Markov chain Monte Carlo methods in atmospheric remote sensing

Johanna Tamminen
johanna.tamminen@fmi.fi

ESA Summer School on
Earth System Monitoring and Modeling
July 30 – Aug 11, 2012, Frascati

July, 31, 2012

Markov chain Monte Carlo method (with applications to atmospheric remote sensing)

Contents of this lecture

- Nonlinear inverse problems
- Introduction to Markov chain Monte Carlo
- Implementing MCMC in practice: adaptive Metropolis algorithm
- Examples of applying MCMC to atmospheric remote sensing

Solving nonlinear problems

- Posterior distribution

$$\pi(x) = p(x \mid y) = \frac{p_{\text{lh}}(y \mid x) p_{\text{pr}}(x)}{\int p_{\text{lh}}(y \mid x) p_{\text{pr}}(x) dx}.$$

can be evaluated pointwise up to the normalizing constant

$$\pi(x) \propto p_{\text{lh}}(y \mid x) p_{\text{pr}}(x).$$

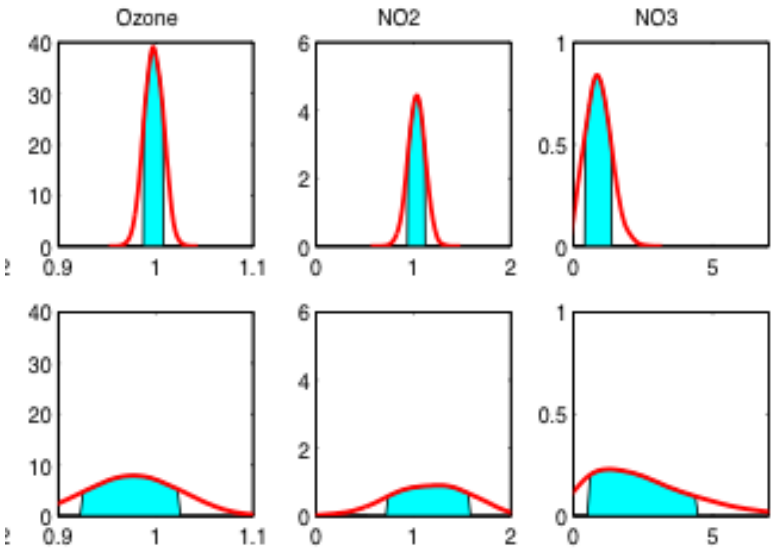
- MAP estimate gives an estimate of 'most probable' value.

- Note notation from now on: we denote with π both the target (posterior) distribution and its pdf.

- What we typically also need is:
  - expectation, mean
  - covariance matrix
  - probability regions (e.g., 90%)
  - how probably $x \in A$?

- In general case these require integration with respect to the posterior distribution:

$$\mathbb{E}(f(x)) = \int f(x)\pi(x)dx$$



Examples of (scaled) probablity regions in GOMOS retrievals

As discussed yesterday, typical ways of solving non-linear problems in practice include

- Linearizing the problem
- Assuming 'close to Gaussian' structure at the estimate
- Iterative finding of the solution

Exploring the posterior vs. optimizing

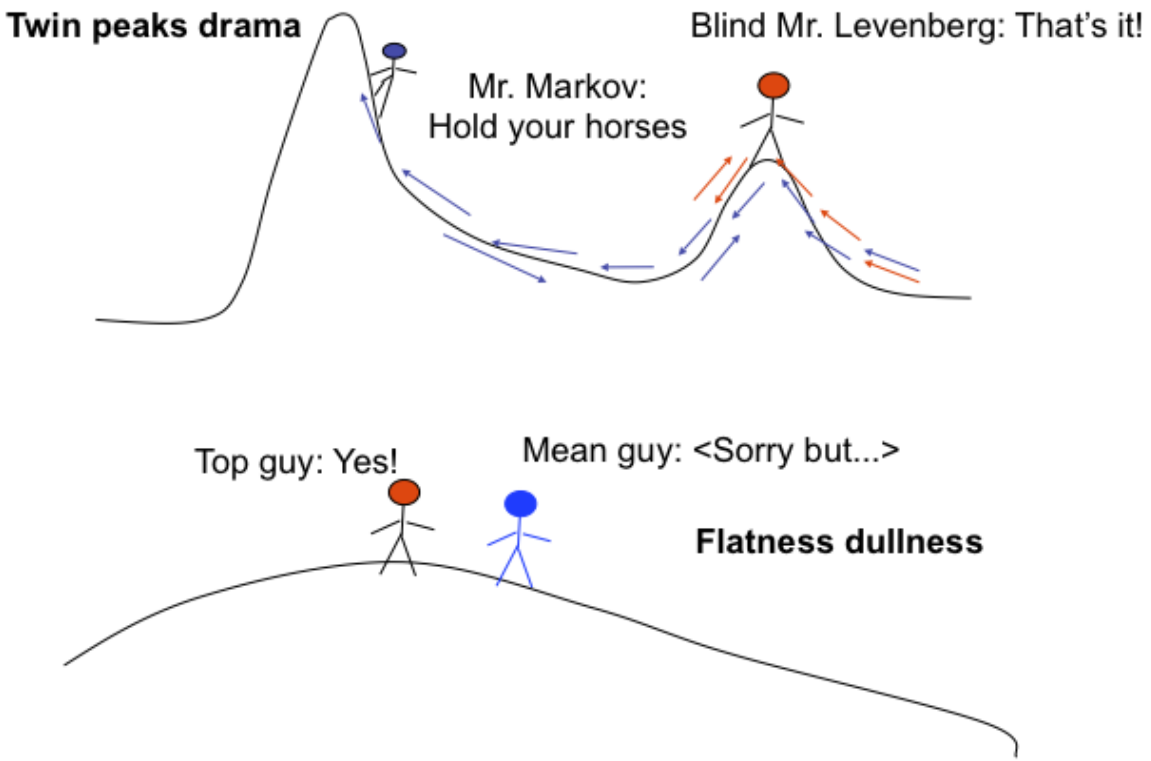
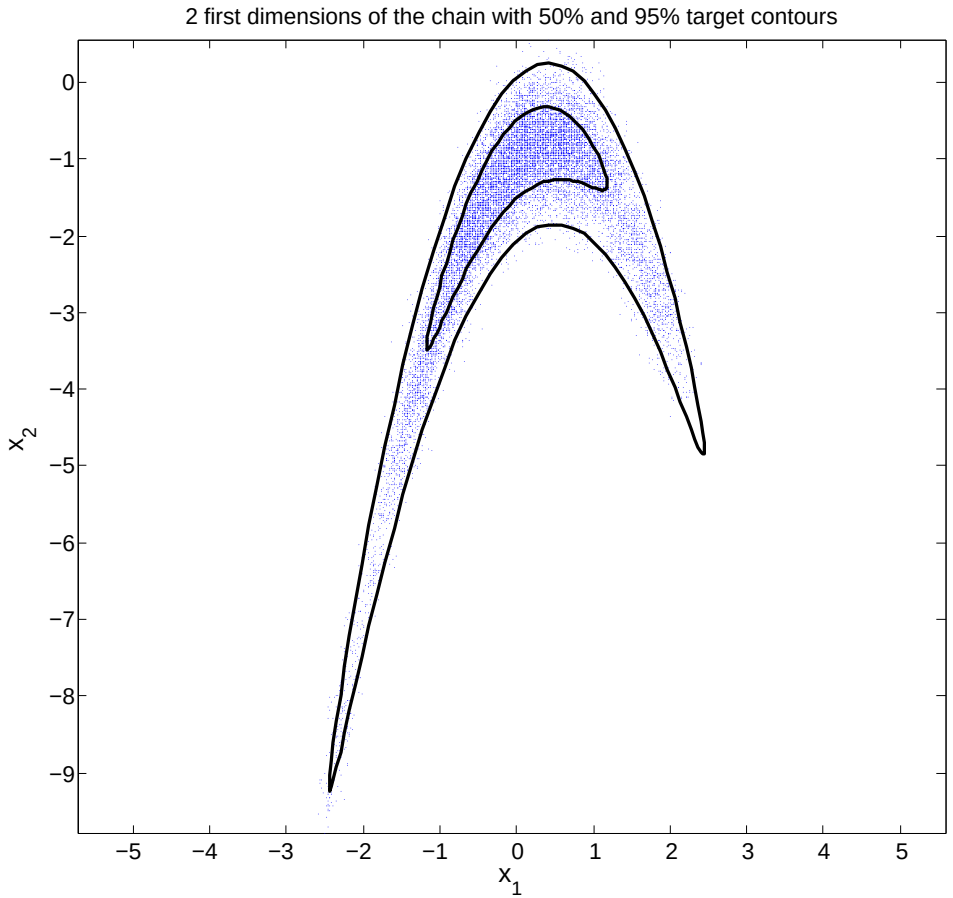


Figure by Erkki Kyrölä (FMI)

What if my posterior distribution is like this?



- In this case Gaussian posterior distribution (mean and covariance) is not a good approximation.

Computation of the posterior distribution

- Systematic exploration of the posterior distribution
  - Typically very time consuming and practically impossible in large dimensons
- Monte Carlo integration
  - Sample randomly $X_1, \ldots, X_N$ from the posterior distribution
  - When samples are independent and identically distributed **law of large numbers** holds

$$\mathbb{E}(f(x)) = \int f(x)\pi(x)dx \approx \frac{1}{N}\sum_{t=1}^{N} f(X_t)$$

  - To study the posterior distribution it requires normally computation of $p(y) = \int p_{\mathrm{lh}}(y \mid x)p_{\mathrm{pr}}(x)dx$
  - it means that the whole space of possible x values needs to be sampled
  - Many variations exist how to make the sampling
- One of the methods is
  Markov chain Monte Carlo (MCMC) - topic of this lecture

Law of Large numbers (LLN)

- LLN says that sample average converges towards the expectation

$$\frac{1}{N}\sum_{t=1}^{N} X_t \longrightarrow \mathbb{E}(X) \text{ when } N \longrightarrow \infty$$

- When LLN holds, we can approximate expectation with sample mean:

$$\mathbb{E}(f(x)) = \int f(x)\pi(x)dx \approx \frac{1}{N}\sum_{t=1}^{N} f(X_t)$$

Markov chain Monte Carlo technique

- MCMC is based on sampling points X_t which are 'cleverly' selected

- In MCMC the sampled points form a Markov chain

Let us have a chain $X = X_1, X_2, \ldots$ now the chain is **Markov chain** if the probability to jump to X_{t+1} depends only on previous point X_t:

$$P(X_{t+1} \mid X_1, X_2, \ldots X_t) = P(X_{t+1} \mid X_t)$$

- The sampled points are thus not independent like in Monte Carlo sampling.

- However, LLN can be shown to hold if the points are selected in a proper way.

Ergodicity of Markov chains

Assuming that the Markov chain:

- has the desired distribution $\pi(x)$ as the stationary distribution (reversibility)
  - Detailed balance equation holds:

$$\pi(X_i)\ P(X_j \mid X_i) = \pi(X_j)\ P(X_i \mid X_j)$$

- samples properly all parts of the state space:
  - is aperiodic - don't repeat itself
  - is irreducible - all places can be reached

then the chain is **ergodic** and the LLN holds

$$\frac{1}{N}\sum_{t=1}^{N} f(X_t) \longrightarrow \mathbb{E}_\pi(f(X)) \text{ when } N \longrightarrow \infty$$

- Standard MCMC algorithms are created so that LLN holds for 'reasonable' target distributions.

MCMC in practice:

Several variants of MCMC exist but most common are:

- Metropolis algorithm *(Metropolis Et. Al. 1953, J.Chem. Phys.)*.
- Metropolis-Hastings algorithm *(Hastings 1970, Biometrika)*.
- Gibbs sampling*(Gelfand and Smith 1990, J. Amer.Stat.Assoc.)*.

Metropolis-Hastings algorithm

At each iteration step t

Step 1) (Proposal step) Sample $Z \sim q(\cdot \mid X_t)$
here Z is a **candidate** point and $q(\cdot \mid X_t)$ is a selected **proposal** distribution

Step 2) (Acceptance step) Accept the candidate point by using the acceptance probability:

$$\alpha = \min\left(1, \frac{\pi(Z)q(X_t \mid Z)}{\pi(X_t)q(Z \mid X_t)}\right) = \min\left(1, \frac{p_{\text{lh}}(y \mid Z)\ p_{\text{pr}}(Z)\ q(X_t \mid Z)}{p_{\text{lh}}(y \mid X_t)\ p_{\text{pr}}(X_t)\ q(Z \mid X_t)}\right)$$

Put $X_{t+1} = Z$ if accepted
$X_{t+1} = X_t$ if rejected.

Metropolis algorithm

When the proposal distribution q is **symmetric** (eg. Gaussian $\mathrm{N}(X_t, C_q)$ centered at the present point X_t):

Step 2) (Acceptance step) Acceptance probability is now simply:

$$\alpha = \min\left(1, \frac{\pi(Z)}{\pi(X_t)}\right) = \min\left(1, \frac{p_{\mathrm{lh}}(y \mid Z)p_{\mathrm{pr}}(Z)}{p_{\mathrm{lh}}(y \mid X_t)p_{\mathrm{pr}}(X_t)}\right)$$

Put $X_{t+1} = Z$ if accepted
$X_{t+1} = X_t$ if rejected.

Note:

- Acceptance probability is selected so that detailed balance $(\pi(x)P(y \mid x)) = \pi(y)P(x \mid y))$ holds. This ensures that the chain converges towards the correct target distribution π
- When the proposal distribution is correctly chosen the LLN assures that the expectation can be approximated by using empirical mean (under mild conditions for π).
- Theoretically, any proposal q having the same support as π should work.
- Usually the proposal distribution q is selected so that it is easy to sample candidate points from it. (eg. Gaussian, fixed spheres or regions around the present point)
- In practice, some proposals q are better than the others.

Note (cont):

- Computing the accptance probability of a posterior distribution requires evaluating

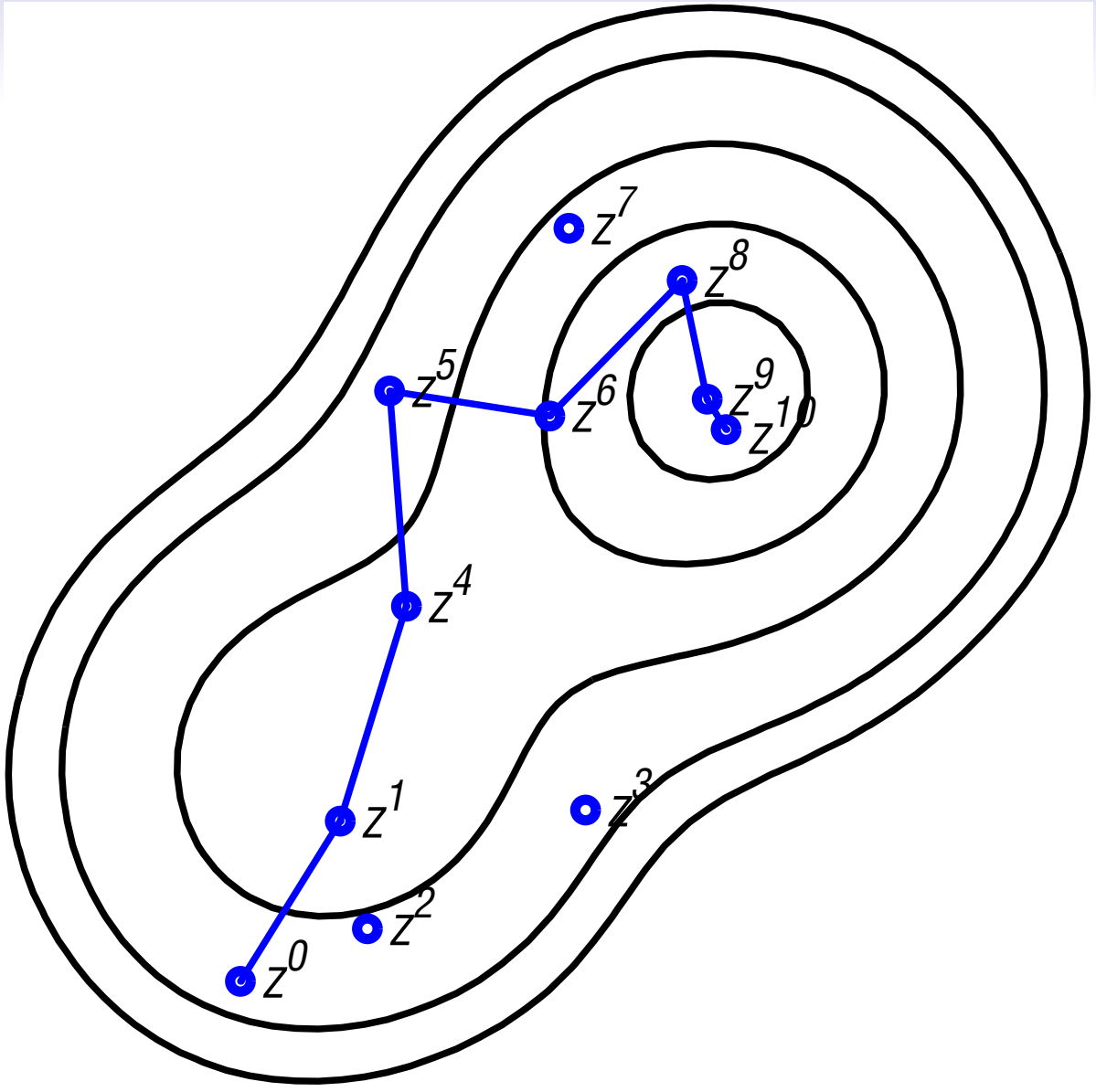
$$\frac{\pi(Z)}{\pi(X_t)} = \frac{p_{\mathrm{lh}}(y \mid Z)\ p_{\mathrm{pr}}(Z)}{p_{\mathrm{lh}}(y \mid X_t)\ p_{\mathrm{pr}}(X_t)}$$

  thus the scaling factor

$$p(y) = \int p_{\mathrm{lh}}(y \mid x) p_{\mathrm{pr}}(x) dx$$

  needs not to be evaluated!

- Candidate point is always accepted if more probable.
- Also non-zero probability to accept less probable values.



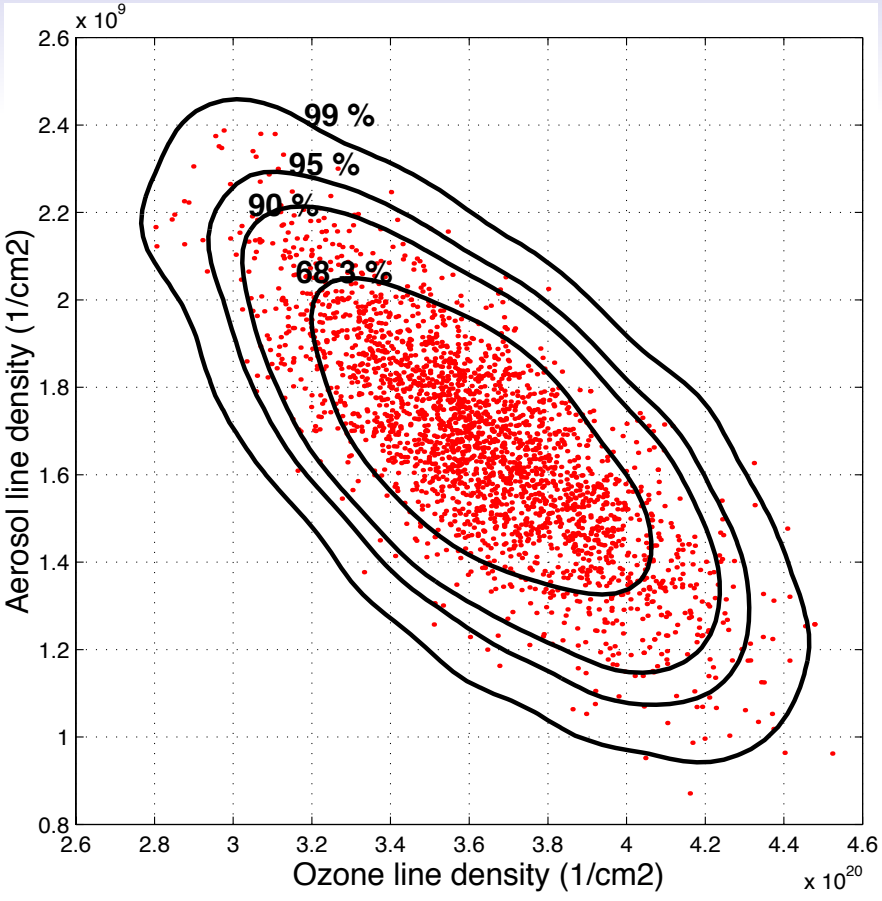
Sampled chain is $z^0, z^1, z^1, z^1, z^4, z^5, z^6, z^6, z^8, z^9, z^{10}, \ldots$

MCMC demo

Posterior distribution by
Monte Carlo sampling

MCMC demo

Posterior distribution by
Monte Carlo sampling:
kernel estimate



Example of marginal posterior distribution of ozone and aerosol horizontal column densities in GOMOS retrieval. The dots indicate sampled points and their distribution indicates the posterior distribution of the solution. The contour curves, computed from sampled points by using Gaussian kernel estimates correspond to 68.3, 90, 95, and 99% probability regions.

MCMC in practice

Even though the basic algorithm is very simple and, in theory, LLN holds for eg. simple Gaussian proposal distribution, few things need to be characterized:

- Select a proposal distribution.

This is an important topic and we'll discuss this more in the following slides

- Select starting point.

In theory, any point will work. In practice it is worth selecting a reasonable point, e.g., the solution of an iterative algorithm.

- How long chain is needed?

This is difficult to say and depends on the target distribution and its dimension. Important to be careful and get hands on experience on this. Several convergence diagnostics also exist and can be sometimes useful.

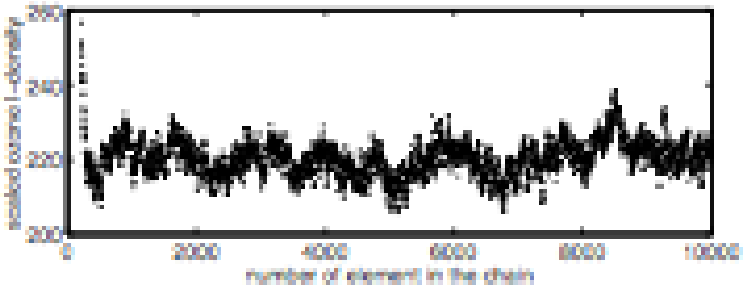
MCMC in practice cont.

- Burn-in

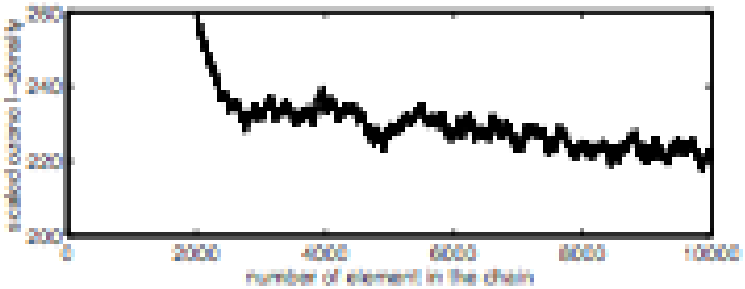
Typically the beginning of the chain is rejected as a 'burn in' period.

Proposal distribution

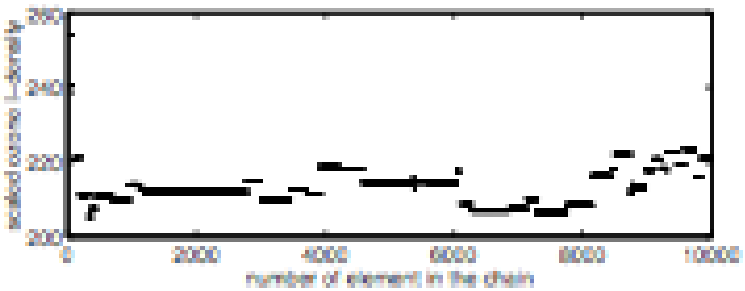
- Efficient sampling requires 'good' propsal distribution



Nice sampling: good proposal



Slowly converging chain: Too small proposal



Chain does not move properly: Too large proposal

Good proposal

- Good balance between accepted and rejected points.

This is typically monitored during sampling.

- In Metropolis algorithm with Gaussian target and Gaussian proposal the optimal acceptance ratio is

$$a_r = \frac{\text{n of accepted points}}{\text{n of proposed points}} \approx 0.234$$

when the dimension of the target is roughly larger than 6. Optimal acceptance ratio is slightly larger in smaller dimensions (eg. in 2D case 0.35). (*See Gelman et al, Efficient Metropolis Jumping Rules, Bayesian Statistics 5, 1996*)

- If the target is close to Gaussian acceptance ratio 0.2–0.3 should be ok.
- If the distribution is strongly non-Gaussian smaller acceptance ratio is obtained even with 'good' proposal.

- For sampling Gaussian distribution $N(\mu, C)$ The optimal Gaussian proposal in Metropolis algorithm is:

$$q(\cdot) \sim N(\cdot, {c_d}^2\ C)$$

where

$$c_d = \frac{2.4}{\sqrt{d}}$$

(*See Gelman et al, Efficient Metropolis Jumping Rules, Bayesian Statistics 5, 1996*)

Applying Metropolis algorithm for GOMOS spectral inversion

Metropolis algorithm:

- Easy technique for sampling points $X_0, \ldots, X_n$ from the target distribution:

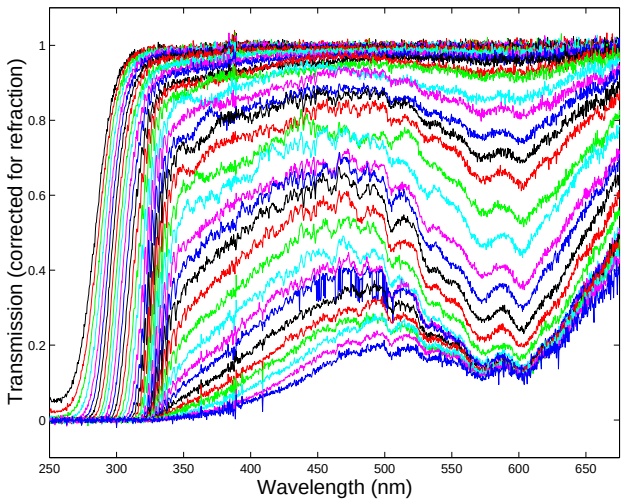
$$E_\pi(f(x)) \approx \frac{1}{n} \sum_{i=1}^{n} f(X_i)$$

In practice:

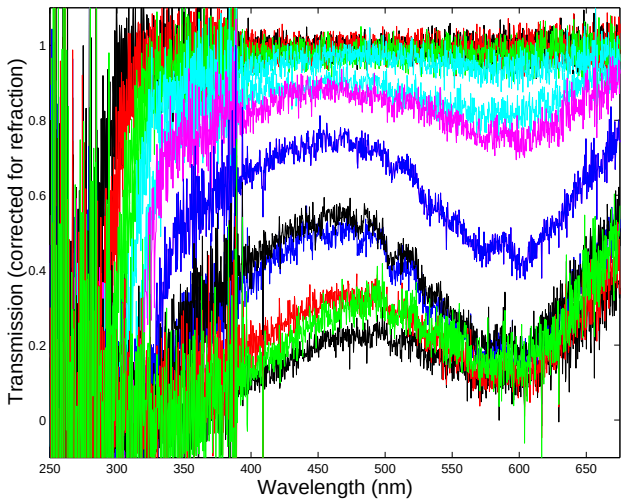
- GOMOS case: the amount of data is enormous, the posterior distributions vary a lot, depending on atmospheric conditions and the noise level of the data $\longrightarrow$ no fixed proposal distribution works effectively for MCMC.
- Manual tuning of proposal distribution impossibe.
- Adaptive and automatic MCMC necessary!

Data/noise

Data using bright star

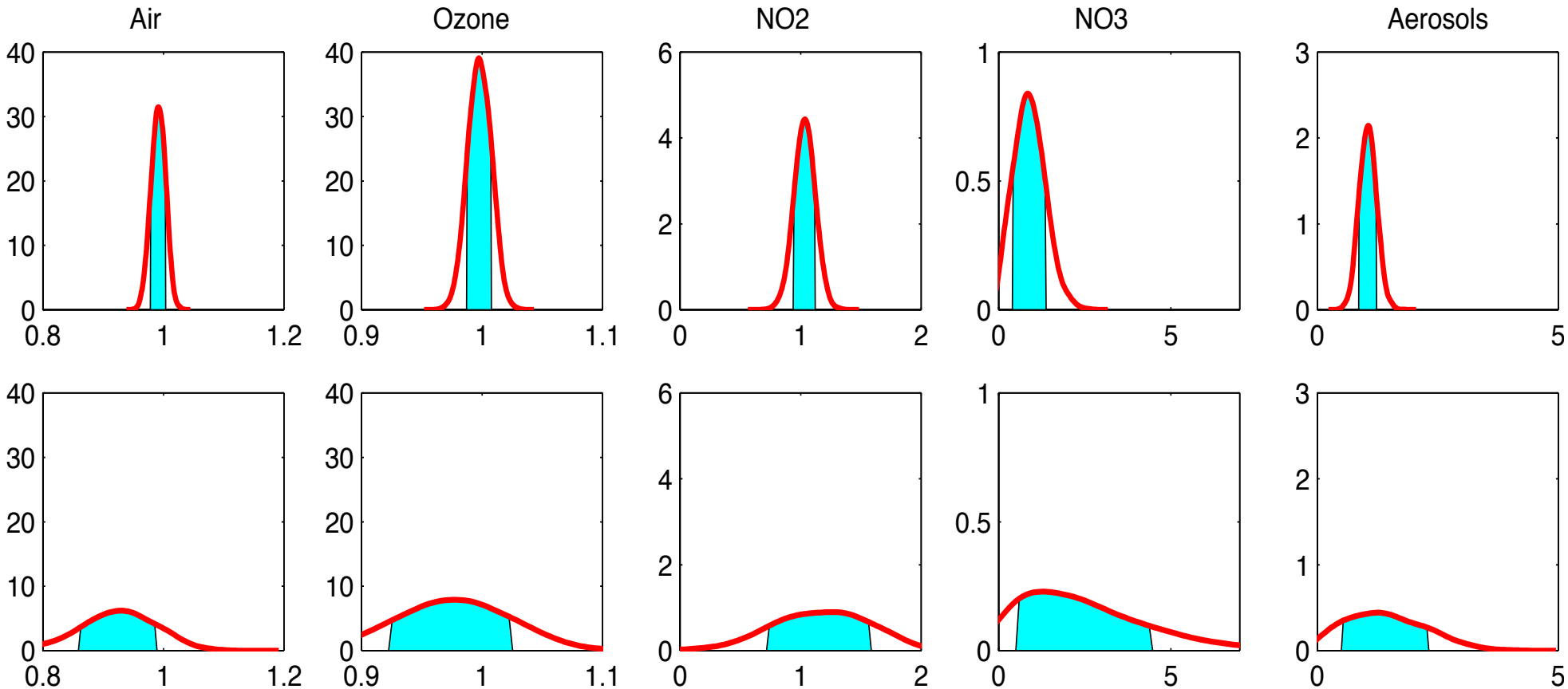


Data using dim star



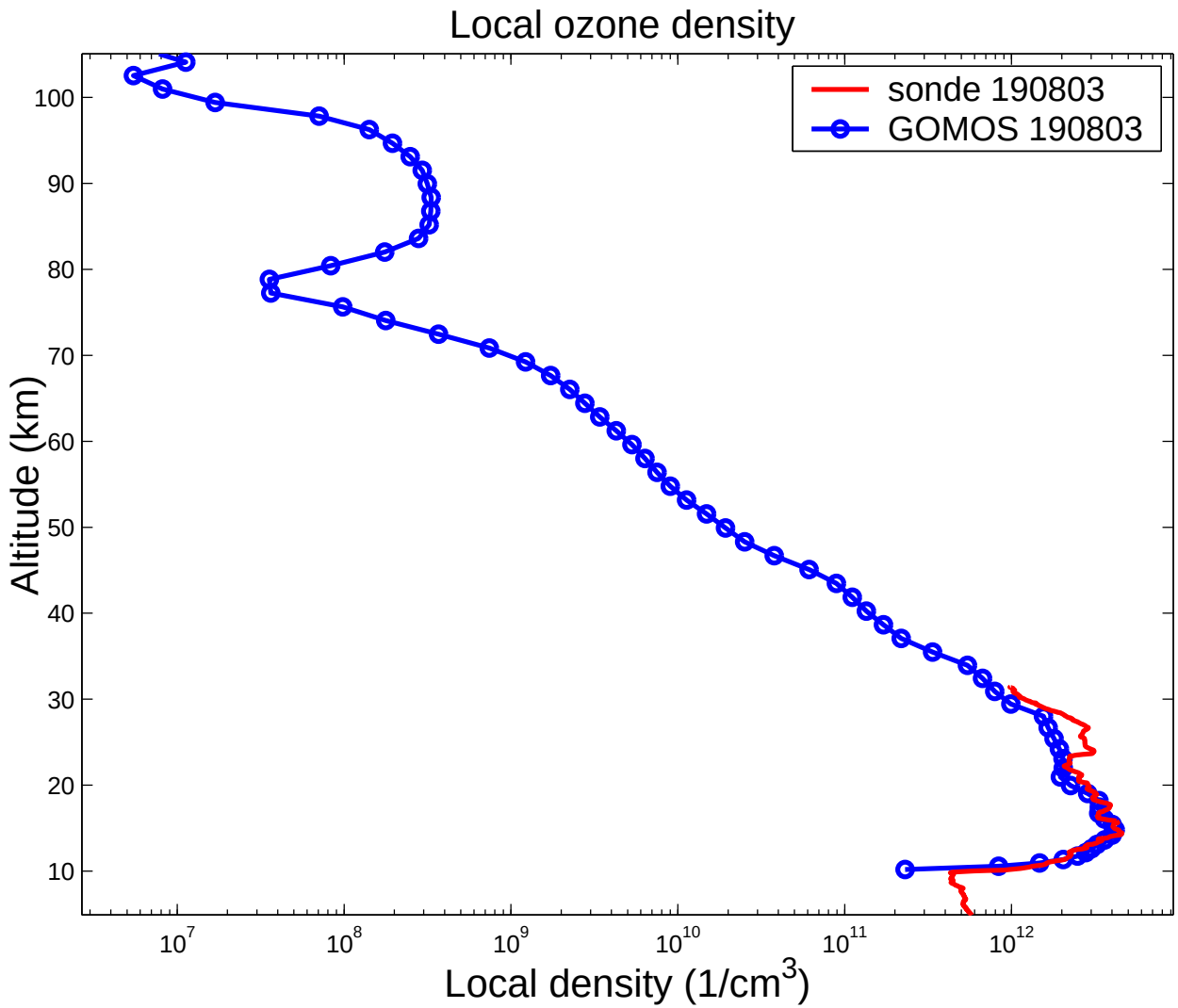
- Strong variability in signal-to-noise ratio depending on the type of star.

Variability in posterior distributions for different cases



Estimated 1σ (68.3%) probability limits for each gas. The top row corresponds to a typical star (magnitude 2) case and the bottom row to a dim star (magnitude 4). The absolute values have been scaled so that the true value corresponds to value 1 and the y axis represents the identifiability.

Several decades of variability depending on the altitudes



Adaptive MCMC

- Idea: optimize proposal distribution during sampling. Learn from past chain.
- One needs to be careful since ergodicity can be lost.
- One of the commonly used approaches is **Adaptive Metropolis algorithm** (*See Haario et al, An adaptive Metropolis algorithm, Bernoulli, 2001*)
- Several variants of this algorithm also exist but we concentrate here mainly on the original version.

Adaptive Metropolis (AM) algorithm - the idea

- AM is simply basic Metropolis algorithm with Gaussian proposal distribution which is adapted during sampling
- In AM the covariance matrix of the proposal distribution is updated based on earlier sampled points.

We define

$$C_t = \begin{cases} C_0, & t \leq t_0 \\ c_d \mathrm{Cov}(X_0, \ldots, X_{t-1}) + c_d \varepsilon, & t > t_0. \end{cases}$$

Here $c_d = 2.4/\sqrt{d}$ as earlier.
The additional term $\varepsilon > 0$ ensures that the distribution does not become singular.

- Empirical covariance matrix determined by points $x_0, \ldots, x_t \in R^d$:

$$\mathrm{Cov}(x_0, \ldots, x_t) = \frac{1}{t}\left(\sum_{i=0}^{t} x_i x_i^T - (t+1)\overline{x}_t\, \overline{x}_t^T\right),$$

where the mean is

$$\overline{x}_t = \frac{1}{t+1}\sum_{i=0}^{t} x_i.$$

Implementing AM algorithm: Recursion formulas for updates

- Recursive formula for the mean $\overline{x}_t$:

$$\overline{x}_{t+1} = \frac{t+1}{t+2}\overline{x}_t + \frac{1}{t+2}x_{t+1}$$

- Recursive formula for the covariance C_t:

$$C_{t+1} = \frac{t-1}{t}C_t + \frac{c_d}{t}\left(t\overline{X}_{t-1}\overline{X}_{t-1}^T - (t+1)\overline{X}_t\overline{X}_t^T + X_tX_t^T\right).$$

Adaptive Metropolis algorithm

At each iteration step $t > t_0$

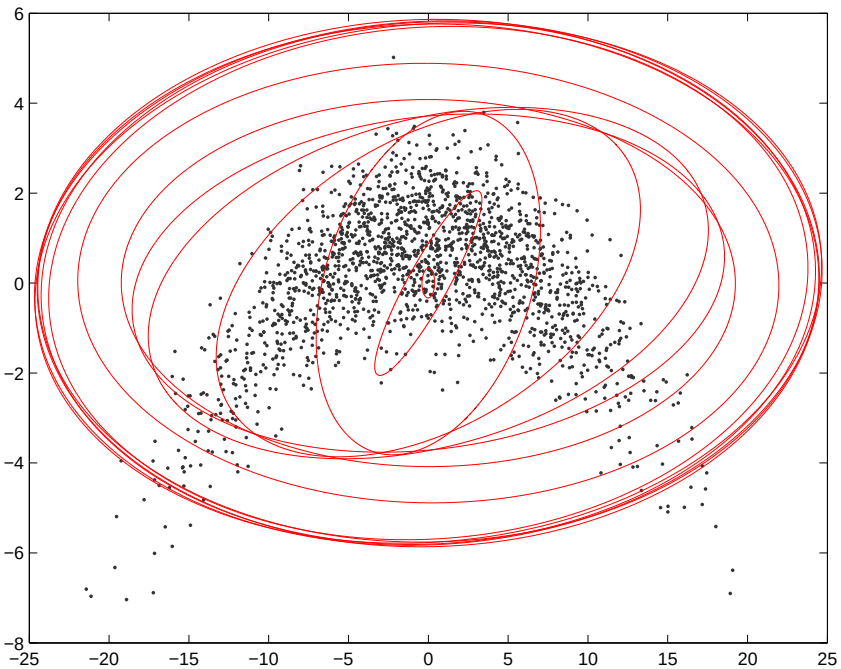
Step 1) (Proposal step) Sample candidate from proposal distribution $\mathrm{N}(X_{t-1}, C_t)$

Step 2) (Acceptance step) Accept using Metropolis algorithm

Step 3) Update proposal distribution by using the recursion formulas.

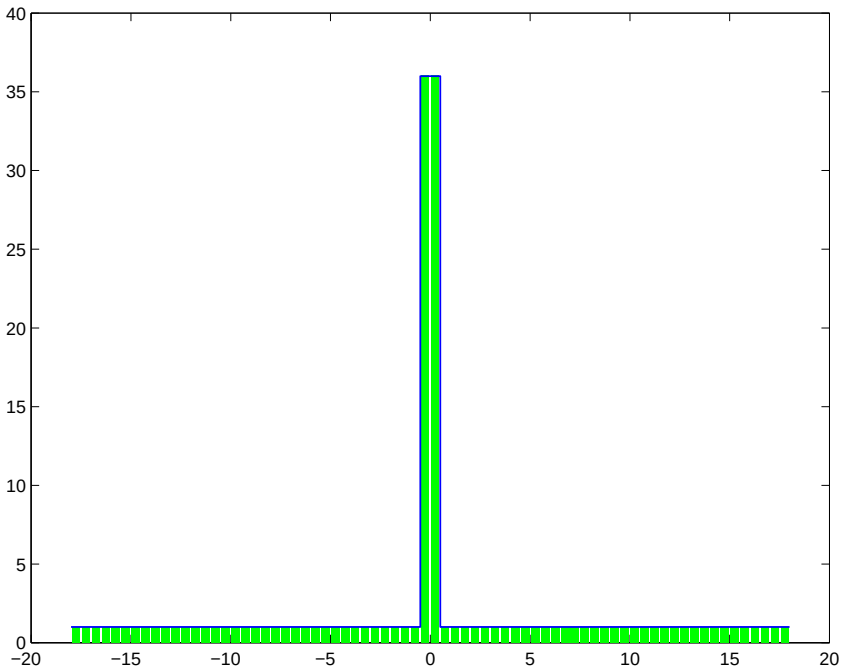
Adaptive Metropolis algorithm cont.

- Proposal distributions which adapt suitable size and/or shape by using the information of the previous states in the chain
- Gaussian proposal distribution $N(X_t, C_t)$, where the covariance C_t depends on time.
- The chain is not a Markov chain but the right ergodic properties can be proved (ref. Haario et al, 2001).



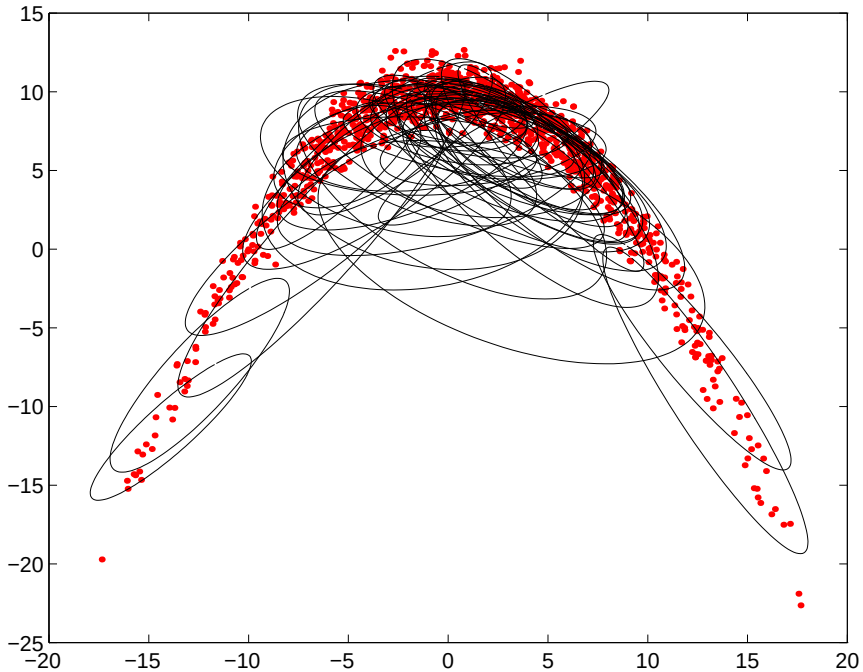
- The 'idea' of the proof is that along the time the proposal distribution converges and the adaptation diminishes.

AM: Adaptive Metropolis algorithm



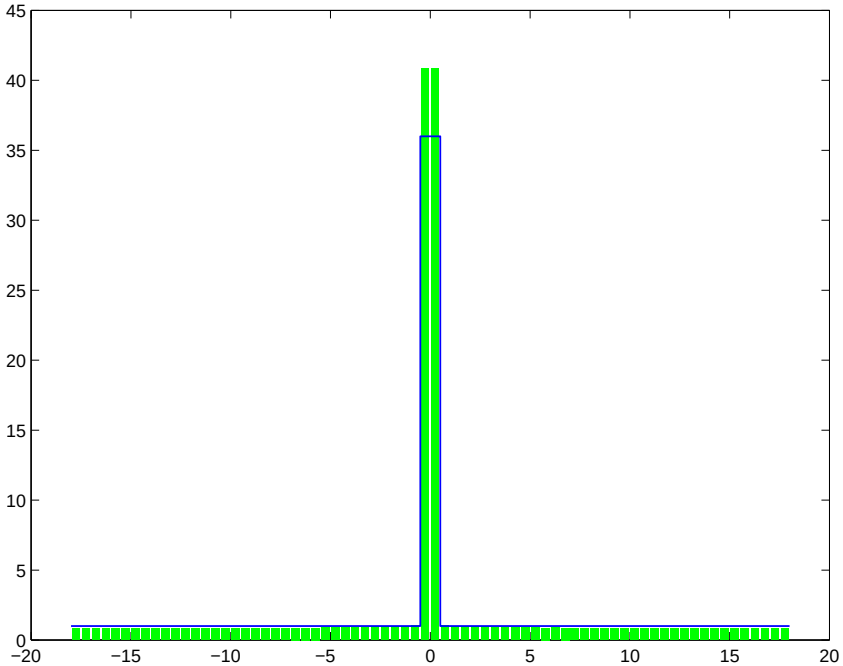
- AM: Convergence towards right target distribution. Green distribution target. Solid line AM.

AP: Adaptive proposal algorithm (Haario et al, 1999)



- AP: adapting proposal distribution during sampling based on only **most recently sampled points**.

AP: Adaptive proposal algorithm



- AP: not right covergence properties if adaptation is continued throughout the sampling.
- If adaptation stopped after the burn-in period then AP algorithm is OK. In practice, AP works well for reasonably 'Gaussian' targets also when adaptation is continued.

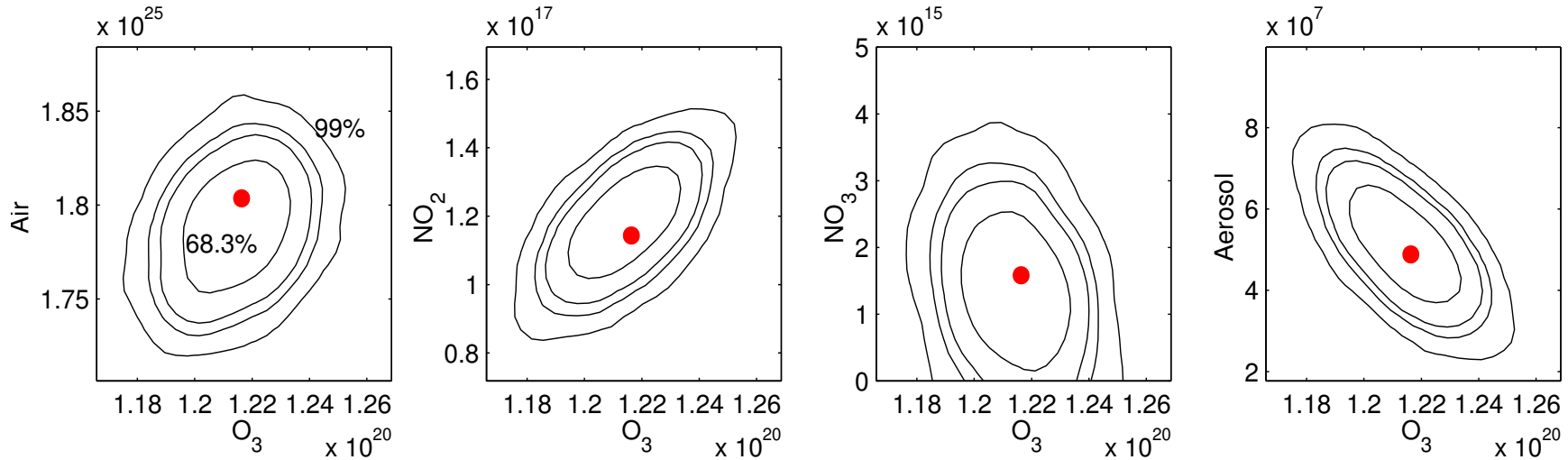
GOMOS: Sampling posterior distribution using MCMC

Motivation:

- Visualizaiton of posterior distribution.
- Freedom in defining a priori: Could be learn more by using more complicated priors than Gaussian? Positivity, for example?
- Freedom in defining noise: Could be learn more by using different measurement noise structures than Gaussian.
- Could expectation be more robust estimate than MAP estimate?
- Can we study the identifiability in non-linear problems?
- Validation of the operative algorithm:
  - Is the posterior distribution really Gaussian - how well does the covariance matrix approximate the posterior distribution?
  - Does the posterior distribution include 'local maximums' - does the iterative Levenberg-Marquardt algorithm get stuck on these points?

Implementation - visualization of posterior distributions

- Automatic and adaptive algorithm essential for implementing MCMC in real GOMOS application. In practice AM (and AP, an earlier version of it) used for studying non-linear GOMOS spectral inversion.



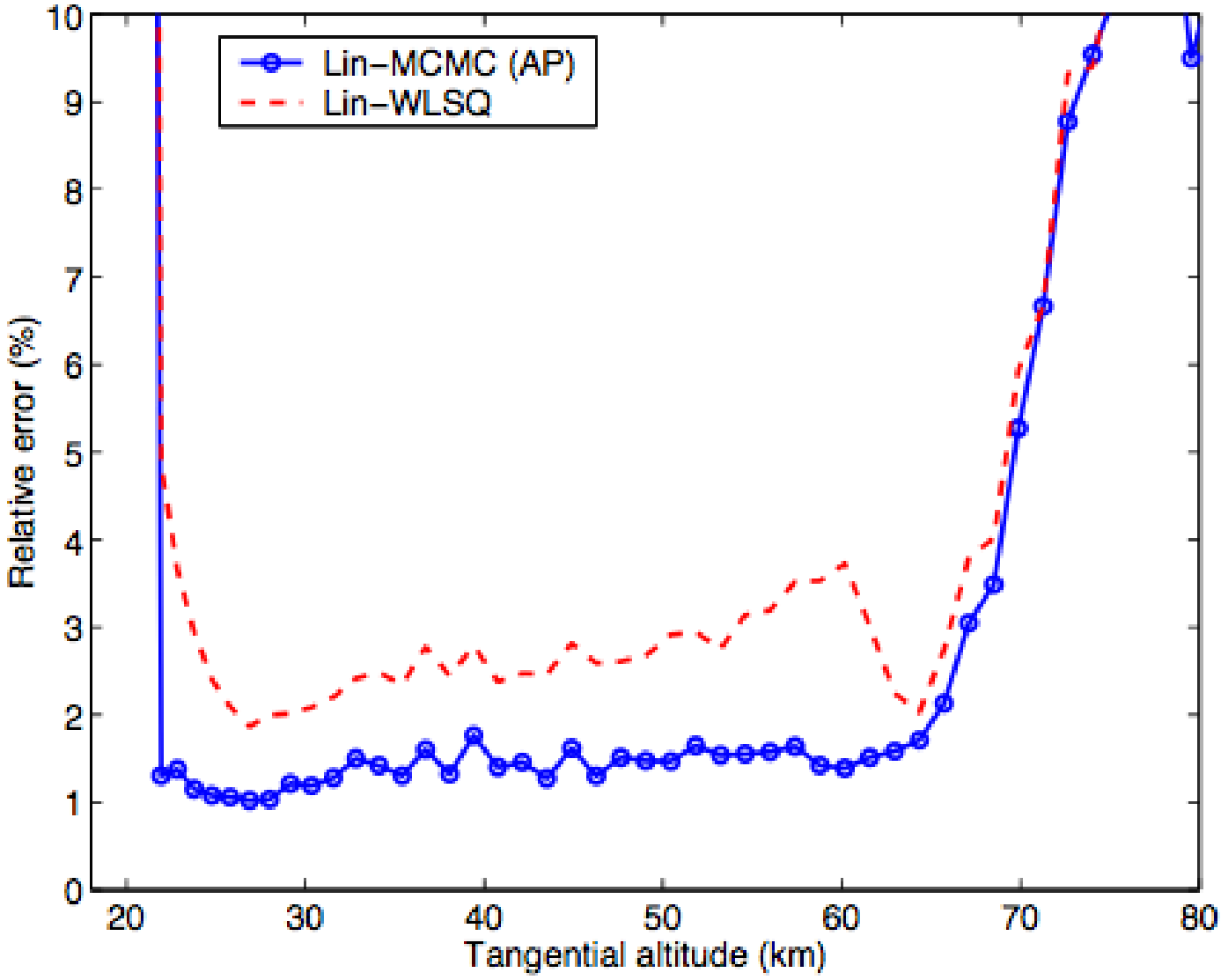
Simulated case: Two-dimensional kernel estimates of the marginal posterior distributions of the gases at 30 km. On *x* axis we present the ozone values and on the *y* axis from left: air, NO_2, NO_3, and aerosol. The contours refer to 68.3, 90, 95, and 99% probability regions. The true values used in the forward model are denoted with dots.

Example of implementing other than Gaussian noise structure

- By linearizing the GOMOS spectral inversion the noise becomes non-Gaussian
- The likelihood function after the linearization is:

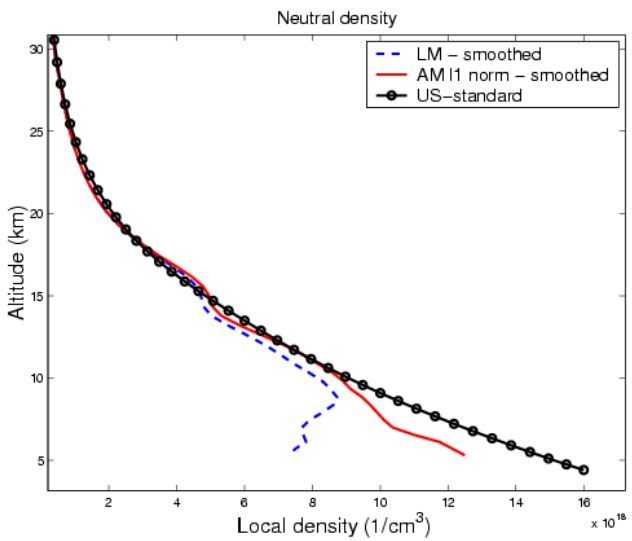
$$p(\widetilde{y} \mid x) = \frac{1}{(2\pi)^{m/2})|C|^{\frac{1}{2}}}$$
$$\cdot \exp\left\{-\frac{1}{2}\sum_{i=1}^{m}\left(\frac{f(x;\lambda_i) - y(\lambda_i)}{\sigma_i^2}\right)^2 - \widetilde{y}\right\}.$$

- This can easlily be implemented in MCMC



- Simulated example: correct error characterization (non-Gaussian) improves results.

Noise statistics: robust inversion

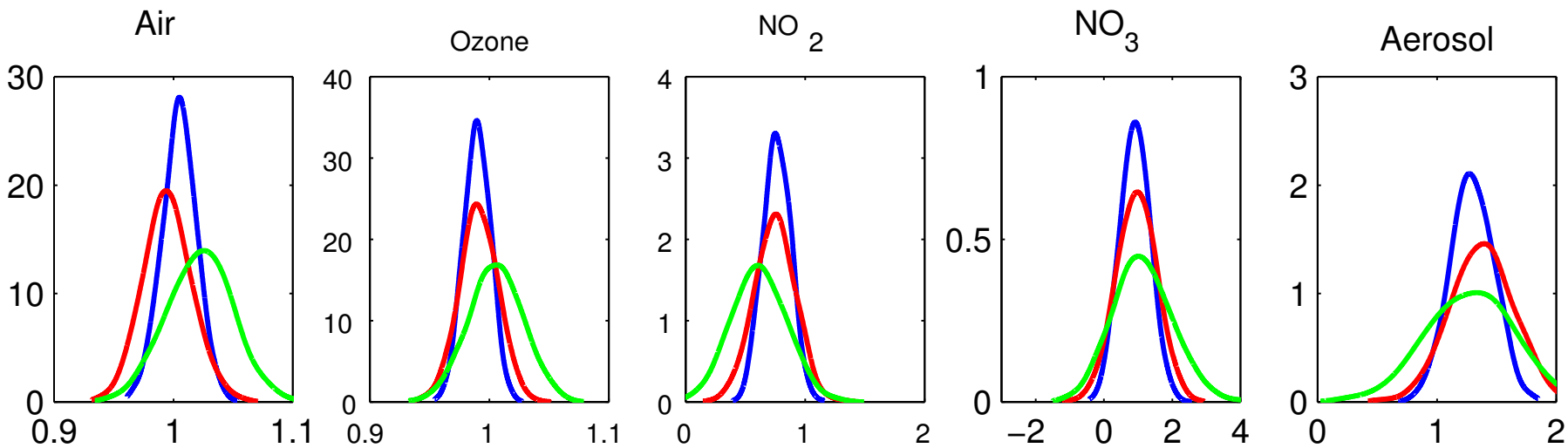


- Uncertainties larger at lower part.
- Robust ℓ_1 norm likehlihood:

$$p(y \mid x) \propto \exp\left(- \sum_{i=1}^{m} \frac{\sqrt{2}\,\left| f_i(x) - y_i \right|}{\sigma_i} \right).$$

- Example: robust inversion improves results at low altitudes

Examples - identifiability in GOMOS retrieval



Marginal posterior densities for various retrieved constituents with varying number of spectral data: all 1417 data points (blue line), every second (red line) and every fourth data point (green line).

MCMC toolbox for matlab

Available at:
`http://helios.fmi.fi/~lainema/mcmc`
Developed by Marko Laine (FMI).