MCMC Sampling and Optimization Methods

(CIV6540 - Probabilistic Machine Learning for Civil Engineers)

Professor: James-A. Goulet

Département des génies civil, géologique et des mines
Polytechnique Montréal

Chapters 5 & 7 – Goulet (2020)
Probabilistic Machine Learning for Civil Engineers
MIT Press

Introduction

Module 4: Methods for estimating parameters, $\boldsymbol{\theta}$

$$\underbrace{f(\boldsymbol{\theta}|\mathcal{D})}_{\text{posterior}} = \frac{\overbrace{f(\mathcal{D}|\boldsymbol{\theta})f(\boldsymbol{\theta})}^{\text{unnormalized posterior}}}{\underbrace{f(\mathcal{D})}_{\text{unknown normalization cte.}}}$$

MCMC: Markov Chain Monte Carlo
Obtain samples from an unnormalized posterior,
i.e. parameter samples density $\propto$ posterior

Gradient-based optimization Methods:
Identify the most likely set of parameter values from an unnormalized posterior

Ethymology

MCMC: Markov Chain Monte Carlo

Markov Hypothesis

Markov Hypothesis:
Given the present, the future is independent of the past

Starting from $X_t \sim p(x_t)$, we can transition to X_{t+1} using a transition function $p(x_{t+1}|x_t)$ that only depends of X_t.

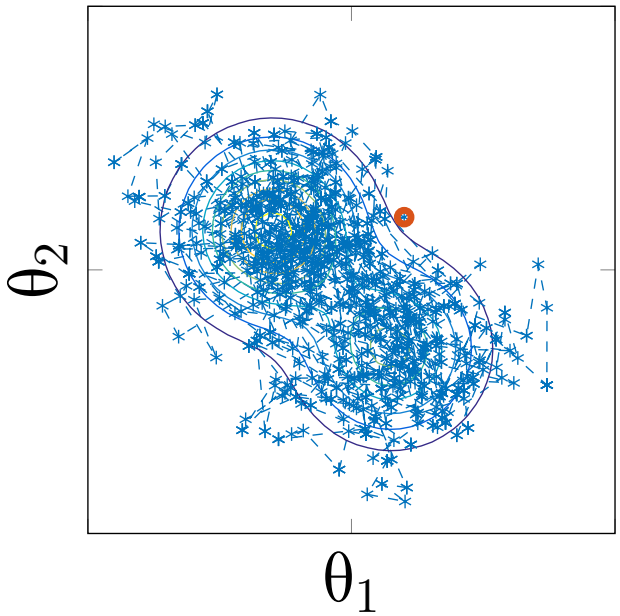
$$p(x_{t+1}|x_t) \equiv p(x_{t+1}|x_1, \cdots, x_t)$$

A **Markov Chain** is defined as the joint distribution

$$p(x_{1:\mathrm{T}}) = p(x_1)p(x_2|x_1)p(x_3|x_2)\cdots = p(x_1)\prod_{t=2}^{\mathrm{T}} p(x_t|x_{t-1})$$

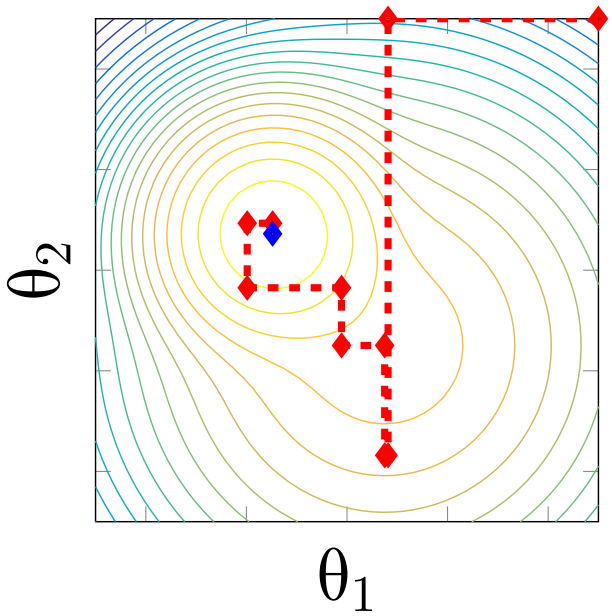
MCMC Preview

Randomly walk through the parameter domain while the **fraction of time spent** at each state $\boldsymbol{\theta}_i$ is $\propto$ **the unnormalized posterior**



Gradient-Based Optimization Preview

Find the **unnormalized posterior** most likely value by employing its **first and second derivatives**



Module #4 Outline

Intro
Metropolis
Convergence
Proposal PDF
Newton & Laplace
Transformation
Example

Topics organization

Background
- 1 Revision probability & linear algebra
- 2 Probability distributions

Machine Learning Basics
- 0 Introduction
- 3 Bayesian Estimation $p(A|B) = \frac{p(B|A)p(A)}{p(B)}$
- 4 MCMC sampling & Newton

Supervised learning
- 5 Regression
- 6 Classification
- 7 LSTM networks for time series

Unsupervised learning
- 7 State-space model for time-series

Decision Making & RL
- 8 Decision Theory
- 9 AI & Sequential decision problems

Section Outline

Metropolis

2.1 Introduction
2.2 Formulation 1D
2.3 Formulation 2D
2.4 What should we do with samples?
2.5 Metropolis-Hastings

History

Metropolis algorithm (Metropolis, 1953) was **developed during WWII** when working on the **Manhattan project** (atomic bomb) at Los Alamos.

Not the most efficient, yet a **simple** and **general** MCMC method.

Other MCMC methods:

- Metropolis-Hasting
- Gibbs Sampling
- Slice Sampling
- Hamiltonian Monte Carlo
- ...

Metropolis algorithm

Initial state: $\boldsymbol{\theta}_0 = [\theta_{1,0}, \theta_{2,0}, \cdots, \theta_{\mathrm{P},0}]^\mathsf{T}$

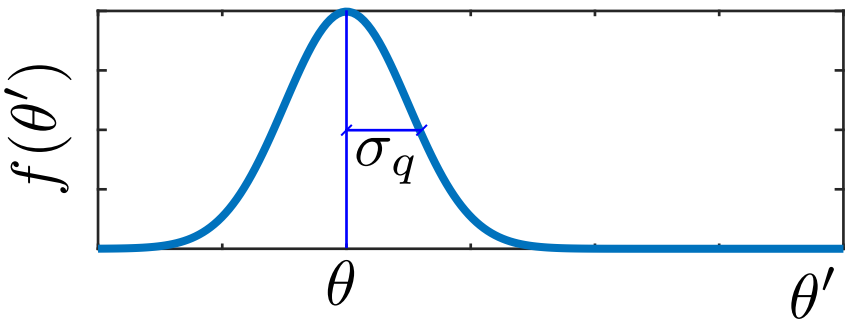
Target distribution: $\tilde{f}(\boldsymbol{\theta}) = f(\mathcal{D}|\boldsymbol{\theta}) f(\boldsymbol{\theta})$
(i.e. unnormalized posterior we want to sample from)

Proposal distribution: $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$
The proposal must have non-zero probability to transition to any state in the target distribution and must be symmetric, i.e.

$$q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\theta}')$$

The **normal distribution** is a general purpose proposal distribution

$$q(\theta'|\theta) = \mathcal{N}(\theta'; \theta, \sigma_q^2)$$



Metropolis algorithm

Initial state: $\boldsymbol{\theta}_0 = [\theta_{1,0}, \theta_{2,0}, \cdots, \theta_{\mathrm{P},0}]^\mathsf{T}$

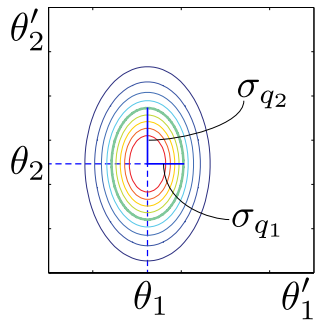
Target distribution: $\tilde{f}(\boldsymbol{\theta}) = f(\mathcal{D}|\boldsymbol{\theta})f(\boldsymbol{\theta})$
(i.e. unnormalized posterior we want to sample from)

Proposal distribution: $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$
The proposal must have non-zero probability to transition to any state in the target distribution and must be symmetric, i.e.

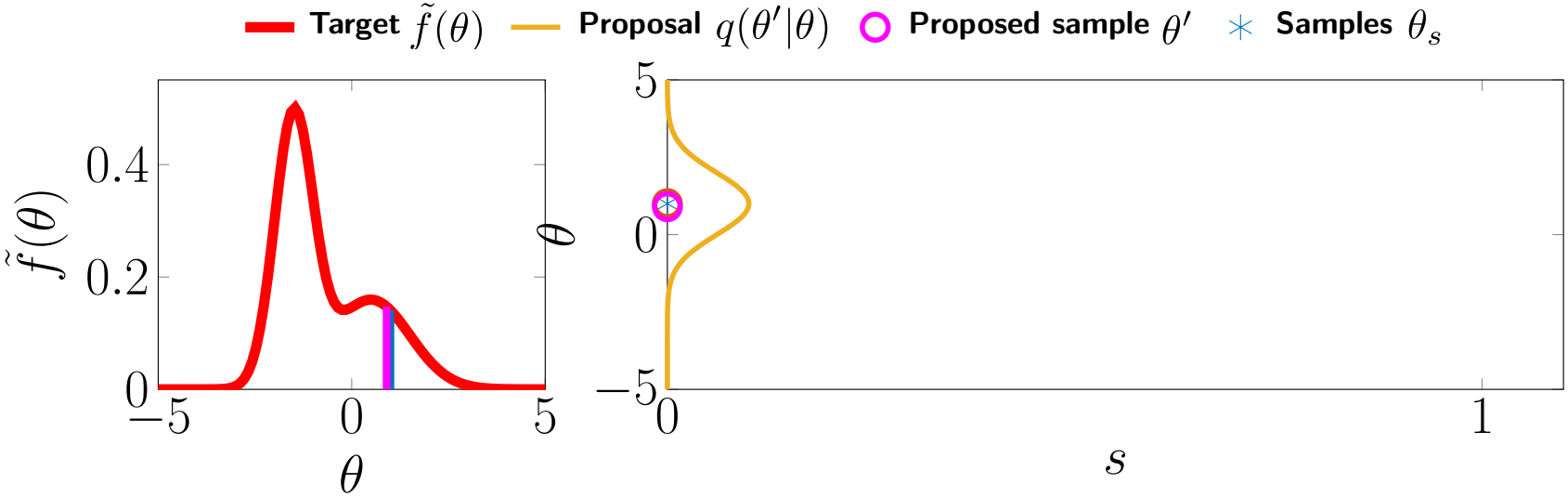
$$q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\theta}')$$

The **normal distribution** is a general purpose proposal distribution

$$q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{\theta}'; \boldsymbol{\theta}, \begin{bmatrix} \sigma_{q_1}^2 & 0 \\ 0 & \sigma_{q_2}^2 \end{bmatrix}\right)$$



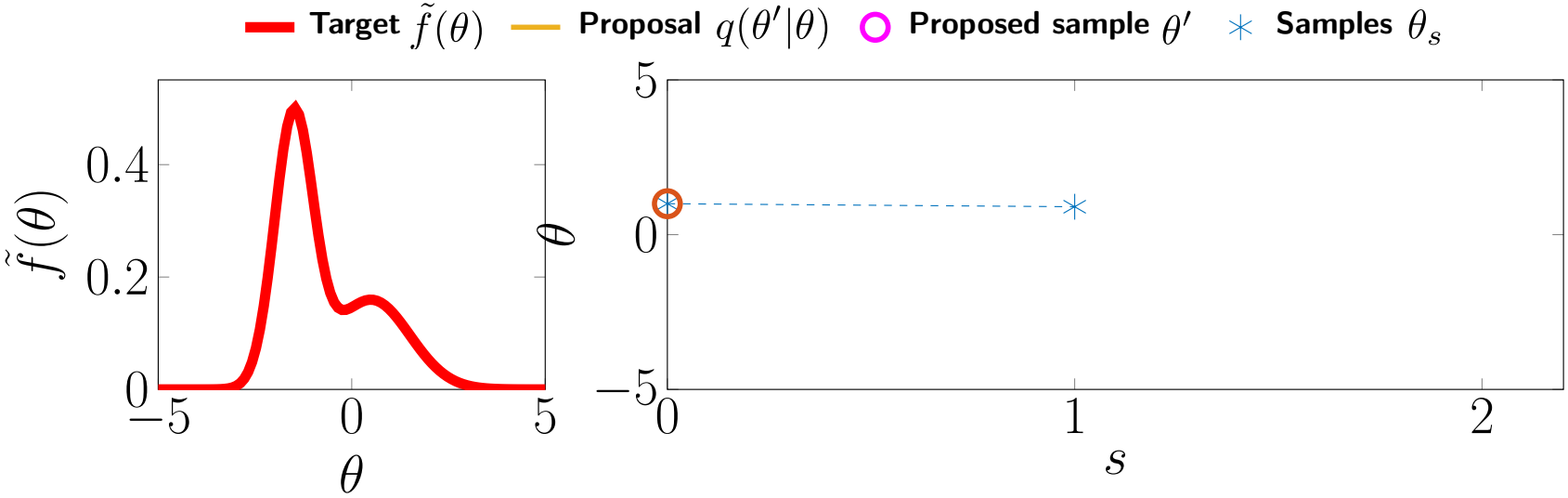
1D Example – Metropolis Algorithm [.m]



$$s = 0$$
$$\theta = \theta_s = 1 \rightarrow \tilde{f}(\theta) = 0.14083$$
$$\theta' = 0.9022 \rightarrow \tilde{f}(\theta') = 0.14718$$
$$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)} = 1.0451$$
$$\alpha \geq 1 \rightarrow \theta_{s+1} = \theta'$$

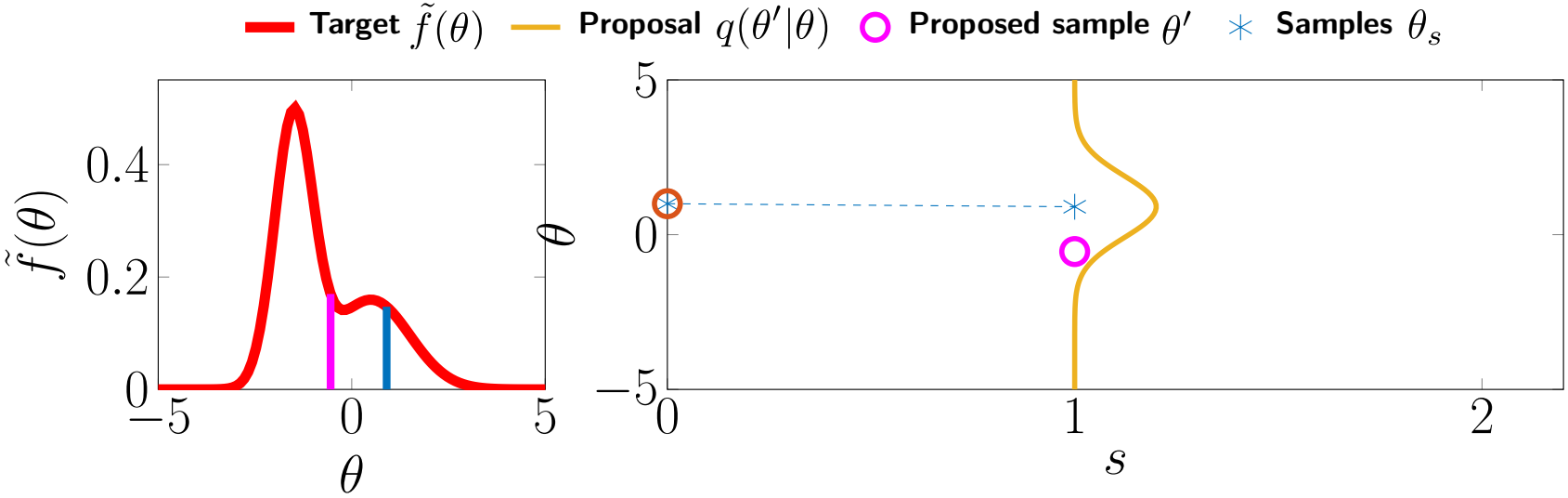
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



[CIVML/MCMC_M_demo_1D.m]

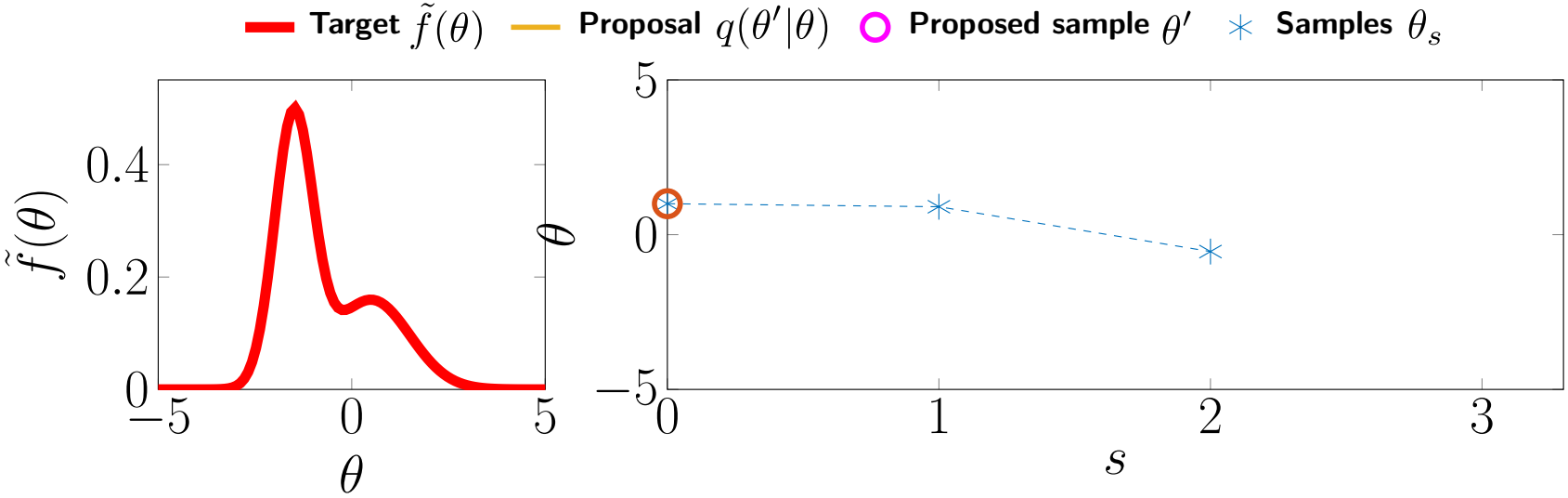
1D Example – Metropolis Algorithm [.m]



$$s = 1$$
$$\theta = \theta_s = 0.9022 \rightarrow \tilde{f}(\theta) = 0.14718$$
$$\theta' = -0.54574 \rightarrow \tilde{f}(\theta') = 0.16984$$
$$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)} = 1.1539$$
$$\alpha \geq 1 \rightarrow \theta_{s+1} = \theta'$$

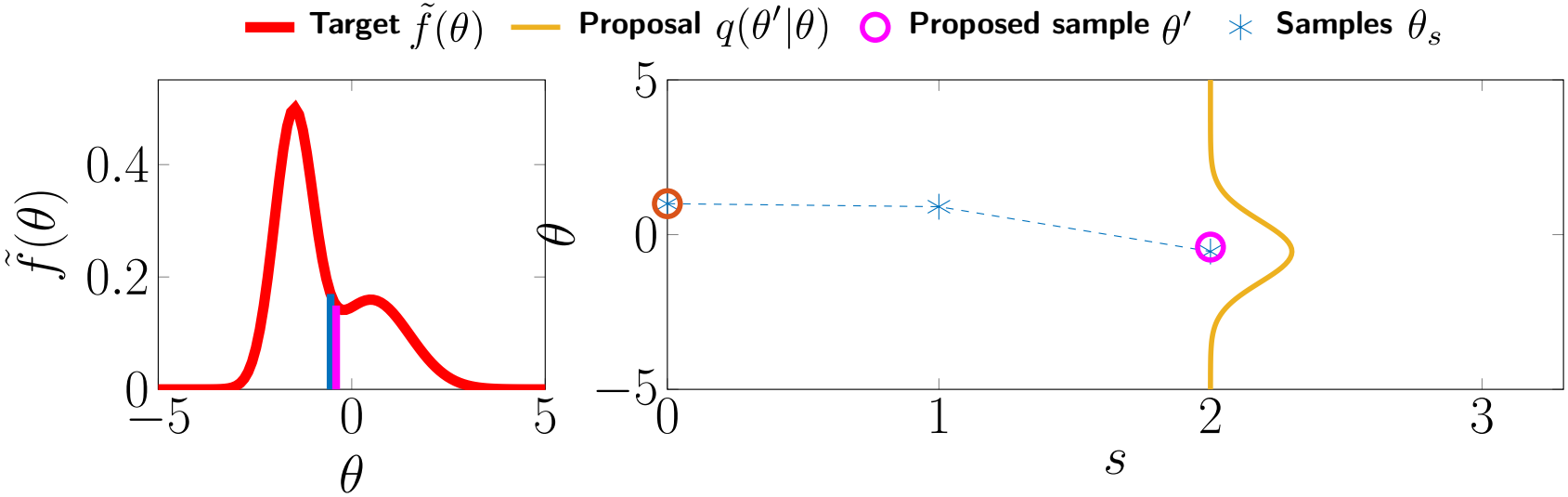
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



[CIVML/MCMC_M_demo_1D.m]

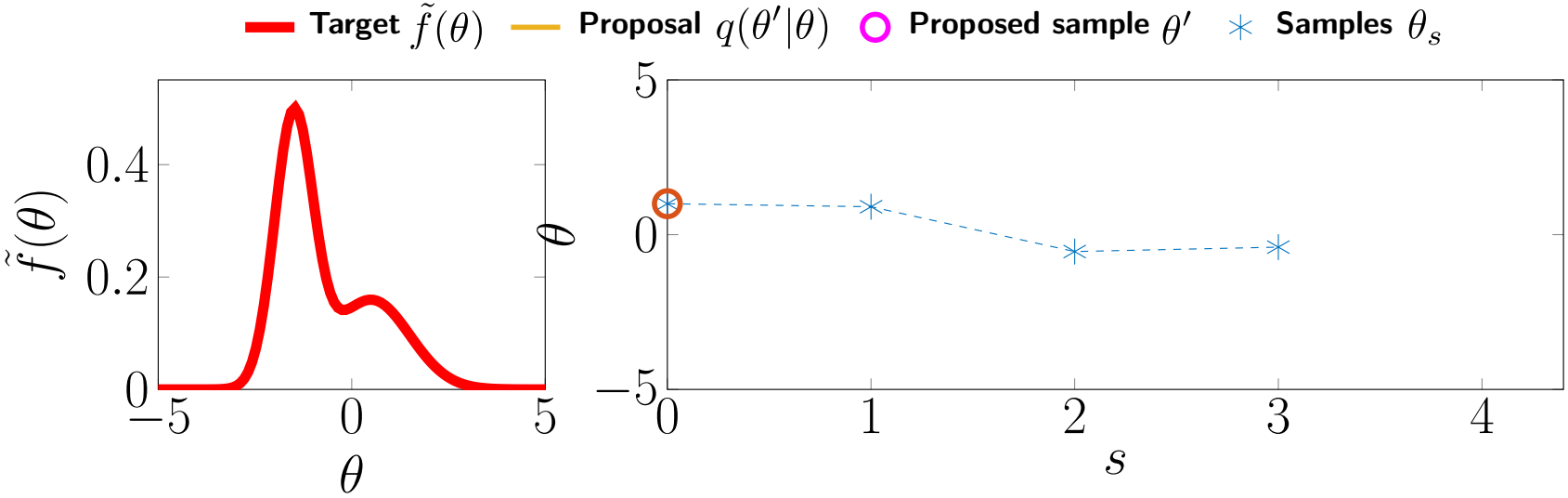
1D Example – Metropolis Algorithm [.m]



$$s = 2$$
$$\theta = \theta_s = \text{-}0.54574 \rightarrow \tilde{f}(\theta) = 0.16984$$
$$\theta' = \text{-}0.40253 \rightarrow \tilde{f}(\theta') = 0.14924$$
$$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)} = 0.87872$$
$$\alpha \leq 1 \rightarrow u = 0.66116$$
$$u < \alpha \rightarrow \theta_{s+1} = \theta'$$

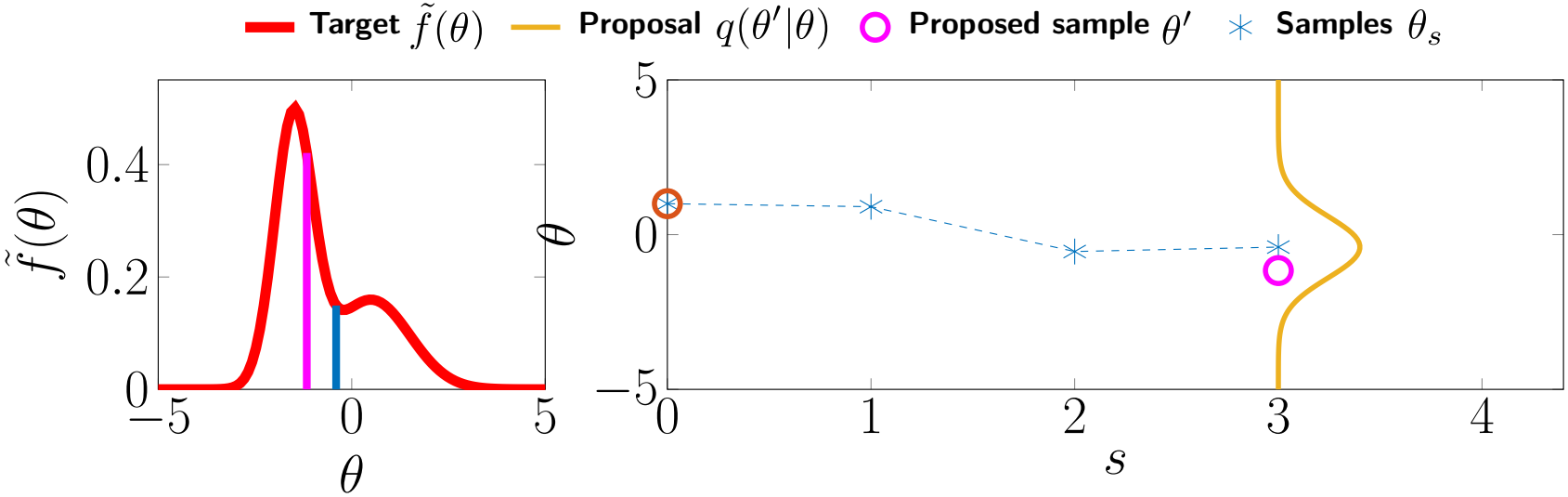
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



s =3

$\theta = \theta_s$ =-0.40253 $\rightarrow$ $\tilde{f}(\theta)$ =0.14924

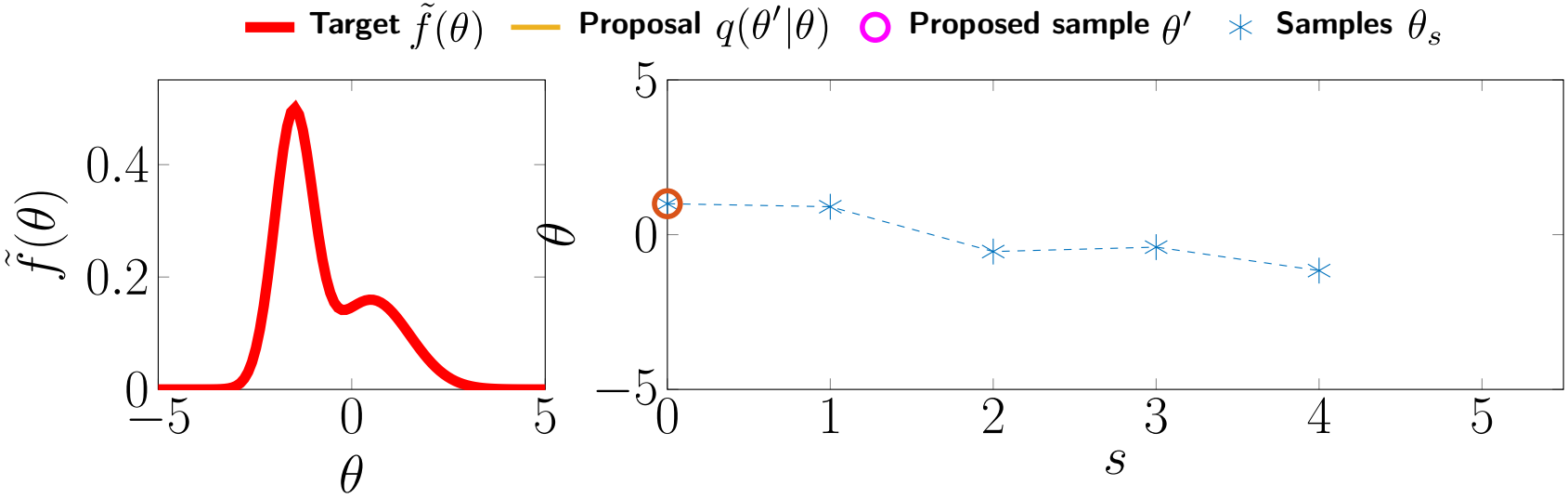
θ' =-1.1613 $\rightarrow$ $\tilde{f}(\theta')$ =0.42072

$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)}$ =2.8191

$\alpha \geq 1 \rightarrow \theta_{s+1} = \theta'$

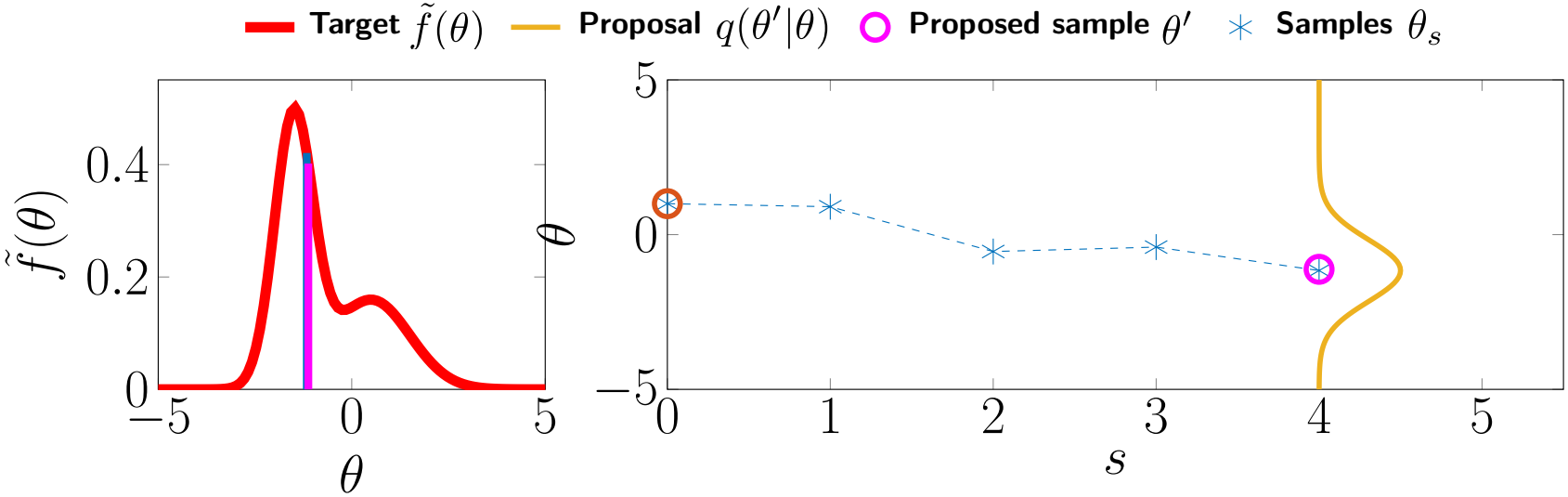
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



[CIVML/MCMC_M_demo_1D.m]

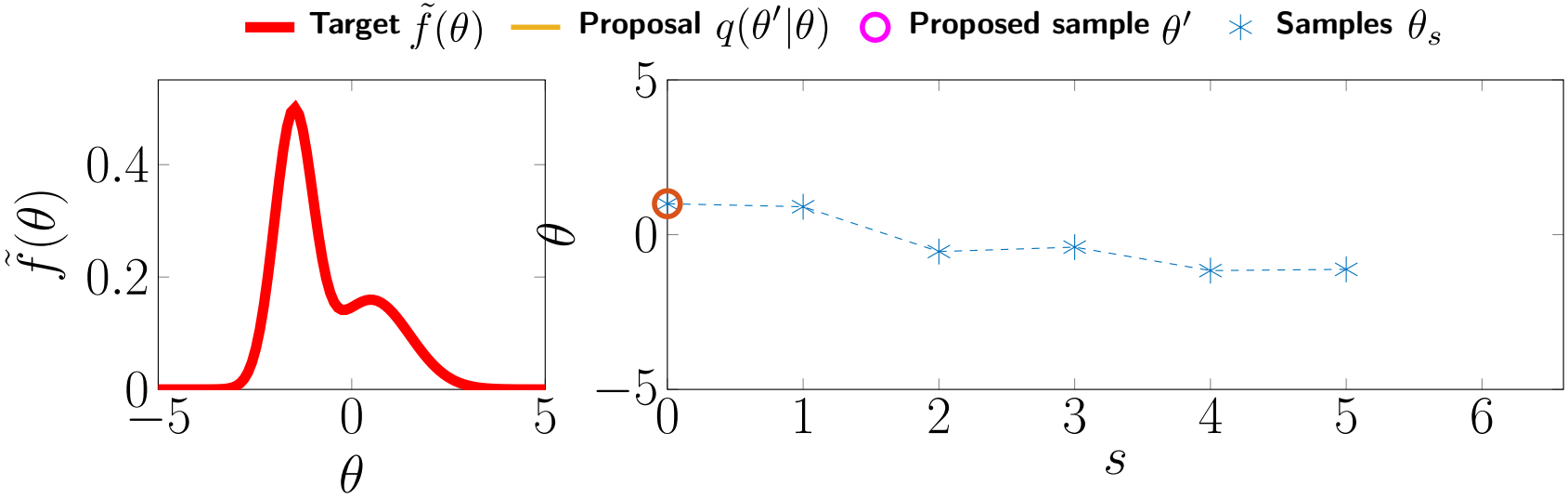
1D Example – Metropolis Algorithm [.m]



$$s = 4$$
$$\theta = \theta_s = \text{-}1.1613 \rightarrow \tilde{f}(\theta) = 0.42072$$
$$\theta' = \text{-}1.1205 \rightarrow \tilde{f}(\theta') = 0.40183$$
$$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)} = 0.95511$$
$$\alpha \leq 1 \rightarrow u = 0.79743$$
$$u < \alpha \rightarrow \theta_{s+1} = \theta'$$

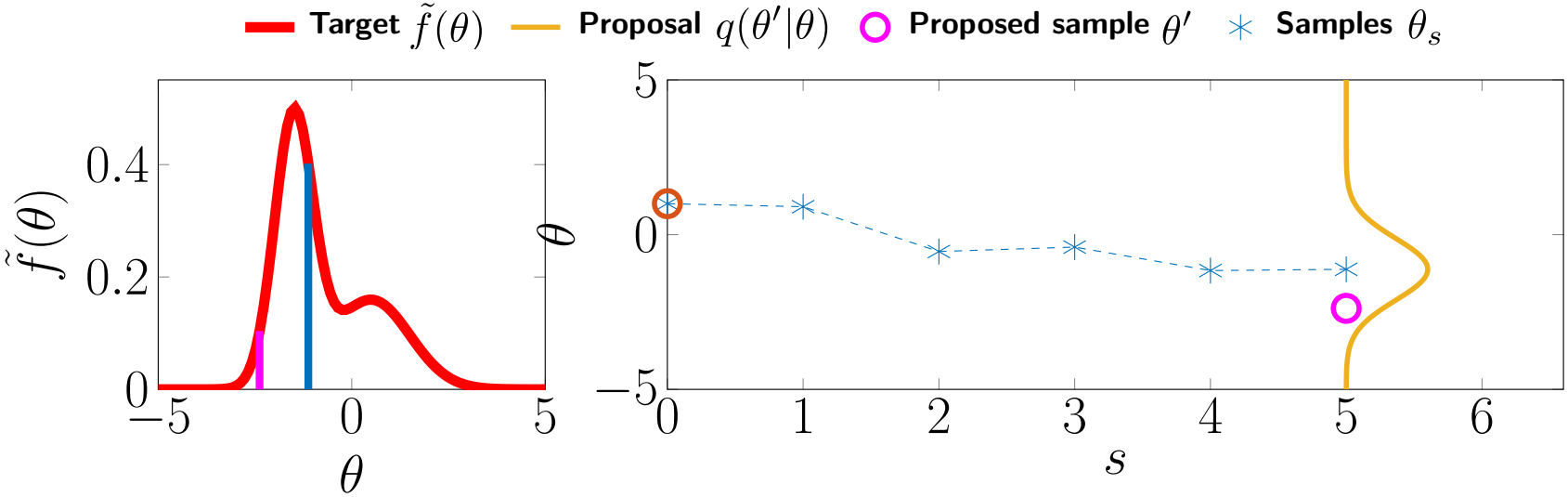
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



[CIVML/MCMC_M_demo_1D.m]

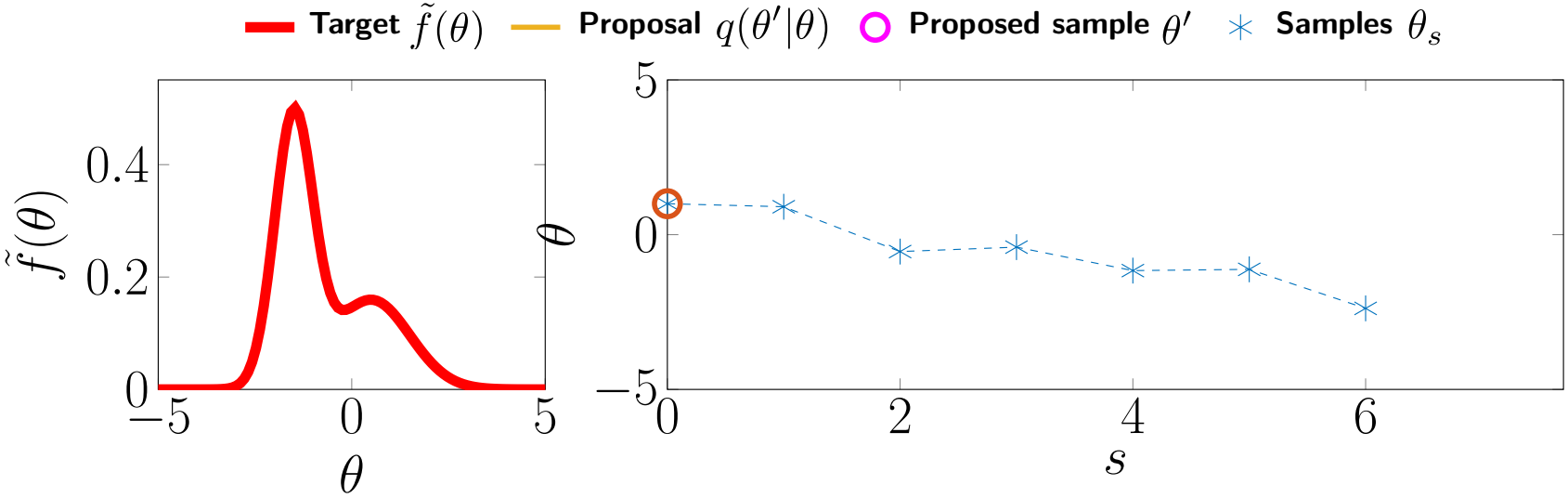
1D Example – Metropolis Algorithm [.m]



$$s = 5$$
$$\theta = \theta_s = \text{-}1.1205 \rightarrow \tilde{f}(\theta) = 0.40183$$
$$\theta' = \text{-}2.3813 \rightarrow \tilde{f}(\theta') = 0.10377$$
$$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)} = 0.25824$$
$$\alpha \leq 1 \rightarrow u = 0.2285$$
$$u < \alpha \rightarrow \theta_{s+1} = \theta'$$

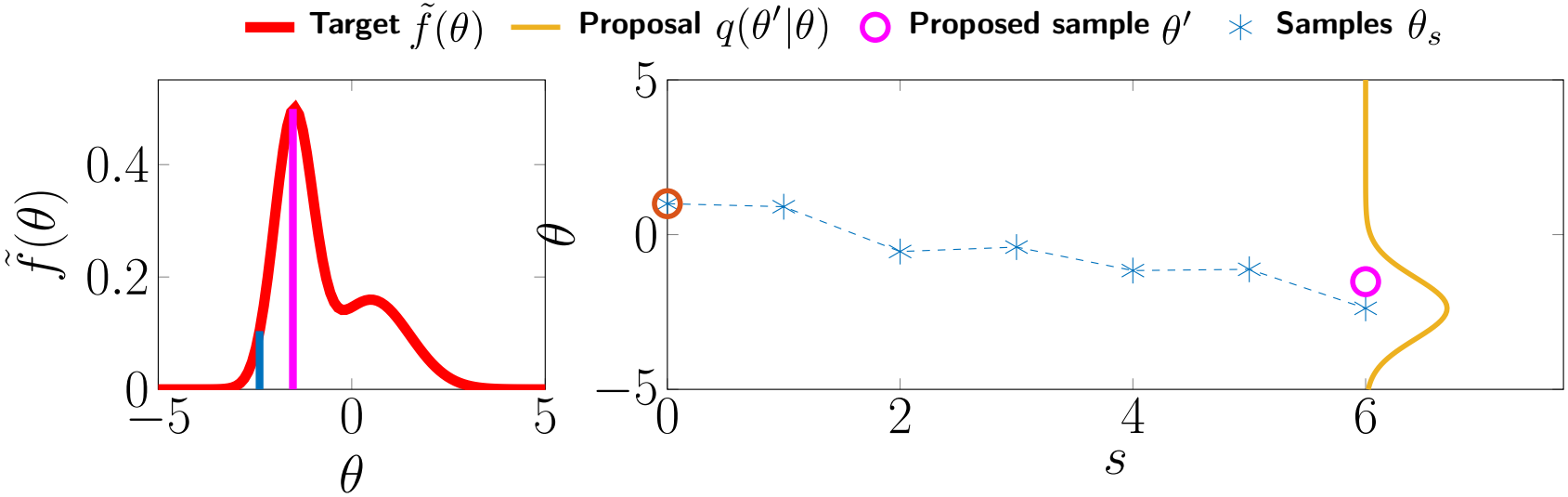
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



s =6

$\theta = \theta_s$ =-2.3813 $\rightarrow$ $\tilde{f}(\theta)$ =0.10377

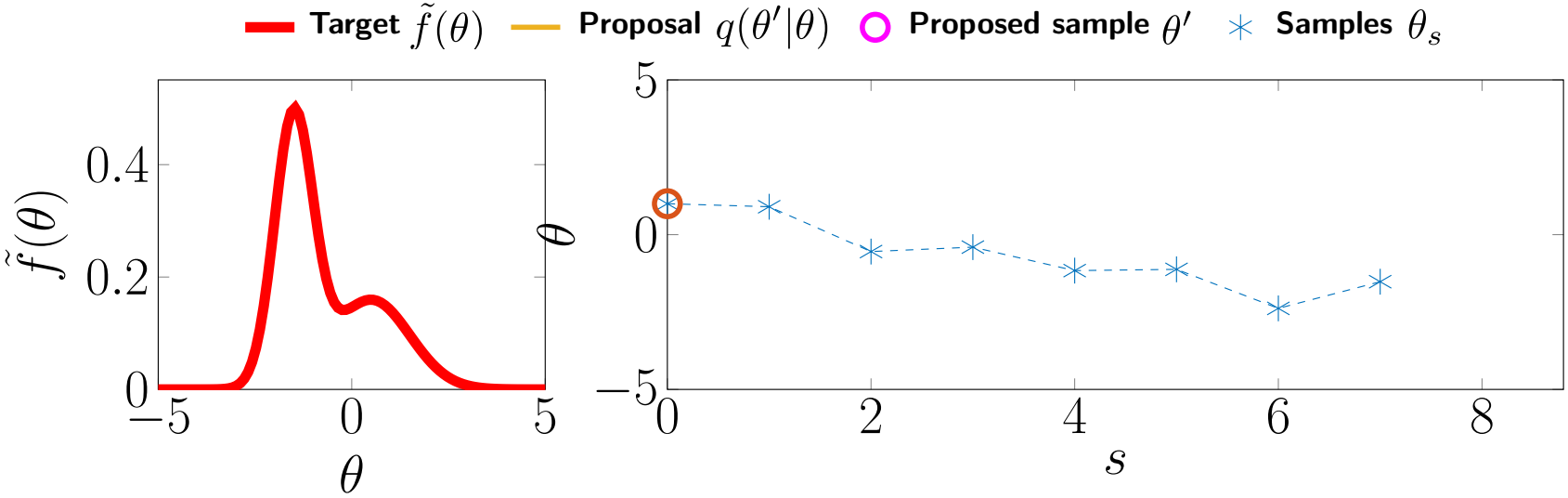
θ' =-1.5199 $\rightarrow$ $\tilde{f}(\theta')$ =0.4991

$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)}$ =4.8097

$\alpha \geq 1 \rightarrow \theta_{s+1} = \theta'$

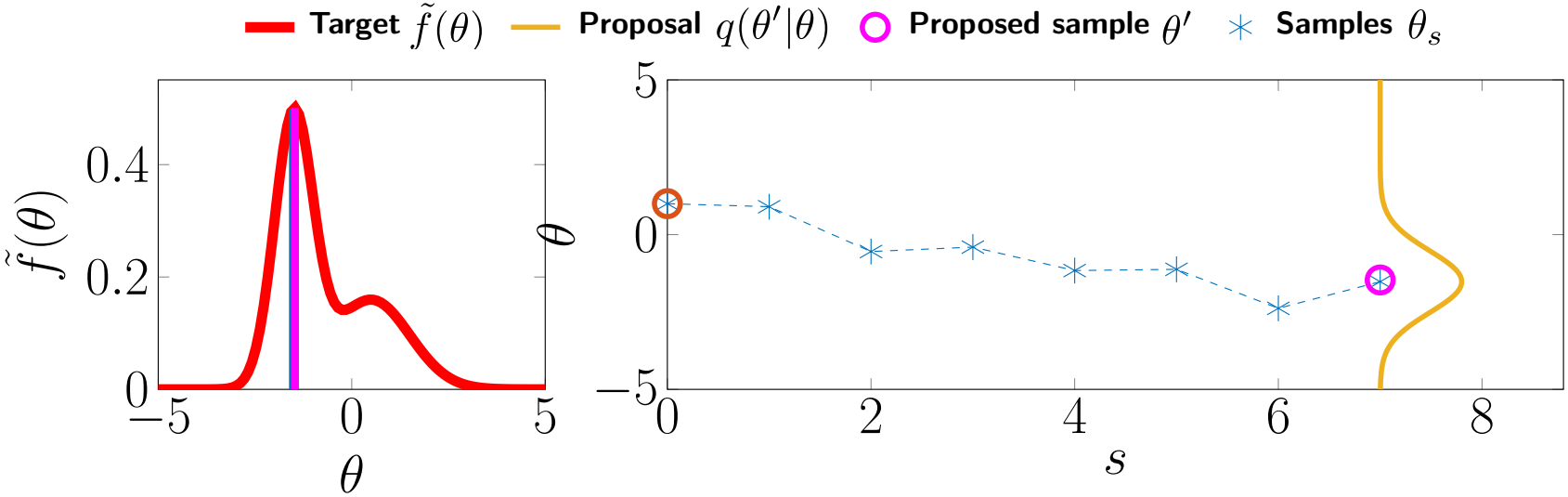
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



s =7

$\theta = \theta_s$ =-1.5199 $\rightarrow$ $\tilde{f}(\theta)$ =0.4991

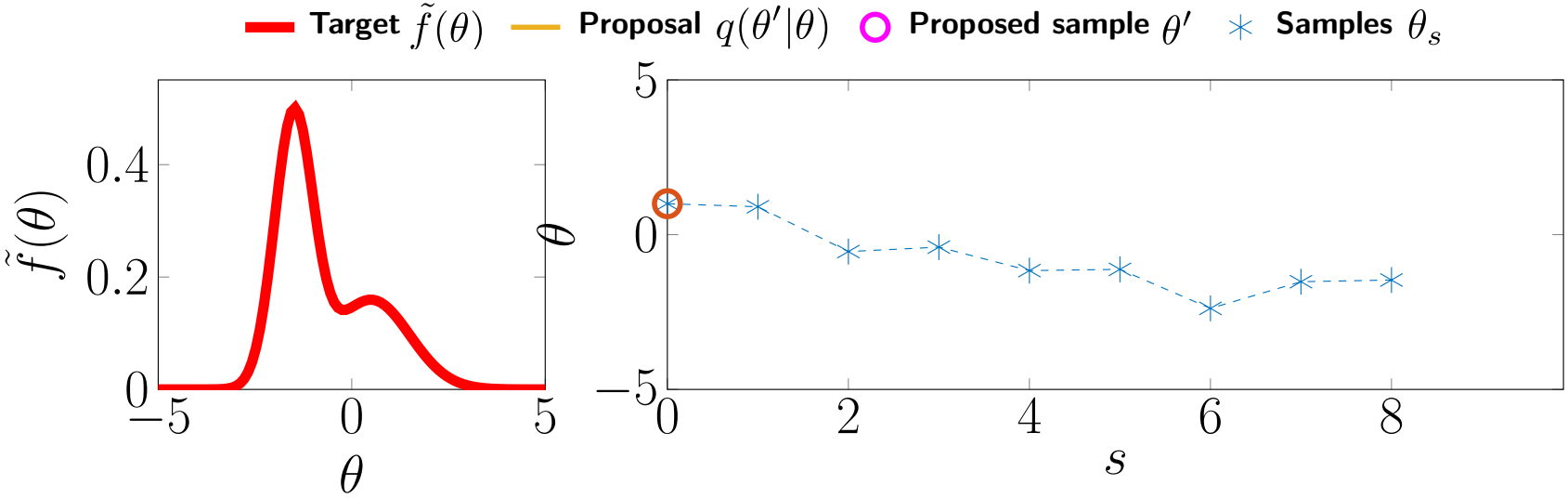
θ' =-1.4648 $\rightarrow$ $\tilde{f}(\theta')$ =0.5007

$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)}$ =1.0032

$\alpha \geq 1 \rightarrow \theta_{s+1} = \theta'$

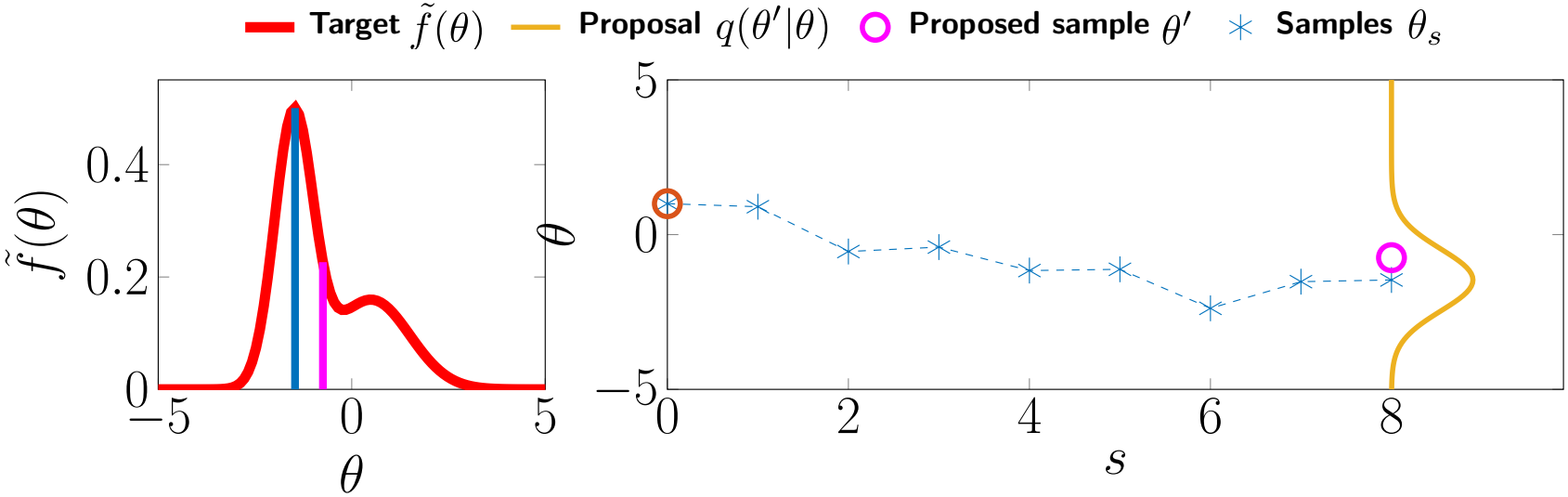
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



[CIVML/MCMC_M_demo_1D.m]

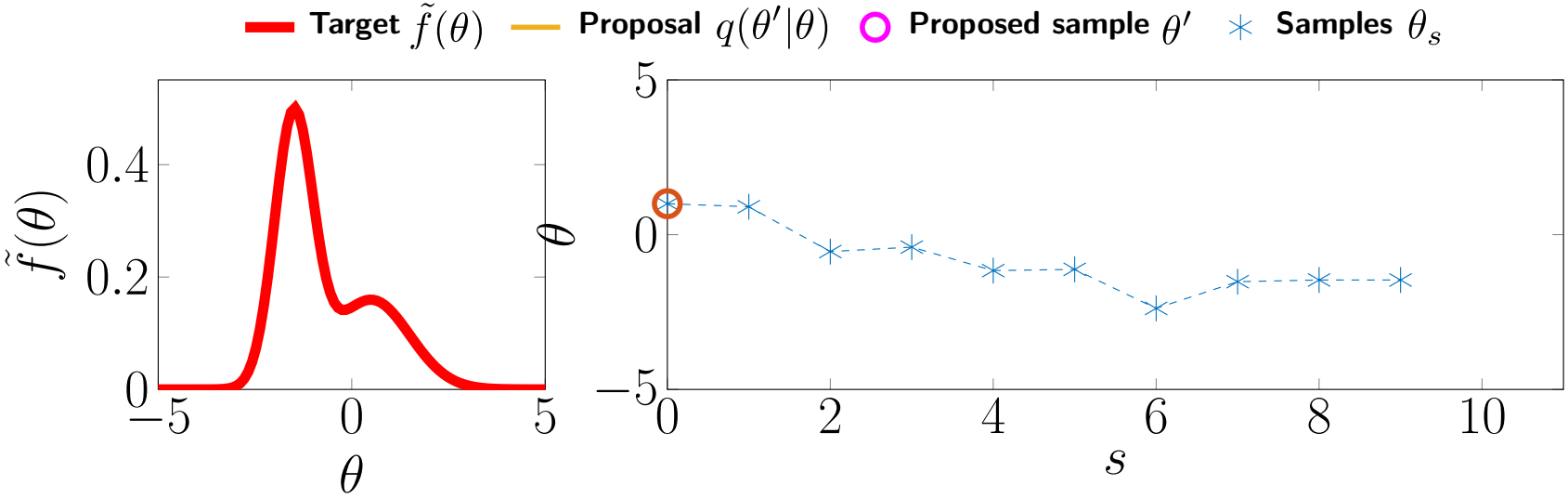
1D Example – Metropolis Algorithm [.m]



$$s = 8$$
$$\theta = \theta_s = \text{-}1.4648 \rightarrow \tilde{f}(\theta) = 0.5007$$
$$\theta' = \text{-}0.74433 \rightarrow \tilde{f}(\theta') = 0.22637$$
$$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)} = 0.4521$$
$$\alpha \leq 1 \rightarrow u = 0.77641$$
$$u \geq \alpha \rightarrow \theta_{s+1} = \theta_s$$

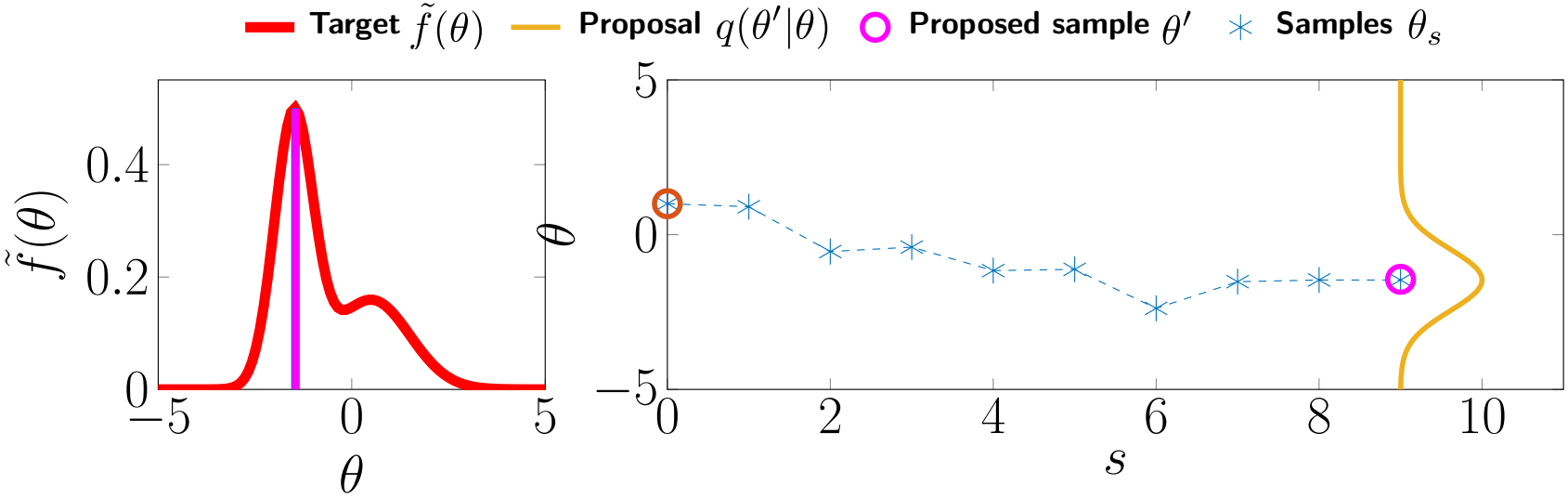
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



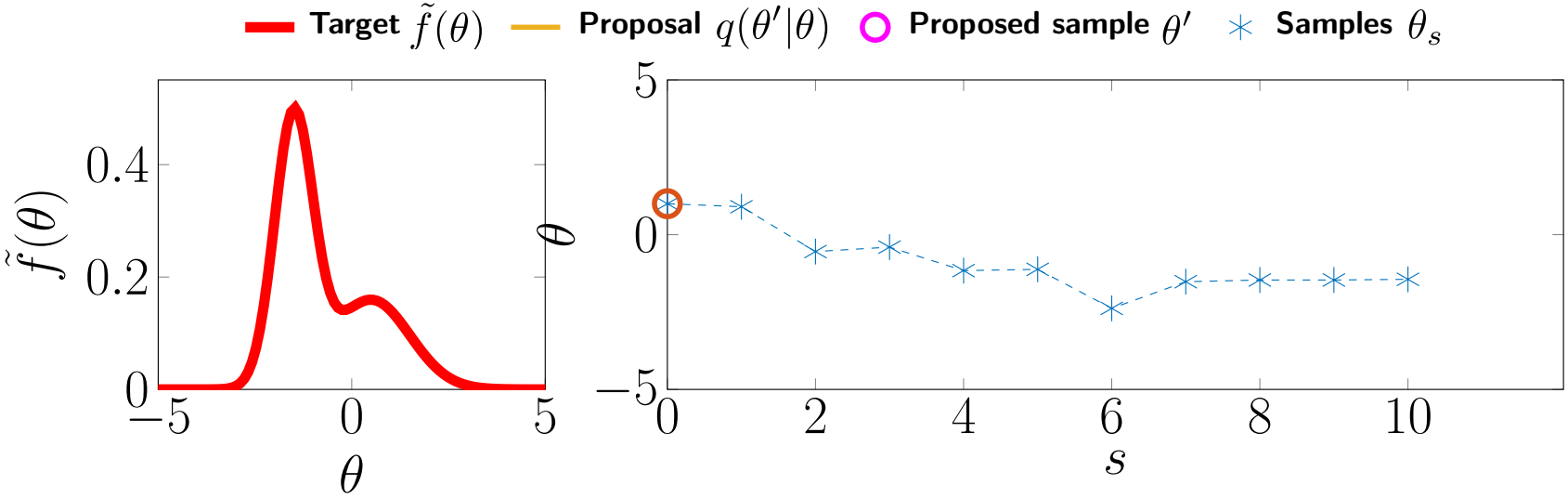
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



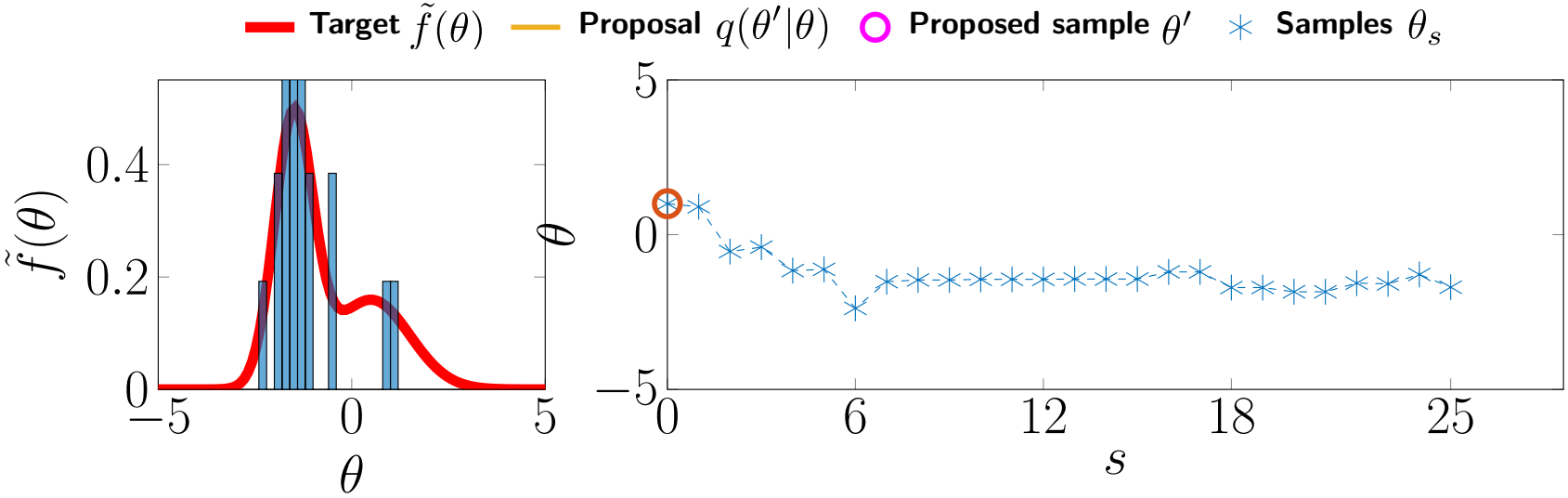
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



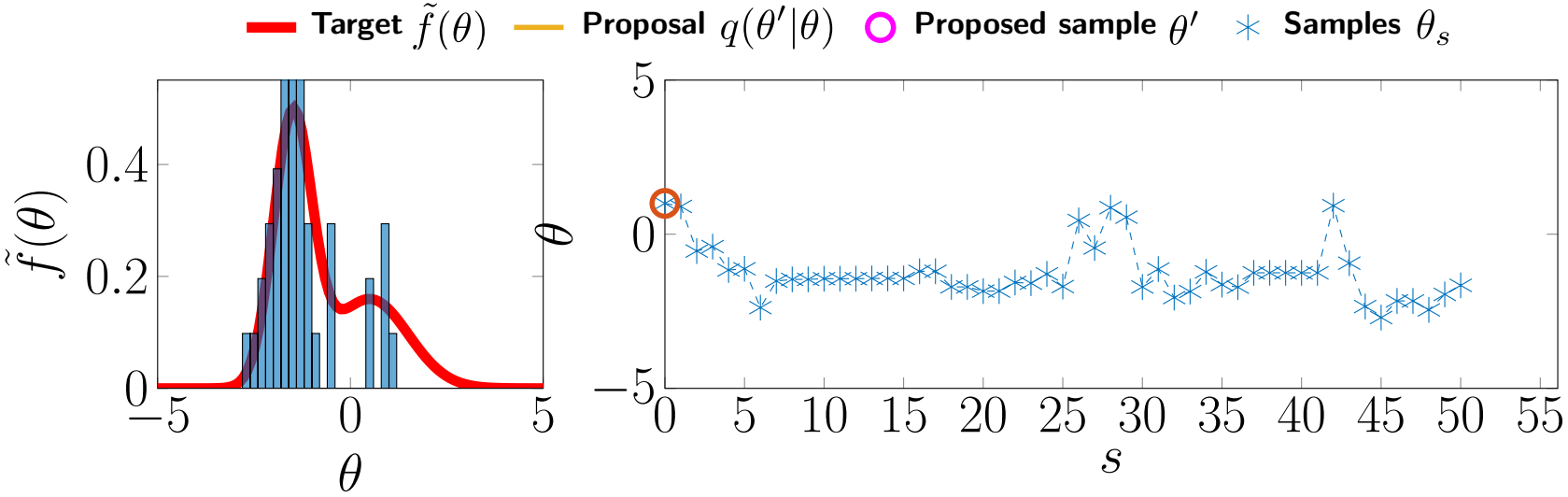
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



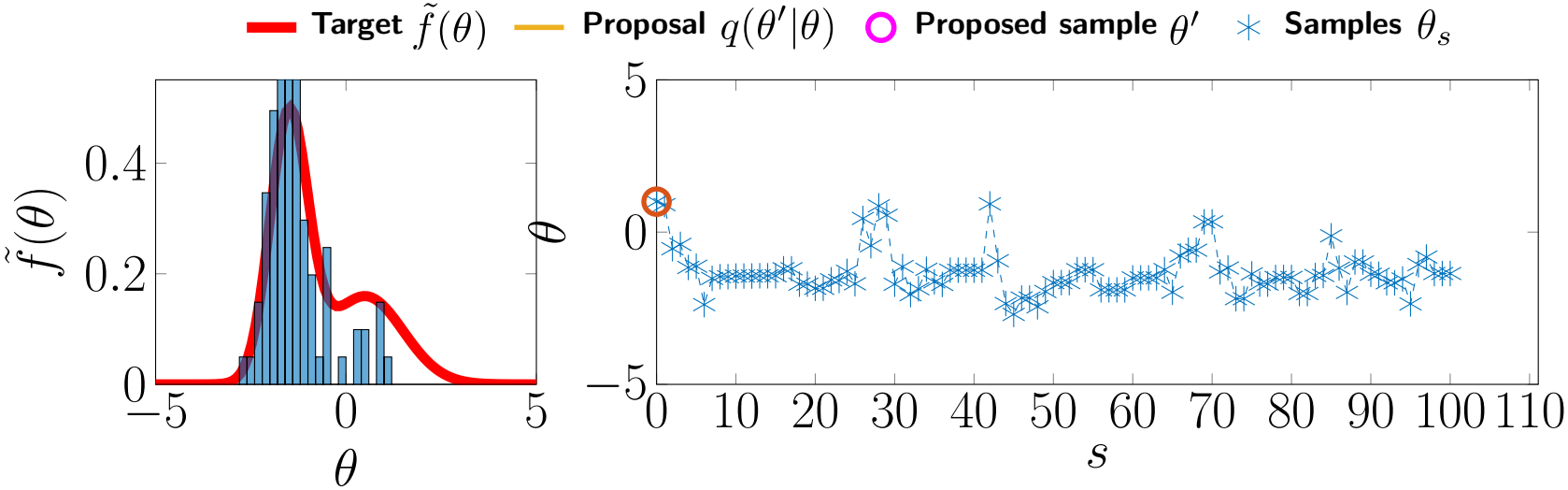
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



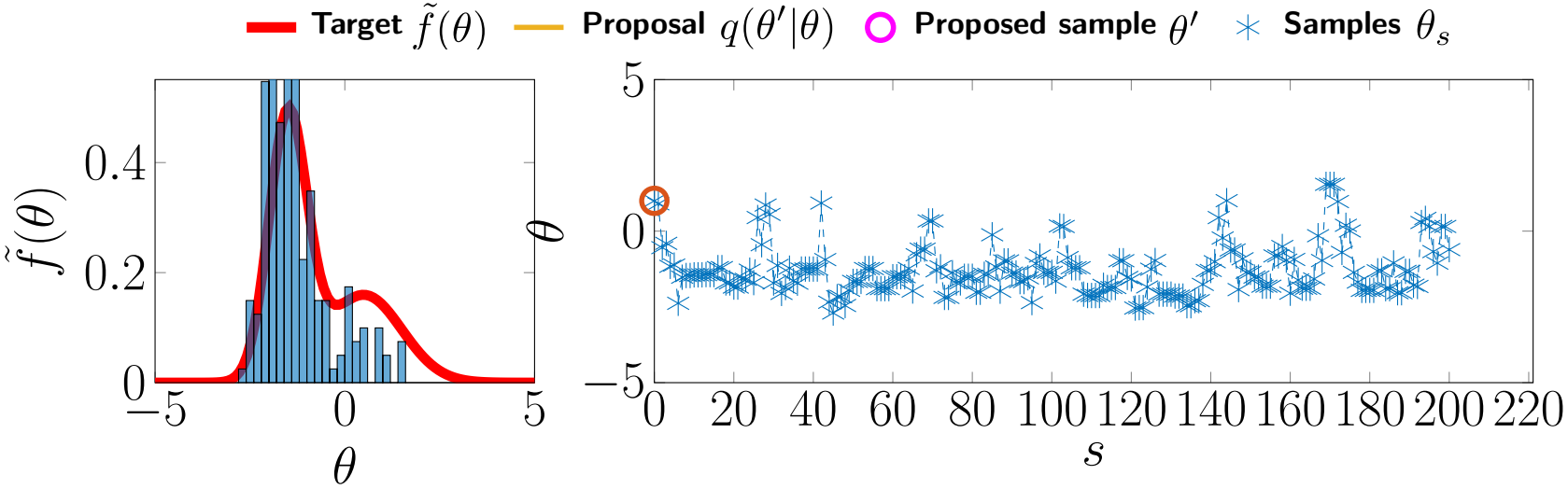
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



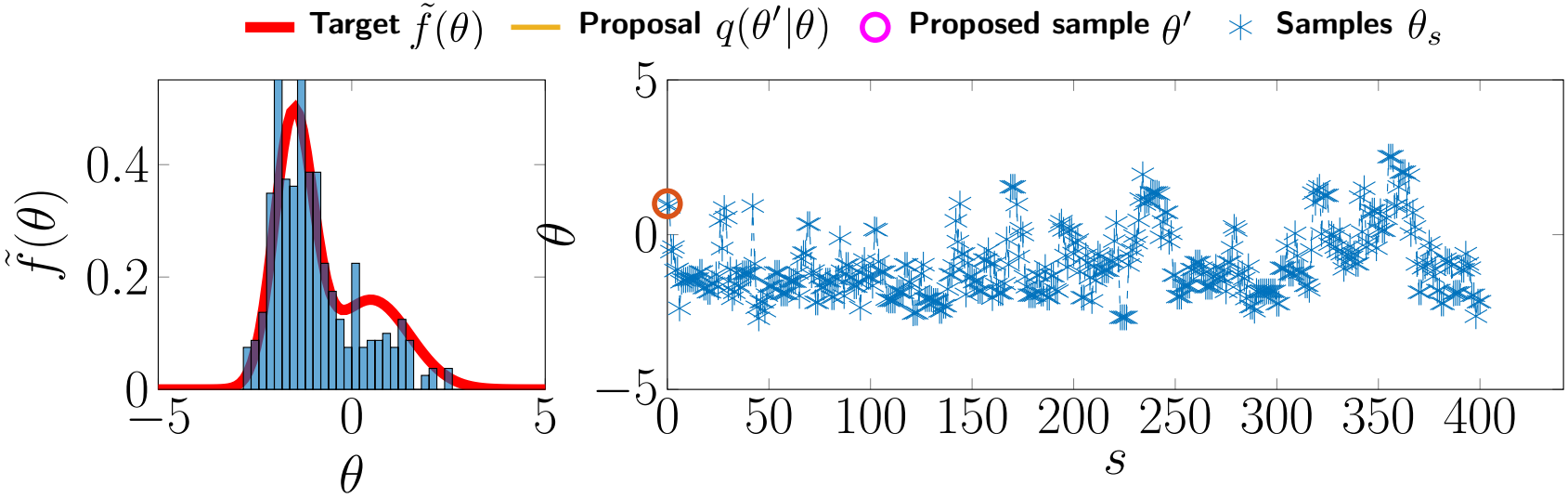
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



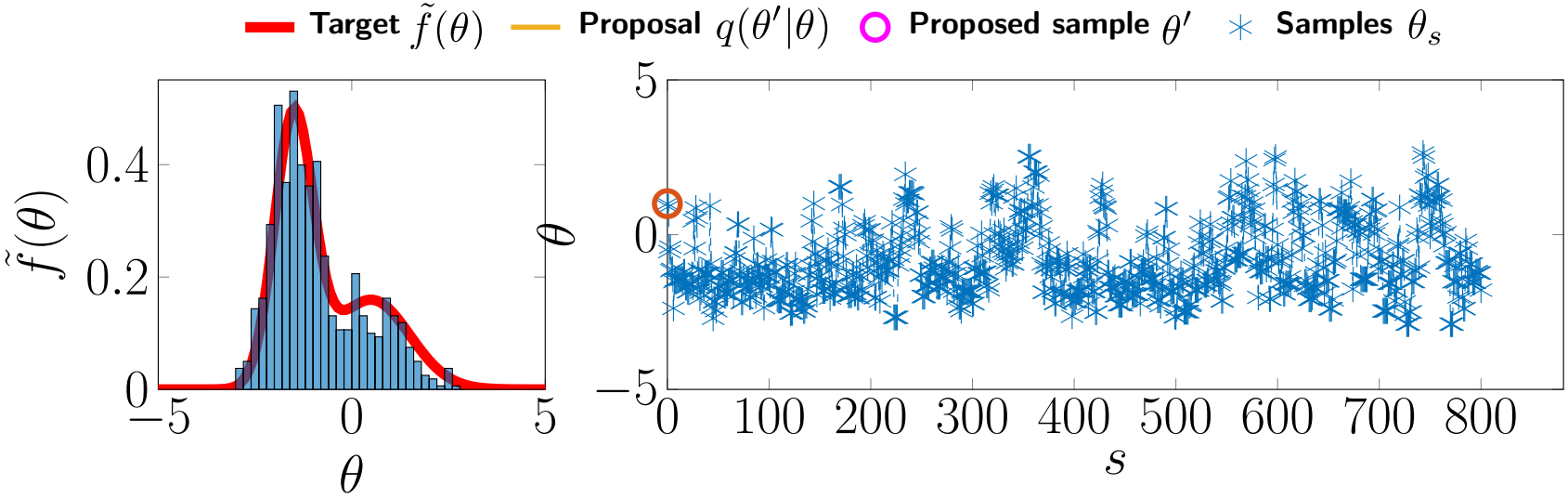
[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



[CIVML/MCMC_M_demo_1D.m]

1D Example – Metropolis Algorithm [.m]



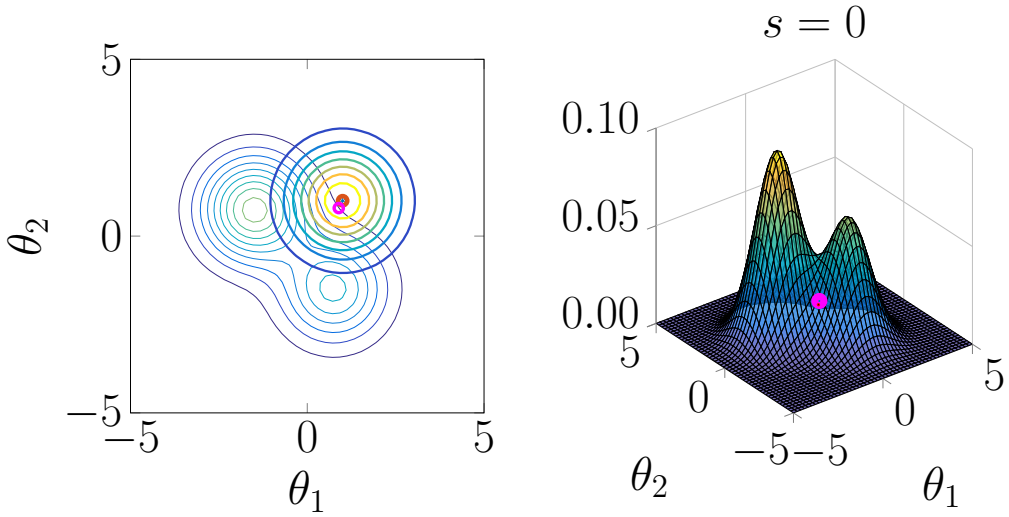
[CIVML/MCMC_M_demo_1D.m]

Algorithm 1: Metropolis sampling

define $\tilde{f}(\boldsymbol{\theta})$, $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$, S
initialize $\boldsymbol{\theta}_0$, $\mathcal{S} = \emptyset$
for $s = \{0, 1, 2, \cdots, \mathrm{S}-1\}$ **do**
    define $\boldsymbol{\theta} = \boldsymbol{\theta}_s$
    sample $\boldsymbol{\theta}' \sim q(\boldsymbol{\theta}'|\boldsymbol{\theta})$
    compute $\alpha = \frac{\tilde{f}(\boldsymbol{\theta}')}{\tilde{f}(\boldsymbol{\theta})}$
    compute $r = \min(1, \alpha)$
    sample $u \sim \mathcal{U}(0, 1)$
    if $u \leq r$ **then**
        $\boldsymbol{\theta}_{s+1} = \boldsymbol{\theta}'$
    else
        $\boldsymbol{\theta}_{s+1} = \boldsymbol{\theta}_s$
    $\mathcal{S} \leftarrow \{\mathcal{S} \cup \{\boldsymbol{\theta}_{s+1}\}\}$

2D Example – Metropolis Algorithm [.m]

Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s

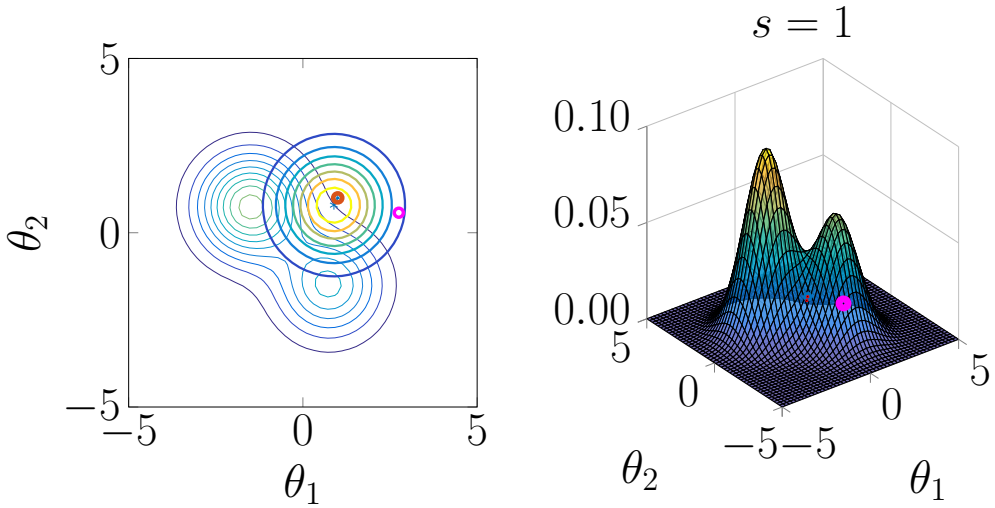


$$s = 0$$
$$\theta = \theta_s = [1,1] \rightarrow \tilde{f}(\theta) = 0.0067776$$
$$\theta' = [0.89041, 0.79237] \rightarrow \tilde{f}(\theta') = 0.010035$$
$$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)} = 1.4806$$
$$\alpha \geq 1 \rightarrow \theta_{s+1} = \theta'$$

[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



$s = 1$

$\theta = \theta_s = [0.89041, 0.79237] \rightarrow \tilde{f}(\theta) = 0.010035$

$\theta' = [2.7523, 0.57414] \rightarrow \tilde{f}(\theta') = 0.0010091$

$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)} = 0.10056$

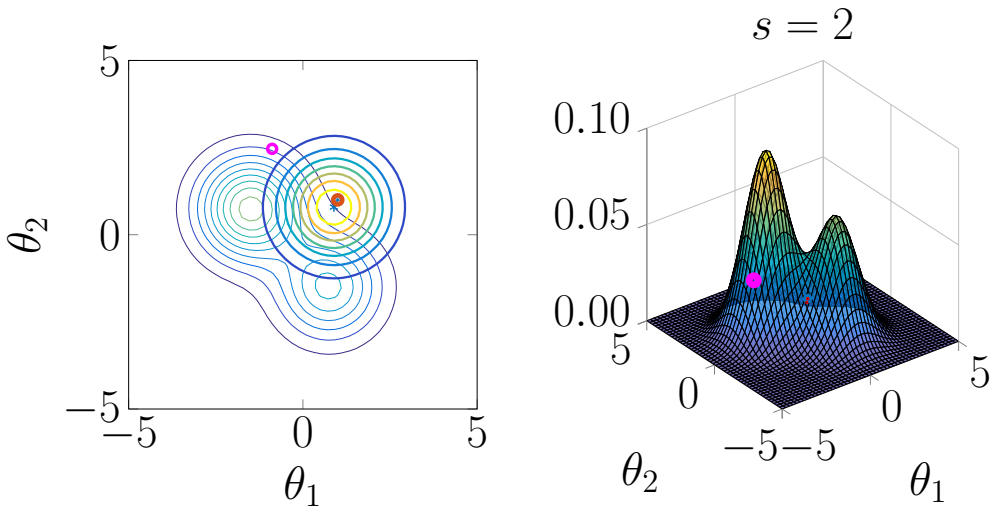
$\alpha \leq 1 \rightarrow u = 0.93366$

$u \geq \alpha \rightarrow \theta_{s+1} = \theta_s$

[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



s =2

$\theta = \theta_s$ = [0.89041,0.79237] $\rightarrow \tilde{f}(\theta)$ =0.010035

θ' = [-0.88274,2.4645] $\rightarrow \tilde{f}(\theta')$ =0.018159

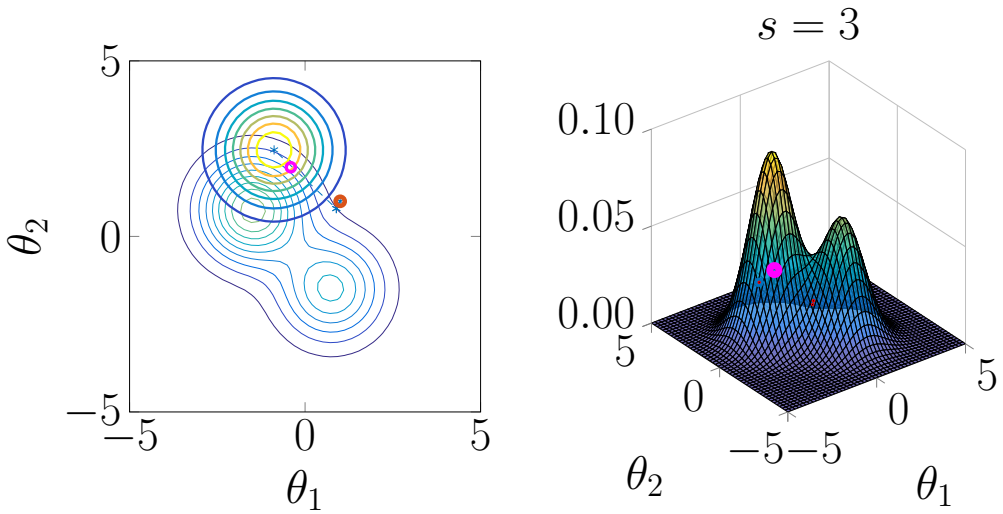
$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)}$ =1.8095

$\alpha \geq 1 \rightarrow \theta_{s+1} = \theta'$

[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



s =3

$\theta = \theta_s$ = [-0.88274,2.4645] $\rightarrow \tilde{f}(\theta)$ =0.018159

θ' = [-0.40663,1.9694] $\rightarrow \tilde{f}(\theta')$ =0.025055

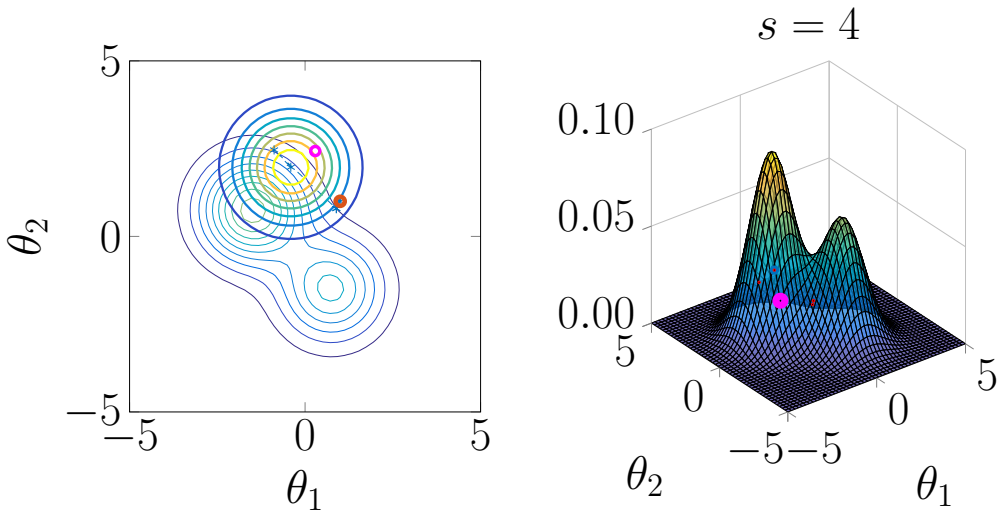
$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)}$ =1.3798

$\alpha \geq 1 \rightarrow \theta_{s+1} = \theta'$

[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



s =4

$\theta = \theta_s$ = [-0.40663,1.9694] $\rightarrow \tilde{f}(\theta)$ =0.025055

θ' = [0.28938,2.4318] $\rightarrow \tilde{f}(\theta')$ =0.0047078

$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)}$ =0.1879

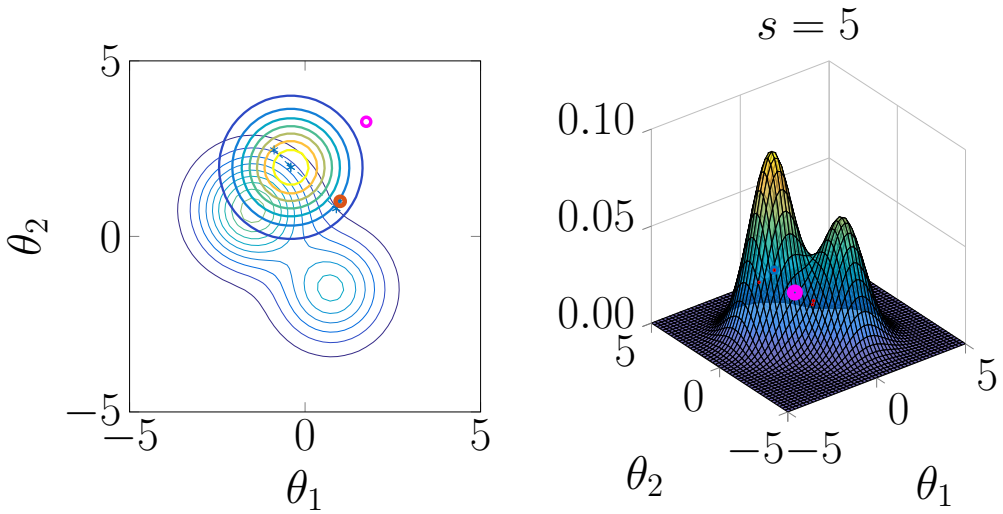
$\alpha \leq 1 \rightarrow u$ =0.87044

$u \geq \alpha \rightarrow \theta_{s+1} = \theta_s$

[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



s =5

$\theta = \theta_s$ = [-0.40663,1.9694] $\rightarrow \tilde{f}(\theta)$ =0.025055

θ' = [1.7418,3.2659] $\rightarrow \tilde{f}(\theta')$ =2.1513e-05

$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)}$ =0.00085862

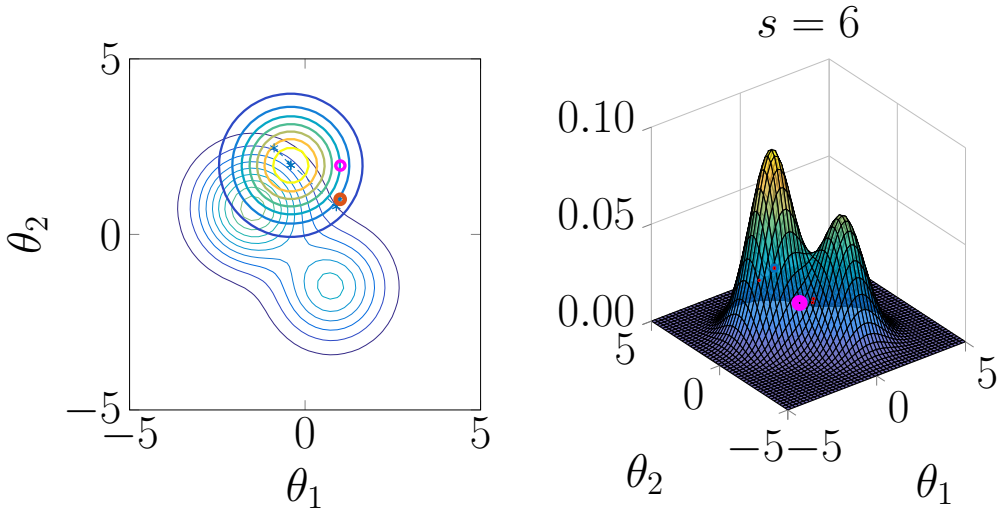
$\alpha \leq 1 \rightarrow u$ =0.043

$u \geq \alpha \rightarrow \theta_{s+1} = \theta_s$

[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



s =6

$\theta = \theta_s$ = [-0.40663,1.9694] $\rightarrow \tilde{f}(\theta)$ =0.025055

θ' = [1.0054,1.962] $\rightarrow \tilde{f}(\theta')$ =0.0021399

$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)}$ =0.085408

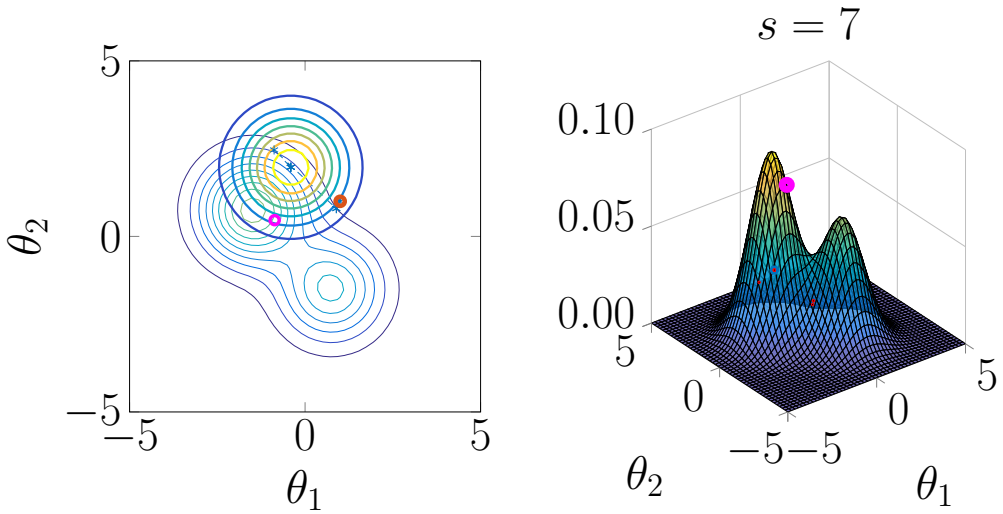
$\alpha \leq 1 \rightarrow u$ =0.75252

$u \geq \alpha \rightarrow \theta_{s+1} = \theta_s$

[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



s =7

$\theta = \theta_s$ = [-0.40663,1.9694] $\rightarrow \tilde{f}(\theta)$ =0.025055

θ' = [-0.86687,0.46298] $\rightarrow \tilde{f}(\theta')$ =0.077505

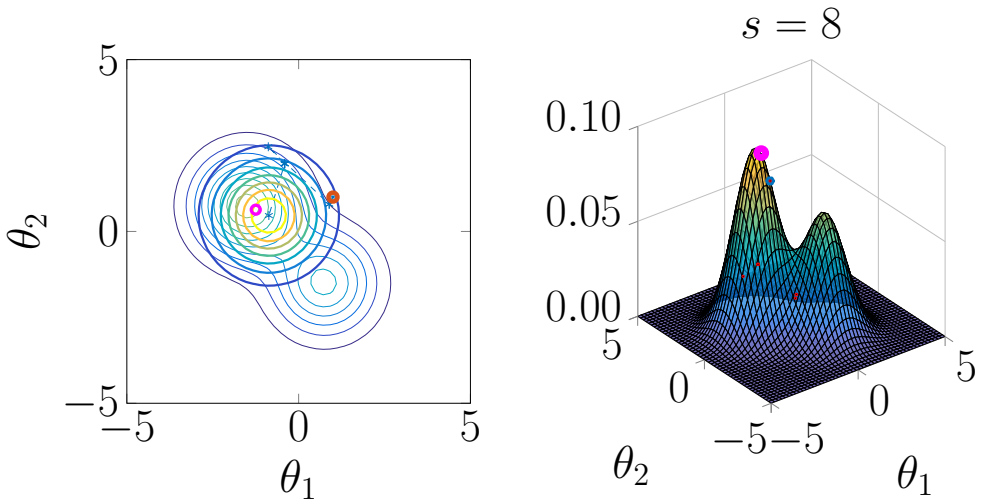
$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)}$ =3.0934

$\alpha \geq 1 \rightarrow \theta_{s+1} = \theta'$

[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



s =8

$\theta = \theta_s$ = [-0.86687,0.46298] $\rightarrow \tilde{f}(\theta)$ =0.077505

θ' = [-1.2473,0.63277] $\rightarrow \tilde{f}(\theta')$ =0.092749

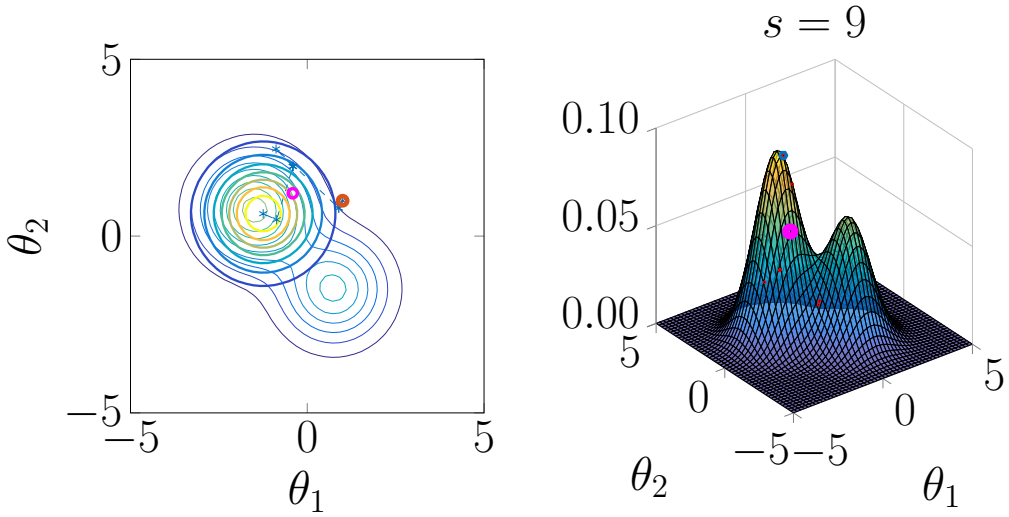
$\alpha = \frac{\tilde{f}(\theta')}{\tilde{f}(\theta)}$ =1.1967

$\alpha \geq 1 \rightarrow \theta_{s+1} = \theta'$

[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

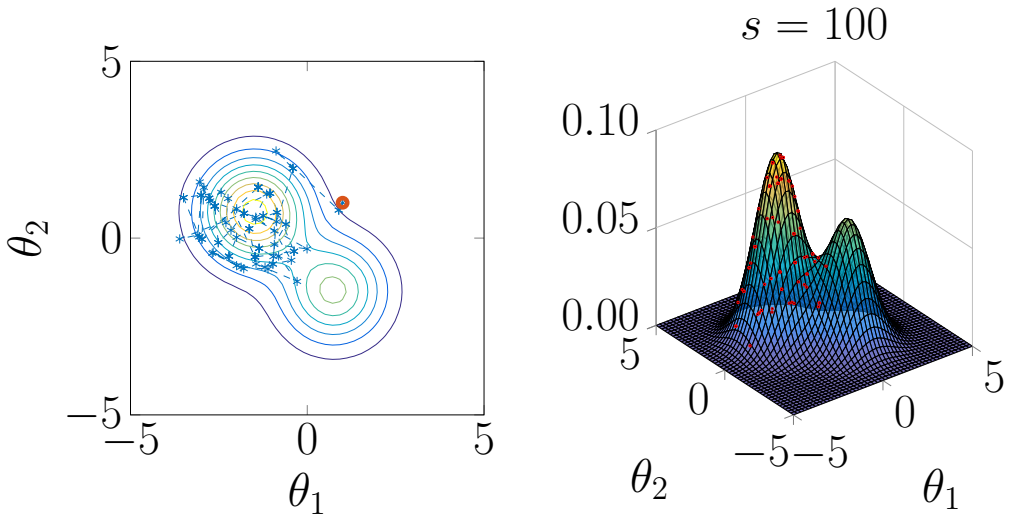
Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

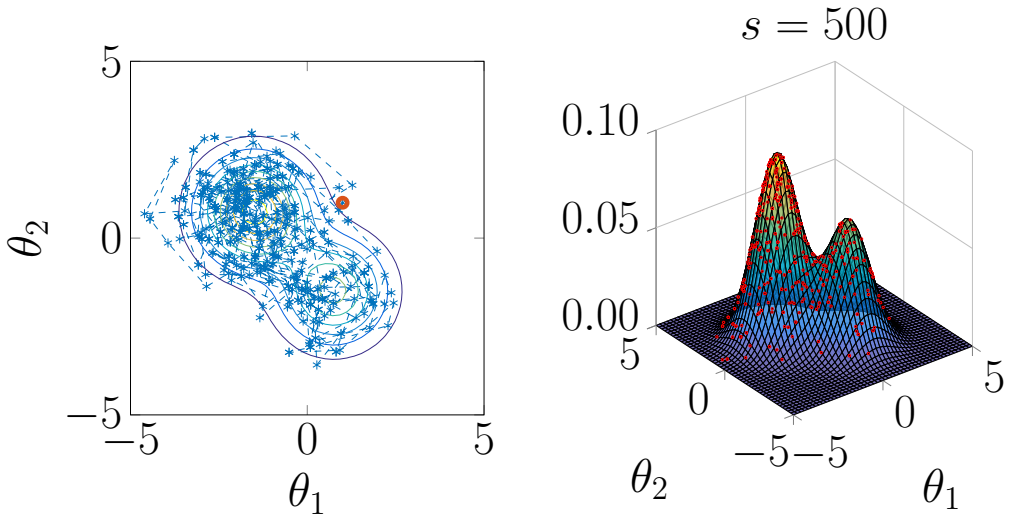
Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

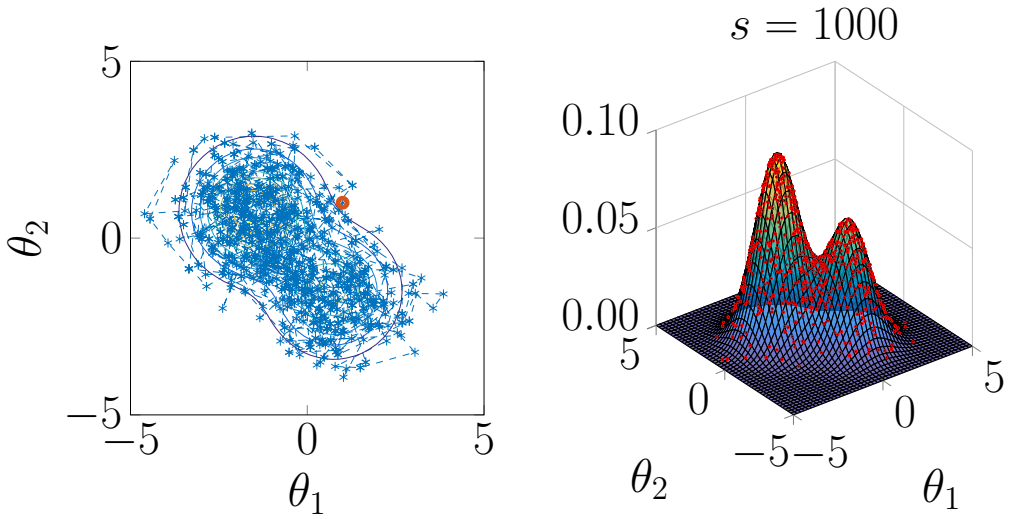
Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

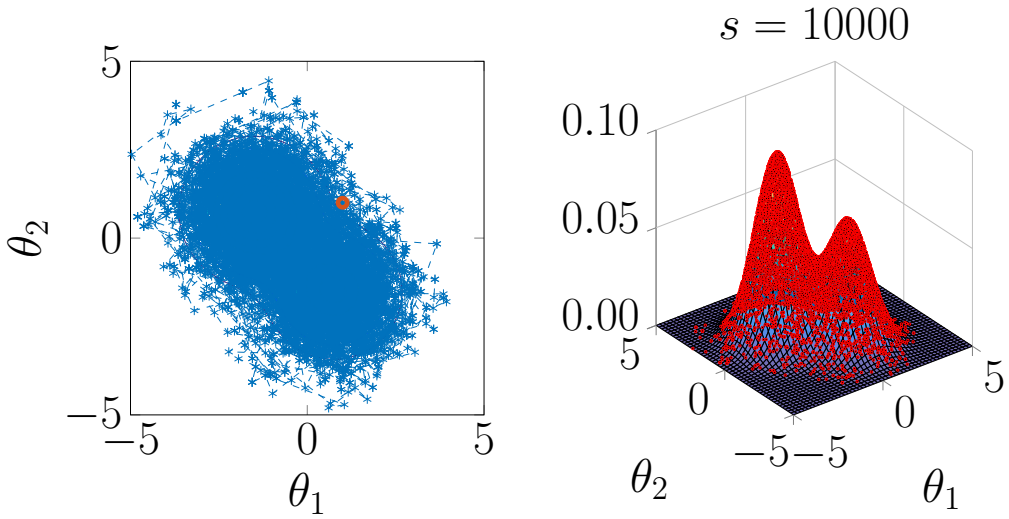
Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



[CIVML/MCMC_M_demo_2D.m]

2D Example – Metropolis Algorithm [.m]

Target $\tilde{f}(\theta)$ **Proposal** $q(\theta'|\theta)$ **Proposed sample** θ' **Samples** θ_s



[CIVML/MCMC_M_demo_2D.m]

Computing with samples – normalization cte.

$$\overbrace{f(\boldsymbol{\theta}|\mathcal{D})}^{\text{Posterior PDF}} = \frac{\overbrace{f(\mathcal{D}|\boldsymbol{\theta})}^{\text{Likelihood}} \cdot \overbrace{f(\boldsymbol{\theta})}^{\text{Prior PDF}}}{\underbrace{f(\mathcal{D})}_{\text{Normalization cte}}}$$

Normalization constant

$$\begin{aligned} f(\mathcal{D}) &= \int_{\boldsymbol{\theta}} f(\mathcal{D}|\boldsymbol{\theta}) f(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ &\approx \left[\frac{1}{\mathrm{S}} \sum_{s=1}^{\mathrm{S}} \frac{1}{f(\mathcal{D}|\boldsymbol{\theta}_s)} \right]^{-1} \end{aligned}$$

⚠ Poor performance in practice

Computing with samples – Posterior PDF

$$\overbrace{f(\boldsymbol{\theta}|\mathcal{D})}^{\text{Posterior PDF}} = \frac{\overbrace{f(\mathcal{D}|\boldsymbol{\theta})}^{\text{Likelihood}} \cdot \overbrace{f(\boldsymbol{\theta})}^{\text{Prior PDF}}}{\underbrace{f(\mathcal{D})}_{\text{Normalization cte}}}$$

Posterior PDF mean

$$\begin{aligned}\mathbb{E}[\boldsymbol{\theta}|\mathcal{D}] &= \int_{\boldsymbol{\theta}} \boldsymbol{\theta} \cdot f(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta} \\ &\approx \frac{1}{\mathrm{S}} \sum_{s=1}^{\mathrm{S}} \boldsymbol{\theta}_s\end{aligned}$$

Posterior PDF covariance

$$\begin{aligned}\mathrm{Cov}[\theta_i, \theta_j|\mathcal{D}] &= \mathbb{E}[(\theta_i - \mathbb{E}[\theta_i|\mathcal{D}])(\theta_j - \mathbb{E}[\theta_j|\mathcal{D}])] \\ &\approx \frac{1}{\mathrm{S}-1} \sum_{s=1}^{\mathrm{S}} (\theta_{i,s} - \mathbb{E}[\theta_i|\mathcal{D}])(\theta_{j,s} - \mathbb{E}[\theta_j|\mathcal{D}])\end{aligned}$$

Computing with samples – Predictive PDF

$$\overbrace{f(\boldsymbol{\theta}|\mathcal{D})}^{\text{Posterior PDF}} = \frac{\overbrace{f(\mathcal{D}|\boldsymbol{\theta})}^{\text{Likelihood}} \cdot \overbrace{f(\boldsymbol{\theta})}^{\text{Prior PDF}}}{\underbrace{f(\mathcal{D})}_{\text{Normalization cte}}}$$

Predictive PDF

$$X|\mathcal{D} \sim f(x|\mathcal{D}) = \int_{\boldsymbol{\theta}} f_X(x;\boldsymbol{\theta}) \overbrace{f(\boldsymbol{\theta}|\mathcal{D})}^{\boldsymbol{\theta}_s} d\boldsymbol{\theta}$$

$$\underbrace{x_s : \; X|\boldsymbol{\theta}_s \sim f_X(x;\boldsymbol{\theta}_s)}_{\text{sample from } X \text{ using MCMC samples } \boldsymbol{\theta}_s}$$

Predictive PDF mean

$$\mathbb{E}[X|\mathcal{D}] = \int_x x \cdot f(x|\mathcal{D}) dx \approx \frac{1}{S}\sum_{s=1}^{S} x_s$$

Predictive PDF variance

$$\text{Var}[X|\mathcal{D}] = \mathbb{E}[(X - \mathbb{E}[X|\mathcal{D}])^2] \approx \frac{1}{S-1}\sum_{s=1}^{S}(x_s - \mathbb{E}[X|\mathcal{D}])^2$$

Computing with samples – Predictive PDF (function)

Function

$$z = h(\boldsymbol{\theta})$$

Predictive PDF

$$\underbrace{z_s : \; Z|\boldsymbol{\theta}_s = h(\boldsymbol{\theta}_s)}_{\text{sample from } Z \text{ using MCMC samples } \boldsymbol{\theta}_s}$$

Predictive PDF mean

$$\begin{aligned} \mathbb{E}[Z|\mathcal{D}] &= \int_x z \cdot f(z|\mathcal{D}) dx \\ &\approx \frac{1}{S} \sum_{s=1}^{S} z_s \end{aligned}$$

Predictive PDF variance

$$\begin{aligned} \text{Var}[Z|\mathcal{D}] &= \mathbb{E}[(Z - \mathbb{E}[Z|\mathcal{D}])^2] \\ &\approx \frac{1}{S-1} \sum_{s=1}^{S} (z_s - \mathbb{E}[Z|\mathcal{D}])^2 \end{aligned}$$

– Computing with samples

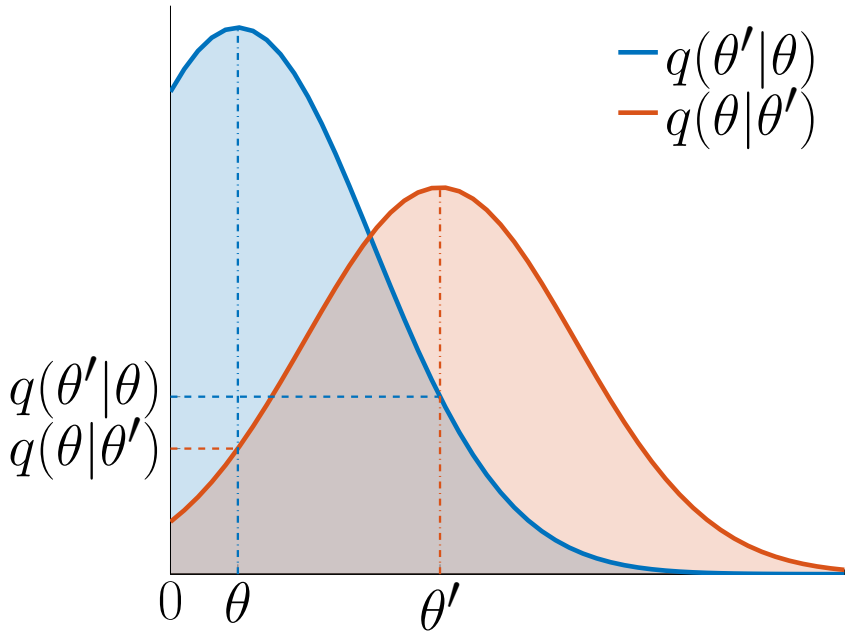
```
import numpy as np

ts   #matrix of S samples times D parameters
fts  #vector of target pdf values -> ts
p_tm = np.mean(ts,0) #posterior mean
p_tcov = np.cov(ts)  #posterior covariance

#  sample xs from ts
xs = np.random.normal(loc = ts[:,0], scale = ts[:,1])
p_xm = np.mean(xs,0) #posterior predictive mean
p_xvar = np.var(xs)  #posterior predictive variance

#  fct z=sin(x)
zs = np.sin(xs)         #function sample
p_zm = np.mean(zs, 0) #posterior predictive mean
p_zvar = np.var(zs)   #posterior predictive variance
```

Metropolis-Hastings

The **Metropolis-Hastings** algorithm is identical to the Metropolis algorithm except that it allows for

non-symmetric transition probabilities

$$q(\boldsymbol{\theta}'|\boldsymbol{\theta}) \neq q(\boldsymbol{\theta}|\boldsymbol{\theta}')$$



When $q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\theta}')$: **Metropolis-Hastings $\equiv$ Metropolis**

Algorithm 2: Metropolis-Hastings sampling

define $\tilde{f}(\boldsymbol{\theta})$, $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$, S
initialize $\boldsymbol{\theta}_0$, $\mathcal{S} = \emptyset$
for $s = \{0, 1, 2, \cdots, \mathrm{S}-1\}$ **do**
    define $\boldsymbol{\theta} = \boldsymbol{\theta}_s$
    sample $\boldsymbol{\theta}' \sim q(\boldsymbol{\theta}'|\boldsymbol{\theta})$
    compute $\alpha = \frac{\tilde{f}(\boldsymbol{\theta}')}{\tilde{f}(\boldsymbol{\theta})} \cdot \frac{q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta})}$
    compute $r = \min(1, \alpha)$
    sample $u \sim \mathcal{U}(0, 1)$
    if $u \leq r$ **then**
        $\boldsymbol{\theta}_{s+1} = \boldsymbol{\theta}'$
    else
        $\boldsymbol{\theta}_{s+1} = \boldsymbol{\theta}_s$
    $\mathcal{S} \leftarrow \{\mathcal{S} \cup \{\boldsymbol{\theta}_{s+1}\}\}$

Section Outline

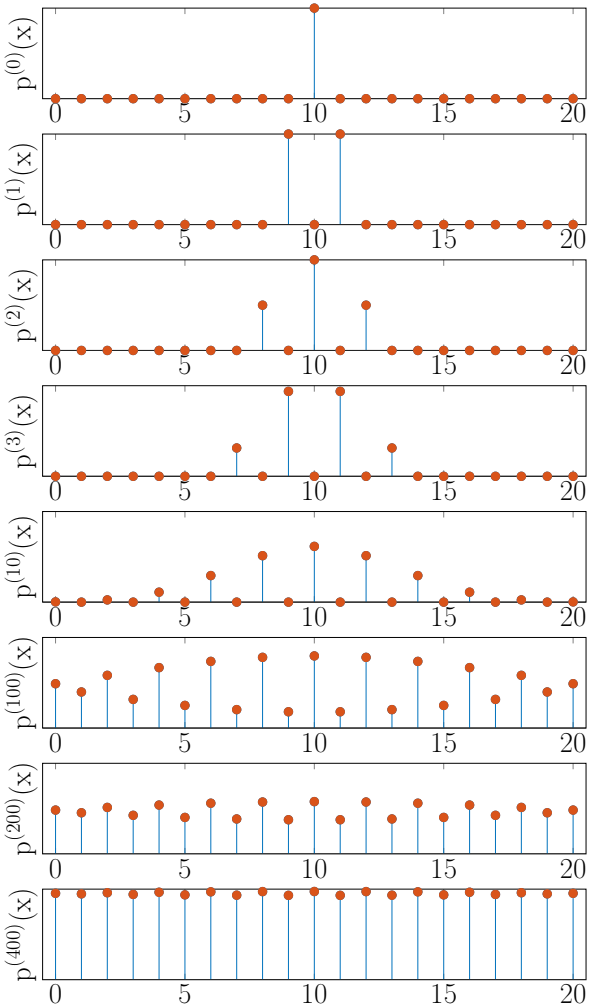
Convergence

3.1 Burn-in phase

3.2 Monitoring convergence

3.3 Example – Convergence

Initial state θ_0 and burn-in phase

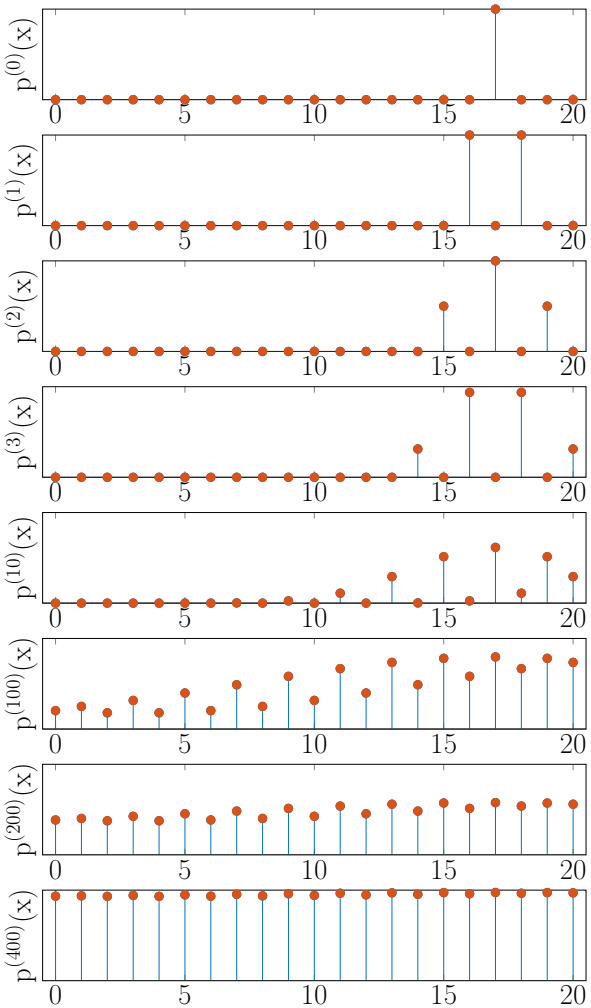


Samples come from the **stationary distribution** when the chain has **forgotten where it started from**

- Samples taken before reaching the **stationary distribution** must be discarded $\rightarrow$ **burn-in phase**
- In practice it is safe to **discard the first half of each chain**

[Murphy, 2012 §24.4.1]

Initial state θ_0 and burn-in phase

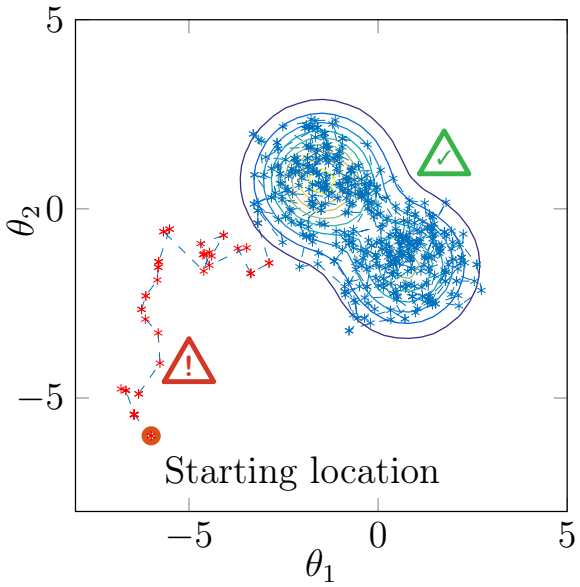


Samples come from the **stationary distribution** when the chain has **forgotten where it started from**

- Samples taken before reaching the **stationary distribution** must be discarded $\rightarrow$ **burn-in phase**
- In practice it is safe to **discard the first half of each chain**

[Murphy, 2012 §24.4.1]

Initial state θ_0 and burn-in phase



Samples come from the **stationary distribution** when the chain has **forgotten where it started from**

- Samples taken before reaching the **stationary distribution** must be discarded $\rightarrow$ **burn-in phase**
- In practice it is safe to **discard the first half of each chain**

[Murphy, 2012 §24.4.1]

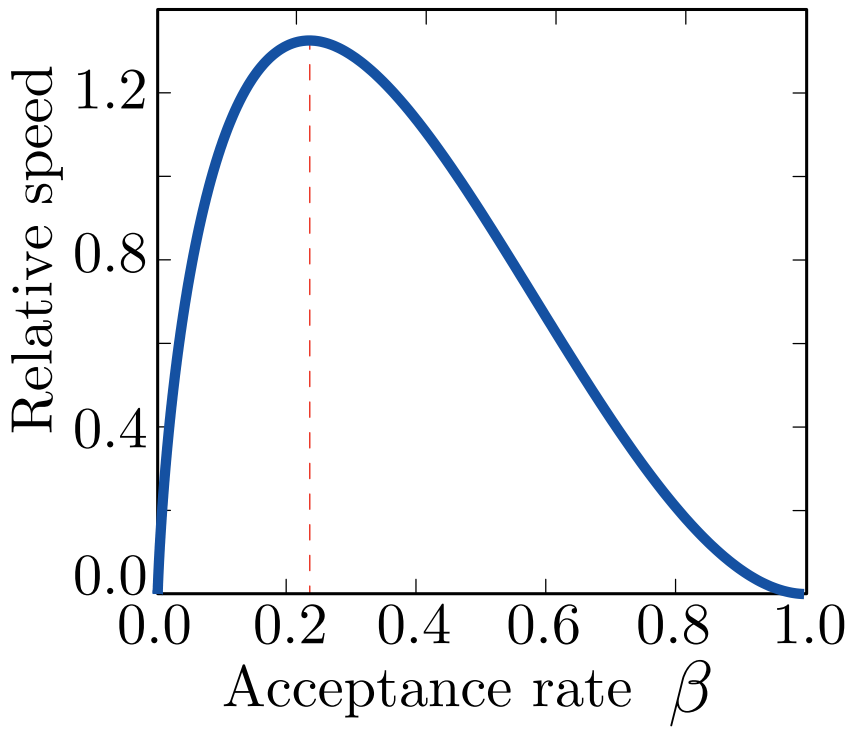
Monitoring convergence

MCMC in 1 dimension: **track convergence to the stationary distribution** by **plotting samples**

MCMC in two or more dimensions:

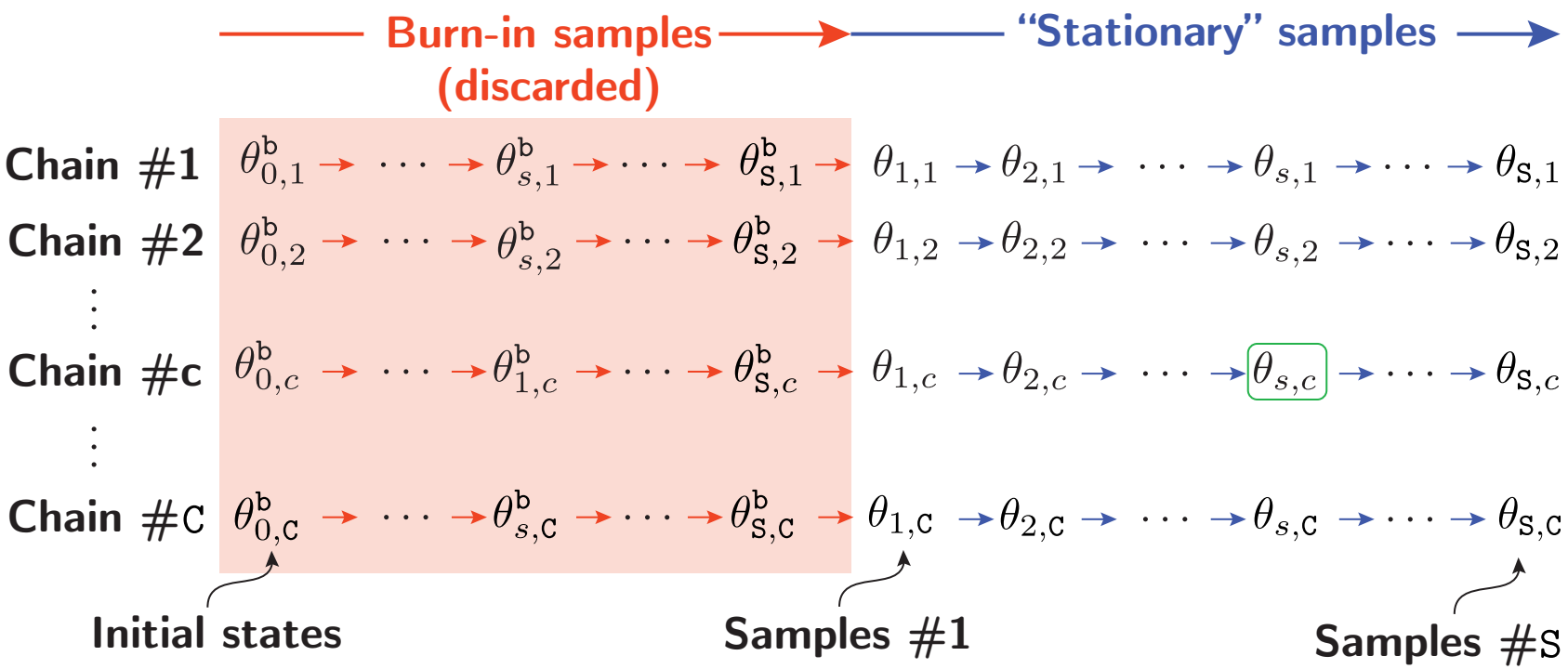
- Sample multiple chains (3-5) each having a different starting point θ_0
- Compare the variance within and between chains using the **estimated potential scale reduction**
- Check **acceptance rate**: 1D$\rightarrow$ $\beta \approx$40%, $\geq$5D$\rightarrow$ $\beta \approx$25%

MCMC efficiency ($\geq$5D) v.s. Acceptance rate β



[Rosenthal, Jeffrey S. *Optimal proposal distributions and adaptive MCMC*. Handbook of Markov Chain Monte Carlo (2011): 93-112.]

Multiple parallel chains



Estimated potential scale reduction – $\hat{R}$

Within-chains

Mean:

$$\bar{\theta}_{\cdot c} = \frac{1}{\mathrm{S}} \sum_{s=1}^{\mathrm{S}} \theta_{s,c}$$

Variance:

$$\underbrace{W = \frac{1}{\mathrm{C}} \sum_{c=1}^{\mathrm{C}} \left[\frac{1}{\mathrm{S}-1} \sum_{s=1}^{\mathrm{S}} (\theta_{s,c} - \bar{\theta}_{\cdot c})^2 \right]}_{\text{Underestimates } \mathrm{Var}[\theta_{s,c}]}$$

Between-chains

Mean:

$$\bar{\theta}_{\cdot\cdot} = \frac{1}{\mathrm{C}} \sum_{c=1}^{\mathrm{C}} \bar{\theta}_{\cdot c}$$

Variance:

$$B = \frac{1}{\mathrm{C}-1} \sum_{c=1}^{\mathrm{C}} (\bar{\theta}_{\cdot c} - \bar{\theta}_{\cdot\cdot})^2$$

$$\overbrace{\hat{V} = \frac{\mathrm{S}-1}{\mathrm{S}} W + B}^{\text{Overestimates } \mathrm{Var}[\theta_{s,c}]} \quad \rightarrow \quad \boxed{\hat{R} = \sqrt{\frac{\hat{V}}{W}}}$$

Estimated potential scale reduction – $\hat{R}$

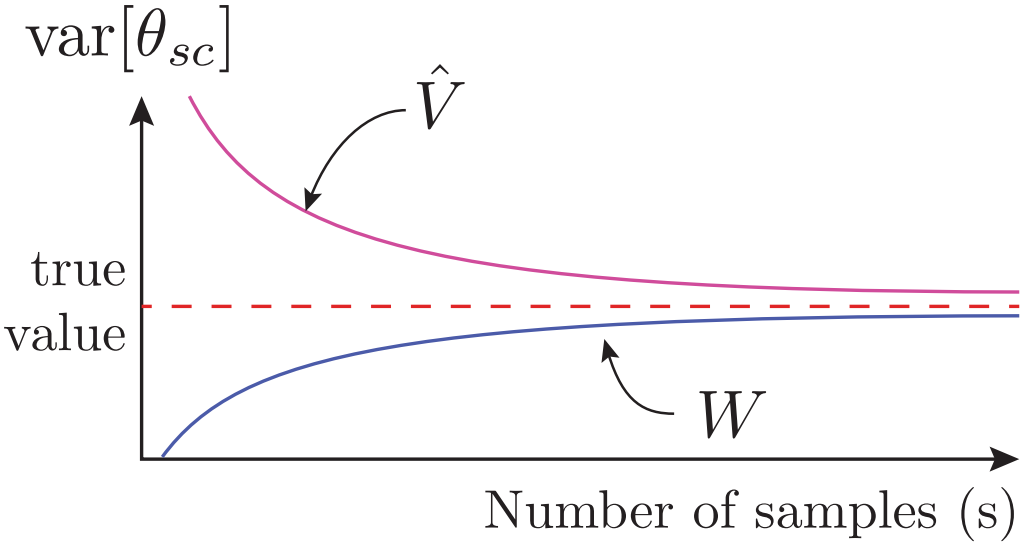
$$\hat{R} = \sqrt{\frac{\hat{V}}{W}}$$

W: **Underestimate** the variance of $\theta_{s,c}$

$\hat{V}$: **Overestimate** the variance of $\theta_{s,c}$

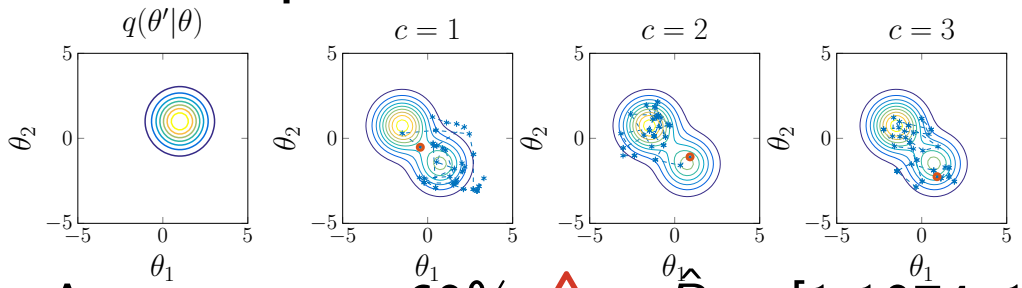
If $\hat{R} \approx 1 \rightarrow$ ✓

else if $\hat{R} > 1 \rightarrow$ ✗ (non-stationary)

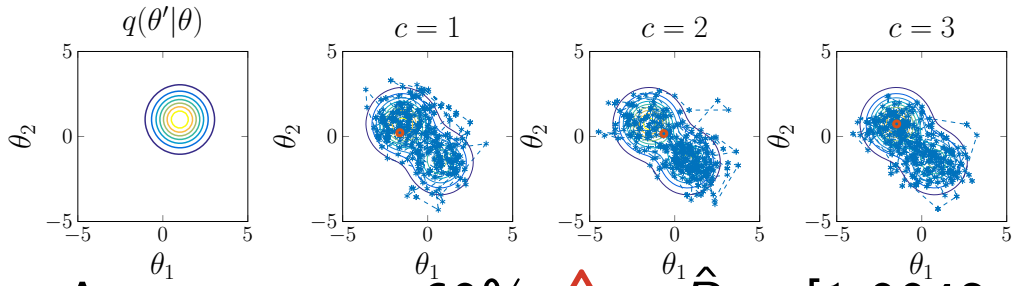


For $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_{\mathrm{P}}]^{\intercal} \rightarrow$ compute $\hat{R}_p, \ \forall p = 1 : \mathrm{P}$

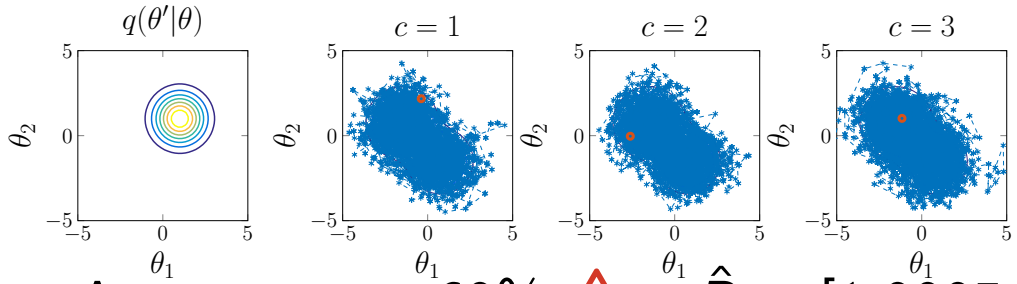
Example – Estimated potential scale reduction [.m]



$S = 100$: Accept. rate: 60% ⚠ , $\hat{R} = [1.1074, 1.0915]$ ⚠



$S = 1000$: Accept. rate: 60% ⚠ , $\hat{R} = [1.0042, 1.0107]$ ⚠

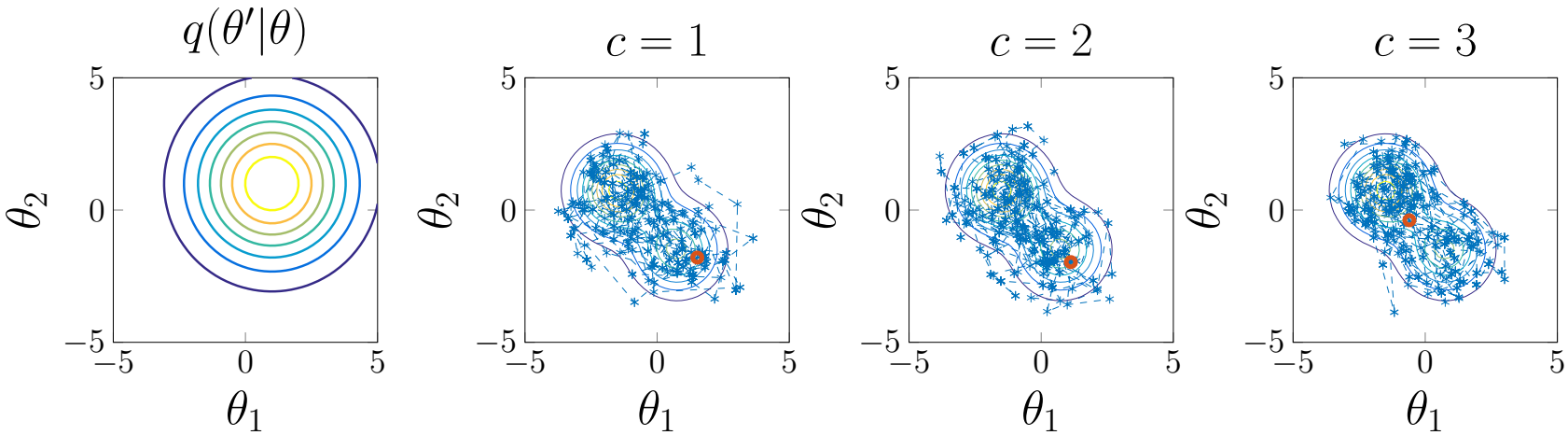


$S = 10000$: Accept. rate: 60% ⚠ , $\hat{R} = [1.0005, 1.0011]$ ✓

[CIVML/MCMC_M_EPSR_2D.m]

Example – Estimated potential scale reduction [.m]

$$\sigma_q = 1 \rightarrow \sigma_q = 2$$



$\mathrm{S} = 1000$: Accept. rate: 38% ✓ , $\hat{R} = [1.0182, 1.0194]$ ✗

[CIVML/MCMC_M_EPSR_2D.m]

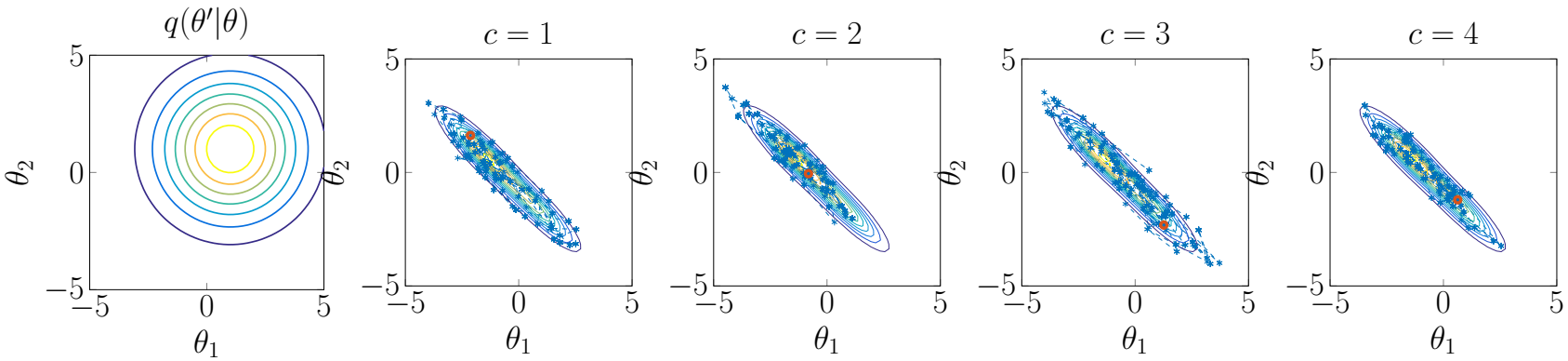
Section Outline

Proposal PDF

4.1 Criteria for a good proposal PDF

4.2 Proposal PDF generic formulation

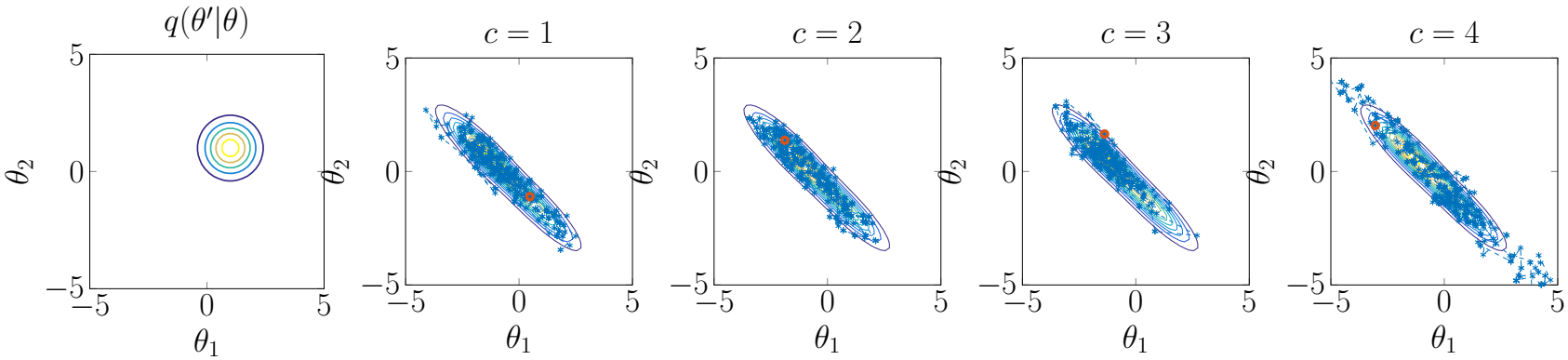
Example – Proposal PDF [.m]



$S = 1000$: Accept. rate: 17% ⚠ , $\hat{R} = [1.0556, 1.049]$ ⚠

[CIVML/MCMC_M_EPSR_2D.m]

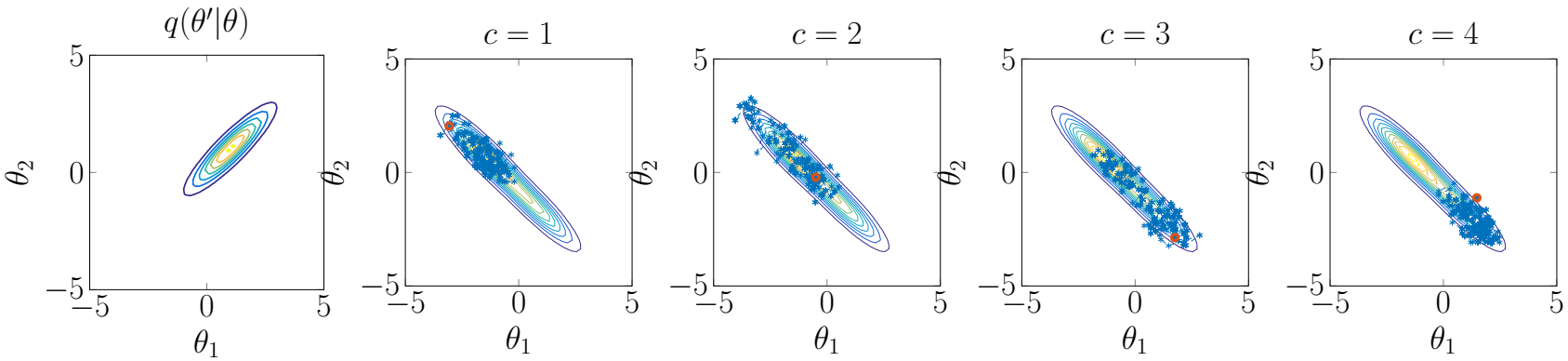
Example – Proposal PDF [.m]



$S = 1000$: Accept. rate: 33% ✓ , $\hat{R} = [1.054, 1.0564]$ ✗

[CIVML/MCMC_M_EPSR_2D.m]

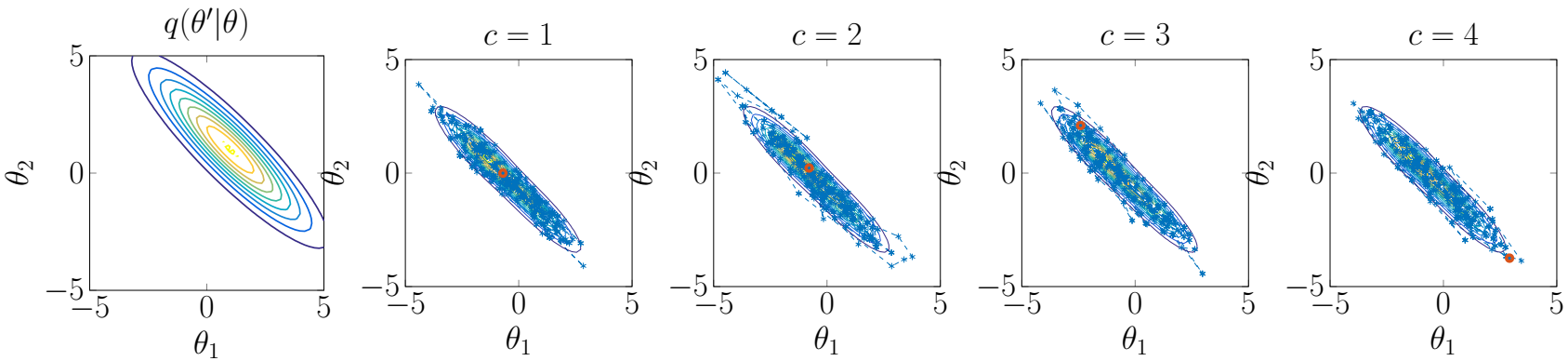
Example – Proposal PDF [.m]



$S = 1000$: Accept. rate: 27% ⚠ , $\hat{R} = [1.3691, 1.3566]$ ⚠

[CIVML/MCMC_M_EPSR_2D.m]

Example – Proposal PDF [.m]



$\mathrm{S} = 1000$: Accept. rate: 36% ✓ , $\hat{R} = [1.0054, 1.0039]$ ✓

In 1/2D, it is easy to identify a correct proposal $q(\theta'|\theta)$
...it is much harder when $> 2D$

[CIVML/MCMC_M_EPSR_2D.m]

Proposal PDF generic formulation

One generic way to define the proposal is using a **normal distribution** centred on the current state $\boldsymbol{\theta}$ and with covariance matrix $\gamma^2 \boldsymbol{\Sigma}_{\boldsymbol{\theta}^*}$

$$q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}'; \boldsymbol{\theta}, \gamma^2 \boldsymbol{\Sigma}_{\boldsymbol{\theta}^*}), \quad \gamma^2 = \frac{2.4^2}{\mathrm{P}}, \quad (\mathrm{P} : \text{nb. parameters})$$

$\boldsymbol{\Sigma}_{\boldsymbol{\theta}^*}$ is estimated using the **Laplace Approximation** calculated for the **maximum a posteriori**-value (MAP), $\boldsymbol{\theta}^*$

The MAP can be estimated using gradient-based optimization techniques such as the **Newton-Raphson** algorithm

[Roberts and Rosenthal, 2001]

Section Outline

Newton & Laplace

5.1 Introduction
5.2 Gradient ascent
5.3 Newton-Raphson's method 1D
5.4 Numerical derivatives
5.5 Newton-Raphson's method >1D : Coordinate ascent
5.6 Laplace approximation

MAP v.s. MLE

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \tilde{f}(\boldsymbol{\theta}) \equiv \arg\max_{\boldsymbol{\theta}} \ln \tilde{f}(\boldsymbol{\theta})$$

$$\underbrace{f(\boldsymbol{\theta}|\mathcal{D})}_{\text{posterior}} = \frac{\overbrace{f(\mathcal{D}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})}^{\text{unnormalized posterior}}}{\underbrace{f(\mathcal{D})}_{\text{normalization cte.}}}$$

Maximum Likelihood Estimate (MLE, $\boldsymbol{\theta}^*$)

$$\tilde{f}(\boldsymbol{\theta}) = \overbrace{f(\mathcal{D}|\boldsymbol{\theta})}^{\text{likelihood}}$$

Note: if $f(\boldsymbol{\theta}) \propto 1$

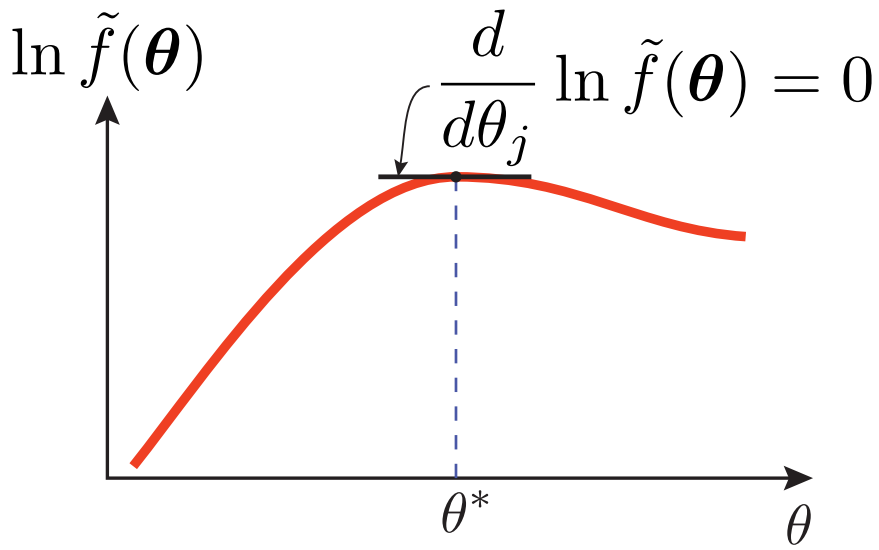
Maximum A Posteriori (MAP, $\boldsymbol{\theta}^*$)

MAP≡MLE

$$\tilde{f}(\boldsymbol{\theta}) = \overbrace{f(\mathcal{D}|\boldsymbol{\theta}) \cdot f(\boldsymbol{\theta})}^{\text{unnormalized posterior}}$$

Introduction to gradient-based optimization

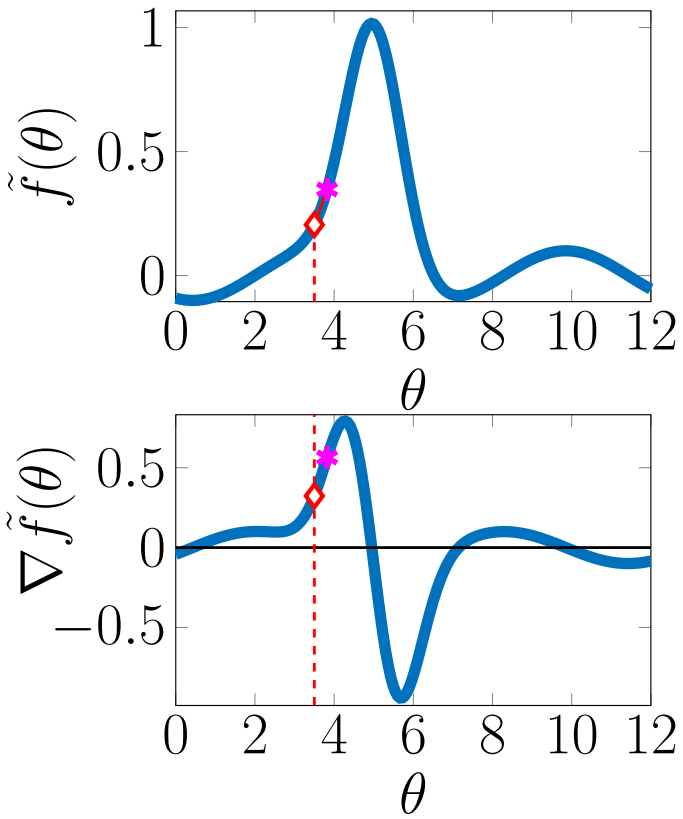
$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \ln \tilde{f}(\boldsymbol{\theta})$$



$$\boldsymbol{\theta}^* : \frac{d}{d\theta_j} \ln \tilde{f}(\boldsymbol{\theta}) = 0 \rightarrow \left\{ \begin{array}{l} \textbf{Gradient ascent} \\ \textbf{Newton-Raphson} \end{array} \right.$$

Example - Gradient ascent [.m]

$$\theta_{\text{new}} = \theta_{\text{old}} + \lambda \cdot \nabla_\theta \tilde{f}(\theta_{\text{old}}) \qquad \left(\nabla \tilde{f}(\theta) \equiv \nabla_\theta \tilde{f}(\theta) \equiv \frac{d\tilde{f}(\theta)}{d\theta} \right)$$



loop: #1

θ_{old} =3.5

$\tilde{f}(\theta_{old})$ =0.20515

$\nabla \tilde{f}(\theta_{old})$ =0.32327

$\theta_{new} = \theta_{old} + \nabla \tilde{f}(\theta_{old})$

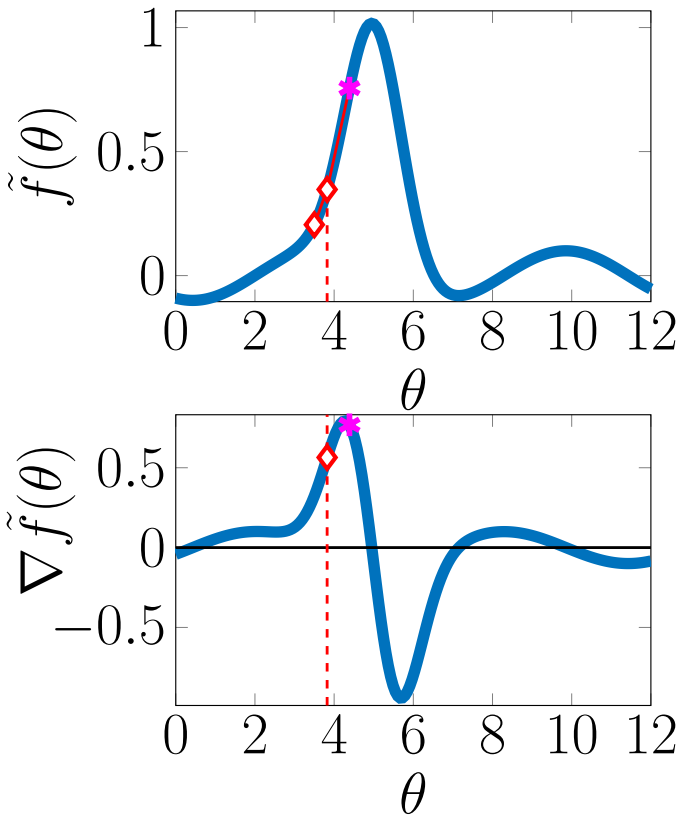
=3.8233

$\tilde{f}(\theta_{new})$ =0.34723

[CIV_ML/Gradient_Ascent_demo.m]

Example - Gradient ascent [.m]

$$\theta_{\text{new}} = \theta_{\text{old}} + \lambda \cdot \nabla_\theta \tilde{f}(\theta_{\text{old}}) \qquad \left(\nabla \tilde{f}(\theta) \equiv \nabla_\theta \tilde{f}(\theta) \equiv \frac{d\tilde{f}(\theta)}{d\theta} \right)$$



loop: #2

θ_{old} =3.8233

$\tilde{f}(\theta_{old})$ =0.34723

$\nabla \tilde{f}(\theta_{old})$ =0.56433

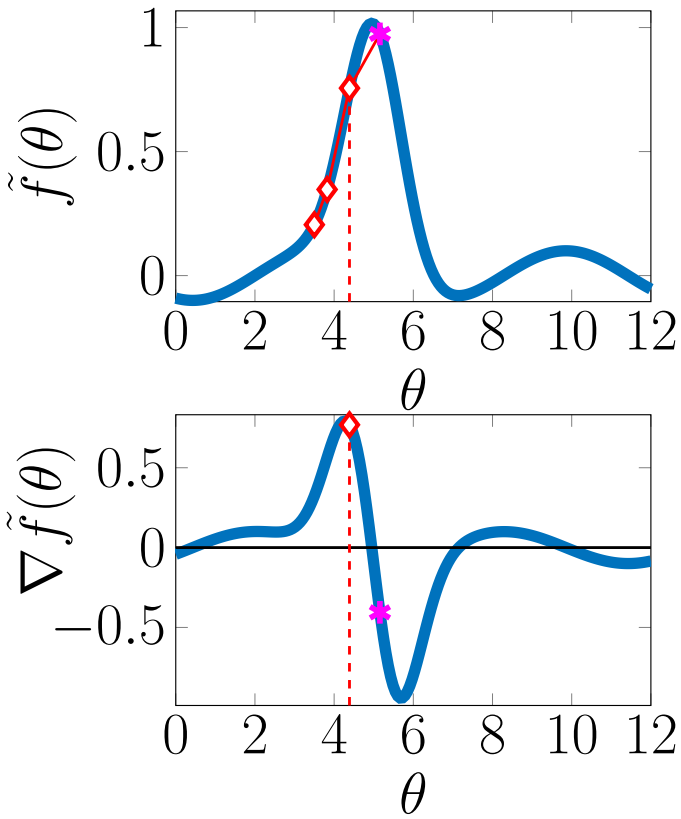
$\theta_{new} = \theta_{old} + \nabla \tilde{f}(\theta_{old})$
=4.3876

$\tilde{f}(\theta_{new})$ =0.75572

[CIV_ML/Gradient_Ascent_demo.m]

Example - Gradient ascent [.m]

$$\theta_{\text{new}} = \theta_{\text{old}} + \lambda \cdot \nabla_\theta \tilde{f}(\theta_{\text{old}}) \qquad \left(\nabla \tilde{f}(\theta) \equiv \nabla_\theta \tilde{f}(\theta) \equiv \frac{d\tilde{f}(\theta)}{d\theta} \right)$$



loop: $\#3$

θ_{old} =4.3876

$\tilde{f}(\theta_{old})$ =0.75572

$\nabla \tilde{f}(\theta_{old})$ =0.76887

$\theta_{new} = \theta_{old} + \nabla \tilde{f}(\theta_{old})$

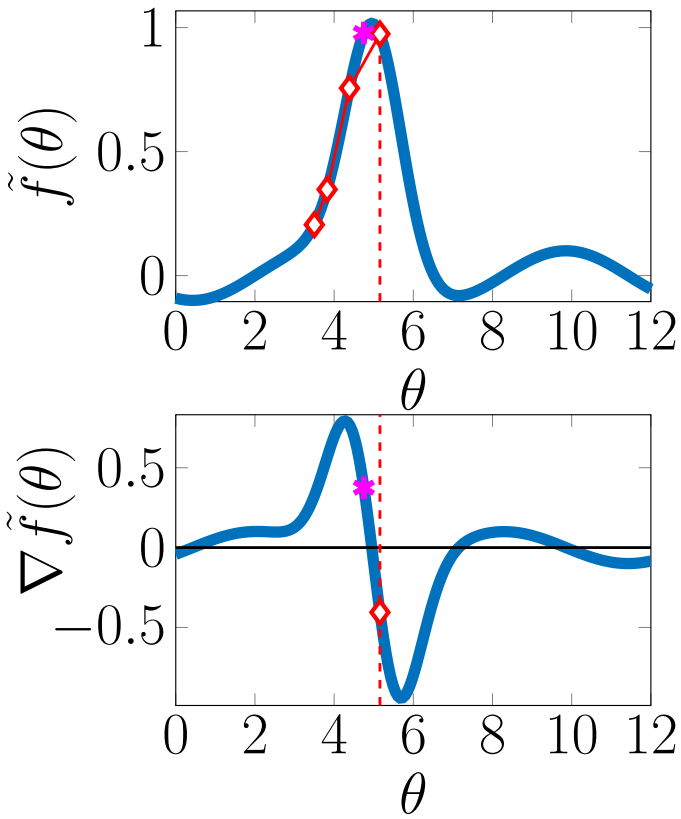
=5.1565

$\tilde{f}(\theta_{new})$ =0.97433

[CIV_ML/Gradient_Ascent_demo.m]

Example - Gradient ascent [.m]

$$\theta_{\text{new}} = \theta_{\text{old}} + \lambda \cdot \nabla_\theta \tilde{f}(\theta_{\text{old}}) \qquad \left(\nabla \tilde{f}(\theta) \equiv \nabla_\theta \tilde{f}(\theta) \equiv \frac{d\tilde{f}(\theta)}{d\theta} \right)$$



loop: $\#4$

θ_{old} =5.1565

$\tilde{f}(\theta_{old})$ =0.97433

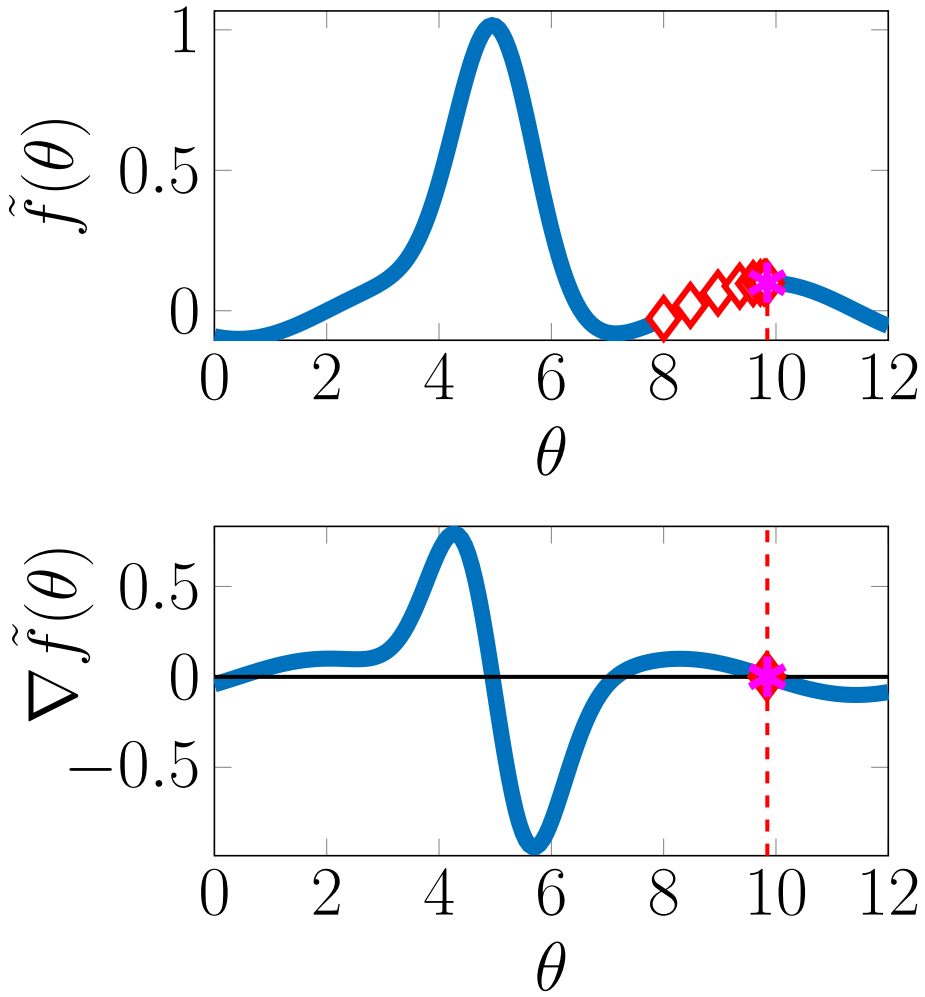
$\nabla \tilde{f}(\theta_{old})$ =-0.40536

$\theta_{new} = \theta_{old} + \nabla \tilde{f}(\theta_{old})$
=4.7511

$\tilde{f}(\theta_{new})$ =0.978

[CIV_ML/Gradient_Ascent_demo.m]

Local maxima - Gradient ascent [.m]



[CIV_ML/Gradient_Ascent_demo.m]

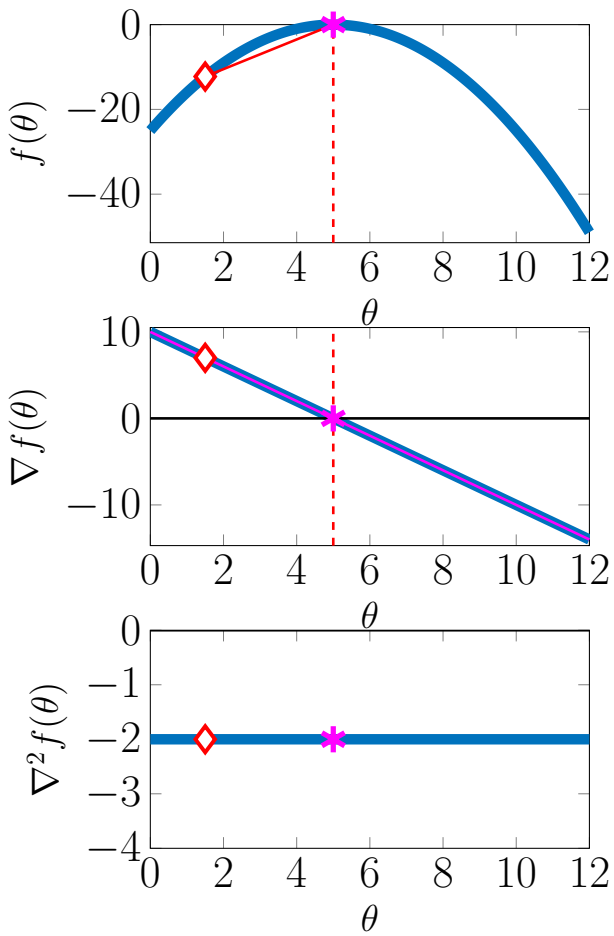
Algorithm 3: Gradient Ascent algorithm

initialize $\lambda = \lambda_0$, $\theta_{\text{old}} = \theta_0$, define ϵ
while $|\nabla_\theta \tilde{f}(\theta_{old})| > \epsilon$ **do**
  compute: $\theta_{\text{old}} \rightarrow \begin{cases} \tilde{f}(\theta_{\text{old}}) & \text{function value} \\ \nabla_\theta \tilde{f}(\theta_{\text{old}}) & \text{Gradient} \end{cases}$
  compute $\theta_{\text{new}} = \theta_{\text{old}} + \lambda \nabla_\theta \tilde{f}(\theta_{\text{old}})$
  if $\tilde{f}(\theta_{old}) > \tilde{f}(\theta_{new})$ **then**
    assign $\lambda = \lambda/2$
    Goto 4
  assign $\lambda = \lambda_0$, $\theta_{\text{old}} = \theta_{\text{new}}$
assign $\theta^* = \theta_{\text{new}}$

Limitation: Finding an efficient value for λ is difficult []

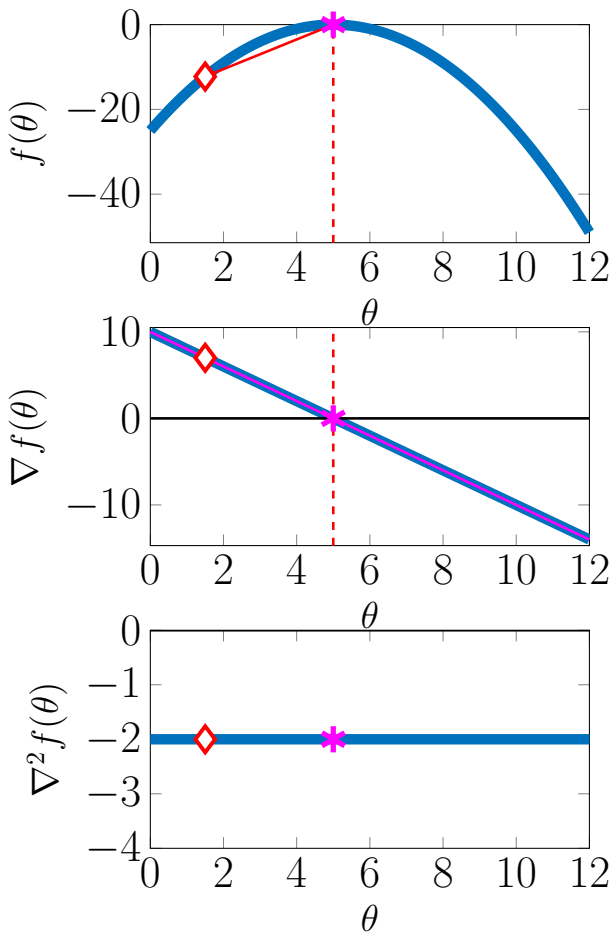
[CIV_ML/Gradient_Ascent_demo.m]

Introduction – Newton-Raphson



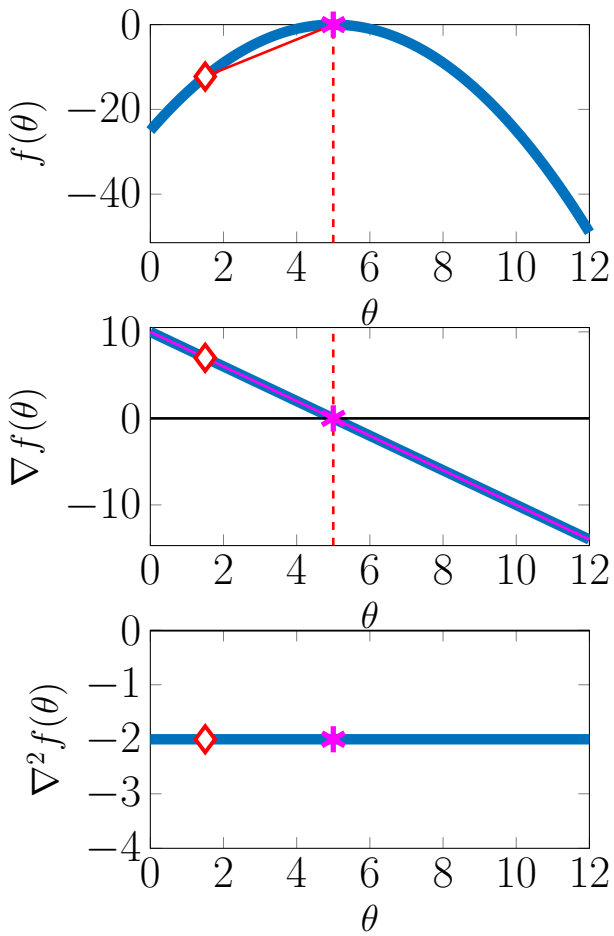
$$\nabla \tilde{f}(\theta) \approx \nabla\nabla \tilde{f}(\theta_{\text{old}}) \cdot (\theta - \theta_{\text{old}}) + \nabla \tilde{f}(\theta_{\text{old}})$$

Introduction – Newton-Raphson



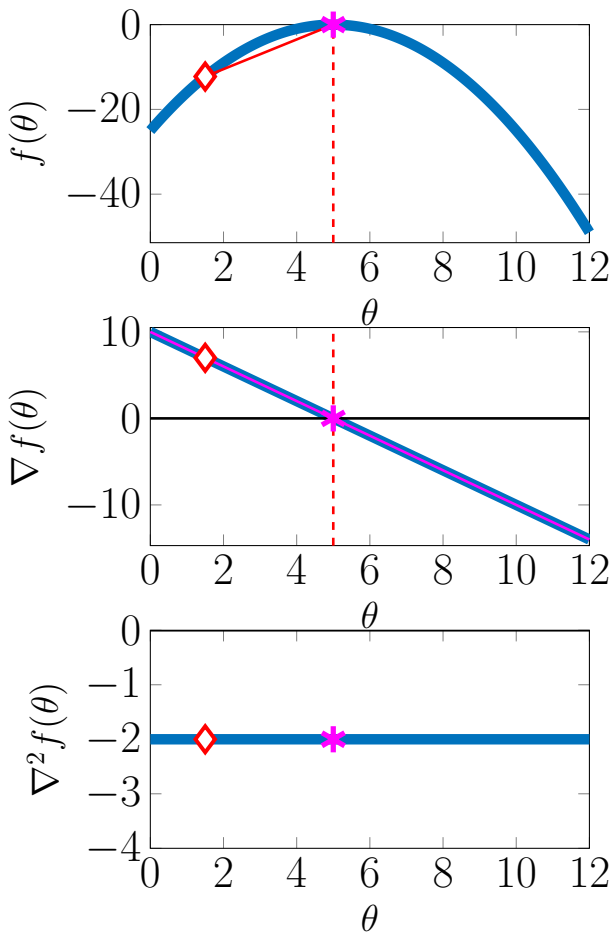
$$0 = \nabla\nabla\tilde{f}(\theta_{\text{old}}) \cdot (\theta - \theta_{\text{old}}) + \nabla\tilde{f}(\theta_{\text{old}})$$

Introduction – Newton-Raphson



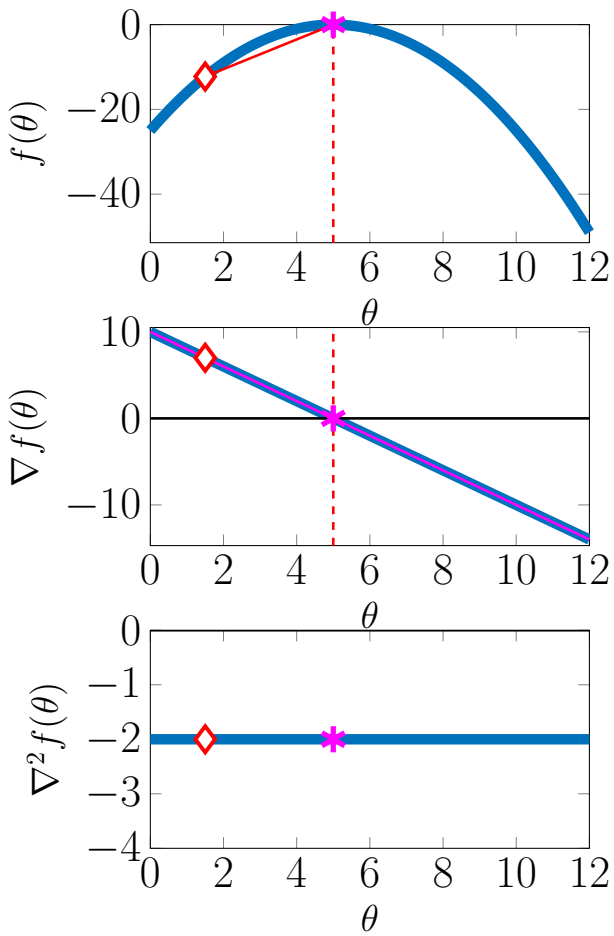
$$-\nabla \tilde{f}(\theta_{\text{old}}) = \nabla\nabla \tilde{f}(\theta_{\text{old}}) \cdot (\theta - \theta_{\text{old}})$$

Introduction – Newton-Raphson



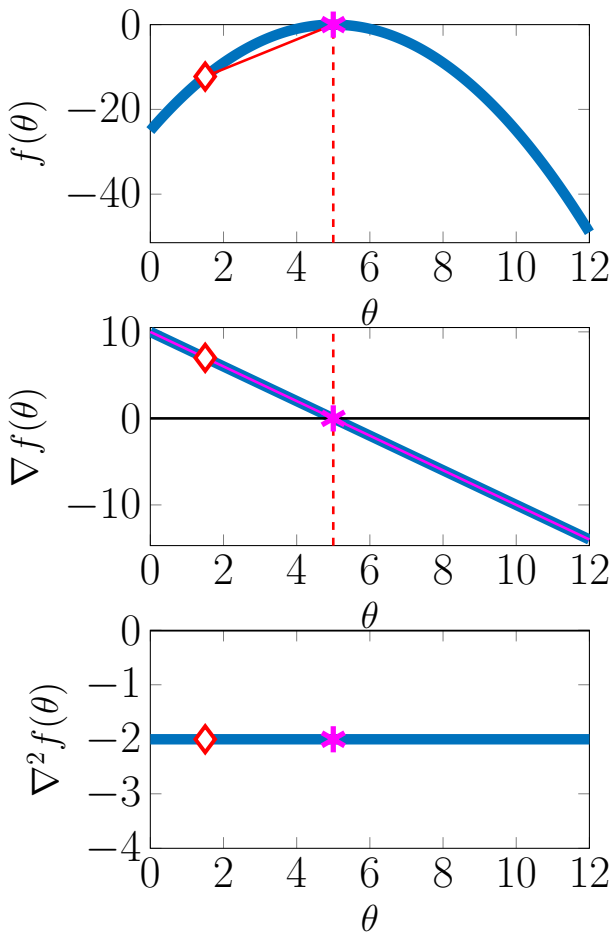
$$-\frac{\nabla \tilde{f}(\theta_{\text{old}})}{\nabla\nabla \tilde{f}(\theta_{\text{old}})} = \theta - \theta_{\text{old}}$$

Introduction – Newton-Raphson



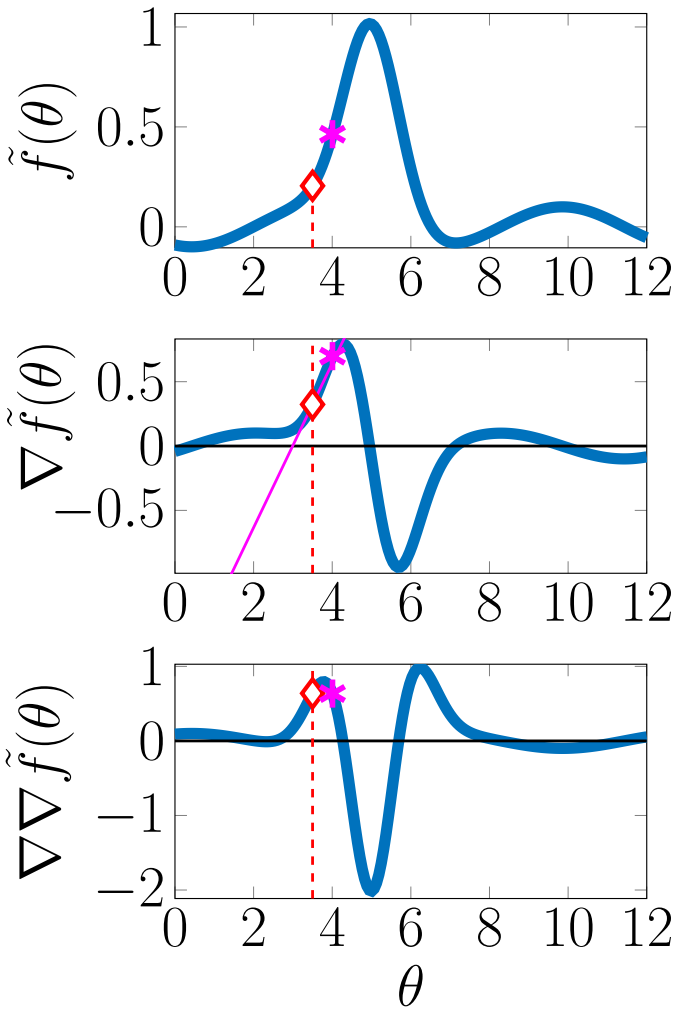
$$\theta_{\text{old}} - \frac{\nabla \tilde{f}(\theta_{\text{old}})}{\nabla\nabla \tilde{f}(\theta_{\text{old}})} = \theta$$

Introduction – Newton-Raphson



$$\theta_{\text{new}} = \theta_{\text{old}} - \frac{\nabla \tilde{f}(\theta_{\text{old}})}{\nabla\nabla \tilde{f}(\theta_{\text{old}})}$$

Example – Newton-Raphson [.m]



loop: #1

$\theta_{old} = 3.5$

$\tilde{f}(\theta_{old}) = 0.20515$

$\lambda = 1$

$\nabla\tilde{f}(\theta_{old}) = 0.32327$

$\nabla\nabla\tilde{f}(\theta_{old}) = 0.63805$

$> 0 \rightarrow \lambda = -\lambda$

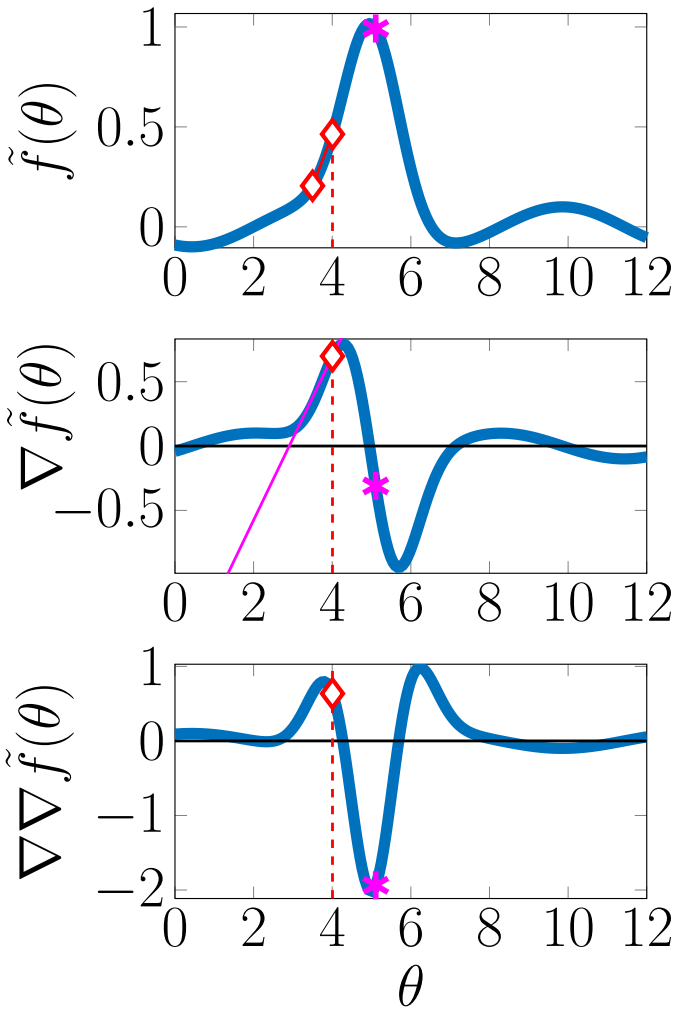
$$\theta_{new} = \theta_{old} - \lambda \frac{\nabla\tilde{f}(\theta_{old})}{\nabla\nabla\tilde{f}(\theta_{old})}$$

$= 4.0067$

$\tilde{f}(\theta_{new}) = 0.46345$

[CIV_ML/NR_demo_book.m]

Example – Newton-Raphson [.m]



loop: #2

θ_{old} =4.0067

$\tilde{f}(\theta_{old})$ =0.46345

λ =1

$\nabla\tilde{f}(\theta_{old}) = 0.69841$

$\nabla\nabla\tilde{f}(\theta_{old}) = 0.63515$

$> 0 \rightarrow \lambda = -\lambda$

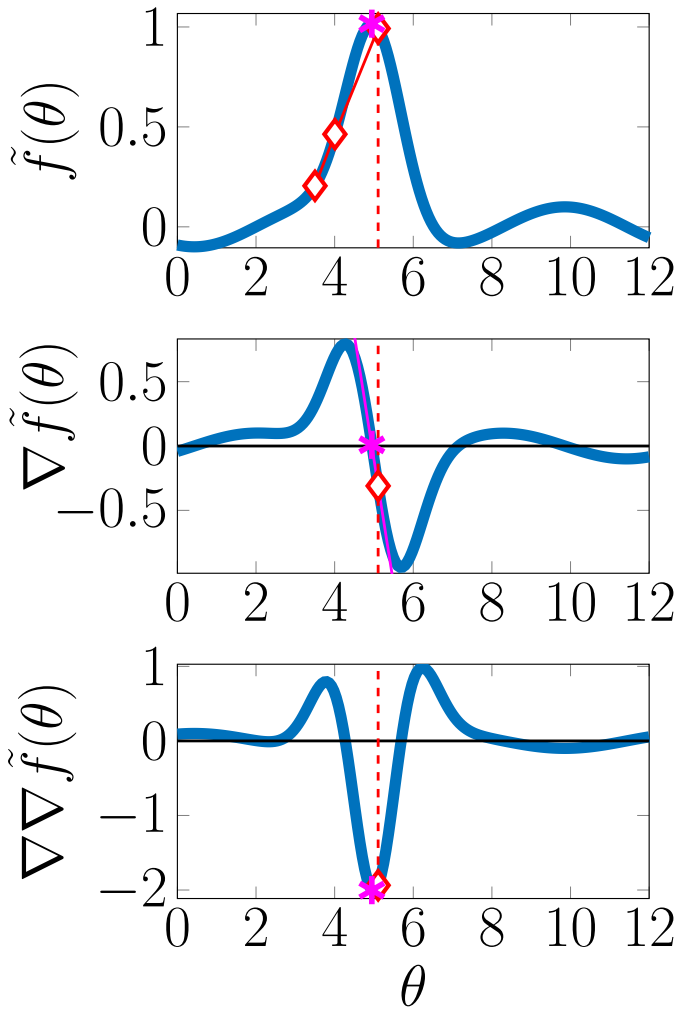
$$\theta_{new} = \theta_{old} - \lambda \frac{\nabla\tilde{f}(\theta_{old})}{\nabla\nabla\tilde{f}(\theta_{old})}$$

=5.1063

$\tilde{f}(\theta_{new})$ =0.99231

[CIV_ML/NR_demo_book.m]

Example – Newton-Raphson [.m]



loop: #3

θ_{old} =5.1063

$\tilde{f}(\theta_{old})$ =0.99231

λ =1

$\nabla\tilde{f}(\theta_{old}) = -0.31007$

$\nabla\nabla\tilde{f}(\theta_{old}) = -1.9364$

$< 0 \rightarrow$ OK

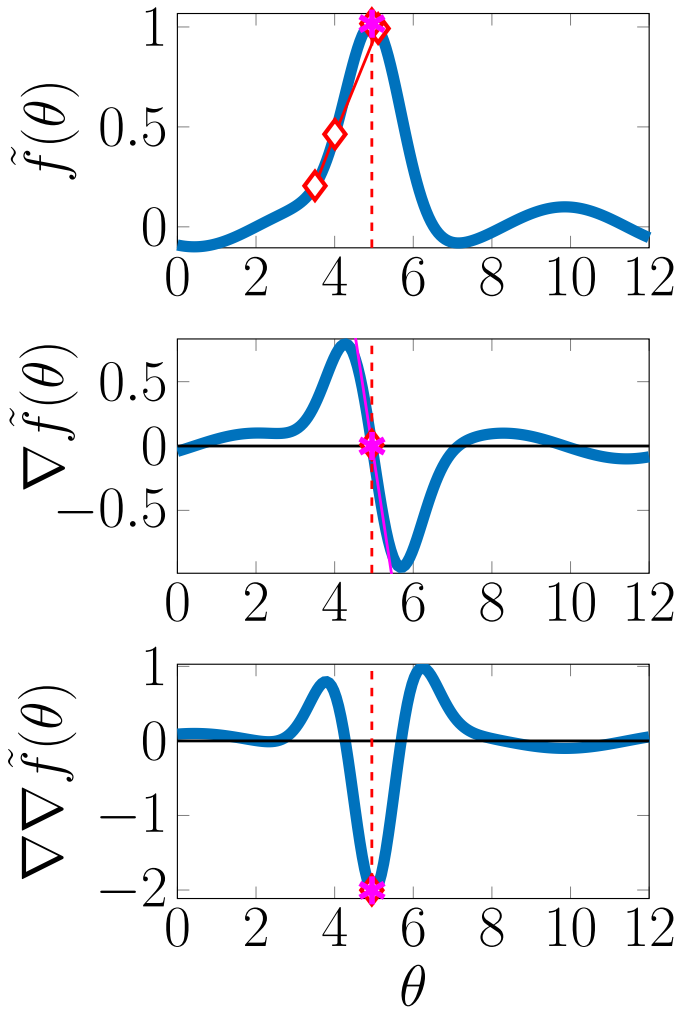
$$\theta_{new} = \theta_{old} - \lambda \frac{\nabla\tilde{f}(\theta_{old})}{\nabla\nabla\tilde{f}(\theta_{old})}$$

=4.9461

$\tilde{f}(\theta_{new})$ =1.0165

[CIV_ML/NR_demo_book.m]

Example – Newton-Raphson [.m]



loop: #4

θ_{old} =4.9461

$\tilde{f}(\theta_{old})$ =1.0165

λ =1

$\nabla\tilde{f}(\theta_{old}) = 0.0093241$

$\nabla\nabla\tilde{f}(\theta_{old}) = -2.0021$

$< 0 \rightarrow$ OK

$$\theta_{new} = \theta_{old} - \lambda \frac{\nabla\tilde{f}(\theta_{old})}{\nabla\nabla\tilde{f}(\theta_{old})}$$

=4.9508

$\tilde{f}(\theta_{new})$ =1.0165

[CIV_ML/NR_demo_book.m]

Algorithm 4: Newton-Raphson 1D

1 initialize $\lambda = \lambda_0 = 1$, $\theta_{\text{old}} = \theta_0$

2 define ϵ, $\tilde{f}(\theta)$

3 **while** $|\nabla_\theta \tilde{f}(\theta_{old})| > \epsilon$ **do**

4 compute: $\theta_{\text{old}} \rightarrow \begin{cases} \tilde{f}(\theta_{\text{old}}) & \text{Function evaluation} \\ \nabla_\theta \tilde{f}(\theta_{\text{old}}) & \text{Gradient} \\ \nabla\nabla_{\theta\theta} \tilde{f}(\theta_{\text{old}}) & \text{1D-Hessian} \end{cases}$

5 **if** $\nabla\nabla_{\theta\theta} \tilde{f}(\theta_{old}) > 0 \rightarrow \lambda = -\lambda$ **then**

6 compute $\theta_{\text{new}} = \theta_{\text{old}} - \lambda \frac{\nabla_\theta \tilde{f}(\theta_{\text{old}})}{\nabla\nabla_{\theta\theta} \tilde{f}(\theta_{\text{old}})}$

7 **if** $\tilde{f}(\theta_{old}) > \tilde{f}(\theta_{new})$ **then**

8 assign $\lambda = \lambda/2$

9 Goto 6

10 assign $\lambda = \lambda_0$, $\theta_{\text{old}} = \theta_{\text{new}}$

11 assign $\theta^* = \theta_{\text{old}}$

Newton-Raphson algorithm – Numerical derivatives

How do we compute our 1st and 2nd derivatives?

Either analytically

... compute analytic formulation

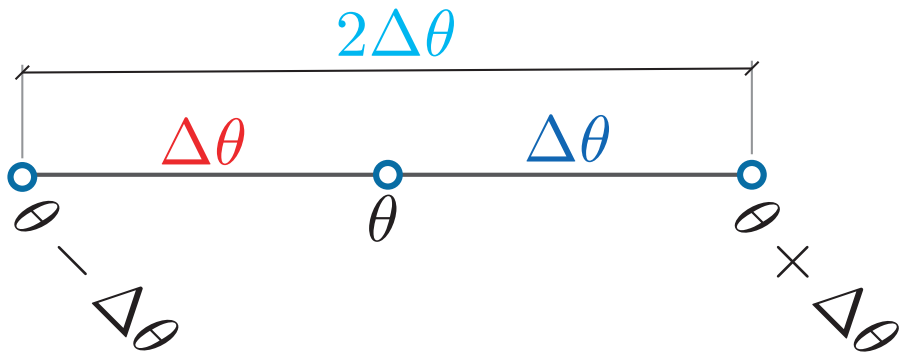
or numerically $(\Delta\theta \ll \theta)$

$$\nabla_{\theta_i} \tilde{f}(\boldsymbol{\theta}) = \frac{\partial \tilde{f}(\boldsymbol{\theta})}{\partial \theta_i} \approx \frac{\tilde{f}(\boldsymbol{\theta} + \mathbb{I}(i)\Delta\theta) - \tilde{f}(\boldsymbol{\theta} - \mathbb{I}(i)\Delta\theta)}{2\Delta\theta}$$

$$\nabla\nabla_{\theta_i\theta_i} \tilde{f}(\boldsymbol{\theta}) = \frac{\partial^2 \tilde{f}(\boldsymbol{\theta})}{\partial \theta_i \partial \theta_i} \approx \frac{\tilde{f}(\boldsymbol{\theta} + \mathbb{I}(i)\Delta\theta) - 2\tilde{f}(\boldsymbol{\theta}) + \tilde{f}(\boldsymbol{\theta} - \mathbb{I}(i)\Delta\theta)}{(\Delta\theta)^2}$$

```
# Index vector
I = lambda i : np.linspace(1,n,n) == i #nx1 indicator vector
# Gradient
grad_fct = lambda x, dx,i : (fx_TR(x+I(i)*dx)-fx_TR(x-I(i)*dx))/(2*dx)
# 1D Hessian
hess_fct = lambda x,dx,i : (fx_TR(x+I(i)*dx)-2*fx_TR(x)+fx_TR(x-I(i)*dx))/dx**2
```

Numerical derivatives



$$\frac{\partial \tilde{f}(\boldsymbol{\theta})}{\partial \theta_i} \approx \frac{\tilde{f}(\boldsymbol{\theta} + \mathbb{I}(i)\Delta\theta) - \tilde{f}(\boldsymbol{\theta} - \mathbb{I}(i)\Delta\theta)}{2\Delta\theta}$$

$$\frac{\partial \tilde{f}(\boldsymbol{\theta})}{\partial \theta_i^2} \approx \frac{\frac{\tilde{f}(\boldsymbol{\theta} + \mathbb{I}(i)\Delta\theta) - \tilde{f}(\boldsymbol{\theta})}{\Delta\theta} - \frac{\tilde{f}(\boldsymbol{\theta}) - \tilde{f}(\boldsymbol{\theta} - \mathbb{I}(i)\Delta\theta)}{\Delta\theta}}{\Delta\theta}$$

$$= \frac{\tilde{f}(\boldsymbol{\theta} + \mathbb{I}(i)\Delta\theta) - 2\tilde{f}(\boldsymbol{\theta}) + \tilde{f}(\boldsymbol{\theta} - \mathbb{I}(i)\Delta\theta)}{(\Delta\theta)^2}$$

Algorithm 5: Newton-Raphson >1D

1 initialize $\lambda = 1$, $\boldsymbol{\theta}_{\text{old}} = \boldsymbol{\theta}_0 = [\theta_1, \theta_2, \cdots, \theta_{\text{P}}]^{\intercal}$

2 define ϵ , $\tilde{f}(\theta)$

3 **while** $|\nabla_\theta \tilde{f}(\theta_{old})| > \epsilon$ **do**

4 **for** $p = 1 : \text{P}$ **do**

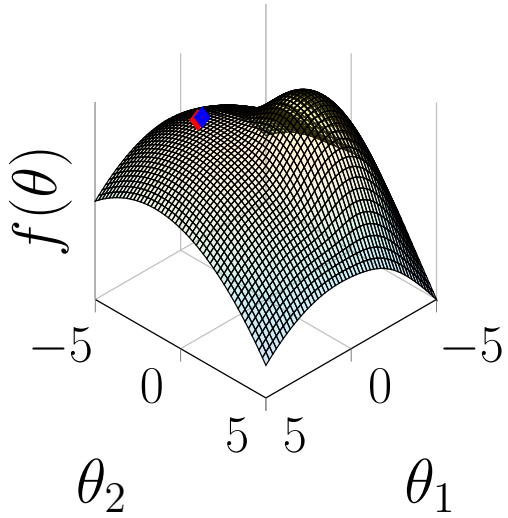
5 compute: $\theta_{p,\text{old}} \rightarrow \begin{cases} \tilde{f}(\theta_{p,\text{old}}) & \text{Function eval.} \\ \nabla_\theta \tilde{f}(\theta_{p,\text{old}}) & \text{Gradient} \\ \nabla\nabla_{\theta\theta} \tilde{f}(\theta_{p,\text{old}}) & \text{1D-Hessian} \end{cases}$

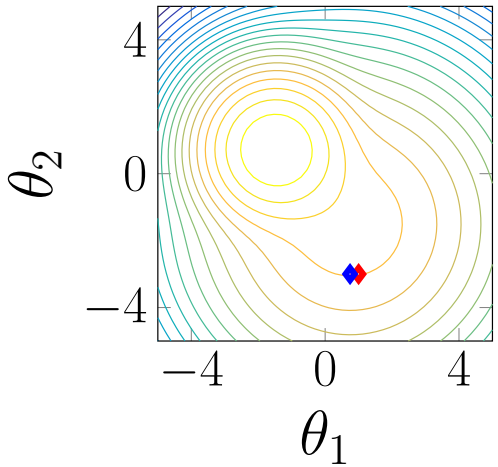
6 ... idem, steps 5-10 from 1D Newton-Raphson

7 assign $\theta_{p,\text{old}} = \theta_{p,\text{new}}$

8 assign $\boldsymbol{\theta}^* = \boldsymbol{\theta}_{\text{old}}$

Example - Newton-Raphson 2D [.m]





loop: $\#1 \rightarrow \theta_1$

θ_{old} =1

λ =1

$f(\theta_{old})$ =-4.4292

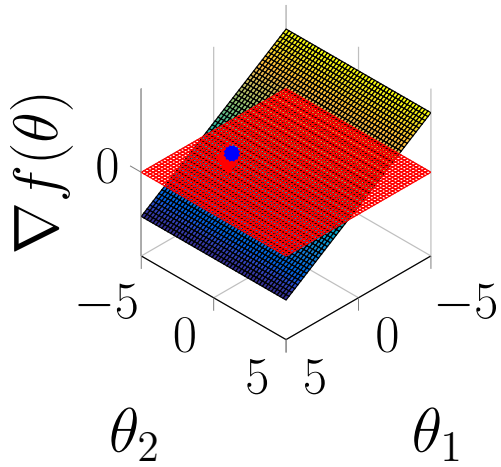
$\nabla f(\theta)$ =-0.063258

$\nabla\nabla f(\theta)$ =-0.2478

$\nabla\nabla f(\theta) < 0 \rightarrow$ OK

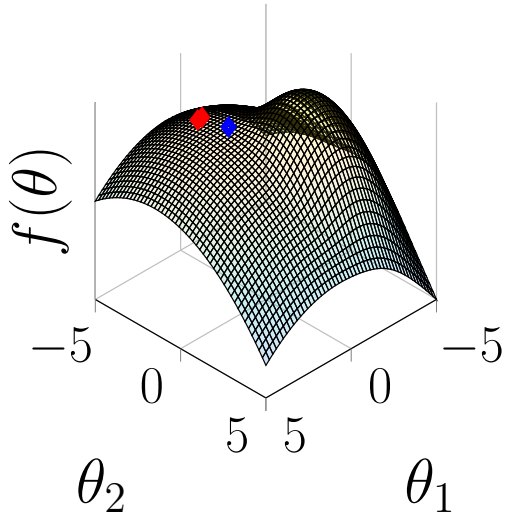
$$\theta_{new} = \theta_{old} - \lambda \frac{\nabla f(\theta)}{\nabla\nabla f(\theta)}$$
$$= 0.74472$$

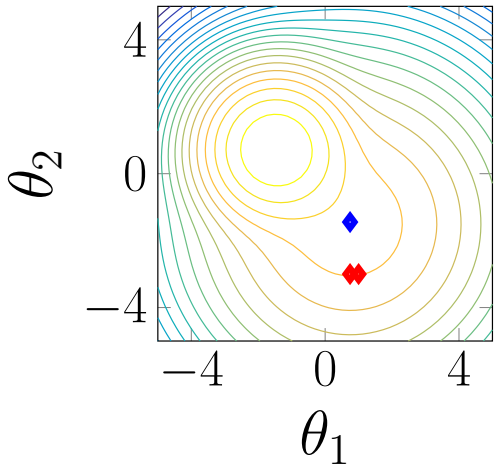
$f(\theta_{new})$ =-4.4212



[CIV_ML/NR_demo_book_2D.m]

Example - Newton-Raphson 2D [.m]





loop: $\#\mathbf{2} \rightarrow \theta_2$

θ_{old} =-3

λ =1

$f(\theta_{old})$ =-4.4212

$\nabla f(\theta)$ =0.37691

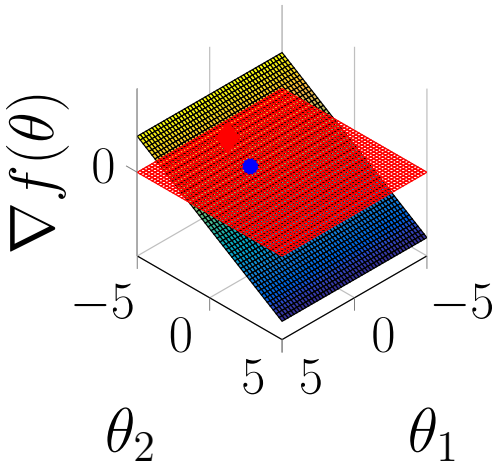
$\nabla\nabla f(\theta)$ =-0.24247

$\nabla\nabla f(\theta) < 0 \rightarrow$ OK

$\theta_{new} = \theta_{old} - \lambda \frac{\nabla f(\theta)}{\nabla\nabla f(\theta)}$

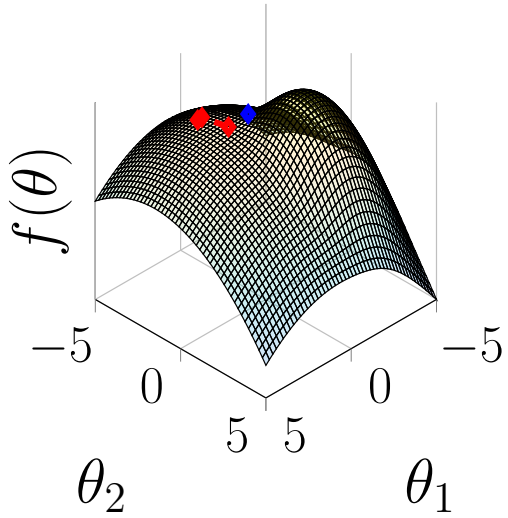
=-1.4456

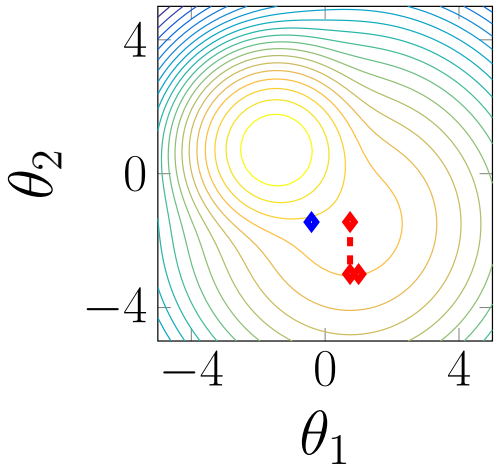
$f(\theta_{new})$ =-4.0984



[CIV_ML/NR_demo_book_2D.m]

Example - Newton-Raphson 2D [.m]





loop: $\#3 \rightarrow \theta_1$

θ_{old} =0.74472

λ =1

$f(\theta_{old})$ =-4.0984

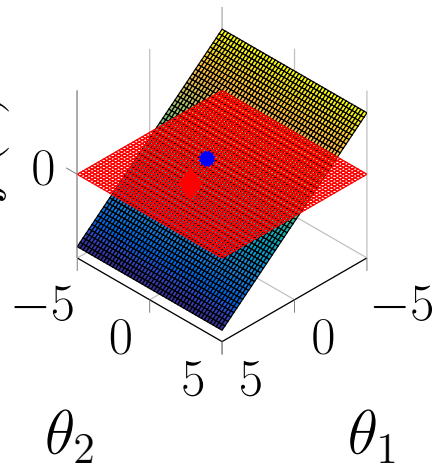
$\nabla f(\theta)$ =-0.092085

$\nabla\nabla f(\theta)$ =-0.079936

$\nabla\nabla f(\theta) < 0 \rightarrow$ OK

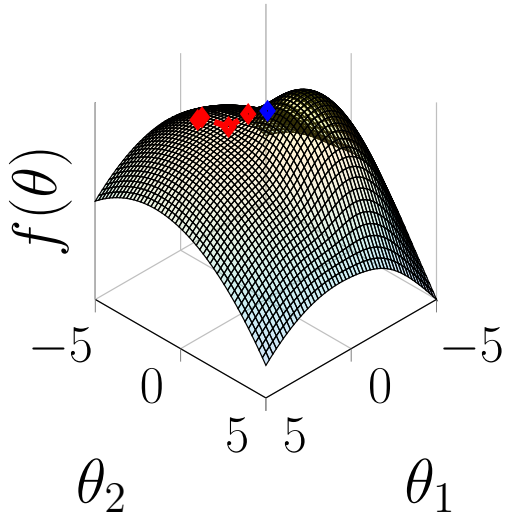
$$\theta_{new} = \theta_{old} - \lambda \frac{\nabla f(\theta)}{\nabla\nabla f(\theta)}$$
$$= \text{-}0.40726$$

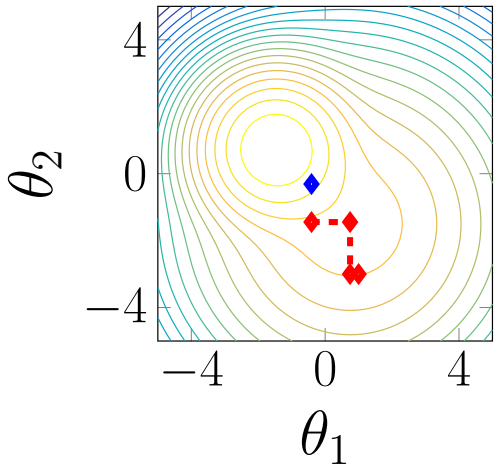
$f(\theta_{new})$ =-4.0076



[CIV_ML/NR_demo_book_2D.m]

Example - Newton-Raphson 2D [.m]





loop: $\#4 \rightarrow \theta_2$

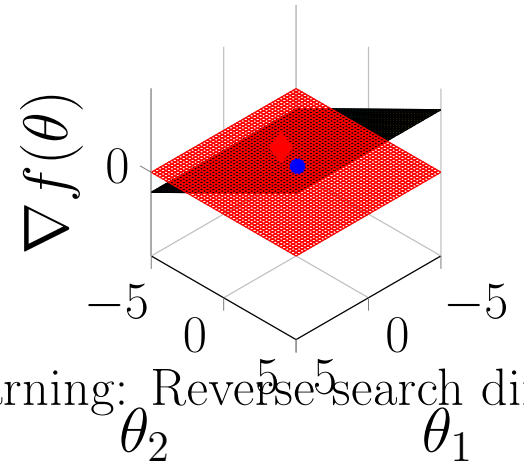
θ_{old} =-1.4456

λ =-1

$f(\theta_{old})$ =-4.0076

$\nabla f(\theta)$ =0.56002

$\nabla\nabla f(\theta)$ =0.49294



$\nabla\nabla f(\theta) > 0 \rightarrow$ Warning: Reverse search direction

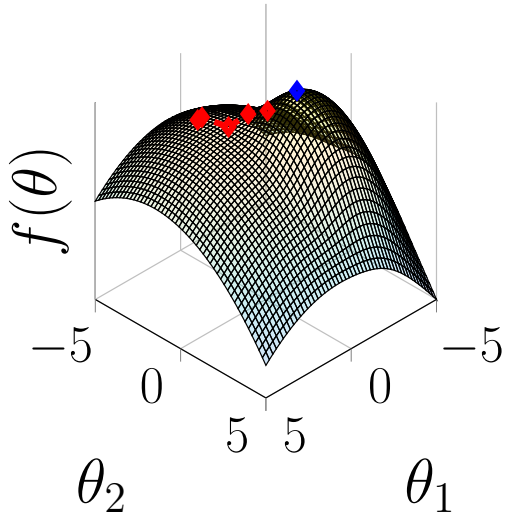
$$\theta_{new} = \theta_{old} + \lambda \frac{\nabla f(\theta)}{\nabla\nabla f(\theta)}$$

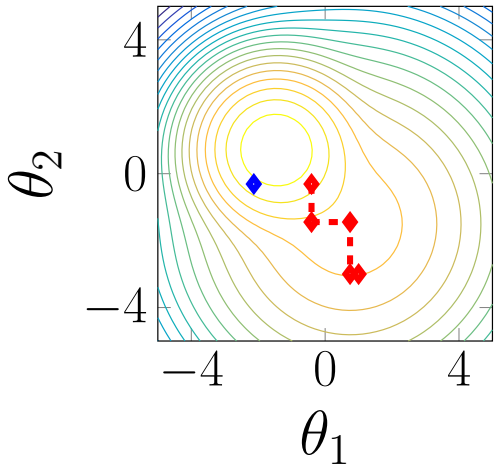
=-0.30949

$f(\theta_{new})$ =-3.1878

[CIV_ML/NR_demo_book_2D.m]

Example - Newton-Raphson 2D [.m]





loop: $\#5 \rightarrow \theta_1$

θ_{old} =-0.40726

λ =1

$f(\theta_{old})$ =-3.1878

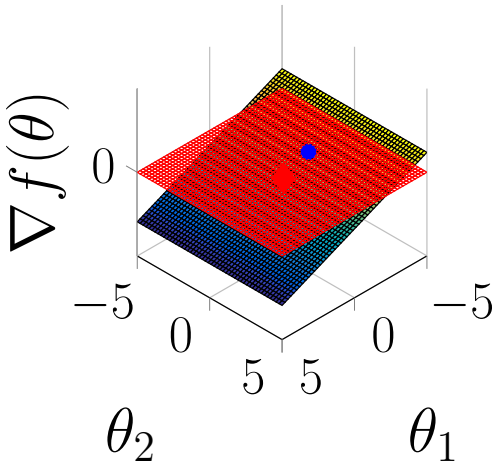
$\nabla f(\theta)$ =-0.71505

$\nabla\nabla f(\theta)$ =-0.41478

$\nabla\nabla f(\theta) < 0 \rightarrow$ OK

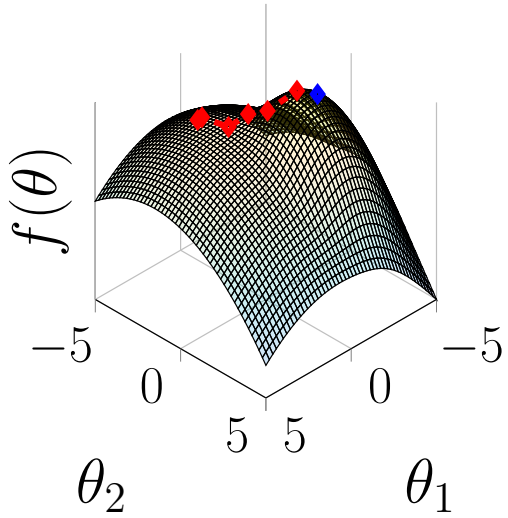
$$\theta_{new} = \theta_{old} - \lambda \frac{\nabla f(\theta)}{\nabla\nabla f(\theta)}$$
$$= \text{-2.1312}$$

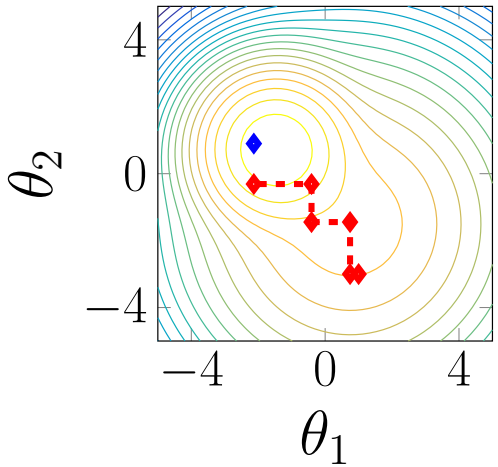
$f(\theta_{new})$ =-3.0086



[CIV_ML/NR_demo_book_2D.m]

Example - Newton-Raphson 2D [.m]





loop: $\#6 \rightarrow \theta_2$

θ_{old} =-0.30949

λ =1

$f(\theta_{old})$ =-3.0086

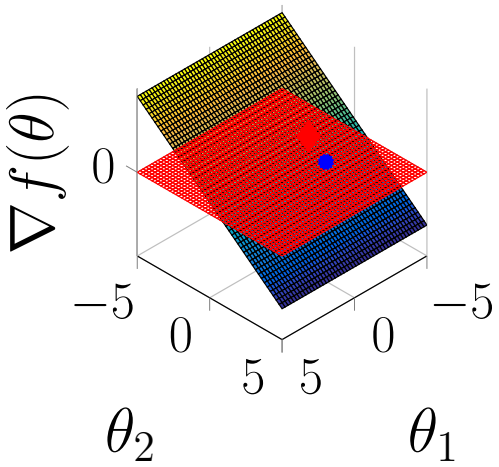
$\nabla f(\theta)$ =0.92963

$\nabla\nabla f(\theta)$ =-0.76961

$\nabla\nabla f(\theta) < 0 \rightarrow$ OK

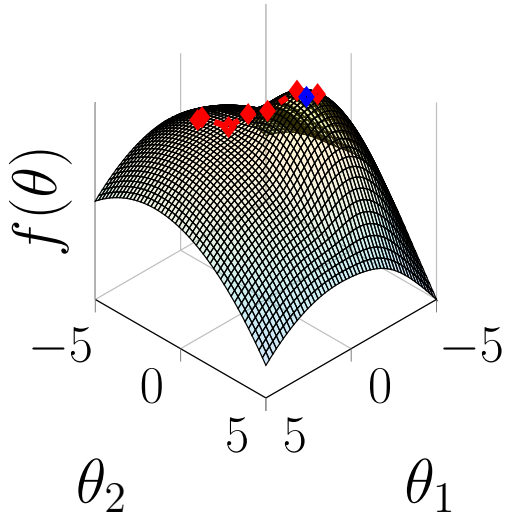
$$\theta_{new} = \theta_{old} - \lambda \frac{\nabla f(\theta)}{\nabla\nabla f(\theta)}$$
$$= 0.89844$$

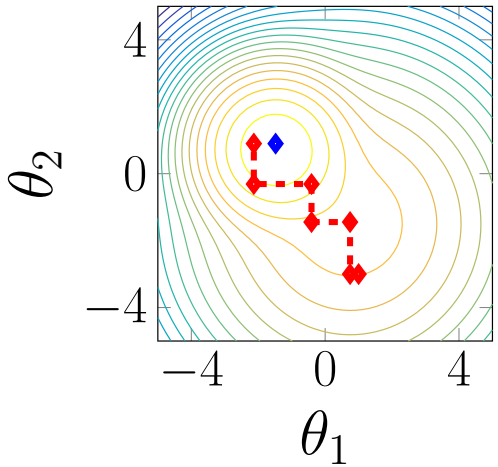
$f(\theta_{new})$ =-2.524



[CIV_ML/NR_demo_book_2D.m]

Example - Newton-Raphson 2D [.m]





loop: $\#7 \rightarrow \theta_1$

θ_{old} =-2.1312

λ =1

$f(\theta_{old})$ =-2.524

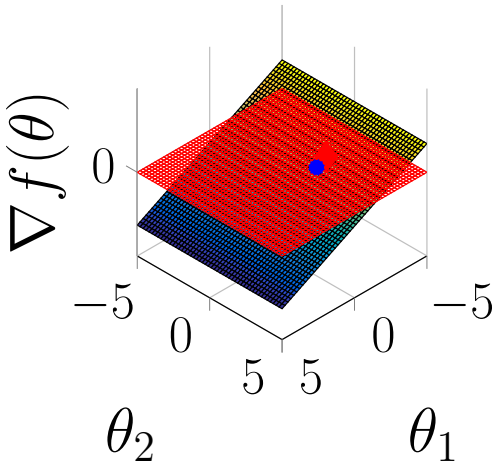
$\nabla f(\theta)$ =0.63424

$\nabla\nabla f(\theta)$ =-0.97433

$\nabla\nabla f(\theta) < 0 \rightarrow$ OK

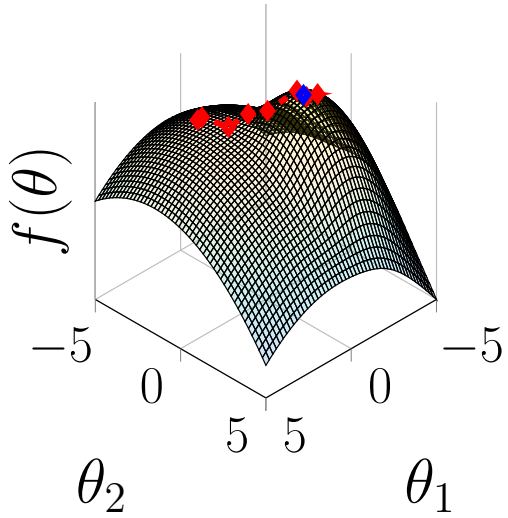
$$\theta_{new} = \theta_{old} - \lambda \frac{\nabla f(\theta)}{\nabla\nabla f(\theta)}$$
$$= \text{-}1.4802$$

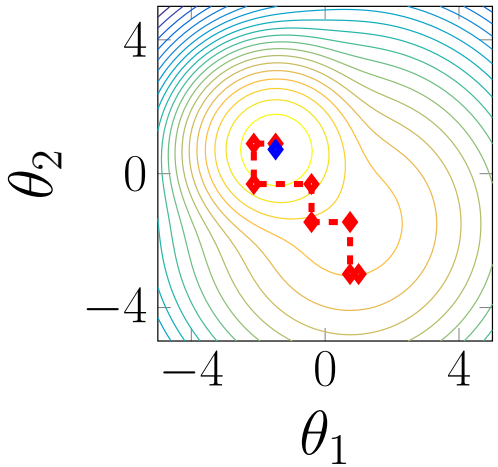
$f(\theta_{new})$ =-2.3168



[CIV_ML/NR_demo_book_2D.m]

Example - Newton-Raphson 2D [.m]





loop: $\#8 \rightarrow \theta_2$

θ_{old} =0.89844

λ =1

$f(\theta_{old})$ =-2.3168

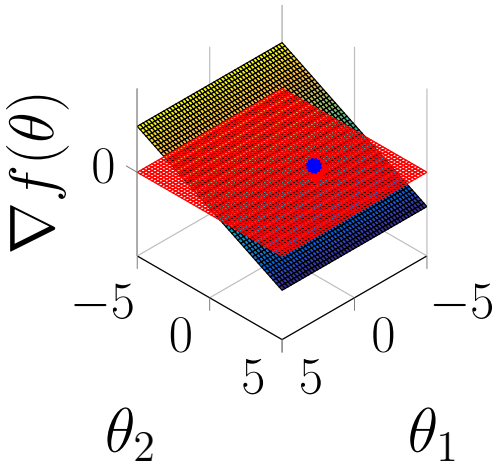
$\nabla f(\theta)$ =-0.1675

$\nabla\nabla f(\theta)$ =-0.96012

$\nabla\nabla f(\theta) < 0 \rightarrow$ OK

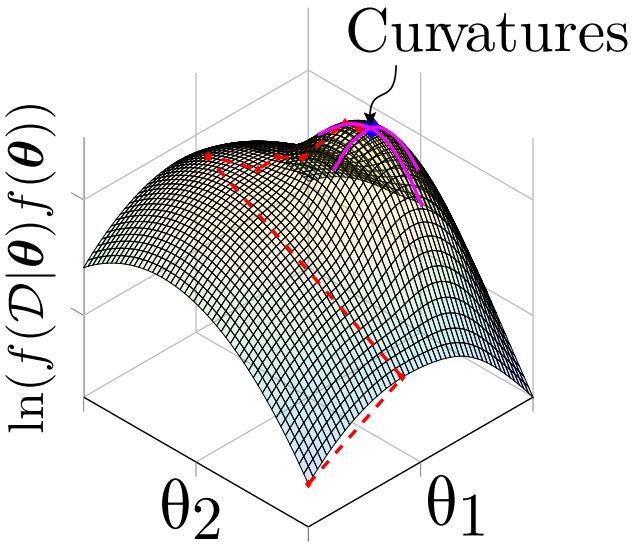
$$\theta_{new} = \theta_{old} - \lambda \frac{\nabla f(\theta)}{\nabla\nabla f(\theta)}$$
$$= 0.72399$$

$f(\theta_{new})$ =-2.3021



[CIV_ML/NR_demo_book_2D.m]

Laplace approximation – Introduction



We can estimate the **covariance matrix** of the **posterior PDF** by calculating the **curvature** of $\ln(\tilde{f}(\boldsymbol{\theta}))$ at MAP optimal values $\boldsymbol{\theta}^*$.

Laplace approximation – Mathematical formulation

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{P/2}(\det \boldsymbol{\Sigma}_{\mathbf{X}})^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{M}_{\mathbf{X}})^{\intercal}\boldsymbol{\Sigma}_{\mathbf{X}}^{-1}(\mathbf{x}-\mathbf{M}_{\mathbf{X}})\right]$$

$$\begin{aligned}
\ln f_{\mathbf{X}}(\mathbf{x}) &= -\tfrac{P}{2}\ln 2\pi - \tfrac{1}{2}\ln|\boldsymbol{\Sigma}_{\mathbf{X}}| - \tfrac{1}{2}(\mathbf{x}-\mathbf{M}_{\mathbf{X}})^{\intercal}\boldsymbol{\Sigma}_{\mathbf{X}}^{-1}(\mathbf{x}-\mathbf{M}_{\mathbf{X}}) \\
\frac{\partial \ln f_{\mathbf{X}}(\mathbf{x})}{\partial \mathbf{x}} &= -\tfrac{1}{2}\frac{\partial\left[(\mathbf{x}-\mathbf{M}_{\mathbf{X}})^{\intercal}\boldsymbol{\Sigma}_{\mathbf{X}}^{-1}(\mathbf{x}-\mathbf{M}_{\mathbf{X}})\right]}{\partial \mathbf{x}} \\
&= -\tfrac{1}{2}\left(2\boldsymbol{\Sigma}_{\mathbf{X}}^{-1}(\mathbf{x}-\mathbf{M}_{\mathbf{X}})\right) \\
&= -\boldsymbol{\Sigma}_{\mathbf{X}}^{-1}(\mathbf{x}-\mathbf{M}_{\mathbf{X}}) \\
\frac{\partial^2 \ln f_{\mathbf{X}}(\mathbf{x})}{\partial^2 \mathbf{x}} &= -\boldsymbol{\Sigma}_{\mathbf{X}}^{-1} \quad \textbf{(Laplace Approximation)}
\end{aligned}$$

Laplace approximation

Assuming that the posterior is approximately **Gaussian**

$$\underbrace{f(\boldsymbol{\theta}|\mathcal{D})}_{\text{posterior}} \approx \mathcal{N}(\boldsymbol{\theta}; \overbrace{\boldsymbol{\theta}^*}^{\text{MAP Estimate}}, \underbrace{\boldsymbol{\Sigma}_{\boldsymbol{\theta}^*}}_{\text{Laplace approximation}})$$

The **posterior covariance is approximated** by

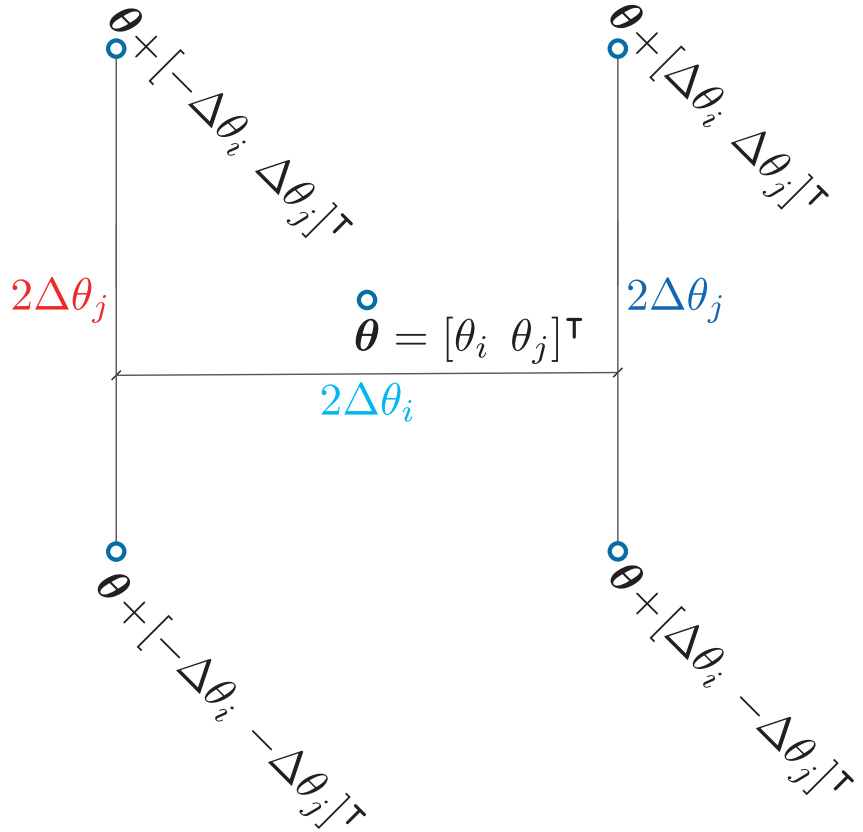
$$\boldsymbol{\Sigma}_{\boldsymbol{\theta}^*} = -\left(\frac{\partial^2 \ln \tilde{f}(\boldsymbol{\theta}^*)}{\partial^2 \boldsymbol{\theta}}\right)^{-1} = \mathbf{H}[-\overbrace{\ln \tilde{f}(\boldsymbol{\theta}^*)}^{\textbf{log-posterior}}]^{-1}$$

where $\mathbf{H}[\ln \tilde{f}(\boldsymbol{\theta}^*)]$ is the **Hessian matrix** evaluated at the most-likely parameter set $\boldsymbol{\theta}^*$

$$\left[\mathbf{H}[-\ln \tilde{f}(\boldsymbol{\theta}^*)]\right]_{ij} = -\frac{\partial^2 \ln \tilde{f}(\boldsymbol{\theta}^*)}{\partial \theta_i \partial \theta_j}$$

Hessian calculation [.m]

$$\Big[\mathbf{H}[\tilde{f}(\boldsymbol{\theta})]\Big]_{ij} \approx \frac{\frac{\partial \tilde{f}(\boldsymbol{\theta}+\Delta\theta)}{\partial \theta_j} - \frac{\partial \tilde{f}(\boldsymbol{\theta}-\Delta\theta)}{\partial \theta_j}}{2\Delta\theta_i}$$

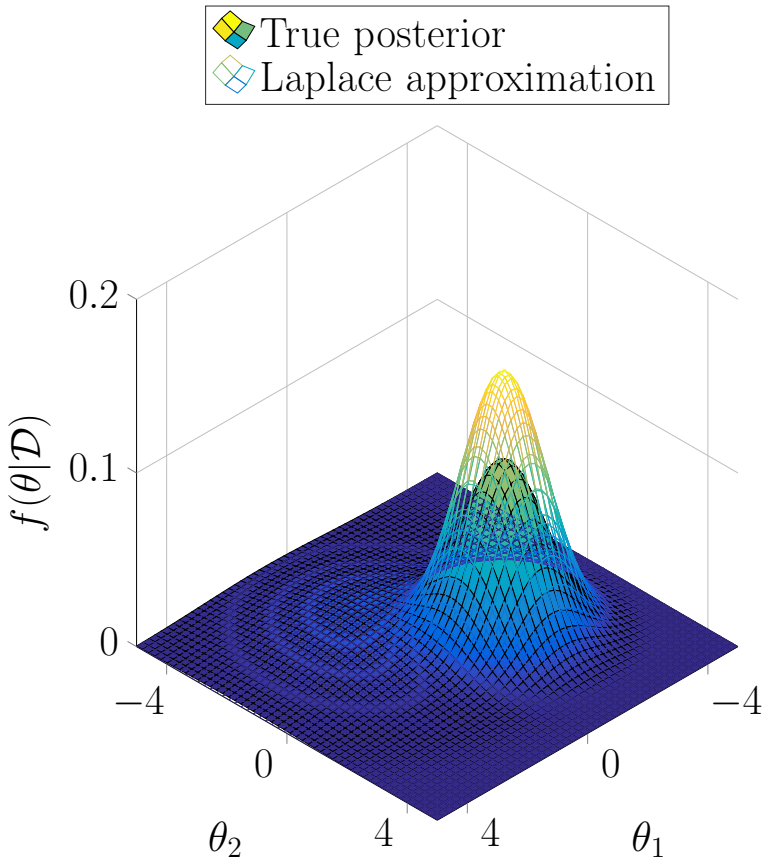


$$\frac{\partial \tilde{f}(\boldsymbol{\theta}+\Delta\theta)}{\partial \theta_j} = \frac{\tilde{f}(\boldsymbol{\theta}+\mathbb{I}(i)\Delta\theta_i+\mathbb{I}(j)\Delta\theta_j) - \tilde{f}(\boldsymbol{\theta}+\mathbb{I}(i)\Delta\theta_i-\mathbb{I}(j)\Delta\theta_j)}{2\Delta\theta_j}$$

$$\frac{\partial \tilde{f}(\boldsymbol{\theta}-\Delta\theta)}{\partial \theta_j} = \frac{\tilde{f}(\boldsymbol{\theta}-\mathbb{I}(i)\Delta\theta_i+\mathbb{I}(j)\Delta\theta_j) - \tilde{f}(\boldsymbol{\theta}-\mathbb{I}(i)\Delta\theta_i-\mathbb{I}(j)\Delta\theta_j)}{2\Delta\theta_j}$$

[CIV_ML/numerical_hessian.m]

Example - Newton-Raphson 2D (cont.) [.m]



$$\mathbf{H}[-\ln f(\boldsymbol{\theta}^*|\mathcal{D})] = \begin{bmatrix} 0.95 & 0.02 \\ 0.02 & 0.95 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\theta}^*} = \mathbf{H}[-\ln f(\boldsymbol{\theta}^*|\mathcal{D})]^{-1} = \begin{bmatrix} 1.05 & -0.02 \\ -0.02 & 1.05 \end{bmatrix}$$

The closer the posterior is to a Gaussian, the better is the Laplace approximation

[CIV_ML/Newton_Raphson_demo_2D.m]

MAP, MLE and Laplace v.s. Bayes

$$\boldsymbol{\theta}^* = \begin{cases} \arg\max_{\boldsymbol{\theta}} f(\mathcal{D}|\boldsymbol{\theta}) & \text{Maximum likelihood estimate (MLE)} \\ \arg\max_{\boldsymbol{\theta}} f(\mathcal{D}|\boldsymbol{\theta}) f(\boldsymbol{\theta}) & \text{Maximum a-posteriori (MAP)} \end{cases}$$

MLE and MAP are often employed instead of a Bayes.

Adequate for *unimodal* problems with large datasets.

Small dataset $\rightarrow$ **Laplace approximation** or **MCMC**

Parameter-space transformation

Both the **Newton-Raphson algorithm** and the **Normal proposal distribution** have issues when parameters are not defined over $\mathbb{R}$

Common cases:

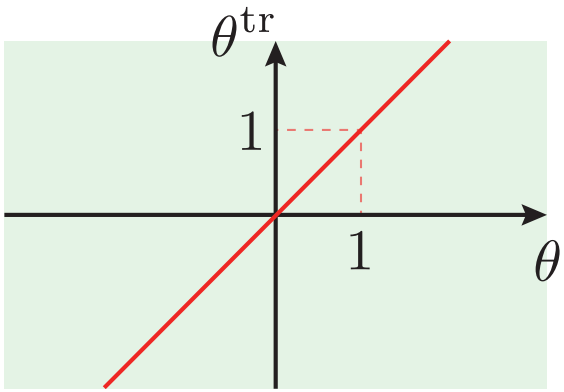
- Mean parameters: $\mu \in \mathbb{R}$
- Standard deviations: $\sigma \in \mathbb{R}^+$
- Correlation coefficients: $\rho \in (-1, 1)$
- Probability: $\Pr(X) \in (0, 1)$

$$\text{Solution} \rightarrow \theta^{\text{tr}} = g(\theta) \in \mathbb{R}$$

Parameter-space transformation – Common cases [.m]

$\theta \in \mathbb{R}$

$$\theta^{\text{tr}} = \theta$$

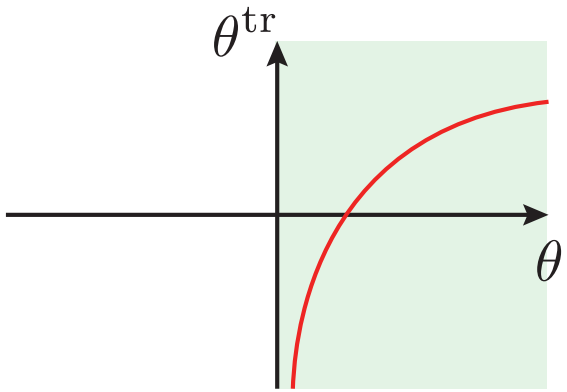


$$\frac{d\theta^{\text{tr}}}{d\theta} = \frac{d\theta}{d\theta^{\text{tr}}} = 1$$

$\theta \in \mathbb{R}^+$

$$\theta^{\text{tr}} = \ln(\theta)$$

$$\theta = e^{\theta^{\text{tr}}}$$

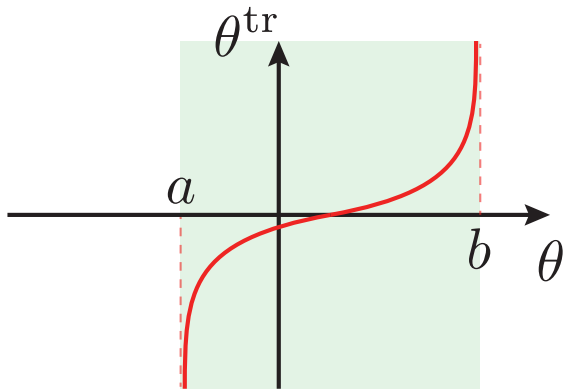


$$\frac{d\theta^{\text{tr}}}{d\theta} = \frac{1}{\theta}, \quad \frac{d\theta}{d\theta^{\text{tr}}} = e^{\theta^{\text{tr}}}$$

$\theta \in (a, b)$

$$\theta^{\text{tr}} = -\ln\left(\frac{b-a}{\theta-a} - 1\right)$$

$$\theta = \frac{b-a}{1+e^{-\theta^{\text{tr}}}} + a$$



$$\frac{d\theta}{d\theta^{\text{tr}}} = \frac{a-b}{(\theta-a)(\theta-b)}$$

$$\frac{d\theta^{\text{tr}}}{d\theta} = \frac{b-a}{(1+e^{-\theta^{\text{tr}}})^2} e^{-\theta^{\text{tr}}}$$

[CIV_ML/transformation_functions.m]

Parameter-space transformation – Derivatives

First and **second derivatives** can directly be computed in the **transformed space**

```
import numpy as np

# Index vector
I = lambda i : np.linspace(1,n,n) == i #nx1 indicator vector
# Gradient
grad_fct = lambda x, dx,i : (fx_TR(x+I(i)*dx)-fx_TR(x-I(i)*dx))/(2*dx)
# 1D Hessian
hessian_fct=lambda x,dx,i:(fx_TR(x+I(i)*dx)-2*fx_TR(x)+fx_TR(x-I(i)*dx))/dx**2
```

Parameter-space transformation – MLE

Calculations for obtaining the MLE using **Newton-Raphson** are perform in the **transformed space** in order to obtain $\boldsymbol{\theta}^{*TR}$;
At the end, transform back to the original space $\boldsymbol{\theta}^* = g^{-1}(\boldsymbol{\theta}^{*TR})$

```
x_MLE=TR_fct_inv(x_MLE_TR)
```

Laplace approximation: perform in the **original space** to estimate the **posterior covariance**

```
H = numerical_hessian(x_MLE,ln_fx)
S_laplace = -np.linalg.inv(H)
S_laplace = (S_laplace + S_laplace.transpose())/2
```

Parameter-space transformation – MCMC []

Proposal distribution: Defined in the transformed space $q(\boldsymbol{\theta}^{TR'}|\boldsymbol{\theta}^{TR}) = \mathcal{N}(\boldsymbol{\theta}^{TR'}; \boldsymbol{\theta}^{TR}, \gamma^2 \boldsymbol{\Sigma}^{TR}_{\boldsymbol{\theta}^*})$. $\boldsymbol{\Sigma}^{TR}_{\boldsymbol{\theta}^*}$ is computed using the Laplace approximation in the transformed space

```
H_TR = numerical_hessian(xMLE_TR ,ln_fxTR)
S_laplace_TR = -np.linalg.inv(H_TR)
```

Target probability: evaluated in the transformed space using the inverse transformation function

$$\tilde{f}(\boldsymbol{\theta}^{\text{tr}}) = \tilde{f}(g^{-1}(\boldsymbol{\theta}^{\text{tr}})) \cdot \prod_{i=1}^{\text{P}} \frac{1}{|\nabla g(\theta_i)|_{\theta_i^{\text{tr}}}}$$

```
det_J = lambda x_TR : prod(dTR_fct_inv(x_TR))
ft_TR = lambda x_TR : det_J(x_TR) @ fx(TR_fct_inv(x_TR))
```

Algorithm 6: Metropolis – Transformed space

initialize $g(\boldsymbol{\theta}) \rightarrow \boldsymbol{\theta}^{\mathrm{TR}},\ g^{-1}(\boldsymbol{\theta}^{\mathrm{TR}}) \rightarrow \boldsymbol{\theta},\ \boldsymbol{\theta}_0$

compute MAP: $\boldsymbol{\theta}^{*TR}$ (Newton-Raphson)

compute $\boldsymbol{\Sigma}^{\mathrm{TR}}_{\boldsymbol{\theta}^*} = \mathbf{H}[-\ln \tilde{f}(\boldsymbol{\theta}^{*TR})]^{-1}$ (Laplace approx.)

for $s = 1, 2, \cdots$ **do**

  define $\boldsymbol{\theta}^{\mathrm{TR}} = g(\boldsymbol{\theta}_s)$

  sample $\boldsymbol{\theta}^{'TR} \sim \mathcal{N}(\boldsymbol{\theta}^{\mathrm{TR}}, \gamma^2 \boldsymbol{\Sigma}^{\mathrm{TR}}_{\boldsymbol{\theta}^*})$

  compute $\alpha = \dfrac{\tilde{f}(g^{-1}(\boldsymbol{\theta}^{'TR}))}{\tilde{f}(g^{-1}(\boldsymbol{\theta}^{\mathrm{TR}}))} \cdot \prod_{i=1}^{\mathrm{P}} \dfrac{|\nabla g(\theta_i)|_{\boldsymbol{\theta}^{\mathrm{TR}}}}{|\nabla g(\theta'_i)|_{\boldsymbol{\theta}^{'TR}}}$

  compute $r = \min(1, \alpha)$

  sample $u \sim \mathcal{U}(0, 1)$

  if $u \leq r$ **then**

    $\boldsymbol{\theta}_{s+1} = g^{-1}(\boldsymbol{\theta}^{'TR})$

  else

    $\boldsymbol{\theta}_{s+1} = \boldsymbol{\theta}_s$

Example – Characterizing concrete resistance

$$\overbrace{f(\mu_R, \sigma_R|\mathcal{D})}^{\text{Posterior PDF}} = \frac{\overbrace{f(Y_1 = y_1, \cdots, Y_N = y_N|\mu_R, \sigma_R)}^{\text{Likelihood}} \cdot \overbrace{f(\mu_R, \sigma_R)}^{\text{Prior PDF}}}{\underbrace{f(Y_1 = y_1, \cdots, Y_N = y_N)}_{\text{Normalization cte}}}$$

Measurement errors: $y = r + v,\ v : V \sim \mathcal{N}(v; 0, 0.01^2)$ MPa

Prior: $f(\mu_R, \sigma_R) = 1/\sigma_R$ (non-informative prior)

Likelihood: $f(y|\boldsymbol{\theta}) = \mathcal{N}(y; \mu_R, \sigma_R^2 + \underbrace{\sigma_V^2}_{=0.01^2})$

$$f(Y_1 = y_1, \cdots, Y_D = y_D|\mu_R, \sigma_R) = \underbrace{\prod_{i=1}^{D} \frac{1}{\sqrt{2\pi}\sqrt{\sigma_R^2 + \sigma_V^2}} \exp\left[-\frac{1}{2}\left(\frac{y_i - \mu_R}{\sqrt{\sigma_R^2 + \sigma_V^2}}\right)^2\right]}_{\text{Normal PDF}}$$

Example – Transformation functions & NR results

Target PDF

$$\tilde{f}(\mu_R, \sigma_R) = f(\mathcal{D}|\mu_R, \sigma_R) \cdot f(\mu_R, \sigma_R)$$

Transformation functions

$$\begin{aligned} \boldsymbol{\theta}^{TR} &= [\theta_1^{TR}, \theta_2^{TR}] &&= [\mu_R, \ln(\sigma_R)] \\ \boldsymbol{\theta} &= [\mu_R, \sigma_R] &&= [\theta_1^{TR}, \exp(\theta_2^{TR})]. \end{aligned}$$

Newton-Raphson results

$$\begin{aligned} \boldsymbol{\theta}^* &= [43.0, 0.6] \\ \boldsymbol{\theta}^{*TR} &= [43.0, 1.8] \\ \mathbf{H}[-\ln \tilde{f}(\boldsymbol{\theta}^{*TR})]^{-1} &= \begin{bmatrix} 1.11 & 0 \\ 0 & 0.17 \end{bmatrix} \end{aligned}$$

Example – Computing with MCMC samples

$$\mathbb{E}[\boldsymbol{\theta}|\mathcal{D}] \approx \frac{1}{\mathrm{S}}\sum_{s=1}^{\mathrm{S}}\boldsymbol{\theta}_s = [42.9, 3.8]$$

$$\mathrm{Cov}[\theta_i, \theta_j|\mathcal{D}] \approx \frac{1}{\mathrm{S}-1}\sum_{s=1}^{\mathrm{S}}(\theta_{i,s} - \mathbb{E}[\theta_i|\mathcal{D}])(\theta_{j,s} - \mathbb{E}[\theta_j|\mathcal{D}])$$

$$= \begin{bmatrix} 8.0 & 2.5 \\ 2.5 & 20.6 \end{bmatrix}$$

$$\mathbb{E}[R|\mathcal{D}] \approx \frac{1}{\mathrm{S}}\sum_{s=1}^{\mathrm{S}} r_s = 42.9$$

$$\mathrm{Var}[R|\mathcal{D}] \approx \frac{1}{\mathrm{S}-1}\sum_{s=1}^{\mathrm{S}}(r_s - \mathbb{E}[R|\mathcal{D}])^2 = 40.5$$

Example - Results visualization [.m]

$$\mathcal{D} = \{43.3, 40.4, 44.8\}$$

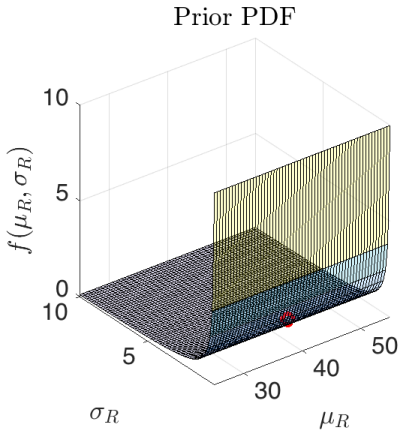
$$\mathsf{s} = 10^4$$

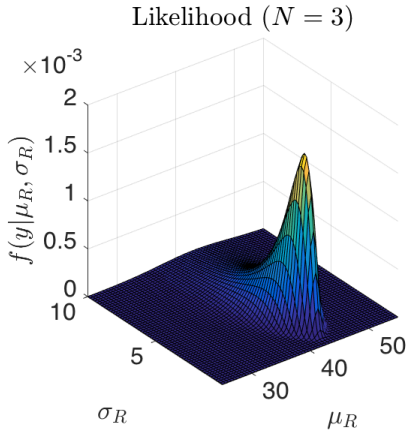
$$\mathsf{c} = 4$$

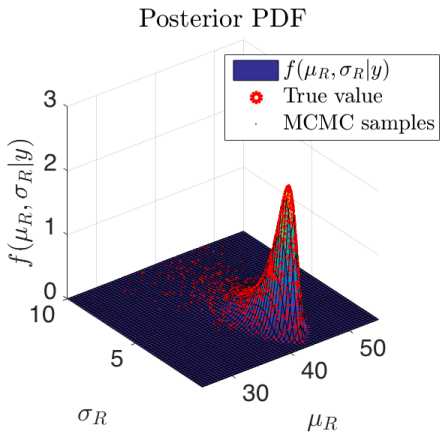
$$\gamma^2 = 1.44$$

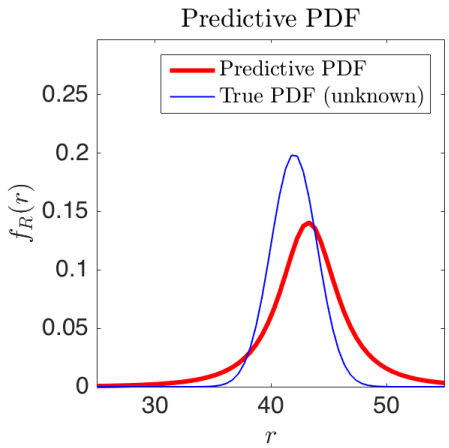
$$\beta = 0.29$$

$$\hat{R} = [1.0009, 1.0017]$$









[CIV_ML/Exemples/MCMC_Bayesian_updating.m]

Summary

Markov Hypothesis:
Given the present, the future is independent of the past

$$p(x_{t+1}|x_t) \equiv p(x_{t+1}|x_{1:t})$$

MCMC: Markov Chain Monte Carlo
Draw samples from any unnormalized PDF $\rightarrow$ posterior

Target distribution: $\tilde{f}(\boldsymbol{\theta}) = f(\mathcal{D}|\boldsymbol{\theta})f(\boldsymbol{\theta})$

Proposal distribution: $q(\boldsymbol{\theta}'|\boldsymbol{\theta})$

Posterior PDF mean:

$$\mathbb{E}[\boldsymbol{\theta}|\mathcal{D}] \approx \frac{1}{\mathrm{S}} \sum_{s=1}^{\mathrm{S}} \boldsymbol{\theta}_s$$

Posterior PDF covariance:

$$\mathrm{Cov}[\theta_i, \theta_j|\mathcal{D}] \approx \frac{1}{\mathrm{S}-1} \sum_{s=1}^{\mathrm{S}} (\theta_{i,s} - \mathbb{E}[\theta_i|\mathcal{D}])(\theta_{j,s} - \mathbb{E}[\theta_j|\mathcal{D}])$$

Burn-in phase:
Samples taken before reaching the stationary distribution must be discarded

Convergence :

- Compare the variance within and between chains using the **estimated potential scale reduction**≈ 1
- Check **acceptance rate**: 1D$\rightarrow$ $\beta \approx$40%, $\geq$5D$\rightarrow$ $\beta \approx$25%

Proposal tuning:

$$q(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}, \gamma^2 \boldsymbol{\Sigma}_{\boldsymbol{\theta}^*}), \quad \gamma^2 = \frac{2.4^2}{\mathrm{P}}, \quad (\mathrm{P}: \text{nb. param})$$

$\boldsymbol{\Sigma}_{\boldsymbol{\theta}^*}$ is estimated using the **Laplace Approximation** calculated for the **maximum a posteriori**-value (MAP), $\boldsymbol{\theta}^*$ (Newton Raphson + Laplace approximation)

Transformation:
Both the gradient-based optimization and the Normal proposal distribution have issues when parameters are not defined over $\mathbb{R}$