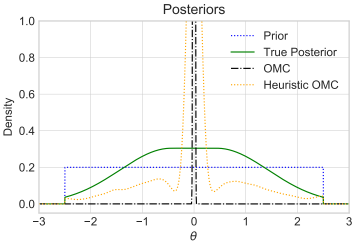
Why did OMC fail?

- The OMC weights $w_i = p(\boldsymbol{\theta}_i^*) * (\det(\mathbf{J}_i^\top \mathbf{J}_i))^{-1/2}$ are unstable (ESS was 1.2!)
- This happens when the (approximate) likelihood function has nearly flat regions so that $\det(\mathbf{J}_i^\top \mathbf{J}_i) \approx 0$.
  Note: stated OMC assumptions are not violated.

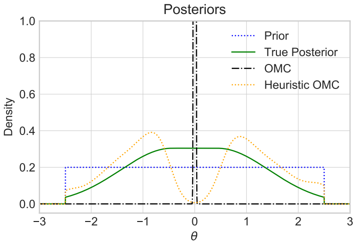


Stabilising the weights/matrices does not help

- Taking the pseudo-inverse or pseudo-determinant of $\mathbf{J}_i^\top \mathbf{J}_i$ does not help.



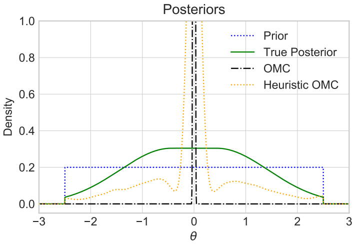
(a) Pseudo-inverse



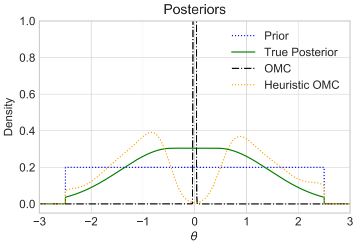
(b) Pseudo-determinant

Stabilising the weights/matrices does not help

- Taking the pseudo-inverse or pseudo-determinant of $\mathbf{J}_i^\top \mathbf{J}_i$ does not help.
- The weights are not the real issue. The problem is more fundamental:
  OMC uses a single point to represent an entire region where the likelihood is (nearly) constant.



(a) Pseudo-inverse



(b) Pseudo-determinant

Contents

Background: Optimisation Monte Carlo

Contribution 1: A failure mode of Optimisation Monte Carlo

Contribution 2: Robust Optimisation Monte Carlo (ROMC)

Key properties of ROMC

(Ikonomov and Gutmann, AISTATS 2020)

1. ROMC generalises OMC.

2. Fixes OMC's failure case: It handles likelihood functions that are (nearly) flat on significant regions in parameter space.

3. Works for general distance functions $d(g(\boldsymbol{\theta}, \mathbf{u}), \mathbf{x}_o)$ and not only Euclidean distances between summary statistics.
   (condition $\dim(\boldsymbol{\theta}) \leq \dim(\Phi(\mathbf{x}))$ disappears)

4. Does not require (approximate) derivatives, while OMC does.

5. Can be run as post-processing to OMC or from scratch.

The ROMC framework

ROMC is a framework for inference. It has three key steps:

1. For $i = 1, \dots n'$, sample $\mathbf{u}_i \sim p(\mathbf{u})$ and determine

$$\boldsymbol{\theta}_i^* = \arg\min_{\boldsymbol{\theta}} d(g(\boldsymbol{\theta}, \mathbf{u}_i), \mathbf{x}_o)$$

   Same as in OMC but we can use general distances $d(\mathbf{x}, \mathbf{x}_o)$.

The ROMC framework

ROMC is a framework for inference. It has three key steps:

1. For $i = 1, \dots n'$, sample $\mathbf{u}_i \sim p(\mathbf{u})$ and determine

$$\boldsymbol{\theta}_i^* = \arg\min_{\boldsymbol{\theta}} d(g(\boldsymbol{\theta}, \mathbf{u}_i), \mathbf{x}_o)$$

   Same as in OMC but we can use general distances $d(\mathbf{x}, \mathbf{x}_o)$.

2. Use the minimal distances $d_i^* = d(g(\boldsymbol{\theta}_i^*, \mathbf{u}_i))$ to choose an acceptance threshold ϵ / keep the n best $\boldsymbol{\theta}_i^*$.

The ROMC framework

ROMC is a framework for inference. It has three key steps:

1. For $i = 1, \dots n'$, sample $\mathbf{u}_i \sim p(\mathbf{u})$ and determine

$$\boldsymbol{\theta}_i^* = \arg\min_{\boldsymbol{\theta}} d(g(\boldsymbol{\theta}, \mathbf{u}_i), \mathbf{x}_o)$$

   Same as in OMC but we can use general distances $d(\mathbf{x}, \mathbf{x}_o)$.

2. Use the minimal distances $d_i^* = d(g(\boldsymbol{\theta}_i^*, \mathbf{u}_i))$ to choose an acceptance threshold ϵ / keep the n best $\boldsymbol{\theta}_i^*$.
3. For each i where $d_i^* \leq \epsilon$, define a proposal distribution q_i on the "acceptance region" $C_\epsilon^i = \{\boldsymbol{\theta} : d(g(\boldsymbol{\theta}, \mathbf{u}_i), \mathbf{x}_o) \leq \epsilon\}$

The ROMC framework

ROMC is a framework for inference. It has three key steps:

1. For $i = 1, \dots n'$, sample $\mathbf{u}_i \sim p(\mathbf{u})$ and determine

$$\boldsymbol{\theta}_i^* = \arg\min_{\boldsymbol{\theta}} d(g(\boldsymbol{\theta}, \mathbf{u}_i), \mathbf{x}_o)$$

   Same as in OMC but we can use general distances $d(\mathbf{x}, \mathbf{x}_o)$.
2. Use the minimal distances $d_i^* = d(g(\boldsymbol{\theta}_i^*, \mathbf{u}_i))$ to choose an acceptance threshold ϵ / keep the n best $\boldsymbol{\theta}_i^*$.
3. For each i where $d_i^* \leq \epsilon$, define a proposal distribution q_i on the "acceptance region" $C_\epsilon^i = \{\boldsymbol{\theta} : d(g(\boldsymbol{\theta}, \mathbf{u}_i), \mathbf{x}_o) \leq \epsilon\}$

Approximate posterior is represented by weighted samples $\boldsymbol{\theta}_{ij}$:

$$\boldsymbol{\theta}_{ij} \sim q_i(\boldsymbol{\theta}) \qquad w_{ij} = \mathbb{1}_{C_\epsilon^i}(\boldsymbol{\theta}_{ij}) \frac{p(\boldsymbol{\theta}_{ij})}{q_i(\boldsymbol{\theta}_{ij})} \qquad \text{(i=1, \dots n; \quad j=1, \dots, m)}$$

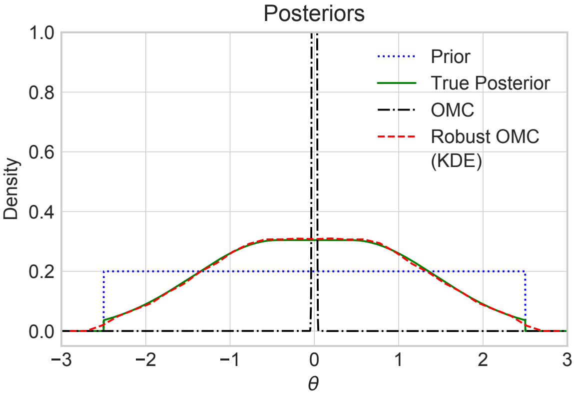
Construction of the proposal distribution

(General idea, see paper for details)

- Using $\boldsymbol{\theta}_i^*$ and the optimisation trajectory, we build a model of the acceptance regions $C_\epsilon^i = \{\boldsymbol{\theta} : d(g(\boldsymbol{\theta}, \mathbf{u}_i), \mathbf{x}_o) \leq \epsilon\}$
- Simple but effective: model C_ϵ^i as a hypercube or ellipse and define q_i to be the uniform distribution on it.

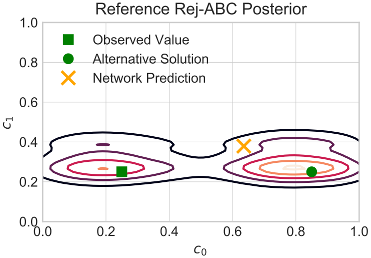
Results on the toy example

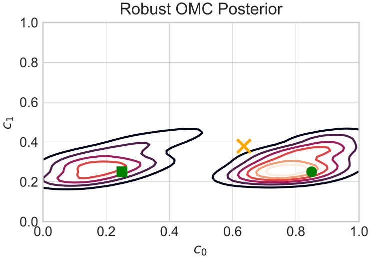
- Acceptance regions C_ϵ^i given by intervals on the line.
- ROMC handles the (nearly) flat likelihood function correctly.
- ROMC accurately represents uncertainty while OMC does not.



Results on the colour inference task

- Setup:
  - Optimisation via a gradient-free method (Bayesian optimisation with GP surrogate modelling)
  - Acceptance regions were modelled as ellipses (derived from the GP surrogate model)
- ROMC posterior matches reference posterior well.
- Effective sample size: 97% (vs. approx. 0.5% for OMC)





Conclusions

- Talk was on Bayesian inference for implicit models.

Conclusions

- Talk was on Bayesian inference for implicit models.
  - Implicit models: models that are defined by a stochastic simulator/data generating process.

Conclusions

- Talk was on Bayesian inference for implicit models.
  - Implicit models: models that are defined by a stochastic simulator/data generating process.
  - Optimisation Monte Carlo (OMC): Bayesian inference method that uses optimisation to increase computational efficiency.

Conclusions

- Talk was on Bayesian inference for implicit models.
  - Implicit models: models that are defined by a stochastic simulator/data generating process.
  - Optimisation Monte Carlo (OMC): Bayesian inference method that uses optimisation to increase computational efficiency.
- We showed that OMC under-estimates posterior uncertainty by collapsing regions of near-constant likelihood into a point.

Conclusions

- Talk was on Bayesian inference for implicit models.
  - Implicit models: models that are defined by a stochastic simulator/data generating process.
  - Optimisation Monte Carlo (OMC): Bayesian inference method that uses optimisation to increase computational efficiency.
- We showed that OMC under-estimates posterior uncertainty by collapsing regions of near-constant likelihood into a point.
- We proposed a robust generalisation of OMC, robust OMC, that explains and corrects this failure mode while maintaining OMC's benefits due to optimisation.

Code

- ROMC has been added to the software package “ELFI: Engine for Likelihood-Free Inference”
  `https://github.com/elfi-dev/elfi`
- Link to collab notebooks available through
  Vasileios Gkolemis, Michael Gutmann
  Extending the statistical software package Engine for Likelihood-Free Inference
  `https://arxiv.org/abs/2011.03977`

ROMC generalises OMC (see paper for the proof)

Theorem: Under the below assumptions, ROMC becomes equivalent to standard OMC as $\epsilon \to 0$.

Assumption 1. The distance $d(g(\boldsymbol{\theta}, \mathbf{u}), \mathbf{x}_o)$ is given by the Euclidean distance between summary statistics $||\mathbf{f}(\boldsymbol{\theta}, \mathbf{u}) - \Phi(\mathbf{x}_o)||$.

Assumption 2. The proposal distribution $q_i(\boldsymbol{\theta})$ is the uniform distribution on C_ϵ^i.

Assumption 3. The acceptance regions C_ϵ^i are approximated by the ellipsoid $C_\epsilon^i = \{\boldsymbol{\theta} : (\boldsymbol{\theta} - \boldsymbol{\theta}_i^*)^\top \mathbf{J}_i{}^\top \mathbf{J}_i(\boldsymbol{\theta} - \boldsymbol{\theta}_i^*) \leq \epsilon\}$ where $\mathbf{J}_i$ is the Jacobian matrix with columns $\partial \mathbf{f}(\boldsymbol{\theta}_i^*, \mathbf{u}_i)/\partial \theta_k$.

Assumption 4. The matrix square root $\mathbf{A}_i$ of $\mathbf{J}_i{}^\top \mathbf{J}_i$ is full rank, i.e. $\text{rank}(\mathbf{A}_i) = \dim(\boldsymbol{\theta})$.

Assumption 5. The prior $p(\boldsymbol{\theta})$ is constant on the acceptance regions C_ϵ^i.

Explanation of the failure case

Identified failure case is due to violation of Assumptions 3 and 4:

Assumption 3. The acceptance regions C_ϵ^i are approximated by the ellipsoid $C_\epsilon^i = \{\boldsymbol{\theta} : (\boldsymbol{\theta} - \boldsymbol{\theta}_i^*)^\top \mathbf{J}_i^\top \mathbf{J}_i (\boldsymbol{\theta} - \boldsymbol{\theta}_i^*) \leq \epsilon\}$ where $\mathbf{J}_i$ is the Jacobian matrix with columns $\partial \mathbf{f}(\boldsymbol{\theta}_i^*, \mathbf{u}_i)/\partial \theta_k$.

Assumption 4. The matrix square root $\mathbf{A}_i$ of $\mathbf{J}_i^\top \mathbf{J}_i$ is full rank, i.e. $\text{rank}(\mathbf{A}_i) = \dim(\boldsymbol{\theta})$.

Explanation of the failure case

Identified failure case is due to violation of Assumptions 3 and 4:

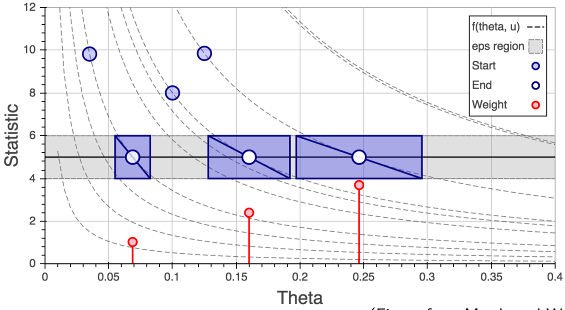
Assumption 3. The acceptance regions C_ϵ^i are approximated by the ellipsoid $C_\epsilon^i = \{\boldsymbol{\theta} : (\boldsymbol{\theta} - \boldsymbol{\theta}_i^*)^\top \mathbf{J}_i^\top \mathbf{J}_i (\boldsymbol{\theta} - \boldsymbol{\theta}_i^*) \leq \epsilon\}$ where $\mathbf{J}_i$ is the Jacobian matrix with columns $\partial \mathbf{f}(\boldsymbol{\theta}_i^*, \mathbf{u}_i)/\partial \theta_k$.

Assumption 4. The matrix square root $\mathbf{A}_i$ of $\mathbf{J}_i^\top \mathbf{J}_i$ is full rank, i.e. $\text{rank}(\mathbf{A}_i) = \dim(\boldsymbol{\theta})$.

For non-uniform priors, one then also risks violating Assumption 5:
Assumption 5. The prior $p(\boldsymbol{\theta})$ is constant on the acceptance regions C_ϵ^i.

Explanation of the failure case

- OMC uses only information at $\boldsymbol{\theta}_i^*$ to approximate C_ϵ^i.
- ROMC in contrast uses information in a non-negligible neighbourhood around $\boldsymbol{\theta}_i^*$ to approximate C_ϵ^i.



(Figure from Meeds and Welling, NIPS 2015)