Year 2020-2021

Estimation and filtering

Bayesian estimation using stochastic simulation: theory e applications

Prof. Gianluigi Pillonetto

SUMMARY

- Fisherian vs Bayesian estimation
- Bayesian estimation using Monte Carlo methods
- Bayesian estimation using Markov chain Monte Carlo
- On-line Bayesian estimation (particle filters)

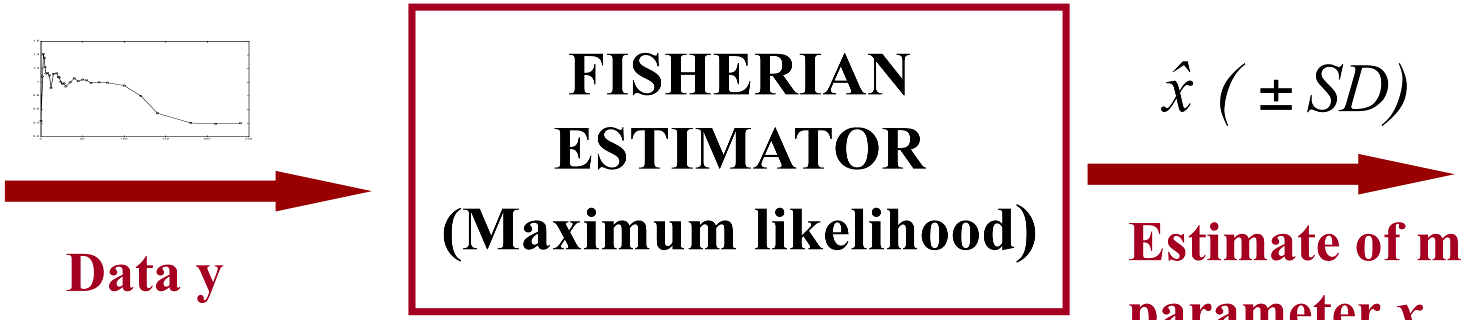
FISHER VS BAYES

Let us consider the model: *y=G(x)+v*

Fisher approach: x, which admits a true deterministic value, is estimated using only the experimental data
e.g. Maximum Likelihood: $\hat{x} = \arg\max p_{y|x}\left(y|x\right)$

Bayes approach: x is random and we estimate one realization using not only the experimental data (posteriori information), but also the a priori information (indipendent of the data)

Fisher approach to parametric estimation



Data y

FISHERIAN ESTIMATOR (Maximum likelihood)

$\hat{x}\ (\pm SD)$

Estimate of model parameter *x*

ADVANTAGES

- require optimization algorithms (e.g. conjugate gradient/Newton) often not so computational expensive

DRAWBACKS

- They are minimum variance estimators only using linear models and Gaussian measurement errors

- They often return non realistic confidence intervals (e.g. containing negative values due to Gaussian approximations of the estimates)

Bayes approach to parametric estimation (1/7)

The starting point is that we have some information on x, indipendent of the data (i.e. "before seeing the data"=*a priori*), and these expectations are summarized in the *a priori* probability density function

$$p_x(x)$$

Such expectations are then modified after seeing the data y, hence one speaks of a posteriori probability density function (=conditional on y)

$$p_{x|y}(x|y)$$

This is the key function obtained by Bayes.
From it, one can obtain point estimates and confidence intervals.

Bayes approach to parametric estimation (2/7)

Why using Bayesian priors

- To onclude all the available information in the estimation process

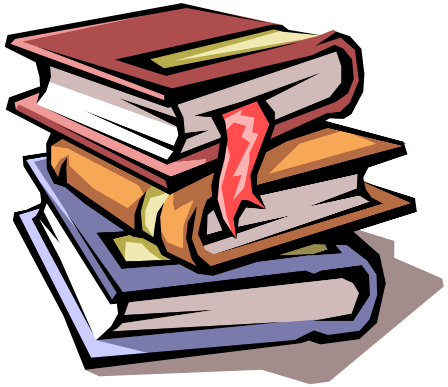
- To extend the complexity of the model
  - Priors on all the unknown parameters

- To improve the parameter estimates
  - Use of population or individual information

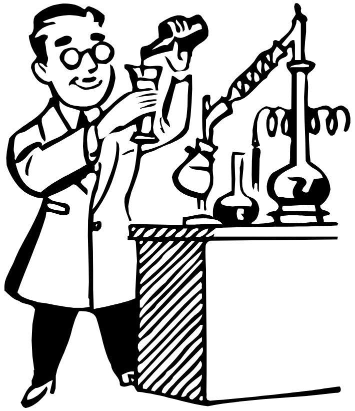
- To analyze sparse data set/high measurement noise
  - "Weak" Likelihood, "strong" prior

How to obtain the prior?

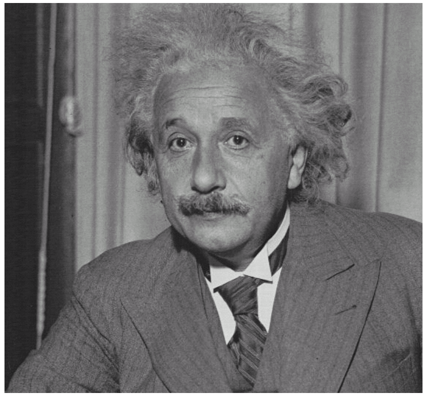
Literature



Previous experiments



Experts



Population studies

Bayes approach to parametric estimation (3/7)

Examples of Bayesian estimators

From $p_{x|y}(x|y)$ one can obtain different estimators.
The most used are:

Posterior mean (minimum variance error)

$$\hat{x} = E\left[x\middle|y\right] = \int x p_{x|y}\left(x \mid y\right) dx$$

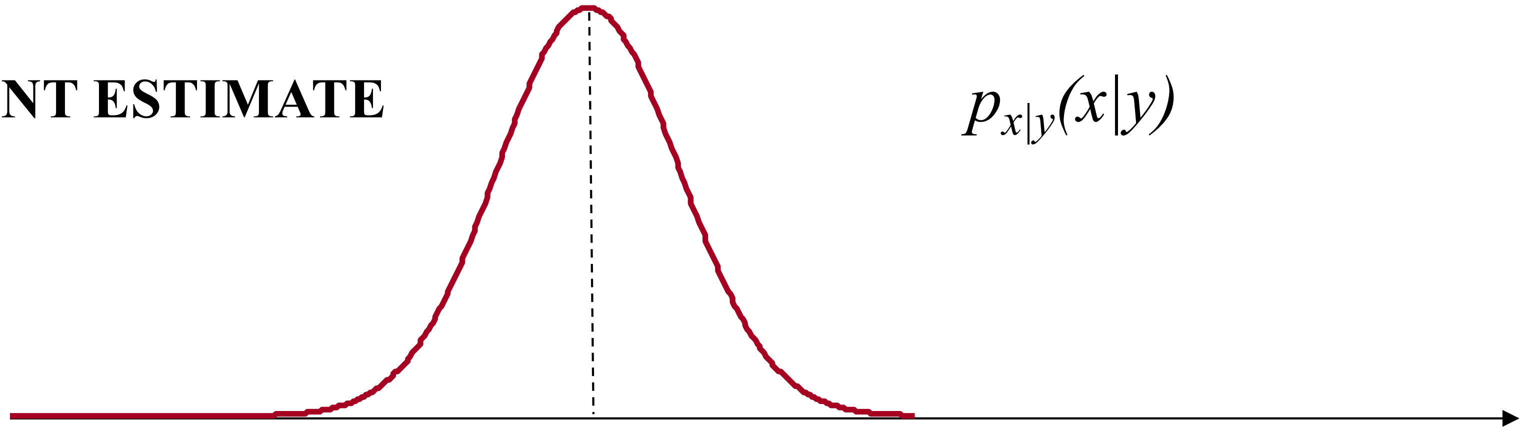
Maximum a posteriori (MAP)

$$\hat{x} = \arg\max p_{x|y}\left(x\middle|y\right)$$

Bayes approach to parametric estimation (4/7)

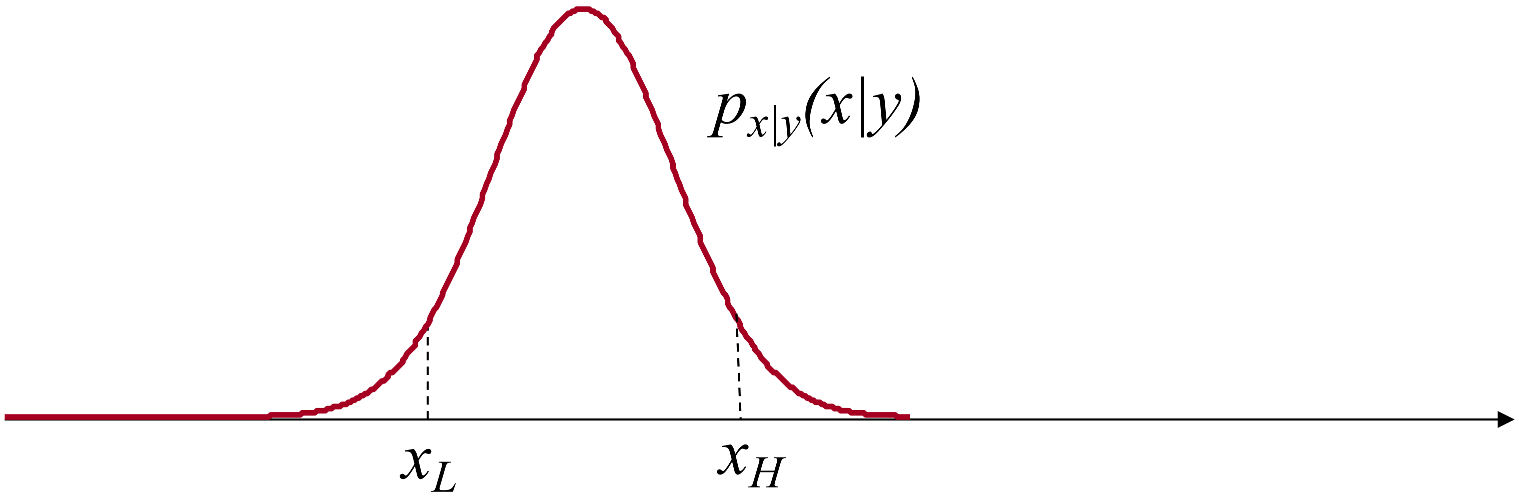
Use of the posterior: example with scalar *x*

POINT ESTIMATE



Here MAP (Maximum a Posteriori) estimate coincides with minima variance estimate

CONFIDENCE INTERVALS



95% CI (mean ± 2SD if $x|y$ is Gaussian)

Bayes approach to parametric estimation (5/7)

We can estimate x from the posterior $p_{x|y}(x|y)$.
But how can we obtain it?

Bayes rule:

$$p_{x|y}(x\,|\,y) = \frac{p_{yx}(y\,|\,x)\,p_x(x)}{p_y(y)}$$

To determine $p_{x|y}(x|y)$ we need:

- the prior density of x, $p_x(x)$
- the likelihood y, $p_{y|x}(y|x)$, computable from the model $G(x)$ and from the statistics of the error ν ($y=G(x)+v$)

Bayes approach to parametric estimation (6/7)

PARTICULAR CASE: *x* e *v* independent Gaussian, linear *G* (*G*(*x*)=*Gx*)

$$p_x(x) = \frac{1}{\left[(2\pi)^M \det(\Sigma_x)\right]^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma_x^{-1}(p-\mu)\right)$$ Prior density

$$p_{y|x}(y \mid x) = \frac{1}{\left[(2\pi)^N \det(\Sigma_v)\right]^{1/2}} \exp\left(-\frac{1}{2}\left[y-Gx\right]^T \Sigma_v^{-1}\left[y-Gx\right]\right)$$ Likelihood

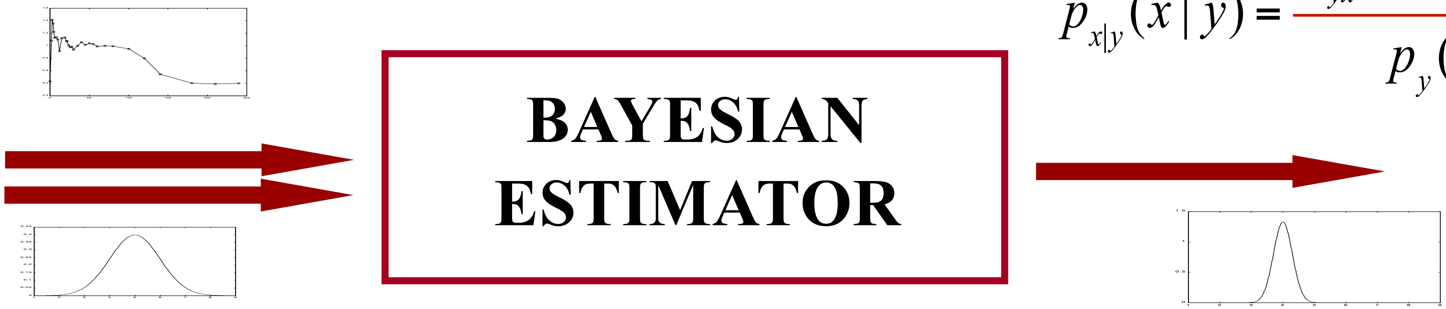
The posterior is also Gaussian and we have:

$$\hat{x}_{MAP} = E\left[x \middle| y\right] = \arg\min_x \left[y-Gx\right]^T \Sigma_v^{-1}\left[y-Gx\right] + \left(x-\mu\right)^T \Sigma_x^{-1}\left(x-\mu\right)$$

Posterior information = data *A priori information*

Bayes approach to parametric estimation (7/7)

$$p_{x|y}(x\,|\,y) = \frac{p_{yx}(y\,|\,x)\,p_x(x)}{p_y(y)}$$



Prior: a priori probability density function of model parameters

Posterior: a posteriori probability density function of model parameters

ADVANTAGES

- Return all the distribution of the estimates (from which e.g. minimum variance estimates and realistic confidence intervals can be obtained)

DRAWBACKS

- Computation of Bayesian point estimates and relative confidence may require solutions of computationally intractable integrals

Bayes approach: computational difficulties (1/2)

Integration plays a fundamental role in Bayesian estimation

- determination of the normalization factor

$$p_{x|y}(x\,|\,y) = \frac{p_{yx}(y\,|\,x)p_x(x)}{\boxed{p_y(y)}} \longrightarrow \int p_{y|x}(y\,|\,x)p_x(x)dx$$

- distribution synthesis $\int g(x)p_{x|y}(x\,|\,y)dx$

Esempi:

$x = \left[x_1\ x_2\ \ldots\ x_d\right]^T$

$g(x) = x_i$: minimum variance estimate of x_i

$A \subset \Re^d$

$$\chi(p \in A) = \begin{cases} 1 \text{ if } p \in A \\ 0 \text{ otherwise} \end{cases}$$

$g(x) = \chi(x \in A)$: probability that x assumes values in A

Bayes approach: computational difficulties (2/2)

- Vector *x* may assume values in high-dimensional spaces and its prior distribution can be non Gaussian

- Nonlinear models may be needed

- Data set size may be poor and the signal to noise ratio can be small

⬇

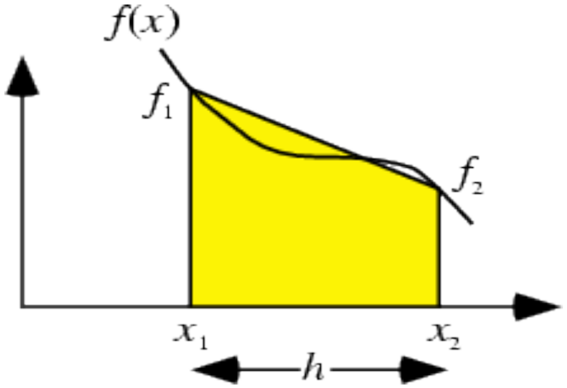
Posterior may be complex,
far from Gaussianity,
hence difficult to integrate

DETERMINISTIC APPROACHES TO THE PROBLEM (1/4)

- Classical numerical methods

Use quadrature rules which approximate the integral using sums of areas of polygons

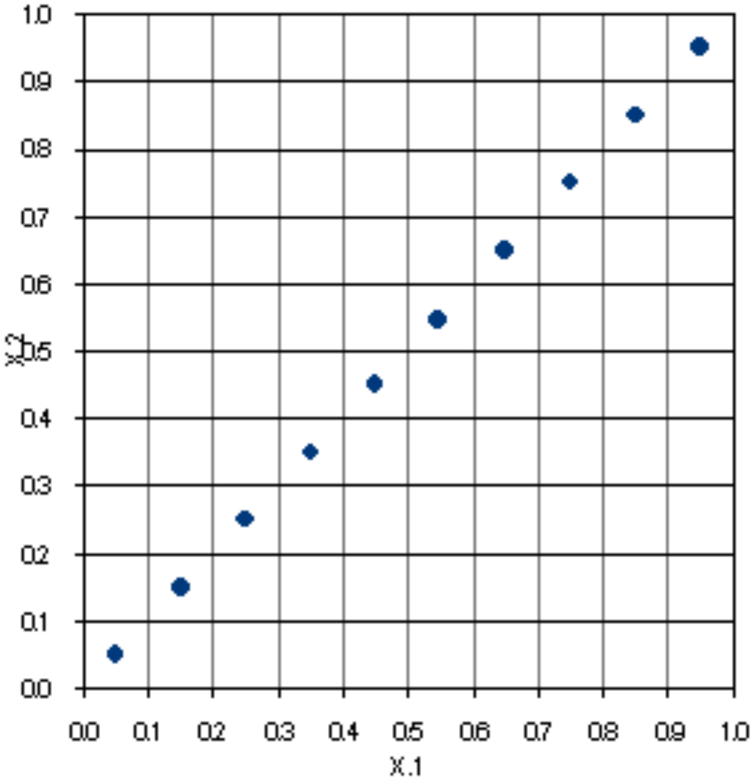
Dimension 1: the integration interval
is divided in pieces of lenght *h*
One obtains polygons which approximate
the function (e.g. lines, Lagrange polynomials)
and then we obtain the area



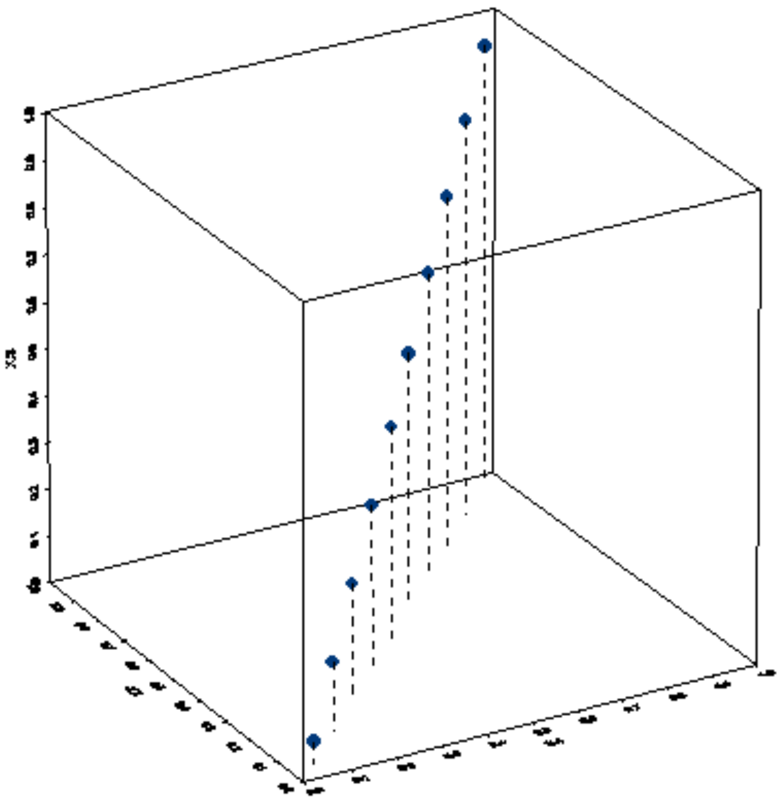
Limits: even if they can provide very accurate results, they are numerical procedures which can be used only in low-dimensional spaces, in practice 2- or 3-dimensional (due to the "curse of dimensionality")

DETERMINISTIC APPROACHES TO THE PROBLEM (2/4)

Curse of dimensionality



10% of coverage



1% of coverage

The number of points has to exponentially increase
to maintain a certain coverage accuracy

DETERMINISTIC APPROACHES TO THE PROBLEM (3/4)

- Asymptotic Laplace approximation

The posterior *π(x)* is approximated by a Gaussian distribution by computing its maximum and its Hessian around the maximum of the log-posterior

$$\hat{x} = \arg\max_x \log\left(\pi\left(x\right)\right)$$

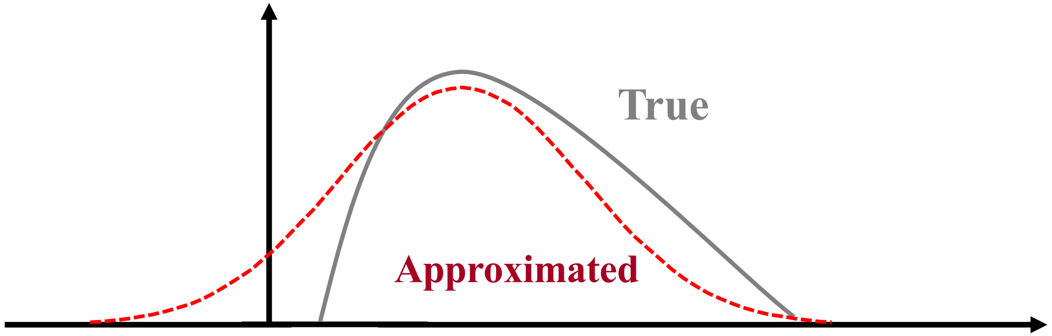
$$\log\pi\left(x\right) \approx \log\pi\left(\hat{x}\right) + \frac{1}{2}\left(x-\hat{x}\right)^T \times \left[\frac{\partial^2 \log\pi}{\partial x^T \partial x}\Big|\hat{x}\right] \times \left(x-\hat{x}\right)$$

$$\boxed{\begin{aligned} &\pi\left(x\right) \approx \frac{1}{\sqrt{\det\left(2\boldsymbol{\pi}\Sigma\right)}} e^{-\frac{1}{2}\left(x-\hat{x}\right)^T \Sigma^{-1}\left(x-\hat{x}\right)} \doteq N\left(\hat{x},\Sigma\right) \\ &\Sigma = -\left[\frac{\partial^2 \log\pi}{\partial x^T \partial x}\Big|\hat{x}\right]^{-1} \end{aligned}}$$

DETERMINISTIC APPROACHES TO THE PROBLEM (4/4)

$$\pi\left(x\right) \approx N\left(\hat{x}, \Sigma\right)$$

$$\Sigma = -\left[\frac{\partial^2 \log \pi}{\partial x^T \partial x}\bigg|\hat{x}\right]^{-1}$$



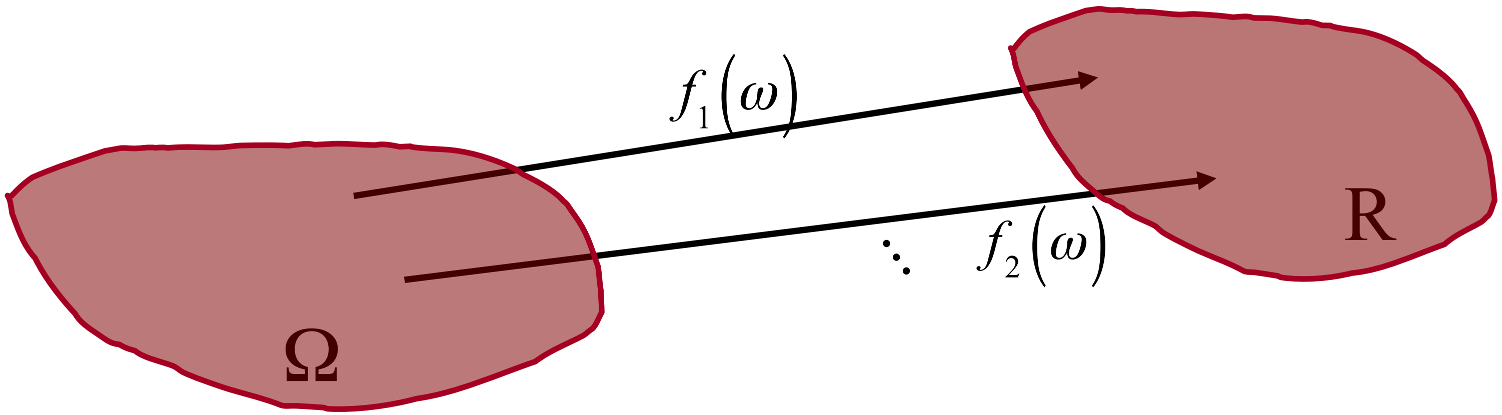
Limits: results are often not so reliable and it is hard to evaluate the goodness of the approximation

SUMMARY

- Fisherian vs Bayesian estimation
- Bayesian estimation using Monte Carlo methods
- Bayesian estimation using Markov chain Monte Carlo
- On-line Bayesian estimation (particle filters)

CONVERGENCE OF RANDOM VARIABLES (1/2)

Consider a sequence of random variables f_n on a sample space Ω with generic element ω

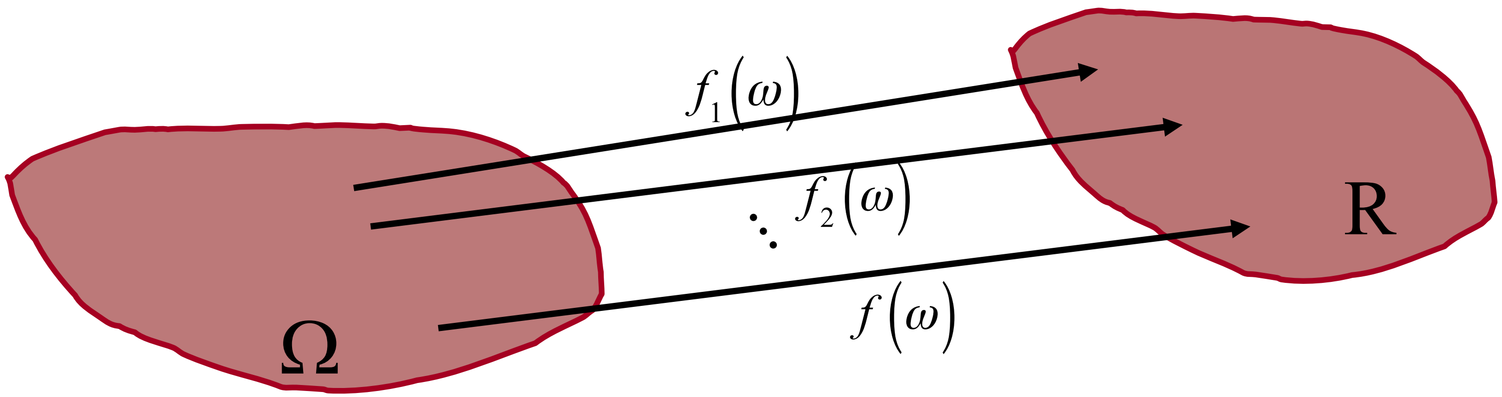


CONVERGENCE OF RANDOM VARIABLES (2/2)

$$\lim_{n\to\infty} f_n \overset{\text{almost surely}}{=} f$$

if

$$\Pr\left(\omega\colon \lim_{\mathrm{n}\to\infty} f_n\left(\omega\right) = f\left(\omega\right)\right) = 1$$



STOCHASTIC APPROACHES: MONTE CARLO SIMULATION

Let us use *π(x)* to denote the posterior:

we are interested in $E_\pi\left(g\right) \doteq \int g\left(x\right)\pi\left(x\right)dx$

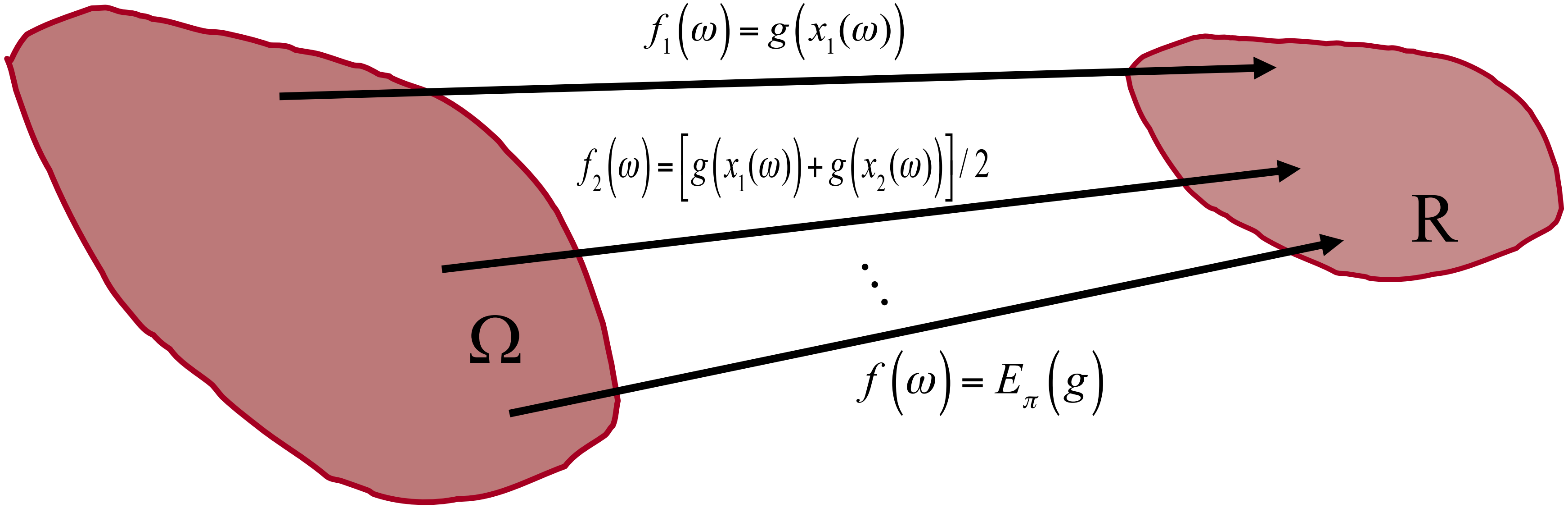
We have x^1, x^2, ..., x^n realizattions i.i.d. from π
Let use define the following Monte Carlo approximation of the integral:

$$E_\pi\left(g\right) \approx \frac{1}{n}\sum_{i=1}^{n} g\left(x_i\right)$$

CONVERGENCE OF A MONTE CARLO ESTIMATOR (1/2)

Strong law of large numbers holds:

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n} g\left(x_i\right) \overset{\text{almost surely}}{=} E_\pi\left(g\right)$$



$f_1\left(\omega\right)=g\left(x_1(\omega)\right)$

$f_2\left(\omega\right)=\left[g\left(x_1(\omega)\right)+g\left(x_2(\omega)\right)\right]/2$

$\ddots$

$f\left(\omega\right)=E_\pi\left(g\right)$

CONVERGENCE OF A MONTE CARLO ESTIMATOR (2/2)

One has:

$$E\left[g\left(x_i\right)\right] = E_\pi\left[g\right]$$

$$\operatorname{var}\left(\frac{1}{n}\sum_{i=1}^{n} g\left(x_i\right)\right) = \frac{1}{n}\int\left(g\left(x\right) - E_\pi\left(g\right)\right)^2 \pi\left(x\right)dx \doteq \frac{\sigma^2}{n}$$

• difference from the true integral value has standard deviation going to zero as $n^{-1/2}$ (indipendent of dimension of x)

• good approximation of the integral requires generation of a large number of realizations/samples from π

Question: is it easy to draw independent samples from π?

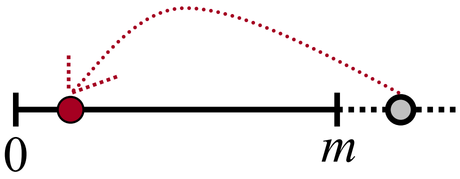
MONTE CARLO SIMULATION: COMPUTATIONAL DIFFICULTIES

Obtaining independent realizations from π is in general simple if we consider univariate distributions

- One obtains samples from uniform random variables over [*0,m*] using recursive methods by computer

$$x_{i+1}=(ax_i+c)\,\mathrm{mod}(m)\qquad a,c\in\mathbb{N}$$

x_0 = generator seed



- Then one uses the inversion method:

$$F(a)=\int_{-\infty}^{a}\pi(x)\,dx$$

If *x* has generic but invertible probability distribution *F*, and *u* is drawn from an uniform random variable over [0,1], *F*$^{-1}$*(u)* is a sample drawn from π.

In fact:

$$\Pr\left(x:=F^{-1}(u)\le a\right)=\Pr\left(u\le F(a)\right)=F(a)$$

MONTE CARLO SIMULATION: COMPUTATIONAL DIFFICULTIES

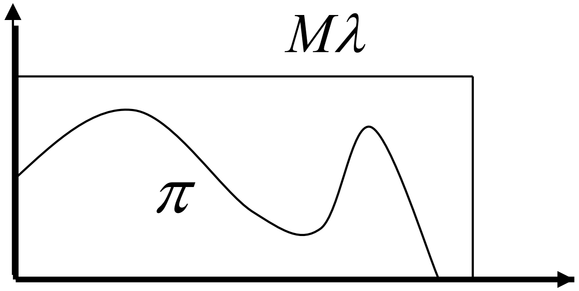
Drawing independent samples from π è is in general a very hard problem if one considers multivariate and non standard probability density functions

- Sample/resample methods
- Ratio of uniform method
- ….
- Rejection sampling

Rejection sampling (acceptance/rejection method)

1) One first obtains samples from a density *λ(x)* different from that of interest assuming that there exists a scalar *M* such that:

$$M\lambda(x) \geq \pi(x)$$



2) Then one obtains a sample from a uniform *u* in [0,1] and accpets the realization *x* from *λ* if

$$u \leq \frac{\pi(x)}{M\lambda(x)}$$

Accepted realizations are i.i.d. samples from π

Rejection sampling: observations

- Two-step method: use of an auxiliary density and then a correction method

- Choice of λ is crucial.
  It must be:

  - easy to simulate
  - easy to evaluate pointwise
  - such that it leads to a small probability of rejecting the sample (similar to π)

Rejection sampling: limitations (1/2)

Probability of accepting the sample from λ:

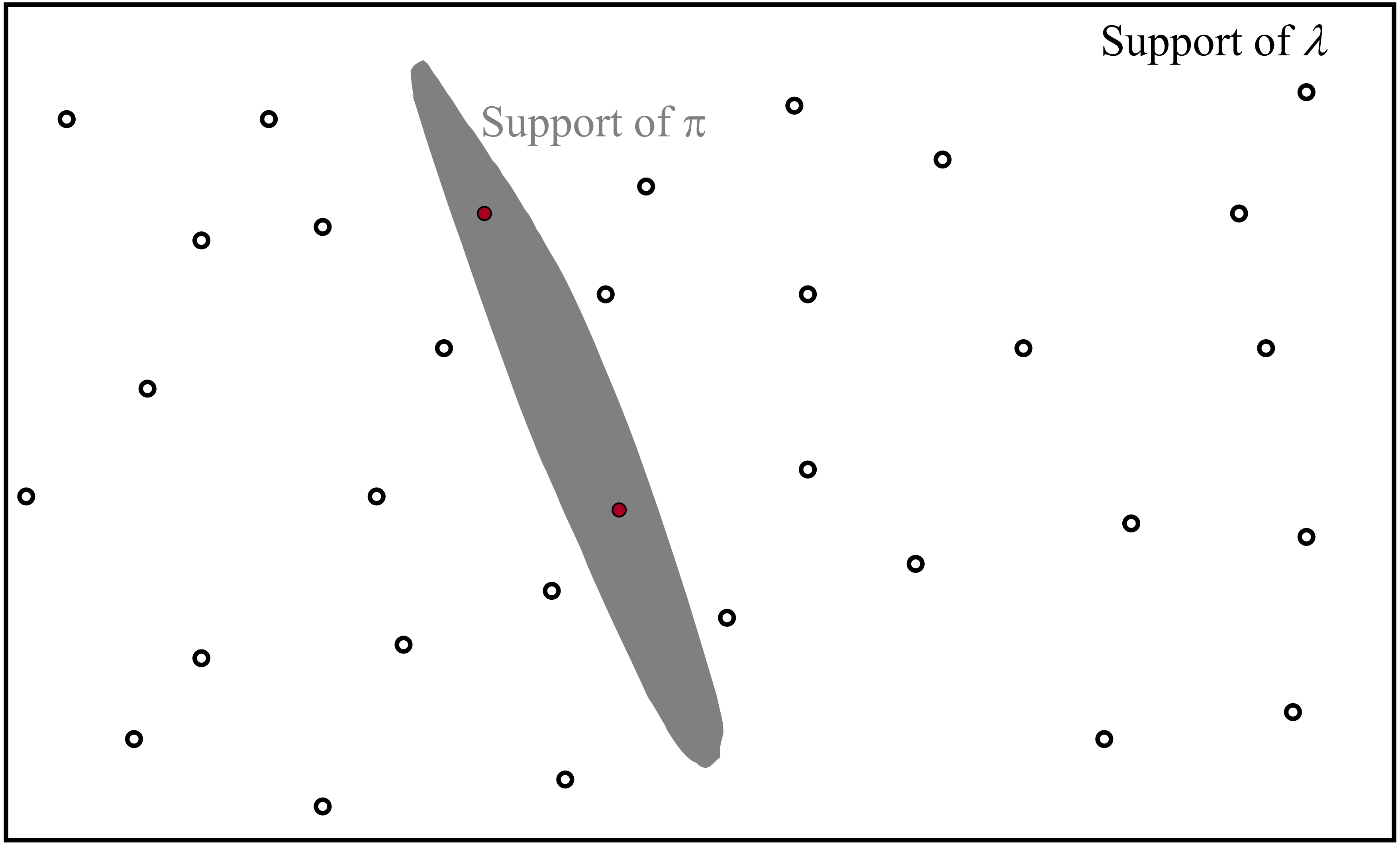
$$\int \Pr\left(u \leq \frac{\pi(x)}{M\lambda(x)} \mid x \right) \lambda(x)dx =$$

$$\int \frac{\pi(x)}{M\lambda(x)} \lambda(x)dx = \frac{1}{M}$$

In practice $M\lambda$ has to be a nice cover of π but its choice is difficult in high-dimension

Rejection sampling: limitations (2/2)

CURSE OF DIMENSIONALITY



Rejection sampling: proof of correctness

Recall that we proved that, if A is the event
`the sample from λ is accepted`,
then **Pr(A)=M^{-1}**

$$\Pr(x|A) = \frac{\boxed{\Pr(x \cap A)}}{\Pr(A)} = M\Pr(x \cap A)$$

Infinitesimal probability of
generating and accepting *x*
using rejection sampling

$$\Pr(x \cap A) = \lambda(x)dx \, \Pr\left(U \leq \frac{\pi(x)}{M\lambda(x)}\right) = \frac{\lambda(x)dx\pi(x)}{M\lambda(x)}$$

Hence, we can conclude that

$$\Pr(x|A) = M\frac{\lambda(x)dx\pi(x)}{M\lambda(x)} = \pi(x)dx$$

GENERALIZATION OF MONTE CARLO SIMULATION

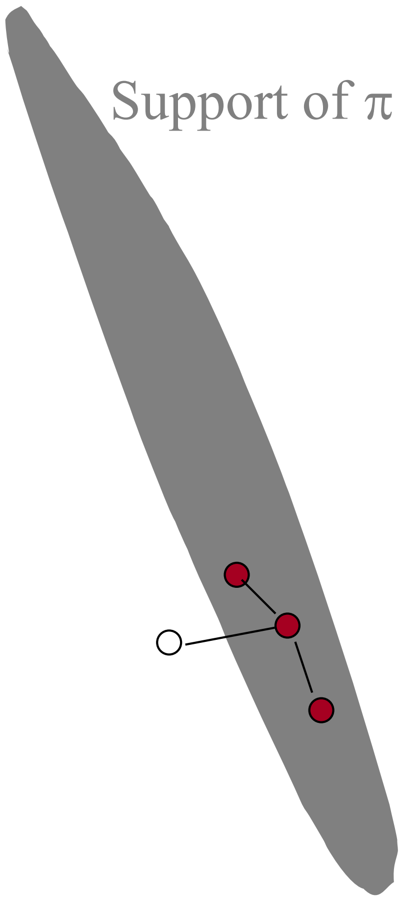
The target is $E_\pi\left(g\right) \doteq \int g\left(x\right)\pi\left(x\right)dx$

We try to extend the use of this estimator

$$E_\pi\left(g\right) \approx \frac{1}{n}\sum_{i=1}^{n} g\left(x_i\right)$$

To the case where $x^1, x^2, \ldots, x^n$ are non independent realizations from π

Advantages



Suppose we have been able to generate a sample from π: I can try to generate the next one close to it

This concept is the basis of the simulation technique called Markov chain Monte Carlo (MCMC)

SUMMARY

- Fisherian vs Bayesian estimation
- Bayesian estimation using Monte Carlo methods
- Bayesian estimation using Markov chain Monte Carlo
- On-line Bayesian estimation (particle filters)

MARKOV CHAINS

Let us consider a collection of random vectors of dimension d

$$\left\{X_t, t = 0,1,2,..\right\}$$

We say it is a Markovian collection if, considering

$$X_t \middle| X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \ldots, X_0 = x_0$$

it holds that

$$\Pr\left(X_t \in A \middle| X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, \ldots, X_0 = x_0\right) = \Pr\left(X_t \in A \middle| X_{t-1} = x_{t-1}\right)$$

$\forall\ A \in \mathrm{B}, \forall t, \forall x$

B=sigma-algebra

STATIONARY MARKOV CHAINS

The chain is stationary if the conditional probability distributions do not vary over time

$$\Pr\left(X_1 \in A \middle| X_0 = x\right) = \Pr\left(X_t \in A \middle| X_{t-1} = x\right) \doteq P\left(A, x\right)$$

$$\forall\ A \in \mathrm{B}, \forall t, \forall x$$

TRANSTION KERNEL OF A STATIONARY MARKOV CHAIN (1/3)

The transition kernel of the chain is that function *k(a,x)* s.t.:

$$P\left(A,x\right)=\int_{A}k\left(a,x\right)da$$

$$k\left(.,.\right)=p_{X_{t+1}|X_{t}}\left(.\,|\,.\right)$$

TRANSTION KERNEL OF A STATIONARY MARKOV CHAIN (2/3)

π_0 (initial probability density) and *k(.,.)* completely define the probability laws of the chain

Example

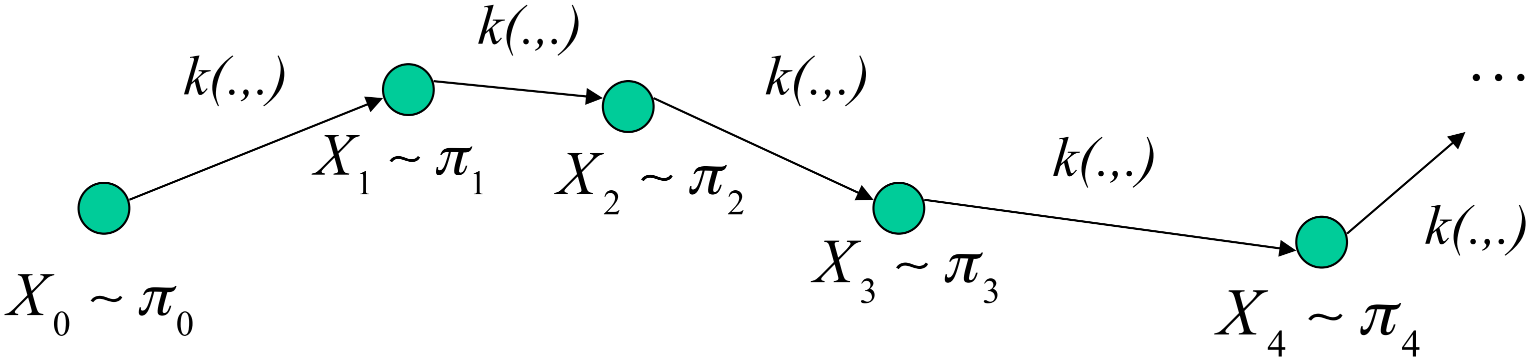
$$p_{x_0,x_1,x_2}\left(x_0,x_1,x_2\right) = p_{x_0}\left(x_0\right)p_{x_1|x_0}\left(x_1\middle|x_0\right)p_{x_2|x_1,x_0}\left(x_2\middle|x_1,x_0\right)$$
$$= \pi_0\left(x_0\right)k\left(x_1,x_0\right)k\left(x_2,x_1\right)$$

For any n-uple of vectors from the chain , the joint probability density can be computed

TRANSTION KERNEL OF A STATIONARY MARKOV CHAIN (3/3)

Assume X_{t-1} has probability density π_{t-1}
If π_t is the probability density of X_t, one has:

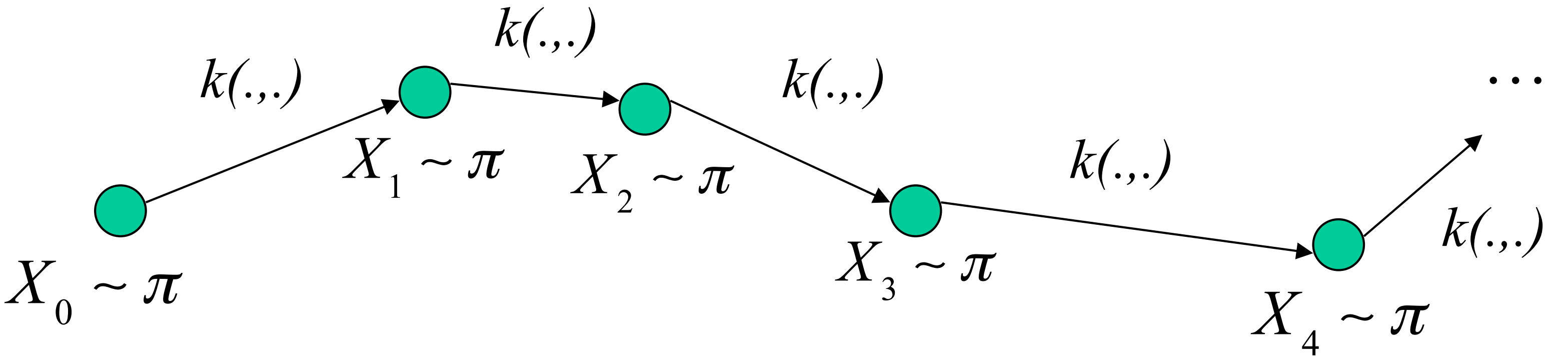
$$\pi_t(a) = \int k(a,x)\pi_{t-1}(x)dx$$



INVARIANT DENSITY OF A STATIONARY MARKOV CHAIN

π is an invariant probability density for the chain if:

$$\pi(a) = \int k(a,x)\pi(x)\,dx$$

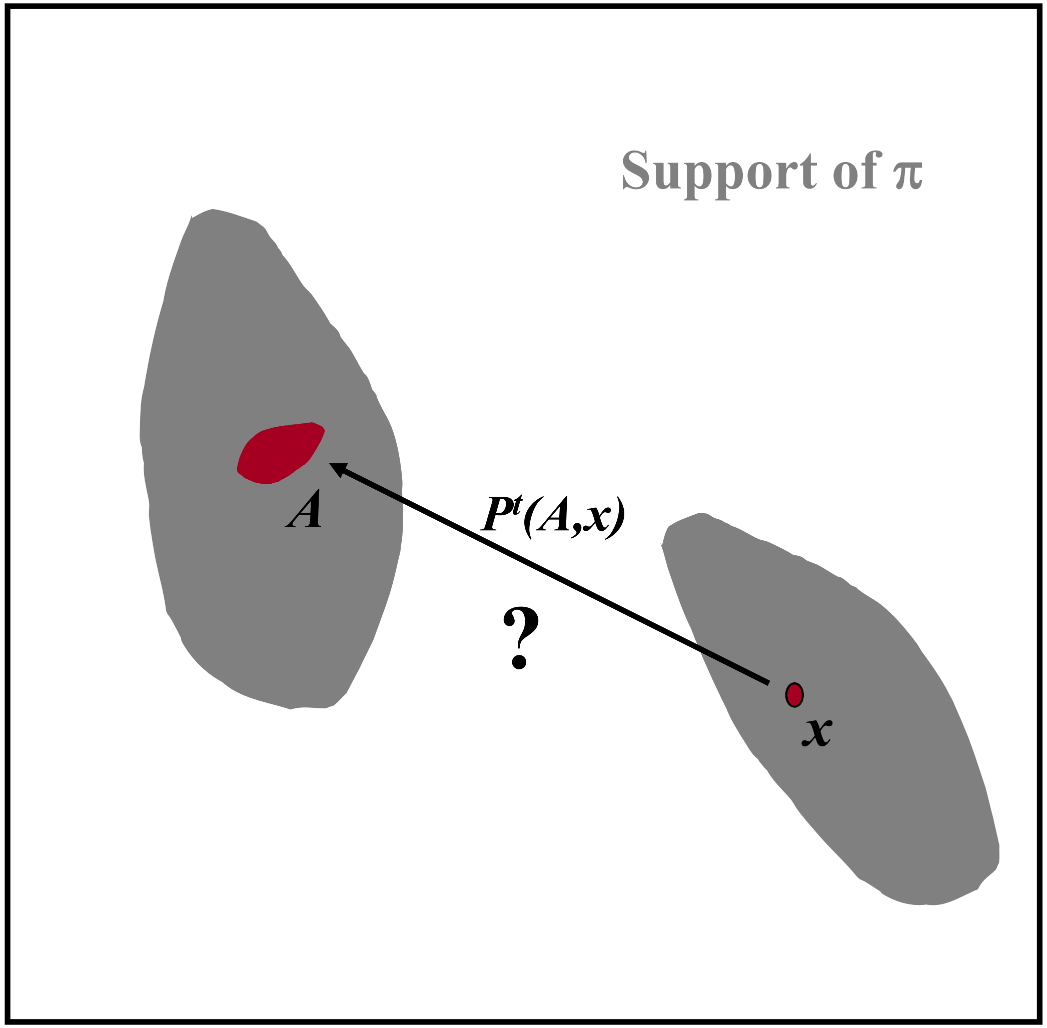


IRREDUCIBLE MARKOV CHAINS (1/2)

Let π be an invariant density for the chain: the chain is irreducible if for any x and A in **B**, with $\int_A \pi(x)dx > 0$, there exists t>0 s.t.

$$\Pr\left(X_t \in A \middle| X_0 = x\right) > 0$$

IRREDUCIBLE MARKOV CHAINS (2/2)



Irreducibility = possibility of visiting all the interesting regions of π starting from any x

STRONG LAW OF LARGE NUMBERS FOR MARKOV CHAINS

Let $\{X_t\}$ be an irreducible Markov chain
having π as invariant density.
One has:

$$\lim_{n\to\infty}\frac{1}{n}\sum_{t=0}^{n} g\left(X_t\right) \overset{q.c.}{=} E_\pi\left(g\right)$$

for any initial state
(except a set of probability zero)

MARKOV CHAIN MONTE CARLO

- Builds an irreducible Markov chain with invariant density equal to the posterior

- Uses Monte Carlo integration to obtain the quantities of interest

The first step of the algorithm can be obtained by using the Metropolis-Hastings algorithm

METROPOLIS-HASTINGS ALGORITHM (1/2)

Current chain state: $X_t=x$

- We propose a new sample $c \sim q\left(.\middle|x\right)$ where $q(.|.)$ is the proposal density of the chain
- with a certain probability $\alpha(c,x)$ we accept the candidate c, i.e. $X_{t+1}=c$
- otherwise $X_{t+1}=x$

ALGORITMO DI METROPOLIS-HASTINGS (2/2)

If the acceptance probability is:

$$\alpha(c,x) = \min\left(1, \frac{\pi(c)q(x|c)}{\pi(x)q(c|x)}\right)$$

π becomes the invariant density of the generated Markov chain

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Preliminary lemma

$$\alpha\left(c,x\right)=\min\left(1,\frac{\pi\left(c\right)q\left(x\middle|c\right)}{\pi\left(x\right)q\left(c\middle|x\right)}\right)$$

$$\Downarrow$$

$$\pi\left(X_t\right)q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)=\pi\left(X_{t+1}\right)q\left(X_t\middle|X_{t+1}\right)\alpha\left(X_t,X_{t+1}\right)$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Preliminary lemma

$$\alpha\left(c,x\right)=\min\left(1,\frac{\pi\left(c\right)q\left(x\middle|c\right)}{\pi\left(x\right)q\left(c\middle|x\right)}\right)$$

⬇

$$\pi\left(X_t\right)q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)=\pi\left(X_{t+1}\right)q\left(X_t\middle|X_{t+1}\right)\alpha\left(X_t,X_{t+1}\right)$$

Proof

Let us show that the equality holds
for any possible couple $(X_t\,,X_{t+1})$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Preliminary lemma

$$\alpha(c,x)=\min\left(1,\frac{\pi(c)q(x|c)}{\pi(x)q(c|x)}\right)$$

⬇

$$\pi(X_t)q(X_{t+1}|X_t)\alpha(X_{t+1},X_t)=\pi(X_{t+1})q(X_t|X_{t+1})\alpha(X_t,X_{t+1})$$

Proof

Let us divide all the possible
couples $(X_t\,,X_{t+1})$
into two groups

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Preliminary lemma

$$\alpha\left(c,x\right)=\min\left(1,\frac{\pi\left(c\right)q\left(x\middle|c\right)}{\pi\left(x\right)q\left(c\middle|x\right)}\right)$$

⬇

$$\pi\left(X_t\right)q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)=\pi\left(X_{t+1}\right)q\left(X_t\middle|X_{t+1}\right)\alpha\left(X_t,X_{t+1}\right)$$

Proof

Group 1: $$\frac{\pi\left(X_{t+1}\right)q\left(X_t\middle|X_{t+1}\right)}{\pi\left(X_t\right)q\left(X_{t+1}\middle|X_t\right)}\leqslant 1$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Preliminary lemma

$$\alpha(c,x) = \min\left(1, \frac{\pi(c)q(x|c)}{\pi(x)q(c|x)}\right)$$

$$\Downarrow$$

$$\pi(X_t)q(X_{t+1}|X_t)\alpha(X_{t+1},X_t) = \pi(X_{t+1})q(X_t|X_{t+1})\alpha(X_t,X_{t+1})$$

Proof

Group 1: $$\frac{\pi(X_{t+1})q(X_t|X_{t+1})}{\pi(X_t)q(X_{t+1}|X_t)} \leqslant 1$$

This implies $\alpha(X_{t+1},X_t) = \dfrac{\pi(X_{t+1})q(X_t|X_{t+1})}{\pi(X_t)q(X_{t+1}|X_t)}$ e $\alpha(X_t,X_{t+1}) = 1$

and the equality immediately follows

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Preliminary lemma

$$\alpha\left(c,x\right)=\min\left(1,\frac{\pi\left(c\right)q\left(x\middle|c\right)}{\pi\left(x\right)q\left(c\middle|x\right)}\right)$$

$$\Downarrow$$

$$\pi\left(X_t\right)q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)=\pi\left(X_{t+1}\right)q\left(X_t\middle|X_{t+1}\right)\alpha\left(X_t,X_{t+1}\right)$$

Proof

Group 2: $\dfrac{\pi\left(X_t\right)q\left(X_{t+1}\middle|X_t\right)}{\pi\left(X_{t+1}\right)q\left(X_t\middle|X_{t+1}\right)}<1$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Preliminary lemma

$$\alpha(c,x) = \min\left(1, \frac{\pi(c)q(x|c)}{\pi(x)q(c|x)}\right)$$

$$\Downarrow$$

$$\pi(X_t)q(X_{t+1}|X_t)\alpha(X_{t+1},X_t) = \pi(X_{t+1})q(X_t|X_{t+1})\alpha(X_t,X_{t+1})$$

Proof

Group 2: $$\frac{\pi(X_t)q(X_{t+1}|X_t)}{\pi(X_{t+1})q(X_t|X_{t+1})} < 1$$

This implies $\alpha(X_t,X_{t+1}) = \dfrac{\pi(X_t)q(X_{t+1}|X_t)}{\pi(X_{t+1})q(X_t|X_{t+1})}$ e $\alpha(X_{t+1},X_t) = 1$

and the equality immediately follows

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right) = q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right) + \delta\left(X_{t+1} = X_t\right)\left(1 - \int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$\underbrace{k\left(X_{t+1}\middle|X_{t}\right)}_{\substack{\text{Kernel of the Markov chain}\\ \text{describing the infinitesimal}\\ \text{probability of going from}\\ X_t \text{ to } X_{t+1}}}=q\left(X_{t+1}\middle|X_{t}\right)\alpha\left(X_{t+1},X_{t}\right)+\delta\left(X_{t+1}=X_{t}\right)\left(1-\int q\left(c\middle|X_{t}\right)\alpha\left(c,X_{t}\right)dc\right)$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=\underbrace{q\left(X_{t+1}\middle|X_t\right)}_{\substack{\text{Infinitesimal probability of}\\ \text{proposing as candidate } X_{t+1}\\ \text{if the current state is } X_t}}\alpha\left(X_{t+1},X_t\right)+\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=q\left(X_{t+1}\middle|X_t\right)\underbrace{\alpha\left(X_{t+1},X_t\right)}_{\substack{\text{Probability of}\\ \text{accepting as candidate } X_{t+1}\\ \text{if the current state is } X_t}}+\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=\underbrace{q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)}+\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)$$

Infinitesimal probability
of going to X_{t+1} from X_t
through the acceptance
of the candidate

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=\underbrace{q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)}+\delta\left(X_{t+1}=X_t\right)\left(1-\underbrace{\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc}\right)$$

Infinitesimal probability of going to X_{t+1} from X_t through the acceptance of the candidate

Probability of accepting a sample (before generating it!) If the current state is X_t (generated by q with acceptance probability given by α)

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=\underbrace{q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)}_{\begin{array}{l}\text{Infinitesimal probability}\\ \text{of going to } X_{t+1} \text{ from } X_t\\ \text{through the acceptance}\\ \text{of the candidate}\end{array}}+\delta\left(X_{t+1}=X_t\right)\underbrace{\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)}_{\begin{array}{c}\text{Probability of}\\ \text{remaining at } X_t\end{array}}$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=\underbrace{q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)}+\delta\left(X_{t+1}=X_t\right)\underbrace{\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)}$$

Infinitesimal probability
of going to X_{t+1} from X_t
through the acceptance
of the candidate

And also Dirac delta area
which is equal to the probability
of going from X_t to X_{t+1}
by refusing the candidate:
contribution to $k(X_{t+1}|X_t)$
only if $X_{t+1}=X_t$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=\underbrace{q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)}+\underbrace{\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)}$$

Infinitesimal probability of going to X_{t+1} from X_t through the acceptance of the candidate

Hence, the second contribution is the Dirac delta with that area and centred on X_t

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=\underbrace{q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)}_{\text{Infinitesimal probability of going to } X_{t+1} \text{ from } X_t \text{ through the acceptance of the candidate}}+\underbrace{\underbrace{\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)}_{\text{Hence, the second contribution is the Dirac delta with that area and centred on } X_t}}_{\text{Symmetric term in } X_{t+1} \text{ and } X_t}$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)+\underbrace{\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)}_{\text{Symmetric term in } X_{t+1} \text{ and } X_t}$$

$+$ (lemma)

$$\pi\left(X_t\right)q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)=\pi\left(X_{t+1}\right)q\left(X_t\middle|X_{t+1}\right)\alpha\left(X_t,X_{t+1}\right)$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)+\underbrace{\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)}_{\text{Symmetric term in } X_{t+1} \text{ and } X_t}$$

$+$ (lemma)

$$\pi\left(X_t\right)q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)=\pi\left(X_{t+1}\right)q\left(X_t\middle|X_{t+1}\right)\alpha\left(X_t,X_{t+1}\right)$$

$$\Downarrow$$

$$\pi\left(X_t\right)k\left(X_{t+1}\middle|X_t\right)=\pi\left(X_{t+1}\right)k\left(X_t\middle|X_{t+1}\right)$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)+\underbrace{\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)}_{\text{Symmetric term in } X_{t+1} \text{ and } X_t}$$

+ (lemma)

$$\pi\left(X_t\right)q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)=\pi\left(X_{t+1}\right)q\left(X_t\middle|X_{t+1}\right)\alpha\left(X_t,X_{t+1}\right)$$

$$\Downarrow$$

$$\pi\left(X_t\right)k\left(X_{t+1}\middle|X_t\right)=\pi\left(X_{t+1}\right)k\left(X_t\middle|X_{t+1}\right)$$

Immediately derives from the symmetry of the term $\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)$
that defines the kernel of the chain

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1} \middle| X_t\right) = q\left(X_{t+1} \middle| X_t\right)\alpha\left(X_{t+1}, X_t\right) + \underbrace{\delta\left(X_{t+1} = X_t\right)\left(1 - \int q\left(c \middle| X_t\right)\alpha\left(c, X_t\right)dc\right)}_{\text{Symmetric term in } X_{t+1} \text{ and } X_t}$$

$+$ (lemma)

$$\pi\left(X_t\right)q\left(X_{t+1} \middle| X_t\right)\alpha\left(X_{t+1}, X_t\right) = \pi\left(X_{t+1}\right)q\left(X_t \middle| X_{t+1}\right)\alpha\left(X_t, X_{t+1}\right)$$

$\Downarrow$

$$\pi\left(X_t\right)k\left(X_{t+1} \middle| X_t\right) = \pi\left(X_{t+1}\right)k\left(X_t \middle| X_{t+1}\right)$$

$$\int \pi\left(X_t\right)k\left(X_{t+1} \middle| X_t\right)dX_t = \pi\left(X_{t+1}\right)\int k\left(X_t \middle| X_{t+1}\right)dX_t$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)+\underbrace{\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)}_{\text{Symmetric term in } X_{t+1} \text{ and } X_t}$$

$+$ (lemma)

$$\pi\left(X_t\right)q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)=\pi\left(X_{t+1}\right)q\left(X_t\middle|X_{t+1}\right)\alpha\left(X_t,X_{t+1}\right)$$

$$\Downarrow$$

$$\pi\left(X_t\right)k\left(X_{t+1}\middle|X_t\right)=\pi\left(X_{t+1}\right)k\left(X_t\middle|X_{t+1}\right)$$

$$\int \pi\left(X_t\right)k\left(X_{t+1}\middle|X_t\right)dX_t=\pi\left(X_{t+1}\right)\boxed{\int k\left(X_t\middle|X_{t+1}\right)dX_t} =1$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)+\underbrace{\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)}_{\text{Symmetric term in } X_{t+1} \text{ and } X_t}$$

$+$ (lemma)

$$\pi\left(X_t\right)q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)=\pi\left(X_{t+1}\right)q\left(X_t\middle|X_{t+1}\right)\alpha\left(X_t,X_{t+1}\right)$$

⬇

$$\int \pi\left(X_t\right)k\left(X_{t+1}\middle|X_t\right)dX_t=\pi\left(X_{t+1}\right)$$

METROPOLIS-HASTINGS ALGORITHM: PROOF OF CORRECTNESS

Kernel of the chain

$$k\left(X_{t+1}\middle|X_t\right)=q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)+\underbrace{\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)}_{\text{Symmetric term in } X_{t+1} \text{ and } X_t}$$

+ (lemma)

$$\pi\left(X_t\right)q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)=\pi\left(X_{t+1}\right)q\left(X_t\middle|X_{t+1}\right)\alpha\left(X_t,X_{t+1}\right)$$

$$\Downarrow$$

$$\int \pi\left(X_t\right)k\left(X_{t+1}\middle|X_t\right)dX_t=\pi\left(X_{t+1}\right)$$

Hence, π is indeed the invariant density

OBSERVATIONS (1/2)

- differently from the rejection sampling:

  - the chain always moves
    (if the sample is refused, the next state is equal to the previous one)

  - in general, the algorithm is able to return correlated
    (but not independent) samples from π

OBSERVATIONS (2/2)

- the target density π can be known apart from a normalization factor

$$\alpha\left(c,x\right)=\min\left(1,\frac{\pi\left(c\right)q\left(x\middle|c\right)}{\pi\left(x\right)q\left(c\middle|x\right)}\right) \qquad \pi\left(x\right)\propto p_{y|x}(y|x)p_x(x)$$

- theoretically, the algorithm works for any *q(.|.)* (if the chain is irreducible), but in practice the choice of *q* is crucial

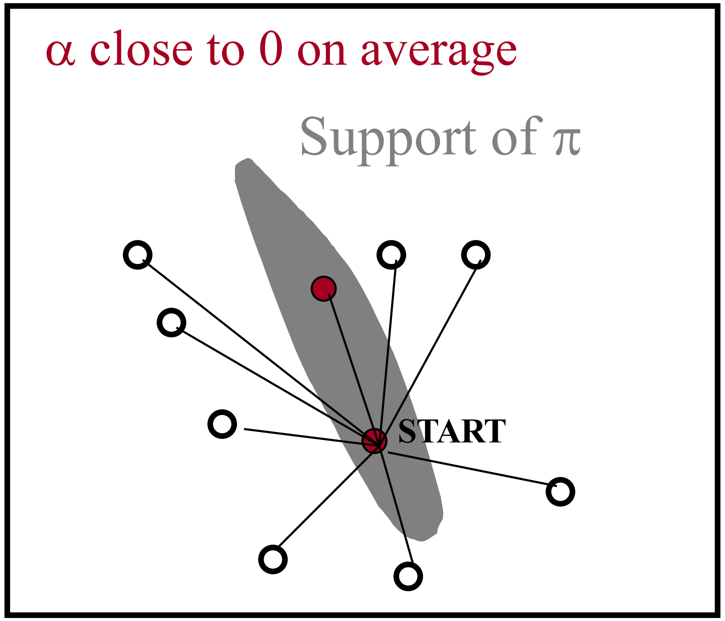
CHOICE OF *q(.|.)* (1/2)

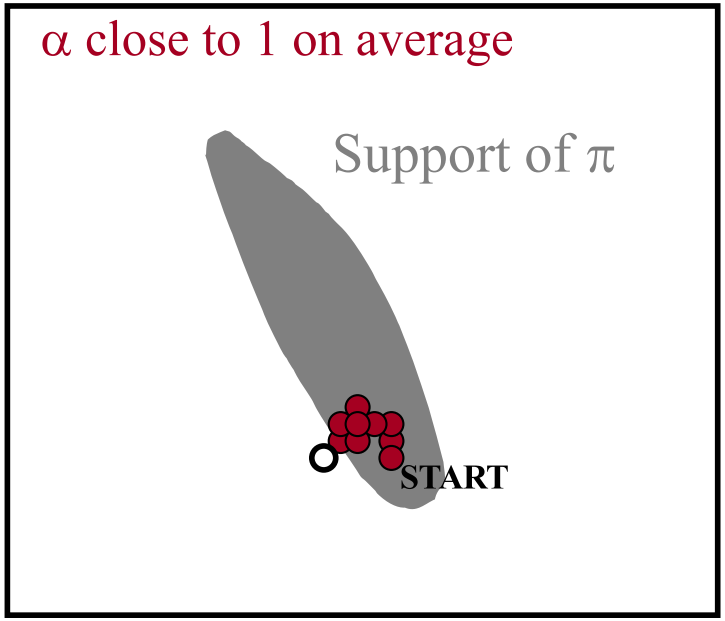
q(.|.) must

- be easy to sample

- be simple to be evaluated pointwise

- able to quickly explore the support of π





CHOICE OF *q(.|.)* (2/2)

Often, it is useful to adopt random-walk proposals

$$q\left(c\middle|x\right) = f\left(\left|c-x\right|\right) = q\left(x\middle|c\right)$$

$$\left.\begin{aligned} &c = x_t + \varepsilon \\ &\varepsilon \sim N\left(0,\Sigma\right) \end{aligned}\right\} \quad q\left(c\middle|x\right) = N\left(x,\Sigma\right)$$

- Σ provides information as how to move locally around the current point

- the acceptance probability becomes

$$\alpha\left(c,x\right) = \min\left(1,\frac{\pi\left(c\right)}{\pi\left(x\right)}\right)$$

Strategies to choose Σ(1/2):

In high-dimension it is worth performing an explorative analysis of π

Example #1:

- Define a diagonal matrix Σ with small variances values

$$\Sigma_{start} = \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \cdots\cdots & \\ & & & \sigma_n^2 \end{bmatrix}$$

- generate Markov chains and monitor the results. Change the variances so as to obtain an acceptance rate around 30-40%

- generate the Markov chain using the matrix Σ obtained by the pilot analysis

Strategies to choose Σ (2/2):

Example #2:

- Calculate the posterior maxima and obtain information on the a posteriori correlation of the components of x

$$\Sigma \propto \left[-\frac{\partial^2 \log \pi}{\partial x^T \partial x} \Big| \hat{x} \right]^{-1}$$

$$\hat{x} = \arg\min_x - \log\left(\pi\right)$$

- the scale factor is chosen so as to obtain an acceptance rate around 30-40%

MCMC: SIMULATED EXAMPLE (1 of 4)

Aim: to reconstruct in sampled form a Gaussian distribution

TARGET

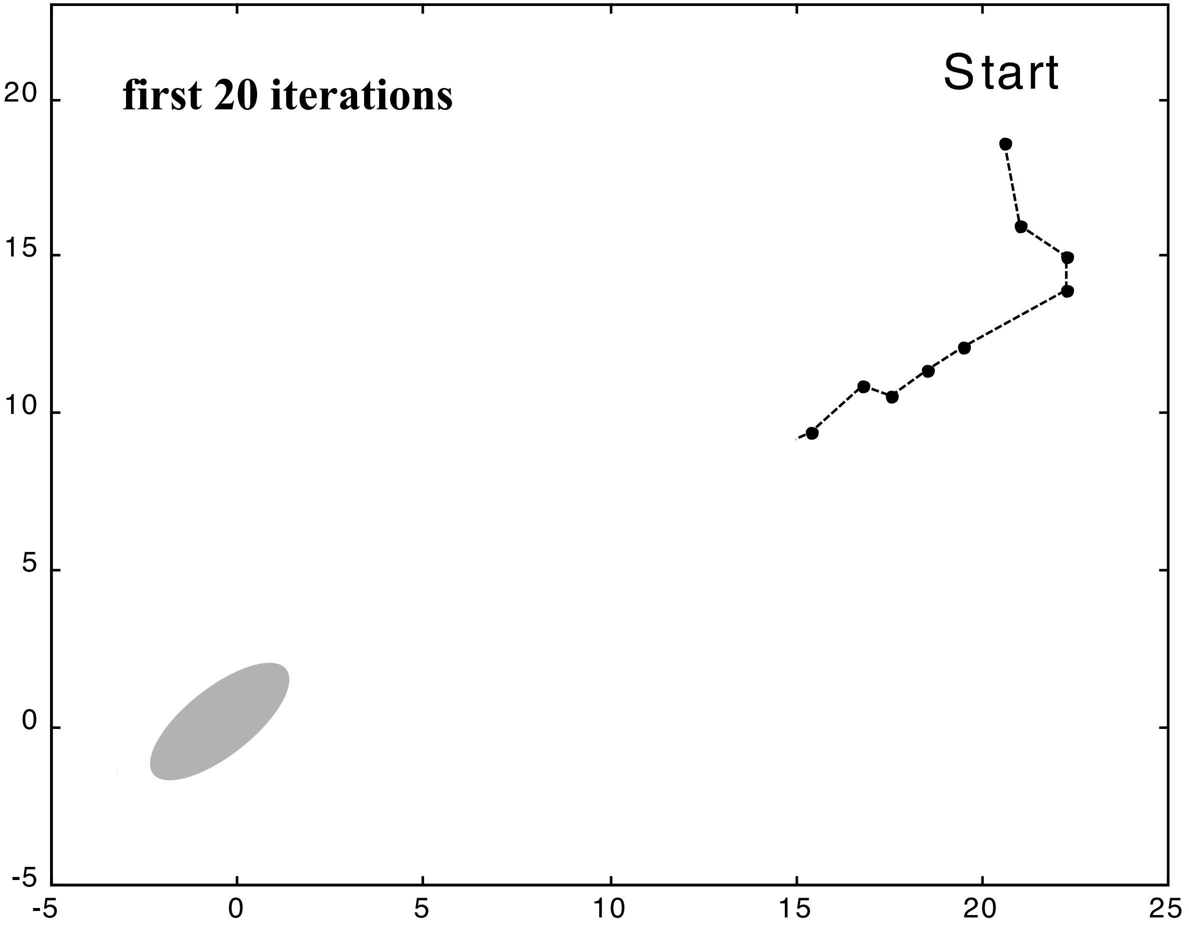
$$\pi = N\left(0, A\right)$$

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$$

PROPOSAL

$$q\left(y \middle| x\right) \sim N\left(x, B\right)$$

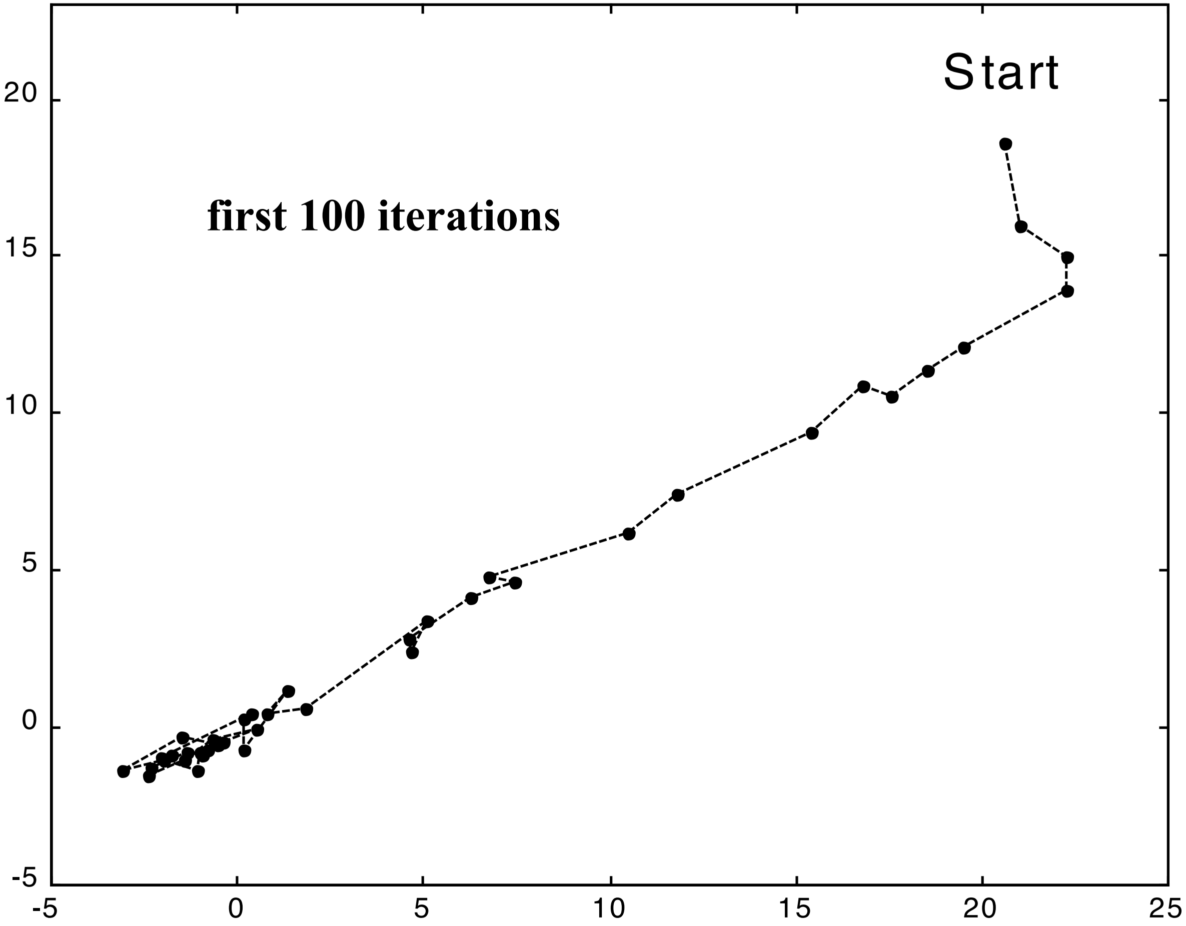
$$B = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$



MCMC: SIMULATED EXAMPLE (2 of 4)

Aim: to reconstruct in sampled form a Gaussian distribution of zero mean and covariance

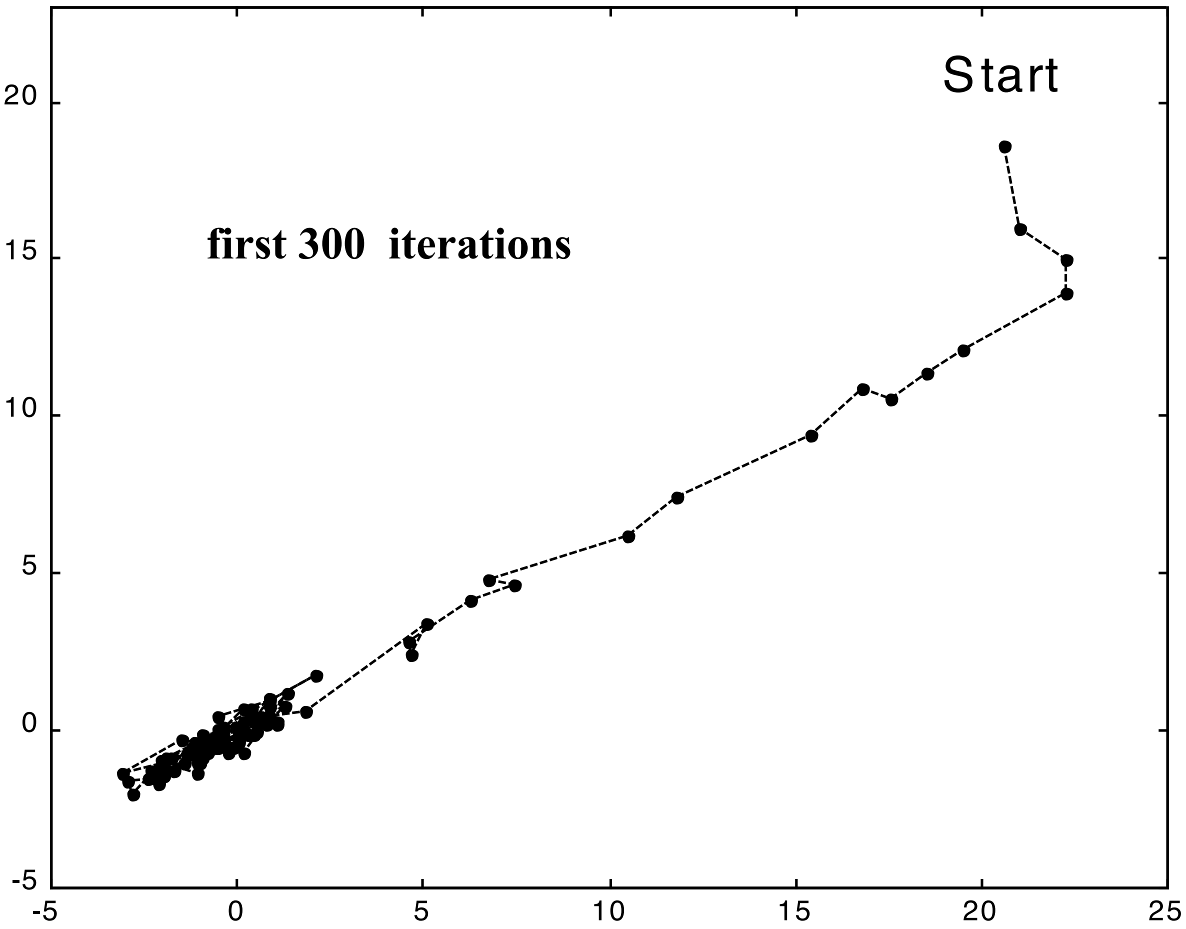
$$A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$$



MCMC: SIMULATED EXAMPLE (3 of 4)

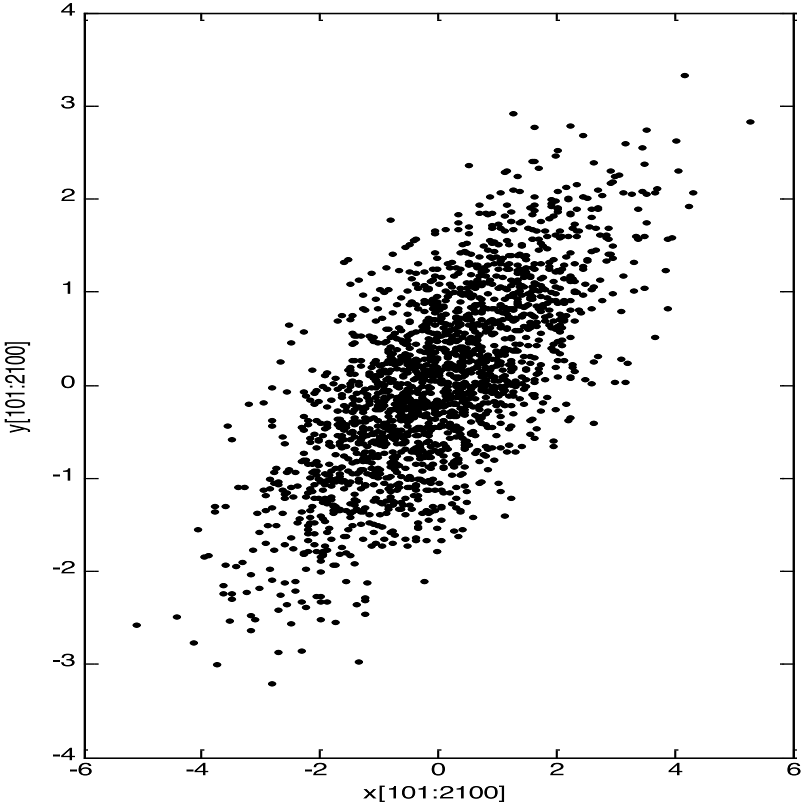
Aim: to reconstruct in sampled form a Gaussian distribution of zero mean and covariance

$$A = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$$

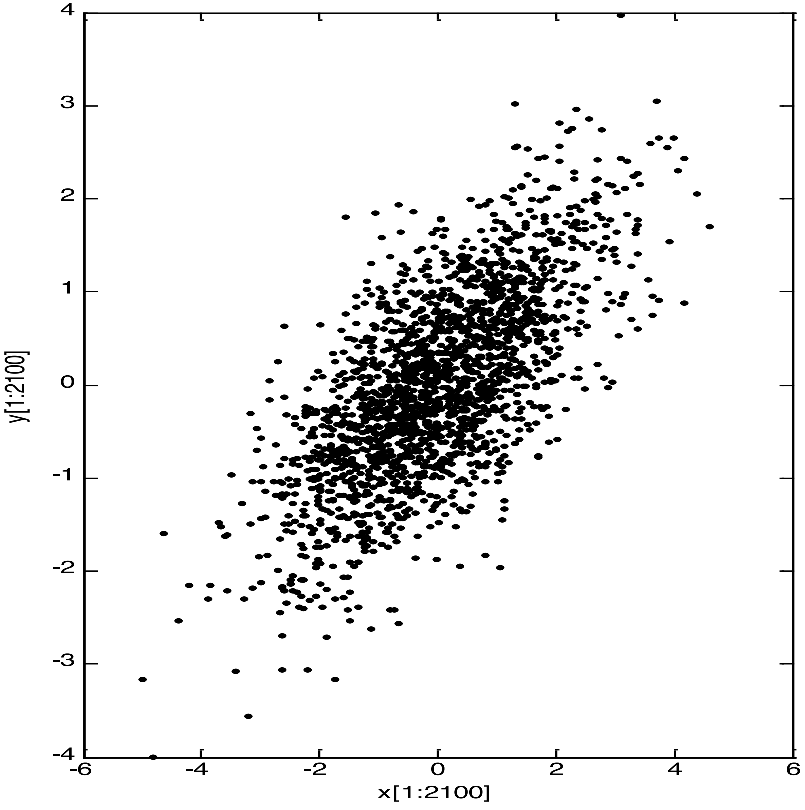


MCMC: SIMULATED EXAMPLE (4 of 4)

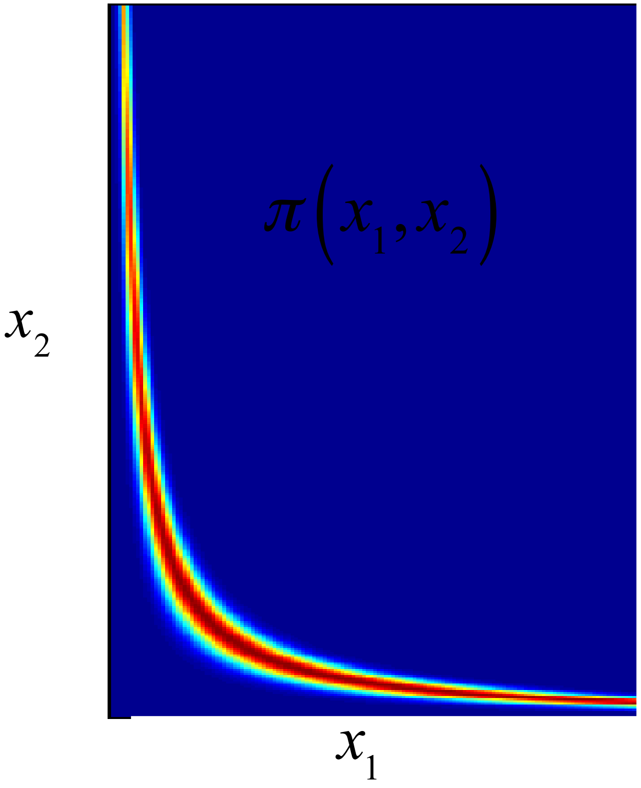
2000 MCMC samples
(iterations 101-2100)



2000 independent samples



CHOICE OF *q(.|.)*: BLOCK SCHEMES



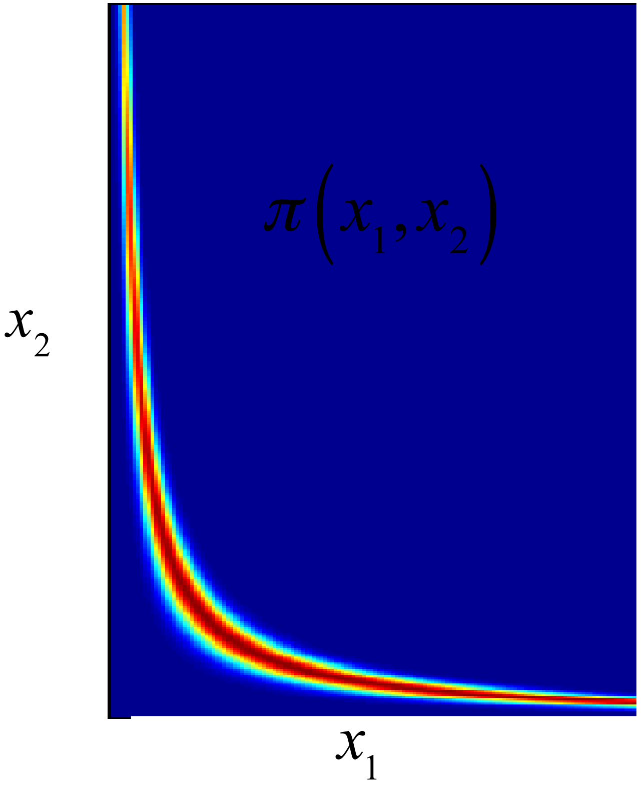
$$x = \left[x_1 \; x_2 \right]$$

Hard case:

- strong a posteriori correlation
- correlation much varies along the parameter space

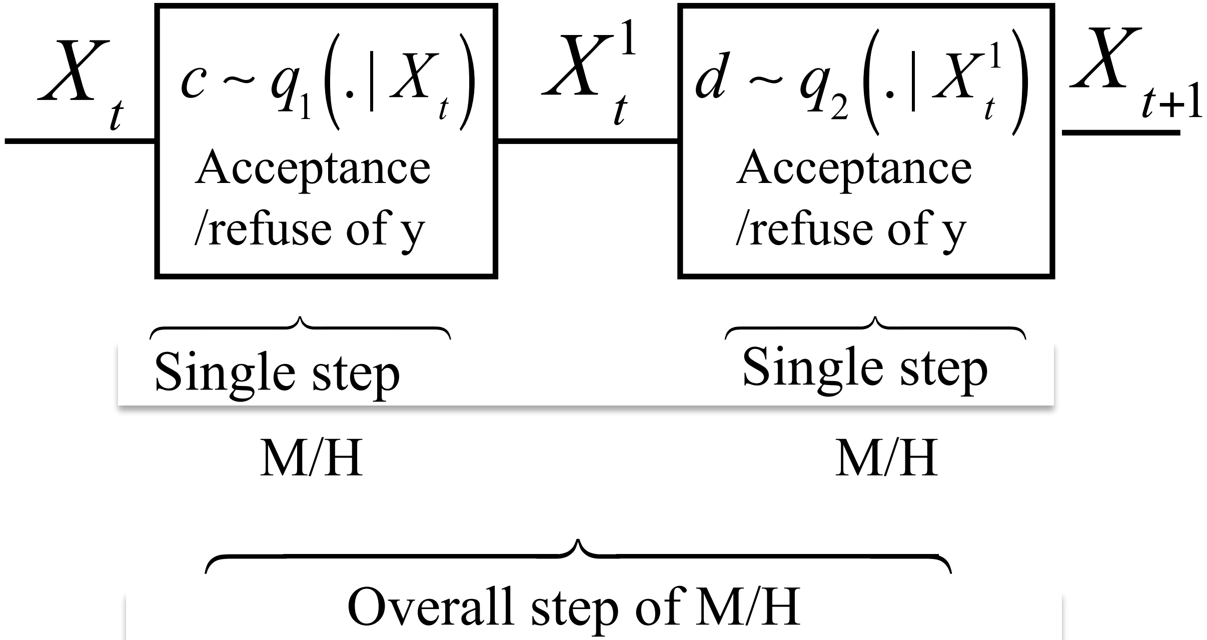
- difficult to move simultaneously *x_1* and *x_2* with a suitable probability c accepting the generated sample

CHOICE OF *q(.|.)*: BLOCK SCHEMES



One solution is to move separately x_1 and x_2 by defining two proposal densities q_1 and q_2

CHOICE OF *q(.|.)*: BLOCK SCHEMES



X_t

$c \sim q_1\left(. \mid X_t\right)$

Acceptance /refuse of y

X_t^1

$d \sim q_2\left(. \mid X_t^1\right)$

Acceptance /refuse of y

X_{t+1}

Single step

Single step

M/H

M/H

Overall step of M/H

CONVERGENCE DIAGNOSTICS (1/2)

Once an MCMC simulation starts,

how many iterations do we need to perform?

- the chain kernel assumes a complex form

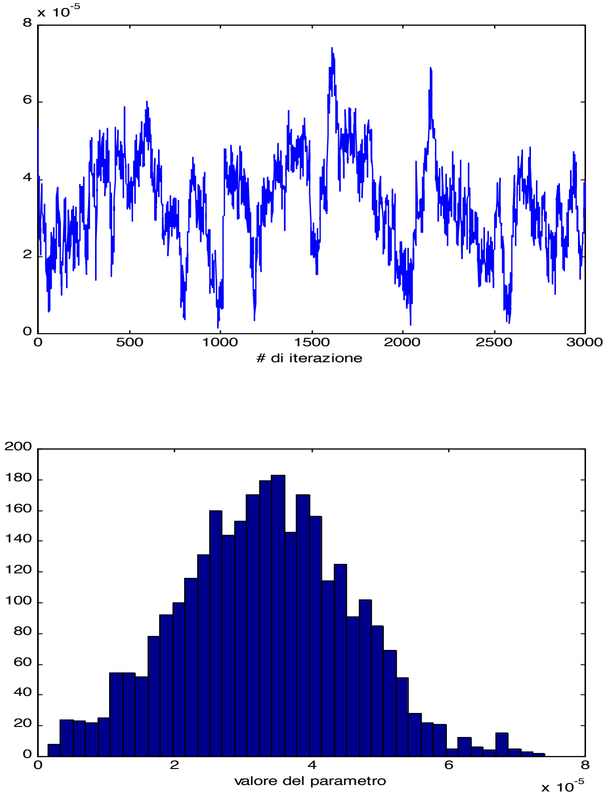
$$k\left(X_{t+1}\middle|X_t\right)=q\left(X_{t+1}\middle|X_t\right)\alpha\left(X_{t+1},X_t\right)+\delta\left(X_{t+1}=X_t\right)\left(1-\int q\left(c\middle|X_t\right)\alpha\left(c,X_t\right)dc\right)$$

and is thus complex to analyze convergence under a theoretical viewpoint

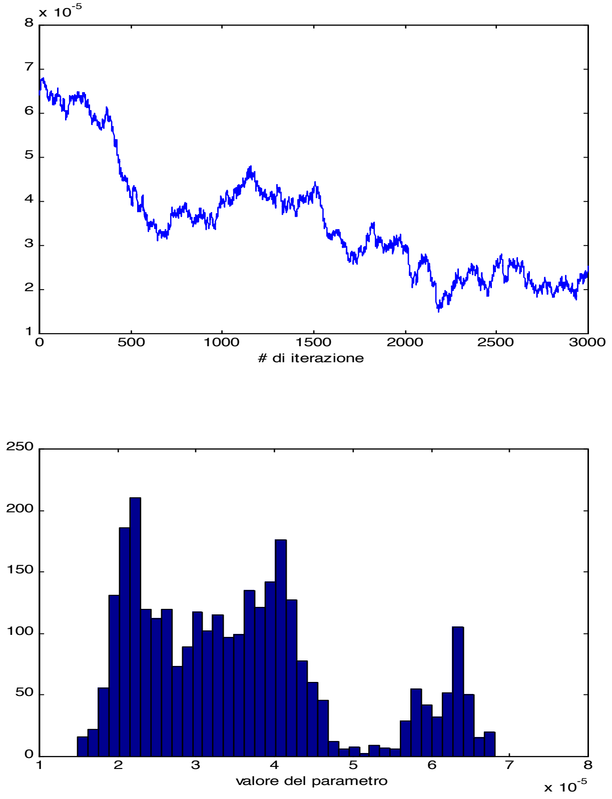
CONVERGENCE DIAGNOSTICS (2/2)

- in practice one obtains information on the Markov chain convergence by analyzing the statistical properties of the generated samples

Good convergence



Bad convergence



MINIMAL MODEL EQUATIONS

G(t) = glucose plasma concentration
I(t) = insulin plasma concentration

$$\dot{G}(t) = -(S_G + X(t))G(t) + S_G G_b \qquad G(0) = G_0$$

$$\dot{X}(t) = -p_2\{X(t) - S_I[I(t) - I_b]\} \qquad X(0) = 0$$

- Ithe model contains 4 parameters that are not directly measurable and have to be estimated from glucose samples

- *I(t)* èis assumed perfectly known by linear interpolation of its noisy samples. The model thus turns out to be a priori identifiable.

MM PARAMETER ESTIMATION USING FISHER

$$y_i = h(t_i, x) + v_i$$

i=1,2,..,N

Gaussian error (CV%=2)

$$v \sim N(0, \Sigma_v) \qquad \Sigma_v(i,i) = \sigma_i^2$$

Glucose prediction

$$x = [S_I, p_2, S_G, G_0]$$

$$L(x) = \frac{1}{(2\pi)^{N/2} \det(\Sigma_v)^{1/2}} e^{-\frac{1}{2}\sum_{i=1}^{N}\left(\frac{y_i - h(t_i, x)}{\sigma_i}\right)^2}$$

LIKELIHOOD

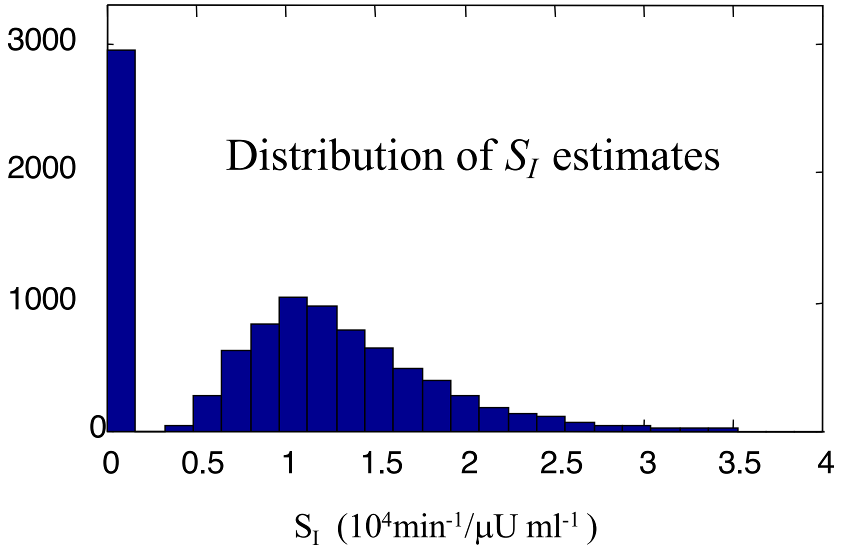
$$x^{ML} = \arg\max_p L(p)$$

MAXIMUM LIKELIHOOD ESTIMATE

DIFFICULTIES ENCOUNTERED BY THE FISHERIAN APPROACH (1/2)

THE S_I=0 PROBLEM

In almost 40% of diabetic subjects the model returns an S_I estimate equal to zero



DIFFICULTIES ENCOUNTERED BY THE FISHERIAN APPROACH (2/2)

OTHER PROBLEMS

- S_I estimate may turn out very small and much uncertain (in particular in diabetic subjects)

- S_I pestimate may turn out much uncertain and not realistic, assuming very large value

- also p_2 estimate may turn out much uncertain

REPRODUCING FISHER DIFFICULTIES VIA COMPUTER SIMULATION

$$y_i = h(t_i, x) + v_i$$

$$x = [S_I, p_2, S_G, G_0]$$

Let us fix these parameters
to realistic values
for a diabetic subject:

$$S_I = 0.7e-4\, min^{-1} / \mu U ml^{-1}$$

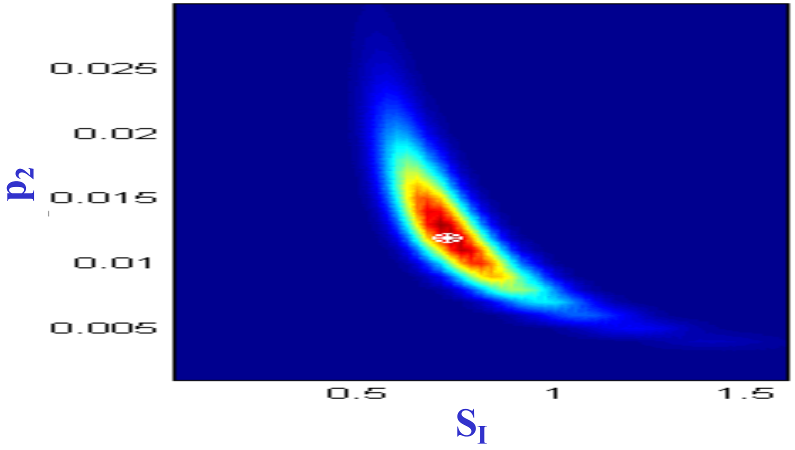
$$p_2 = 0.01 min^{-1}$$

We generate 1000 realizations
of the measurement error

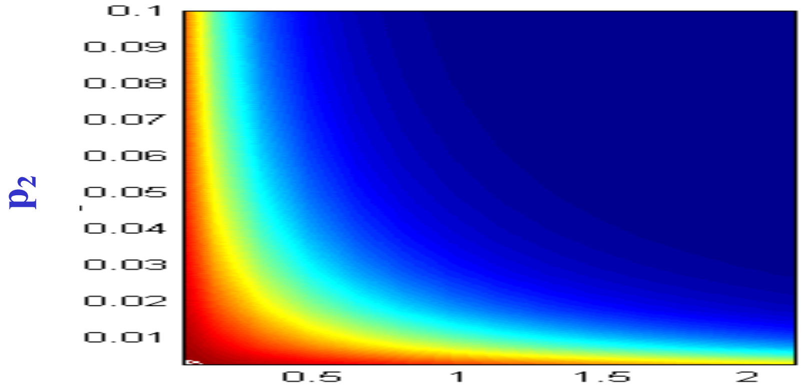
$$v \sim N(0, \Sigma_v)$$

and after each noise realization
we obtain the maximum likelihood
estimate of the MM parameters

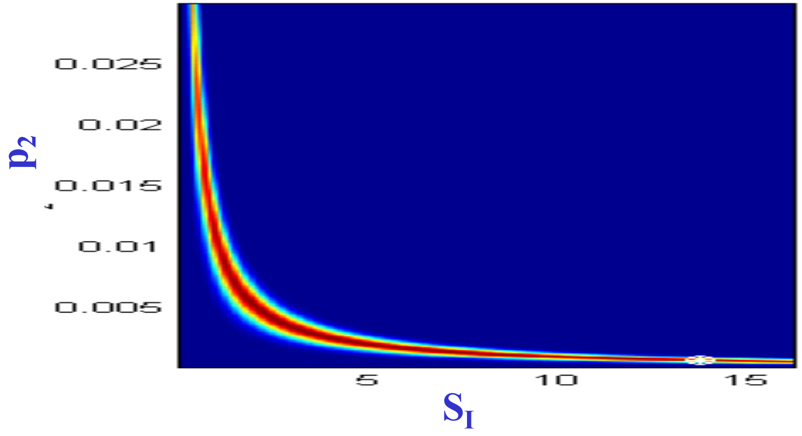
Likelihood shapes in 3 significant cases



Works well

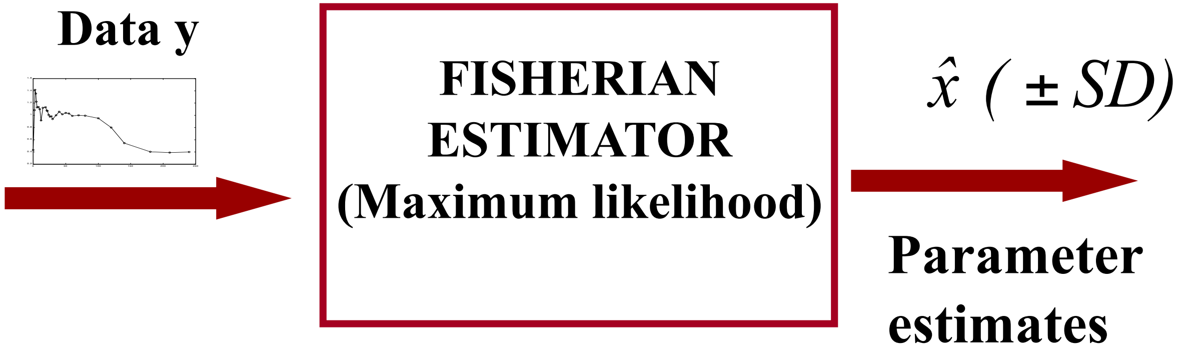


S_I=0 and also *p_2* is much uncertain



Large *S_I*, not realistic, and small *p_2* with large uncertainty

Question: passing from Fisher….



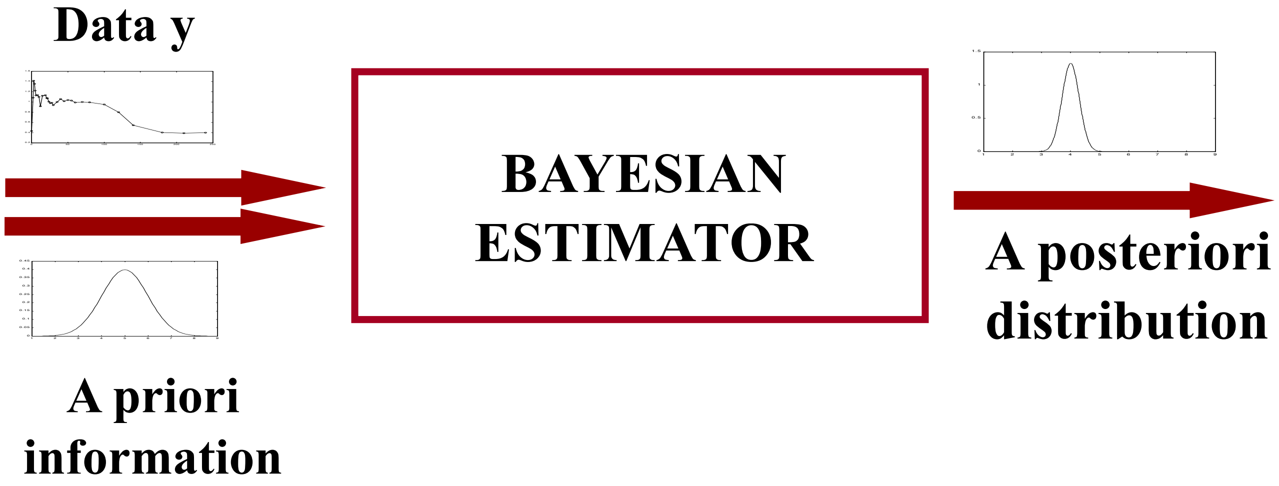
Data y

FISHERIAN ESTIMATOR (Maximum likelihood)

$\hat{x}\ (\pm SD)$

Parameter estimates

to Bayes



Data y

BAYESIAN ESTIMATOR

A priori information

A posteriori distribution

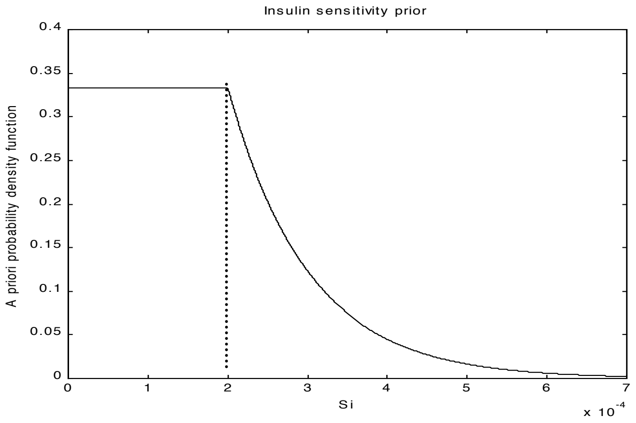
can we overcome the identification problems?

Bayesian strategy: definition of the prior

1. Let us define a prior for S_I based on the many studies reported in the literature

$$p_{S_I}\left(S_I\right) \propto \begin{cases} 0 & \text{se } S_I < 0 \\ 1 & \text{se } 0 \le S_I \le 2e-4 \\ e^{-\frac{\left(S_I-\left(2e-4\right)\right)}{1e-4}} & \text{se } S_I > 2e-4 \end{cases}$$

Prior for S_I

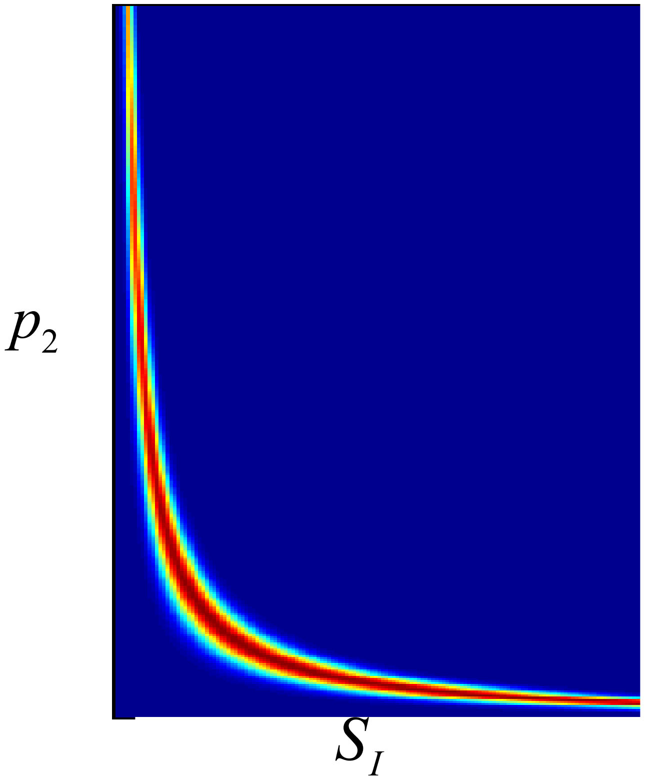


S_I (10^4min^{-1}/μU ml^{-1})

2. The prior is then poorly informative regarding S_G, p_2 e G_0 including just nonnegativity information

$$p_{S_I,S_G,p_2,G_0} \propto p_{S_I}\left(S_I\right)\chi\left(S_G \ge 0\right)\chi\left(p_2 \ge 0\right)\chi\left(G_0 \ge 0\right)$$

Bayesian strategy:
definition of the MCMC scheme



S_I and p_2 are often strongly correlated a posteriori. It is convenient to update them separately by defining two proposal densities:

$$\Sigma_1 = \begin{pmatrix} \sigma_{S_I}^2 & 0 & 0 \\ 0 & \sigma_{G_0}^2 & 0 \\ 0 & 0 & \sigma_{S_G}^2 \end{pmatrix}$$

$$q_1\left(S_I^{new}, G_0^{new}, S_G^{new} \mid S_I^{old}, G_0^{old}, S_G^{old}\right)$$

$$= N\left([S_I^{old} \;\; G_0^{old} \;\; S_G^{old}], \Sigma_1\right)$$

$$\Sigma_2 = \left(\sigma_{p_2}^2\right)$$

$$q_2\left(p_2^{new} \mid p_2^{old}\right)$$

$$= N\left(p_2^{old}, \Sigma_2\right)$$

COMPUTATIONAL COMPLEXITY

Related to the posterior evaluation at any MCMC iteration, i.e. to the cost of solving the differential equations of the model for any new proposed sample

$$\dot{G}(t) = -(S_G + X(t))G(t) + S_G G_b \qquad G(0) = G_0$$

$$\dot{X}(t) = -p_2\{X(t) - S_I[I(t) - I_b]\} \qquad X(0) = 0$$

Define:

$$Z(t) = \int_0^t X(t)\,dt$$

$$= \int_0^t \int_0^t S_I p_2 e^{-p_2(t-\tau)} \left(I(\tau) - I_b\right) d\tau\, dt = \int_0^t S_I \left(1 - e^{-p_2(t-\tau)}\right)\left(I(\tau) - I_b\right) d\tau$$

One has: $G(t) = G_0 e^{-S_G t - Z(t)} + S_G G_b \int_0^t e^{-S_G(t-\tau) - Z(t) + Z(\tau)}\, d\tau$

Glucose prediction in closed form

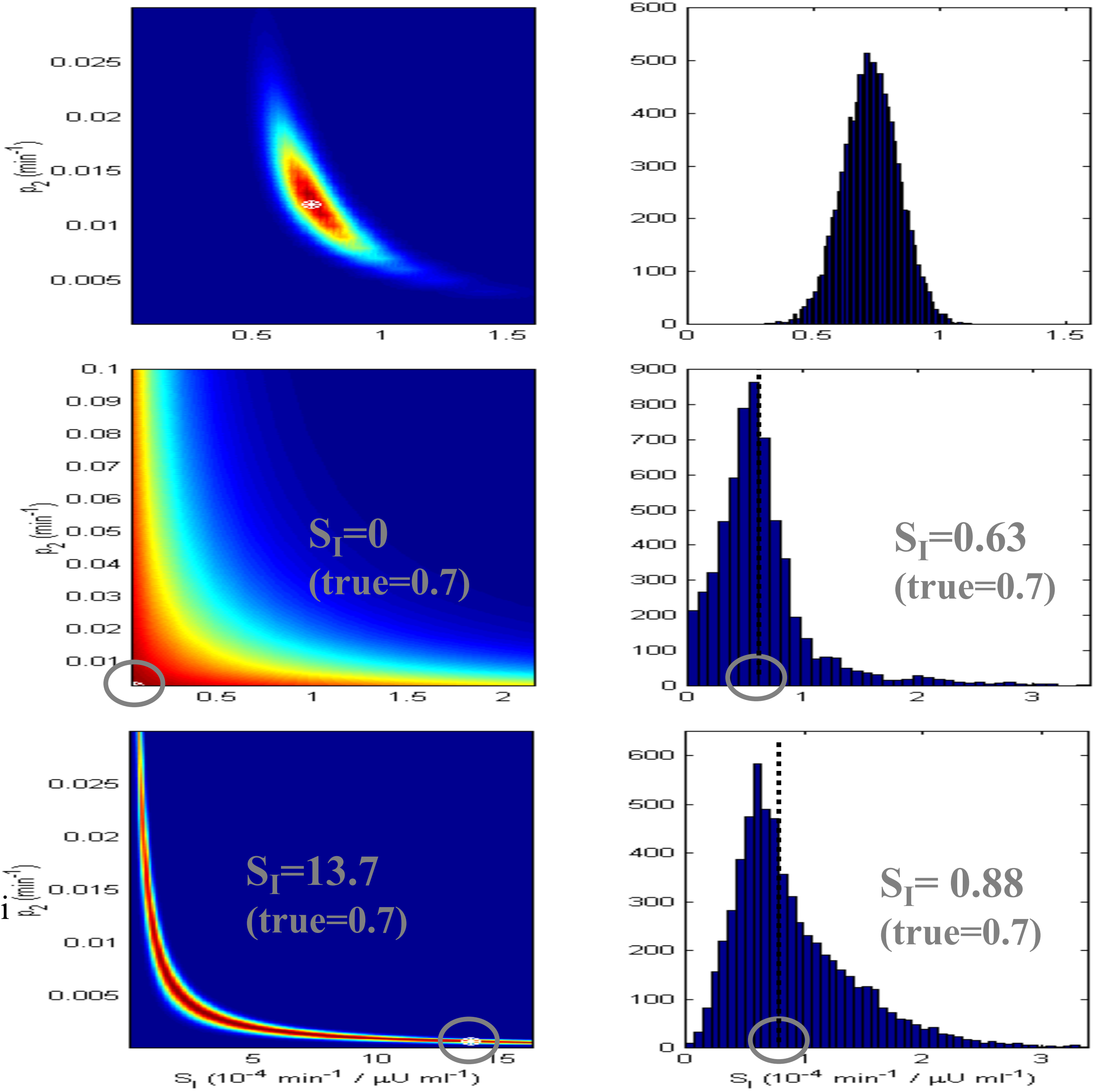
RESULTS

FISHER (ML)

$\hat{x} = \arg\max p_{y|p}\left(y|p\right)$

BAYES S_I POSTERIOR

$E\left[x|y\right] = \int x p_{x|y}\left(x|y\right)dx$



Use of a Bayesian estimator id key in the last two situations

Pillonetto G. , G. Sparacino and C. Cobelli
Numerical non identifiability regions of the minimal model of glucose kinetics: superiority of Bayesian estimation,
Mathematical Biosciences, 2003

SUMMARY: 1000 SYNTHETIC SUBJECTS

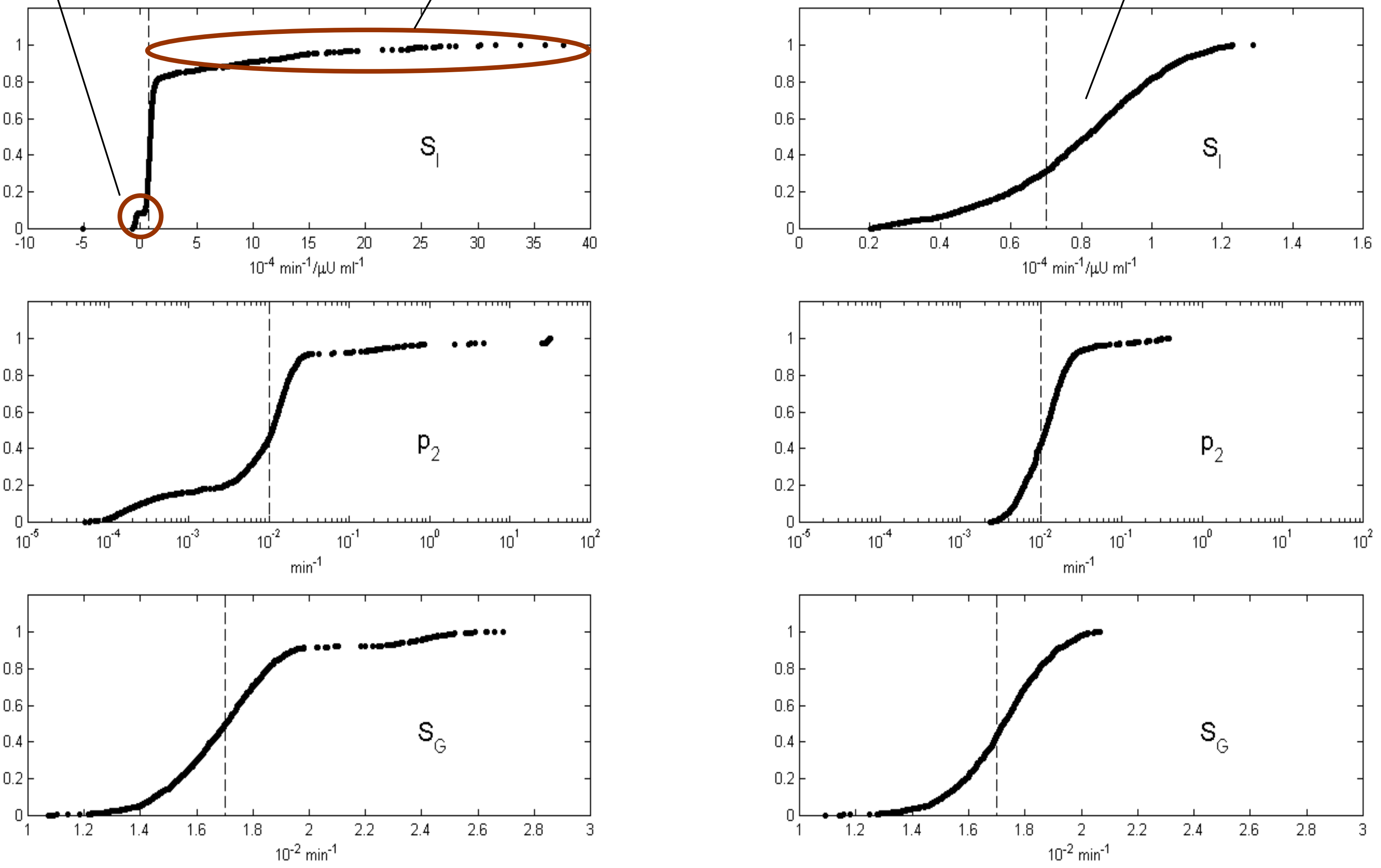
Unrealistically large in another 10-20% dei casi

S_I<0 in 10% of the cases

Always close to zero

Fisher (ML)

Bayes (minimum variance)



CONCLUSIONI

- Mathematical description and identification of a physical system is often a complex task
  (introduction of nonlinearities complicates the estimation process, e.g. nonnegativity constraints)

- Fisher approaches sometimes are not suited to face such difficulties, differently from the Bayesian approaches which appear more powerful alternatives but also more difficult to implement

- MCMC is currently the most powerful approach to face the computational difficulties related to the use of a Bayesian estimator