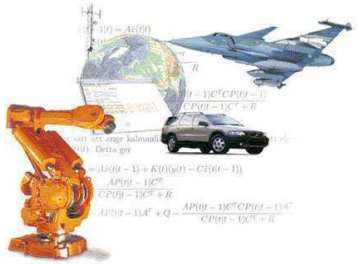
Part 2 - EM and Monte Carlo methods explained via linear system identification

"it is our firm belief that an understanding of linear models is essential for understanding nonlinear ones"



Thomas Schön

Division of Automatic Control
Linköping University
Sweden

The aim – Part 2

The **aim in Part 2** is to introduce expectation maximisation (EM) and Markov chain Monte Carlo (MCMC).

This will be done by showing how simple linear system identification problems can be solved using these methods.

In Part 4 we will then show how EM and MCMC can be used to solve more challenging nonlinear system identification problems.

In other words, we present the methods in this part and hint (there is still much more that remain to be discovered here) at their real potential for nonlinear system identification developed in Part 4.

Outline

1. **Maximum likelihood modelling**
2. Expectation maximisation (EM)
   a) Introduction and derivation
   b) Identifying LGSS models using EM
3. **Bayesian modelling**
4. The Monte Carlo idea
5. Markov chain Monte Carlo (MCMC)
   a) Identifying LGSS models using the Gibbs sampler
   b) The Metropolis Hastings sampler
   c) The Gibbs sampler

Maximum Likelihood (ML) modeling

Maximum likelihood provides a systematic way of computing **point estimates** of the unknown parameters θ in a given model, by exploiting the information present in the measurements $\{y_t\}_{t=1}^T$ and the corresponding inputs $\{u_t\}_{t=1}^T$ (if present).

Computing ML estimates of the parameters in an SSM amounts to:

1. Model the obtained measurements $y_1, \ldots, y_T$ as a realisation from the stochastic variables $Y_1, \ldots, Y_T$.
2. Assume $y_t \mid x_t \sim h_\theta(y_t \mid x_t, u_t)$ and $x_t \mid x_{t-1} \sim f_\theta(x_t \mid x_{t-1}, u_t)$.
3. Assume that the stochastic variables $Y_1, \ldots, Y_T$ are conditionally iid.

Maximum likelihood – goal

The **goal** in maximum likelihood is to find the θ that best describes the distribution from which the data comes from.

Alternatively this can be interpreted as finding the parameter θ that makes the available measurements as likely as possible.

Maximum likelihood – the likelihood function

Definition ((log-)likelihood function)

The likelihood function $L_\theta(y_{1:T})$ is the pdf of the measurements $Y_{1:T}$, with the values for the obtained measurements $y_{1:T}$ inserted,

$$L_\theta(y_{1:T}) \triangleq p_\theta(Y_{1:T} = y_{1:T})$$

and

$$\ell_\theta(y_{1:T}) = \log L_\theta(y_{1:T})$$

is referred to as the log-likelihood.

Latent variables – example

A **latent variable** is a variable that is not directly observed. Other common names are hidden variables, unobserved variables or missing data.

The latent variables in an SSM

$$\begin{aligned} x_{t+1} &\sim f_\theta(x_{t+1} \mid x_t), \\ y_t &\sim h_\theta(y_t \mid x_t), \end{aligned}$$

are given by the unknown states, i.e., $Z = x_{1:T}$.

Expectation Maximization (EM) – strategy and idea

The strategy underlying the EM algorithm is to separate the original ML problem into **two linked problems**, each of which is hopefully easier to solve than the original problem.

This separation is accomplished by exploiting the **structure** inherent in the probabilistic model.

The **key idea** is to consider the joint log-likelihood function of both the observed variables $Y \triangleq y_{1:T}$ and the latent variables Z,

$$\ell_\theta(Z, Y) = \log p_\theta(Z, Y).$$

Expectation Maximization (EM) – the algorithm

Algorithm 1 Expectation Maximization (EM)

1. **Initialise:** Set $i = 1$ and choose an initial θ^1.
2. **While** not converged **do:**

(a) **Expectation (E) step:** Compute

$$\mathcal{Q}(\theta, \theta^i) = \mathrm{E}_{\theta^i}\left[\log p_\theta(Z, Y) \mid Y\right] = \int \log p_\theta(Z, Y) p_{\theta^i}(Z \mid Y) \mathrm{d}Z$$

(b) **Maximization (M) step:** Compute

$$\theta^{i+1} = \arg\max_{\theta \in \Theta} \mathcal{Q}(\theta, \theta^i)$$

(c) $i \leftarrow i + 1$

EM example – LGSS learning

Consider the following scalar LGSS model

$$\begin{aligned} x_{t+1} &= \theta x_t + v_t, \\ y_t &= \frac{1}{2} x_t + e_t, \end{aligned} \qquad \begin{pmatrix} v_t \\ e_t \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \right).$$

For simplicity, let the initial state be fully known, $x_1 = 0$. Finally, the true parameter value for θ is given by $\theta^\star = 0.9$.

The learning problem is now to determine the parameter θ on the basis of the observations $Y = \{y_1, \ldots, y_T\}$ and the above model, using the EM algorithm.

The latent variables Z are given by the states

$$Z = X \triangleq \{x_1, \ldots, x_{T+1}\}.$$

EM example – LGSS learning

The expectation (E) step:

$$\mathcal{Q}(\theta, \theta^i) \triangleq \mathrm{E}_{\theta^i}\left[\log p_\theta(X, Y) \mid Y\right] = \int \log p_\theta(X, Y) p_{\theta^i}(X \mid Y) dX.$$

Let us start investigating $p_\theta(X, Y)$.

$$\begin{aligned} p_\theta(X, Y) &= p_\theta(x_{T+1}, X_T, y_T, Y_{T-1}) \\ &= p_\theta(x_{T+1}, y_T \mid X_T, Y_{T-1}) p_\theta(X_T, Y_{T-1}), \end{aligned}$$

According to the Markov property we have

$$p_\theta(x_{T+1}, y_T \mid X_T, Y_{T-1}) = p_\theta(x_{T+1}, y_T \mid x_T),$$

resulting in

$$p_\theta(X, Y) = p_\theta(x_{T+1}, y_T \mid x_T) p_\theta(X_T, Y_{T-1}).$$

EM example – LGSS learning

Repeated use of the above ideas straightforwardly yields

$$p_\theta(X, Y) = p_\theta(x_1) \prod_{t=1}^{T} p_\theta(x_{t+1}, y_t \mid x_t).$$

According to the model, we have

$$p_\theta\left(\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix} \mid x_t \right) = \mathcal{N}\left(\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix} ; \begin{pmatrix} \theta \\ 1/2 \end{pmatrix} x_t, \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix} \right).$$

EM example – LGSS learning

The resulting $\mathcal{Q}$-function is

$$\begin{aligned}\mathcal{Q}(\theta,\theta^i) &\propto -\mathrm{E}_{\theta^i}\left[\sum_{t=1}^{T} x_t^2 \mid Y\right]\theta^2 + 2\,\mathrm{E}_{\theta^i}\left[\sum_{t=1}^{T-1} x_t x_{t+1} \mid Y\right]\theta \\ &= -\varphi^i\theta^2 + 2\psi^i\theta,\end{aligned}$$

where we have defined

$$\varphi^i \triangleq \sum_{t=1}^{T} \mathrm{E}_{\theta^i}\left[x_t^2 \mid Y\right], \qquad \psi^i \triangleq \sum_{t=1}^{T-1} \mathrm{E}_{\theta^i}\left[x_t x_{t+1} \mid Y\right].$$

There exists explicit expressions (linear state smoothing problem) for these expected values (see the lecture notes for details).

EM example – LGSS learning

The maximization (M) step:

$$\theta^{i+1} = \arg\max_{\theta}\ \mathcal{Q}(\theta,\theta^i).$$

Hence, the M step simply amounts to solving the following quadratic problem,

$$\theta^{i+1} = \arg\max_{\theta}\ -\varphi^i\theta^2 + 2\psi^i\theta,$$

which results in

$$\theta^{i+1} = \frac{\psi^i}{\varphi^i}.$$

EM example – LGSS learning

Algorithm 2 EM for LGSS

1. **Initialise:** Set $i = 1$ and initialise $\theta^1 = 0.1$ and $\theta^0 = 0.6$.
2. **While** $|\ell_{\theta^i}(Y) - \ell_{\theta^{i-1}}(Y)| \geq 10^{-6}$ **do:**

(a) **Expectation (E) step:** Compute

$$\varphi^i = \sum_{t=1}^{T} \mathrm{E}_{\theta^i}\left[x_t^2 \mid Y\right], \qquad \psi^i = \sum_{t=1}^{T} \mathrm{E}_{\theta^i}\left[x_t x_{t+1} \mid Y\right].$$

(b) **Maximization (M) step:** Find the next iterate according to

$$\theta^{i+1} = \frac{\psi^i}{\varphi^i}.$$

(c) $i \leftarrow i + 1$

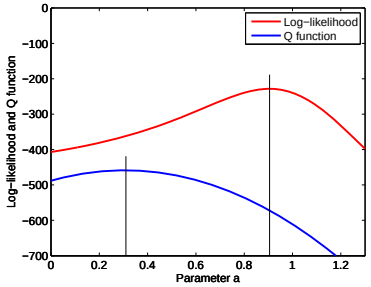
EM example – LGSS learning

- Different number of samples T used.
- Monte Carlo studies, each using 1000 realisations of data.
- Initialize the parameter at $\theta^1 = 0.1$.

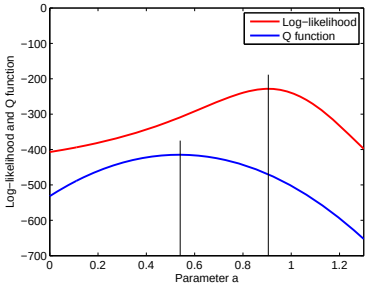
T	100	200	500	1000	2000	5000	10000
$\widehat{\theta}$	0.8716	0.8852	0.8952	0.8978	0.8988	0.8996	0.8998

No surprise, since ML is asymptotically efficient.

EM example – LGSS learning

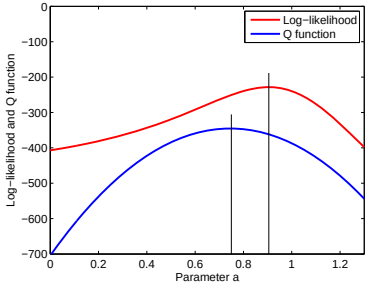


(a) Iteration 1

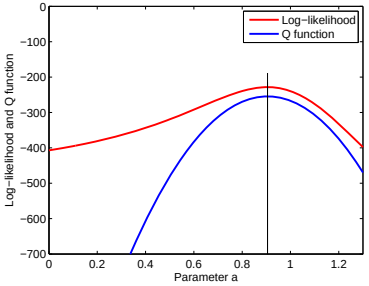


(b) Iteration 2

EM example – LGSS learning



(c) Iteration 3



(d) Iteration 11

EM – fully parameterised LGSS model

Consider a fully parameterised LGSS model

$$\underbrace{\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix}}_{\xi_t} = \underbrace{\begin{pmatrix} A & B \\ C & D \end{pmatrix}}_{\Gamma} \underbrace{\begin{pmatrix} x_t \\ u_t \end{pmatrix}}_{z_t} + \begin{pmatrix} v_t \\ e_t \end{pmatrix}, \quad \begin{pmatrix} v_t \\ e_t \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \underbrace{\begin{pmatrix} Q & S \\ S^{\mathsf{T}} & R \end{pmatrix}}_{\Pi} \right),$$

or more compactly, $\xi_t \mid x_t \sim \mathcal{N}\left(\xi_t \mid \Gamma z_t, \Pi\right)$.

The initial state x_1 distributed according to $\mathcal{N}(x_1 \mid \mu, P_1)$. The parameters to be identified are (using set notation)

$$\theta = \{\Gamma, \mu, \Pi, P_1\}.$$

Follow exactly the same strategy used in previous example (see lecture notes for details).

Bayesian modeling

The **goal** in Bayesian modeling is to compute the posterior $p(\underbrace{\theta, x_{1:T}}_{\triangleq \eta} \mid y_{1:T}) = p(\eta \mid y_{1:T})$ (or one of its marginals).

Bayesian modeling amounts to

1. Find an expression for the likelihood $p(y_{1:T} \mid \eta)$.
2. Assign priors $p(\eta)$ to all unknown stochastic variables η present in the model.
3. Determine the posterior distribution $p(\eta \mid y_{1:T})$.

Posterior predictive distribution

In many applications we are not directly interested in the values of the parameters θ. Instead we are interested (for example) in being able to make predictions.

The **posterior predictive distribution** $p(y_{T+1} \mid y_{1:T})$ is found by marginalising $p(y_{T+1}, \eta_{\mathsf{p}} \mid y_{1:T})$ w.r.t. $\eta_{\mathsf{p}} \triangleq \{\eta, x_{T+1}\} = \{\theta, x_{1:T+1}\}$,

$$\begin{aligned} p(y_{T+1} \mid y_{1:T}) &= \int p(y_{T+1}, \eta_{\mathsf{p}} \mid y_{1:T}) \mathrm{d}\eta_{\mathsf{p}} \\ &= \int h(y_{T+1} \mid x_{T+1}, \theta) p(\eta_{\mathsf{p}} \mid y_{1:T}) \mathrm{d}\eta_{\mathsf{p}}, \end{aligned}$$

where

$$p(\eta_{\mathsf{p}} \mid y_{1:T}) = f(x_{T+1} \mid x_T, \theta) p(\eta \mid y_{1:T}).$$

A first Bayesian model

Let us consider the following scalar LGSS model

$$\begin{aligned} x_{t+1} &= \theta_1 x_t + 0.5 u_t + v_t, & v_t &\sim \mathcal{N}(0, \theta_2), \\ y_t &= 0.5 x_t + e_t, & e_t &\sim \mathcal{N}(0, 0.1). \end{aligned}$$

The input sequence generated as $u_t \sim \mathcal{N}(0, 0.1)$ is assumed known.

(Unrealistic) assumption: the states $x_{1:T+1}$ are available.

Task: Find the posterior distribution for the unknown parameters in the above LGSS model,

$$p(\theta_1, \theta_2 \mid \mathcal{D}) = p(\theta \mid \mathcal{D}),$$

where $\mathcal{D} = \{y_{1:T}, x_{1:T+1}\}$.

Bayesian modeling (again)

Recall that Bayesian modeling amounts to,

1. Find an expression for the likelihood $p(\mathcal{D} \mid \theta)$.
2. Assign priors $p(\theta)$ to all unknown stochastic variables θ present in the model.
3. Determine the posterior distribution $p(\theta \mid \mathcal{D})$.

The **aim** is to write the likelihood $p(\mathcal{D} \mid \theta)$ in such a way that we can easily assign a prior to θ that has the same functional form (w.r.t. θ) as the likelihood.

Likelihood computations, continued (I/II)

$$p(\mathcal{D} \mid \theta) \propto \cdots \propto \theta_2^{-\frac{T-1}{2}} \exp\left(-\frac{\varphi}{2\theta_2}\right) \frac{1}{\sqrt{\theta_2}} \exp\left(-\frac{\sigma}{2\theta_2}\left({\theta_1}^2 - 2\frac{\gamma}{\sigma}\theta_1\right)\right)$$

where we have defined

$$\begin{aligned} \sigma &\triangleq \sum_{t=1}^{T} x_t^2, \\ \gamma &\triangleq \sum_{t=1}^{T} \left(x_{t+1} x_t - 0.5 x_t u_t\right), \\ \varphi &\triangleq \sum_{t=1}^{T} \left(x_{t+1}^2 + 0.25 u_t^2 - x_{t+1} u_t\right). \end{aligned}$$

Question: guided by this, how do we chose the prior $p(\theta)$ to have the same functional form as $p(\mathcal{D} \mid \theta)$?

Likelihood computations, continued (II/II)

Completing the squares results in

$$p(\mathcal{D} \mid \theta) \propto \underbrace{\frac{1}{\sqrt{\theta_2/\sigma}} \exp\left(-\frac{1}{2\theta_2/\sigma}\left(\theta_1 - \frac{\gamma}{\sigma}\right)^2\right)}_{\propto \mathcal{N}\left(\theta_1 \mid \frac{\gamma}{\sigma}, \frac{1}{\sigma}\theta_2\right)}$$
$$\times \underbrace{\theta_2^{-\frac{T-1}{2}} \exp\left(-\frac{1}{\theta_2}\left(\frac{\varphi}{2} - \frac{\gamma^2}{2\sigma}\right)\right)}_{\propto \mathcal{IG}\left(\theta_2 \mid \frac{T-3}{2}, \frac{\varphi}{2} - \frac{\gamma^2}{2\sigma}\right)}$$
$$\propto \mathcal{N}\left(\theta_1 \mid \frac{\gamma}{\sigma}, \frac{1}{\sigma}\theta_2\right) \mathcal{IG}\left(\theta_2 \mid \frac{T-3}{2}, \frac{\varphi}{2} - \frac{\gamma^2}{2\sigma}\right)$$

Reminder – the inverse Gamma distribution

The inverse gamma distribution is defined on the positive real line and it is characterised by the so called shape parameter a and the scale parameter b,

$$x \sim \mathcal{IG}(a, b), \qquad a > 0, b > 0.$$

The pdf is given by

$$\mathcal{IG}(x \mid a, b) = \frac{b^a}{\Gamma(a)} x^{-(a+1)} \exp\left(-\frac{b}{x}\right), \qquad x > 0,$$

where $\Gamma(a)$ is the gamma function, i.e., $\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} \mathrm{d}t$.

Assign prior and compute posterior

Our strategy dictates that we should choose the prior such that it has the same functional form as the likelihood $p(\mathcal{D} \mid \theta)$, i.e., normal inverse gamma,

$$p(\theta_1, \theta_2) = p(\theta_1 \mid \theta_2) p(\theta_2) = \mathcal{N}(\theta_1 \mid m, c\theta_2)\, \mathcal{IG}(\theta_2 \mid a, b)$$
$$= \mathcal{NIG}(\theta_1, \theta_2 \mid m, c, a, b),$$

Compute the posterior

$$p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta) p(\theta)$$
$$\propto \mathcal{NIG}\left(\theta_1, \theta_2 \mid \widetilde{m}, \widetilde{c}, \widetilde{a}, \widetilde{b}\right) \mathcal{NIG}(\theta_1, \theta_2 \mid m, c, a, b)$$
$$\propto \mathcal{NIG}(\theta_1, \theta_2 \mid m^\star, c^\star, a^\star, b^\star)$$

(see the lecture notes for details)

Conjugate priors

The posterior distribution $p(\eta \mid \mathcal{D})$ and the prior distribution $p(\eta)$ are said to be **conjugate distributions** if they are both distributed according to the same distribution.

The prior is then referred to as the **conjugate prior** for the present likelihood $p(\mathcal{D} \mid \eta)$.

Put in slightly different words, if the posterior distribution and the prior distribution have the same functional form, the prior is said to be the conjugate prior for the underlying likelihood.

Monte Carlo methods

Motivation – MCMC

In solving inference problems we are sooner or later typically faced with various **integration problems**, which tend to live in high dimensional spaces.

This hols for both Maximum likelihood and Bayesian approaches.

To be concrete, we have the following three classes

1. Expectation
2. Normalisation
3. Marginalisation

MCMC motivation 1 – expectation

An expected value often provides an interesting (and interpretable) point estimate.

Computing an expectation amounts to solving the following integral

$$\mathrm{E}\left[g(z)\right] = \int_{\mathcal{Z}} g(z) p(z \mid y_{1:T}) \mathrm{d}z,$$

for some function $g : \mathcal{Z} \to \mathbb{R}^{n_g}$.

Examples: Computing a point estimate of the state ($z = x_t$ and $g(x_t) = x_t$). Computing the conditional mean estimate of the parameters θ in a dynamic system given the measurements ($z = \theta, g(\theta) = \theta$)

MCMC motivation 2 – Normalisation

Computing the marginal likelihood $p(y_{1:T})$ (i.e., the normalization factor) has to be done if the posterior distribution is needed.

The corresponding integral is

$$p(y_{1:T}) = \int_{\mathcal{Z}} p(y_{1:T} \mid z) p(z) \mathrm{d}z.$$

Examples: Used in empirical Bayes (type 2 maximum likelihood, evidence approximation) for finding an initial parameter guess.

MCMC motivation 3 – Marginalisation

If we are interested in the properties of a stochastic variable z_1 and have access to the pdf $p(z_1, z_2 \mid y_{1:T})$, then we can marginalize out the variable z_2, resulting in $p(z_1 \mid y_{1:T})$.

$$p(z_1 \mid y_{1:T}) = \int_{\mathcal{Z}_2} p(z_1, z_2 \mid y_{1:T}) \mathrm{d}z_2$$

Examples: We have algorithms targeting $p(\theta, x_{1:T} \mid y_{1:T})$, but often we are only interested in $p(\theta, x_{1:T} \mid y_{1:T})$. As another example (in using the EM algorithm for nonlinear ML identification) we need the two-step smoothing densities $p(x_{t:t+1} \mid y_{1:T})$, whereas several smoothing algorithms provides the entire joint smoothing density $p(x_{1:T} \mid y_{1:T})$.

Approximation methods

Many of the models we are currently interested in do **not** allow for closed form expressions. We are forced to approximations. Broadly speaking there are two classes,

1. **Deterministic analytical approximations:** Either approximate the model or restrict the solution to belong to an analytically tractable form. Examples, variational Bayes (VB), expectation propagation (EP).
2. **Stochastic approximations:** Keep the model and approximate the solution without imposing any restrictions other than the computational resources available.

Analytical approximations of the model and/or the solution have been/are very common.
In this course we work with stochastic approximations.

Monte Carlo methods

Monte Carlo methods provides **computational solutions**, where the obtained accuracy is only limited by our computational resources.

Monte Carlo methods respects the model and the general solution. The approximation does not impose any restricting assumptions on the model or the solution.

The Monte Carlo idea

The integral

$$I(g(z)) \triangleq \mathrm{E}_{\pi(z)}\left[g(z)\right] = \int_{\mathcal{Z}} g(z)\pi(z)\mathrm{d}z.$$

is approximated by

$$\widehat{I}_M(g(z)) = \frac{1}{M}\sum_{i=1}^{M} g(z^i).$$

The strong law of large numbers tells us that

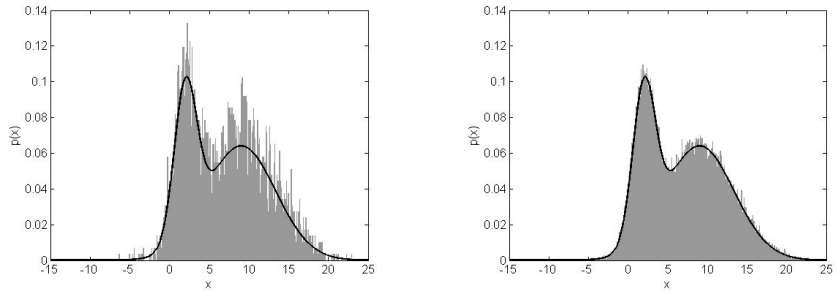
$$\widehat{I}_M(g(z)) \xrightarrow{\text{a.s.}} I(g(z)), \qquad M \to \infty,$$

and the central limit theorem state that

$$\frac{\sqrt{M}\left(\widehat{I}_M(g(z)) - I(g(z))\right)}{\sigma_g} \xrightarrow{\text{d}} \mathcal{N}(0,1), \qquad M \to \infty.$$

The Monte Carlo idea – illustration

$$\pi(z) = 0.3\mathcal{N}(z \mid 2, 2) + 0.7\mathcal{N}(z \mid 9, 19)$$



5000 samples 50000 samples

Obvious problem: In general we are **not** able to directly sample from the density we are interested in.

Background: representing an LGSS model (I/II)

An LGSS model is defined by

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + v_t, \\ y_t &= Cx_t + Du_t + e_t, \end{aligned}$$

where

$$\begin{pmatrix} x_1 \\ v_t \\ e_t \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mu \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} P_1 & 0 & 0 \\ 0 & Q & S \\ 0 & S^{\mathsf{T}} & R \end{pmatrix} \right).$$

which equivalently can be written as

$$\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix} \mid x_t \sim \mathcal{N}\left(\begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix} \mid \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} x_t \\ u_t \end{pmatrix}, \begin{pmatrix} Q & S \\ S^{\mathsf{T}} & R \end{pmatrix} \right)$$
$$x_1 \sim \mathcal{N}(x_1 \mid \mu, P_1)$$

Background: representing an LGSS model (II/II)

Introducing the following notation

$$\xi_t \triangleq \begin{pmatrix} x_{t+1} \\ y_t \end{pmatrix}, \qquad z_t \triangleq \begin{pmatrix} x_t \\ u_t \end{pmatrix}, \qquad \Gamma \triangleq \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \qquad \Pi \triangleq \begin{pmatrix} Q & S \\ S^{\mathsf{T}} & R \end{pmatrix}$$

allows us to write the LGSS model more compactly,

$$\begin{aligned} \xi_t \mid x_t &\sim \mathcal{N}(\xi_t \mid \Gamma z_t, \Pi), & \xi_t &= \Gamma z_t + w_t, & w_t &\sim \mathcal{N}(0, \Pi), \\ x_1 &\sim \mathcal{N}(x_1 \mid \mu, P_1). & x_1 &\sim \mathcal{N}(x_1 \mid \mu, P_1). \end{aligned}$$

The parameters are defined as (using set notation)

$$\theta = \{\Gamma, \Pi, \mu, P_1\}.$$

Problem formulation and solution idea

Task: Identify the LGSS model by computing $p(\theta \mid Y)$ where $\theta = \{\Gamma, \Pi\}$ and $Y \triangleq y_{1:T}$ (for simplicity, we assume that the initial state is known)

Let us consider the following extended task, where we are trying to compute $p(\theta, X \mid Y)$, with $X \triangleq x_{1:T+1}$. Note that $p(\theta \mid Y)$ is a marginal of $p(\theta, X \mid Y)$.

Solution idea: Obtain samples θ^k, X^k from the posterior pdf by iterating the following two steps

1. Given θ^k, generate a sample from the state trajectory
$$X^k \sim p(X \mid Y, \theta^k).$$
2. Then, given X^k generate a sample θ^{k+1}
$$\theta^{k+1} \sim p(\theta \mid X^k, Y).$$

Definition – Matrix Normal random matrix

The matrix valued normal distribution is a generalisation of the vector valued normal distribution.

Definition (Matrix normal distribution)

The random matrix $X \in \mathbb{R}^{d\times m}$ has a matrix normal distribution with mean matrix $M \in \mathbb{R}^{d\times m}$ and covariance matrix $\Lambda^{-1} \otimes \Sigma$, where $\Lambda^{-1} \succ 0 \in \mathbb{R}^{m\times m}$ and $\Sigma \succ 0 \in \mathbb{R}^{d\times d}$ if

$$\operatorname{Vec}(X) \sim \mathcal{N}\left(X \mid \operatorname{Vec}(M), \Lambda^{-1} \otimes \Sigma\right).$$

The pdf is given by

$$\mathcal{MN}(X \mid M, \Lambda, \Sigma) = \frac{|\Lambda|^{d/2}}{(2\pi)^{dm/2}|\Sigma|^{m/2}} \exp\left(-\frac{1}{2}\operatorname{Tr}\left((X-M)^{\mathsf{T}}\Sigma^{-1}(X-M)\Lambda\right)\right)$$

Sampling the parameters

The first step of the Bayesian principle is done and the likelihood is

$$p(\mathcal{D} \mid \Gamma, \Pi) = \mathcal{MN}(\Xi \mid \Gamma Z, I, \Pi)$$

The second step is to decide on a suitable prior. We will be pragmatic and make use of a conjugate prior, the **matrix normal inverse Wishart ($\mathcal{MNIW}$)** prior (the generalisation of the $\mathcal{NIG}$ prior).

It is a hierarchical prior that makes use of the fact that $p(\Gamma, \Pi) = p(\Gamma \mid \Pi)p(\Pi)$ and places an $\mathcal{MN}$ prior on Γ conditioned on Π and an $\mathcal{IW}$ prior on Π.

(See lecture notes for detailed derivations of the $\mathcal{MNIW}$ posterior distribution.)

LGSS example (I/II)

1. Given θ^k, generate a sample from the state trajectory

$$X^k \sim p(X \mid Y, \theta^k).$$

2. Then, given X^k generate a sample θ^{k+1}

$$\theta^{k+1} \sim p(\theta \mid X^k, Y)$$

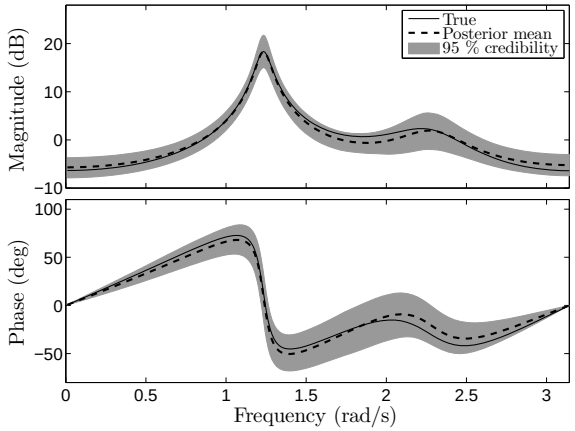
Let us now try this solution using $T = 3000$ samples from

$$x_{t+1} = \begin{pmatrix} 0.37 & 0.89 & 0.52 & 0.56 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} x_t + w_t, \quad w_t \sim \mathcal{N}(0, 0.05 I_4),$$

$$y_t = \begin{pmatrix} 1 & 0.1 & -0.49 & 0.01 \end{pmatrix} x_t + e_t, \qquad e_t \sim \mathcal{N}(0, 0.01).$$

LGSS example (II/II)

Initialize using a subspace algorithm. Run the loop 10000 times.



LGSS identification – Gibbs sampler

So far, only pragmatic, but it seems to work! This results in many questions, for example,

1. Was this just luck?
2. Does it always work?
3. Can we prove that it will always work?

It is a so called **Gibbs sampler** that provably converge to the target distribution!!

This can be used to answer some otherwise challenging questions, for more details see (and its references)

Adrian Wills, Thomas B. Schön, Fredrik Lindsten and Brett Ninness, **Estimation of Linear Systems using a Gibbs Sampler**. *Proceedings of the 16th IFAC Symposium on System Identification (SYSID)*, Brussels, Belgium, July 2012.

Example – AR(1)

Consider the following LGSS model (an autoregressive (AR) model of order 1, i.e., an AR(1) model),

$$x^{t+1} = ax^t + v^t, \qquad v^t \sim \mathcal{N}(0, q),$$
$$x^1 \sim \mu^1 = \mathcal{N}\left(x^1 \mid x_1, p_1\right),$$

where $|a| < 1$.

This can equivalently be defined as a Markov chain $\{x^t\}_{t\geq 1}$ with initial distribution

$$\mu^1 = \mathcal{N}\left(x^1 \mid x_1, p_1\right),$$

and transition kernel

$$K(x^{t+1} \mid x^t) = \mathcal{N}\left(x^{t+1} \mid ax^t, q\right).$$

Example – AR(1)

What happens when $t \to \infty$?

Everything is Gaussian and only linear transformations are involved, which implies that everything remains Gaussian.

Mean value:

$$\mathrm{E}\left[x^{t+1}\right] = \mathrm{E}\left[ax^t + v^t\right] = a\,\mathrm{E}\left[x^t\right] = \cdots = a^t x_1,$$

Variance:

$$p_{t+1} \triangleq \mathrm{Var}\left[x^{t+1}\right] = \mathrm{E}\left[\left(x^{t+1} - \mathrm{E}\left[x^{t+1}\right]\right)^2\right] = \cdots = a^2 p_t + q.$$

Example – AR(1)

When $t \to \infty$ we have to solve

$$\bar{p} = a^2\bar{p} + q,$$

which straightforwardly results in

$$\bar{p} = \frac{q}{1 - a^2}.$$

We have now showed that the Markov chain converge to the following **stationary distribution**

$$\pi^{\mathrm{s}}(x) = \mathcal{N}\left(x \mid 0, \frac{q}{1 - a^2}\right).$$

as $t \to \infty$.

What's the point with the AR(1) example? (I/II)

Task: How do we generate samples from the stationary distribution $\pi^{\mathrm{s}}(x) = \mathcal{N}\left(x \mid 0, \frac{q}{1-a^2}\right)$? Put in other words, the target distribution $\pi(x)$ is given by the stationary distribution $\pi^{\mathrm{s}}(x)$, i.e., $\pi(x) = \pi^{\mathrm{s}}(x)$.

Two solutions for this problem:

1. Simulate sufficiently many samples from the Markov chain and discard the initial samples. The remaining samples will then be approximately distributed according to the target distribution (we just proved that x^t is distributed according to $\pi(x)$ for a large enough t).
2. We proved that the stationary distribution is Gaussian. Generate samples directly from this distribution.

Clearly a somewhat contrived example (obviously solution 2 is preferred), **but** solution 1 is a simple illustration of the strategy underlying all MCMC methods.

What's the point with the AR(1) example? (II/II)

In the example, the Markov chain was fully specified and it was possible to explicitly compute the stationary distribution.

We are of course interested in the **reverse** situation, where we want to generate samples from a (typically rather complicated) target distribution $\pi(z)$.

The task is now to find a transition kernel such that the resulting Markov chain has the target distribution $\pi(z)$ as its stationary distribution.

This can be done in many different ways and **constructive strategies** for doing this are provided by the Gibbs sampler and the Metropolis Hastings sampler.

Metropolis Hastings (MH) sampler

The Metropolis Hastings (MH) sampler provides a **constructive way** of producing a Markov chain that can be used to obtain samples approximately distributed according to the target distribution.

More pragmatically speaking, the MH sampler generates samples $\{z^m\}_{m=1}^M$ which can for example be used to approximately compute integrals.

Metropolis Hastings (MH) sampler – intuition

The basic idea underlying the Metropolis Hastings sampler is surprisingly simple.

Starting from an initial state of the Markov chain z^1, a new candidate sample z^* is generated using a **proposal distribution** $z^* \sim q(z \mid z^1)$.

This proposed sample z^* is then accepted with a certain probability, the so called **acceptance probability**

$$a(z^*, z^m) = \min\left(1, \frac{\pi(z^*)q(z^m \mid z^*)}{\pi(z^m)q(z^* \mid z^m)}\right).$$

If the sample is accepted, the new state of the Markov chain is set to the proposed sample $z^2 = z^*$, otherwise it is simply set to the previous value, $z^2 = z^1$.

Metropolis Hastings (MH) sampler – algorithm

Algorithm 3 Metropolis Hastings (MH) sampler

1. **Initialise:** Set the initial state of the Markov chain z^1.
2. **For** $m = 1$ **to** M**, iterate:**
   a. Sample $z^* \sim q(z \mid z^m)$.
   b. Sample $u \sim \mathcal{U}[0,1]$.
   c. Compute the acceptance probability

$$a(z^*, z^m) = \min\left(1, \alpha(z^*, z^m)\right), \text{where } \alpha(z^*, z^m) = \frac{\pi(z^*)q(z^m \mid z^*)}{\pi(z^m)q(z^* \mid z^m)}$$

   d. Set the next state z^{m+1} of the Markov chain according to

$$z^{m+1} = \begin{cases} z^* & u \leq a(z^*, z^m) \\ z^m & \text{otherwise} \end{cases}$$

Metropolis Hastings (MH) sampler

Note that the MH sampler only requires two things,

1. It requires the definition of a proposal distribution $q(\cdot \mid \cdot)$ that can be used to generate candidate samples.
2. It must be possible to point-wise evaluate the target distribution up to a possibly unknown normalization factor.

Point-wise evaluation of the target density $\pi(\theta)$ for a specific $\theta = \bar{\theta}$

$$\pi(\bar{\theta}) = p(\bar{\theta} \mid y_{1:T}) = \frac{p(y_{1:T} \mid \bar{\theta})p(\bar{\theta})}{p(y_{1:T})}.$$

Example – MH sampler for LGSS identification

Consider the following LGSS model

$$\begin{aligned} x_{t+1} &= \theta x_t + 0.5u_t + v_t, & v_t &\sim \mathcal{N}(0, 0.1), \\ y_t &= 0.5x_t + e_t, & e_t &\sim \mathcal{N}(0, 0.1), \\ p(\theta) &= \mathcal{U}[-1, 1], \end{aligned}$$

where the input sequence $u_t \sim \mathcal{N}(0, 0.1)$ is assumed to be known.

Task: set up an MH sampler targeting $p(\theta \mid y_{1:T})$. In other words, simulate a Markov chain with $p(\theta \mid y_{1:T})$ as its stationary distribution.

The first task it to decide on a proposal distribution, let us use a so called random walk proposal,

$$\theta^* = \theta^m + v_m, \qquad v_m \sim \mathcal{N}(0, Q),$$

or put in other words, $q(\theta^* \mid \theta^m) = \mathcal{N}(\theta^* \mid \theta^m, Q)$.

Example – MH sampler for LGSS identification

The second task is to find an expression for the acceptance probability, which boils down to computing

$$\alpha(\theta^*, \theta^m) = \frac{\pi(\theta^*)q(\theta^m \mid \theta^*)}{\pi(\theta^m)q(\theta^* \mid \theta^m)} = \frac{\pi(\theta^*)}{\pi(\theta^m)} = \frac{p(\theta^* \mid y_{1:T})}{p(\theta^m \mid y_{1:T})},$$

The resulting expression for the acceptance probability is

$$\alpha(\theta^*, \theta^m) = \frac{p(y_{1:T} \mid \theta^*)p(\theta^*)}{p(y_{1:T} \mid \theta^m)p(\theta^m)} = \frac{p(\theta^*)}{p(\theta^m)} \prod_{t=1}^{T} \frac{p(y_t \mid y_{1:t-1}, \theta^*)}{p(y_t \mid y_{1:t-1}, \theta^m)},$$

where the required one step prediction densities are straightforwardly provided by the KF according to

$$p(y_t \mid y_{1:t-1}, \bar{\theta}) = \mathcal{N}\left(y_t \mid 0.5\hat{x}_{t|t-1}(\bar{\theta}), 0.5^2 P_{t|t-1}(\bar{\theta}) + 0.1\right),$$

where $\bar{\theta}$ is used as a placeholder for θ^* or θ^m, respectively.

The Gibbs sampler is a particularly popular **special case** of the Metropolis Hastings sampler, applicable when the conditional distributions

$$\pi_l(z_l \mid z_{-l})$$

are tractable and easy to sample from. Here, z_{-l} denotes all the elements in z, but the l^{th} one.

Algorithm 4 Gibbs sampler (GS)

1. **Initialise:** Set the initial state $z^1 = \left(z_1^1, z_2^1, \ldots, z_K^1\right)$.
2. **For** $m = 1$ **to** M**, iterate:**
   1. Draw $z_1^{m+1} \sim p(z_1 \mid z_2^m, \ldots, z_K^m)$
   2. Draw $z_2^{m+1} \sim p(z_2 \mid z_1^{m+1}, z_3^m, \ldots, z_K^m)$

      $\vdots$

   K. Draw $z_K^{m+1} \sim p(z_K \mid z_1^{m+1}, \ldots, z_K^m)$

See the lecture note for properties of the MH and the Gibbs samplers.