Introduction to Numerical Bayesian Methods

A. Taylan Cemgil

Signal Processing and Communications Lab.

UNIVERSITY OF CAMBRIDGE

Department of Engineering

IEE Professional Development Course on Adaptive Signal Processing,
1-3 March 2006, Birmingham, UK

Thanks to

- Nick Whiteley
- Simon Godsill
- Bill Fitzgerald

Latest Version of the tutorial slides are available from my homepage under *Quick Links* (or type cemgil to google)

`http://www-sigproc.eng.cam.ac.uk/~atc27/`
`http://www-sigproc.eng.cam.ac.uk/~atc27/papers/cemgil-iee-pres.pdf`

Outline

- Introduction, Bayes' Theorem, Sample applications
- Deterministic Inference Techniques
  - Variational Methods: Variational Bayes, EM, ICM
- Stochastic (Sampling Based) Methods
  - Markov Chain Monte Carlo (MCMC)
  - Importance Sampling
- Online Inference
  - Sequential Monte Carlo
- Summary and Remarks

Bayes' Theorem [4, 5]

Thomas Bayes (1702-1761)

What you know about a parameter θ after the data $\mathcal{D}$ arrive is what you knew before about θ and what the data $\mathcal{D}$ told you.

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}$$

$$\text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

An application of Bayes' Theorem: "Parameter Estimation"

Given two fair dice with outcomes λ and y,

$$\mathcal{D} = \lambda + y$$

What is λ when $\mathcal{D} = 9$?

An application of Bayes' Theorem: "Parameter Estimation"

$$\mathcal{D} = \lambda + y = 9$$

$\mathcal{D} = \lambda + y$	$y = 1$	$y = 2$	$y = 3$	$y = 4$	$y = 5$	$y = 6$
$\lambda = 1$	2	3	4	5	6	7
$\lambda = 2$	3	4	5	6	7	8
$\lambda = \mathbf{3}$	4	5	6	7	8	**9**
$\lambda = \mathbf{4}$	5	6	7	8	**9**	10
$\lambda = \mathbf{5}$	6	7	8	**9**	10	11
$\lambda = \mathbf{6}$	7	8	**9**	10	11	12

Bayes theorem "upgrades" $p(\lambda)$ into $p(\lambda|\mathcal{D})$.

But you have to provide an observation model: $p(\mathcal{D}|\lambda)$

Another application of Bayes' Theorem: "Model Selection"

Given an unknown number of fair dice with outcomes $\lambda_1, \lambda_2, \ldots, \lambda_n$,

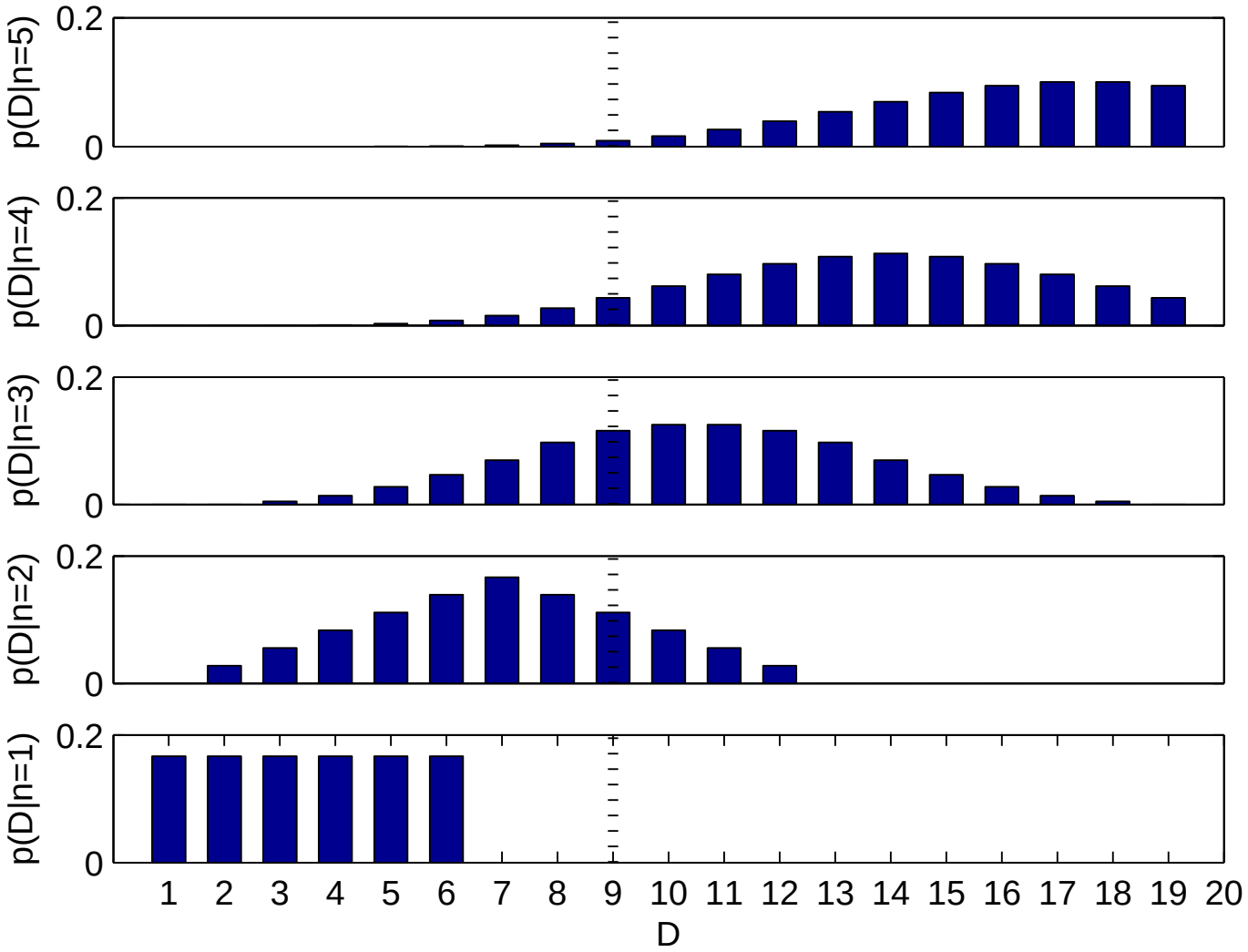
$$\mathcal{D} = \sum_{i=1}^{n} \lambda_i$$

How many dice there are when $\mathcal{D} = 9$?

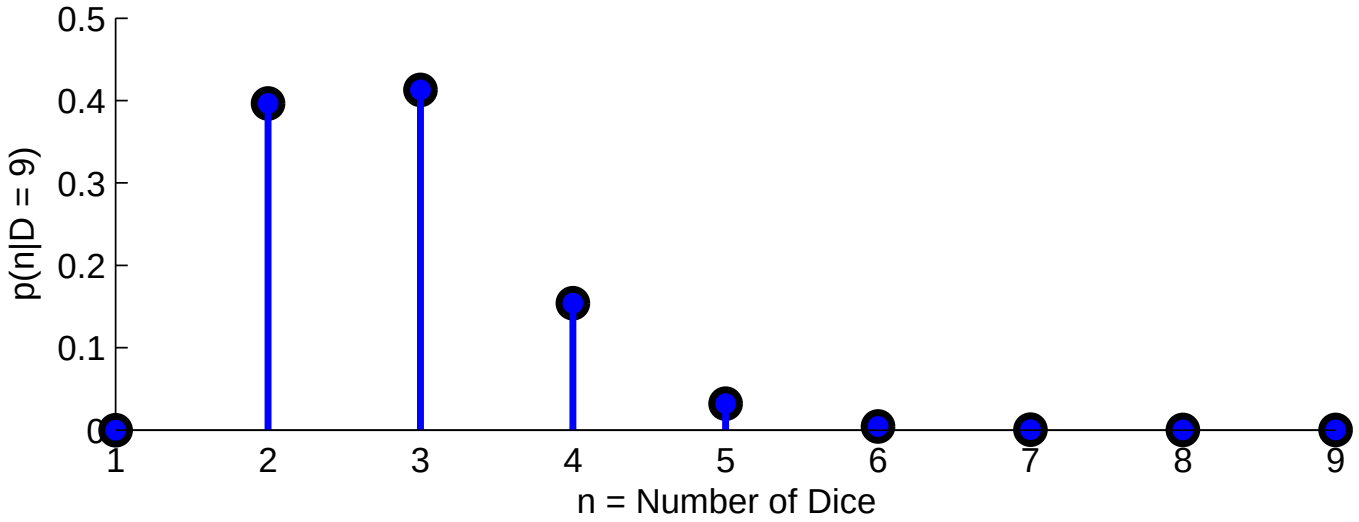
Given all n are equally likely (i.e., $p(n)$ is flat), we calculate (formally)

$$\begin{aligned} p(n|\mathcal{D}=9) &= \frac{p(\mathcal{D}=9|n)p(n)}{p(\mathcal{D})} \propto p(\mathcal{D}=9|n) \\ &\propto \sum_{\lambda_1,\ldots,\lambda_n} p(\mathcal{D}|\lambda_1,\ldots,\lambda_n) \prod_{i=1}^{n} p(\lambda_i) \end{aligned}$$

$$p(\mathcal{D}|n) = \sum_{\boldsymbol{\lambda}} p(\mathcal{D}|\boldsymbol{\lambda}, n) p(\boldsymbol{\lambda}|n)$$



Another application of Bayes' Theorem: "Model Selection"



- Complex models are more flexible but they spread their probability mass
- Bayesian inference inherently prefers "simpler models" – occam's razor
- Computational burden: We need to sum over all parameters λ

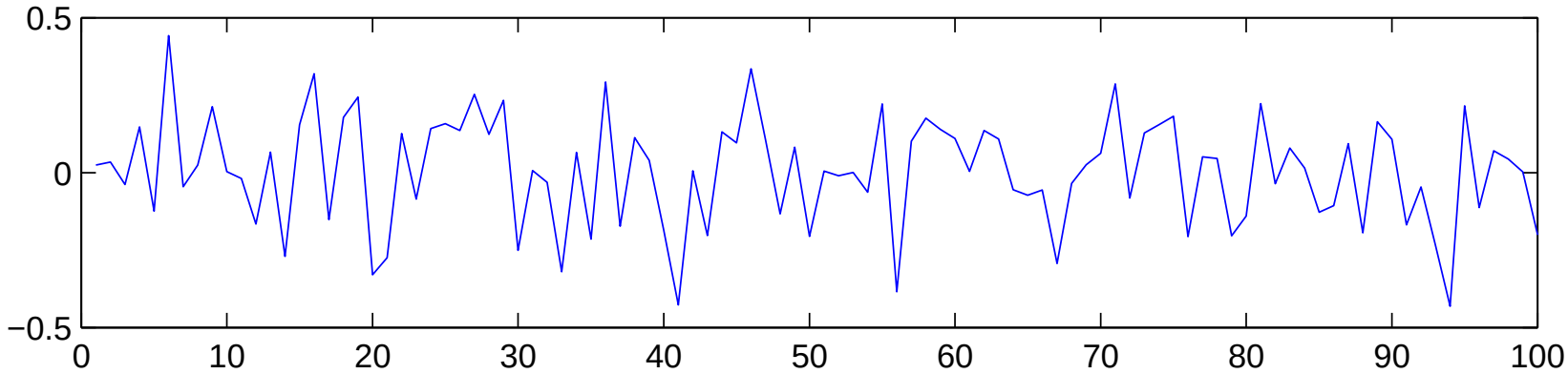
Example: AR(1) model

$$x_k = Ax_{k-1} + \epsilon_k \qquad\qquad k = 1 \ldots K$$

ϵ_k is i.i.d., zero mean and normal with variance R.

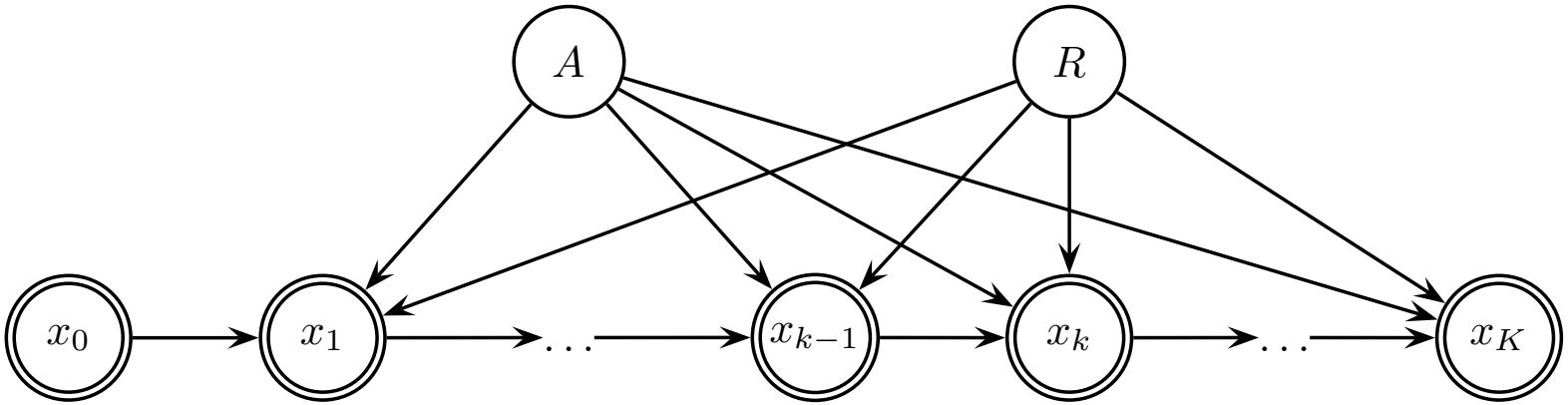
Estimation problem:

Given $x_0, \ldots, x_K$, determine coefficient A and variance R (both scalars).



AR(1) model, Generative Model notation

$$
\begin{aligned}
A &\sim \mathcal{N}(A; 0, P) \\
R &\sim \mathcal{IG}(R; \nu, \beta/\nu) \\
x_k | x_{k-1}, A, R &\sim \mathcal{N}(x_k; Ax_{k-1}, R) \qquad x_0 = \hat{x}_0
\end{aligned}
$$



Gaussian : $\mathcal{N}(x; \mu, V) \equiv |2\pi V|^{-\frac{1}{2}} \exp(-\frac{1}{2}(x-\mu)^2/V)$

Inverse-Gamma distribution: $\mathcal{IG}(x; a, b) \equiv \Gamma(a)^{-1} b^{-a} x^{-(a+1)} \exp(-1/(bx)) \quad x \geq 0$

Observed variables are shown with double circles

Bayesian Posterior Inference

$$p(A, R|x_0, x_1, \ldots, x_K) \quad \propto \quad p(x_1, \ldots, x_K|x_0, A, R)p(A, R)$$

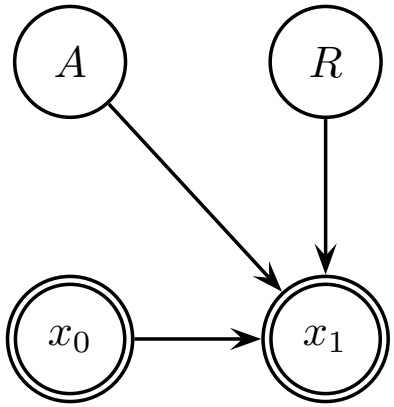
$$\text{Posterior} \quad \propto \quad \text{Likelihood} \times \text{Prior}$$

Using the Markovian (conditional independence) structure we have

$$p(A, R|x_0, x_1, \ldots, x_K) \quad \propto \quad \left(\prod_{k=1}^{K} p(x_k|x_{k-1}, A, R) \right) p(A)p(R)$$

Numerical Example

Suppose $K = 1$,

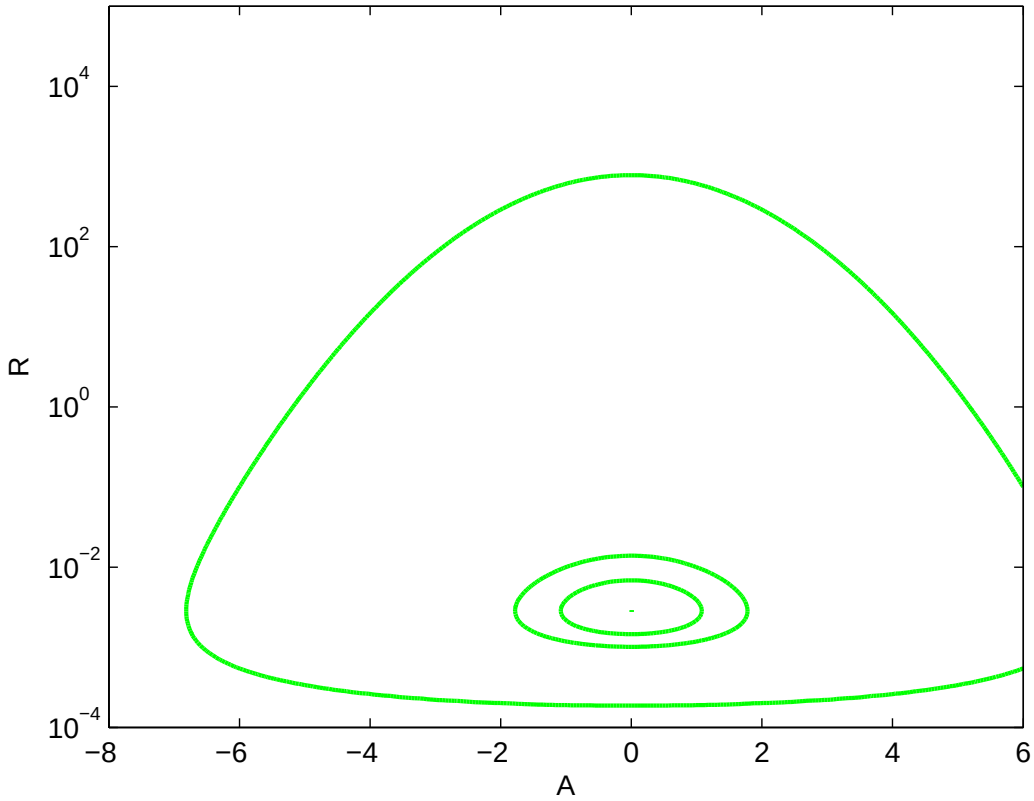


By Bayes' Theorem and the structure of AR(1) model

$$\begin{aligned} p(A, R|x_0, x_1) &\propto p(x_1|x_0, A, R)p(A)p(R) \\ &= \mathcal{N}(x_1; Ax_0, R)\mathcal{N}(A; 0, P)\mathcal{IG}(R; \nu, \beta/\nu) \end{aligned}$$

Numerical Example, the prior $p(A, R)$

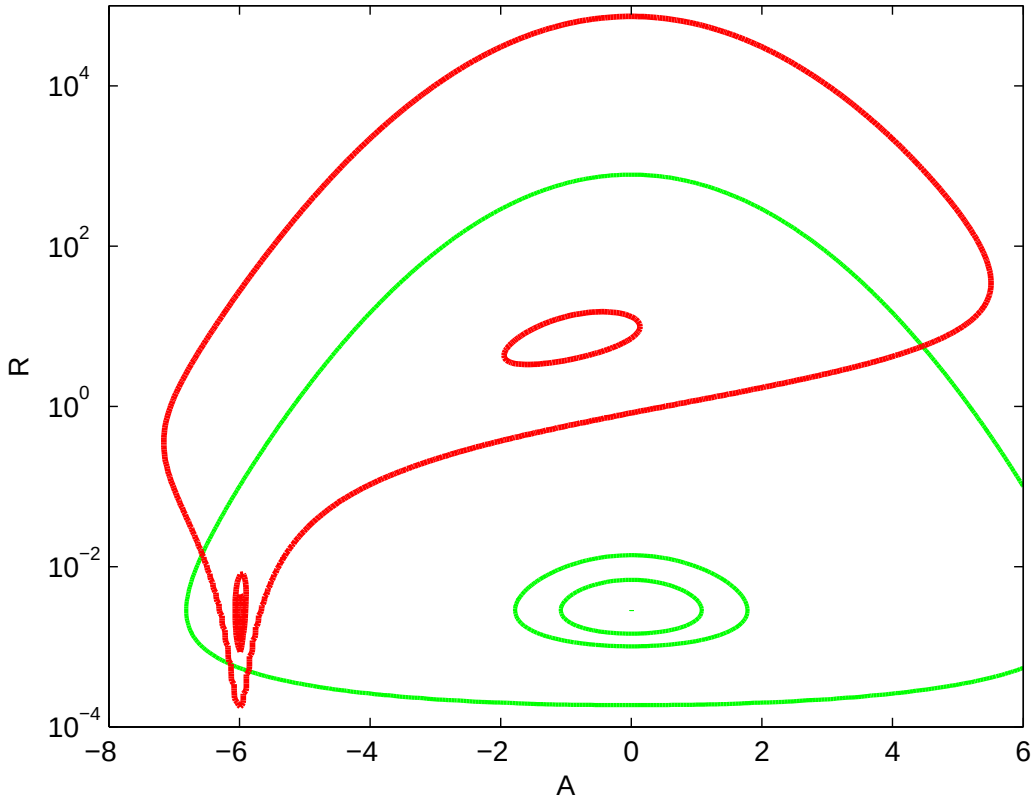
Equiprobability contour of $p(A)p(R)$



$$A \sim \mathcal{N}(A; 0, 1.2) \qquad R \sim \mathcal{IG}(R; 0.4, 250)$$

Suppose: $x_0 = 1$ $\qquad x_1 = -6$ $\qquad x_1 \sim \mathcal{N}(x_1; Ax_0, R)$

Numerical Example, the posterior $p(A, R|x)$



Note the bimodal posterior with $x_0 = 1, x_1 = -6$

- $A \approx -6 \Leftrightarrow$ low noise variance R.
- $A \approx 0 \Leftrightarrow$ high noise variance R.

Remarks

- The maximum likelihood solution (or any other point estimate) is not always representative about the solution
- (Unfortunately), exact posterior inference is only possible for few special cases
- Even very simple models can lead easily to complicated posterior distributions
- *A-priori* independent variables often become dependent *a-posteriori* (“Explaining away”)
- Ambiguous data usually leads to a multimodal posterior, each mode corresponding to one possible explanation
- The complexity of an inference problem depends, among others, upon the particular “parameter regime” and observed data sequence

Probabilistic Inference

A huge spectrum of applications – all boil down to computation of

- **expectations** of functions under probability distributions: **Integration**

$$\langle f(x) \rangle \quad = \quad \int_{\mathcal{X}} dx p(x) f(x)$$

- **modes** of functions under probability distributions: **Optimization**

$$x^* \quad = \quad \operatorname*{argmax}_{x \in \mathcal{X}} p(x) f(x)$$

- any “mix” of the above: e.g.,

$$x^* \quad = \quad \operatorname*{argmax}_{x \in \mathcal{X}} p(x) = \operatorname*{argmax}_{x \in \mathcal{X}} \int_{\mathcal{Z}} dz p(z) p(x|z)$$

Divide and Conquer

Probabilistic modelling provides a methodology that puts a clear division between

- What to solve : Model Construction
  - Both an Art and Science
  - Highly domain specific
- How to solve : Inference Algorithm
  - (In principle) Mechanical
  - Generic

"*An approximate solution of the exact problem is often more useful than the exact solution of an approximate problem*",

J. W. Tukey (1915-2000).

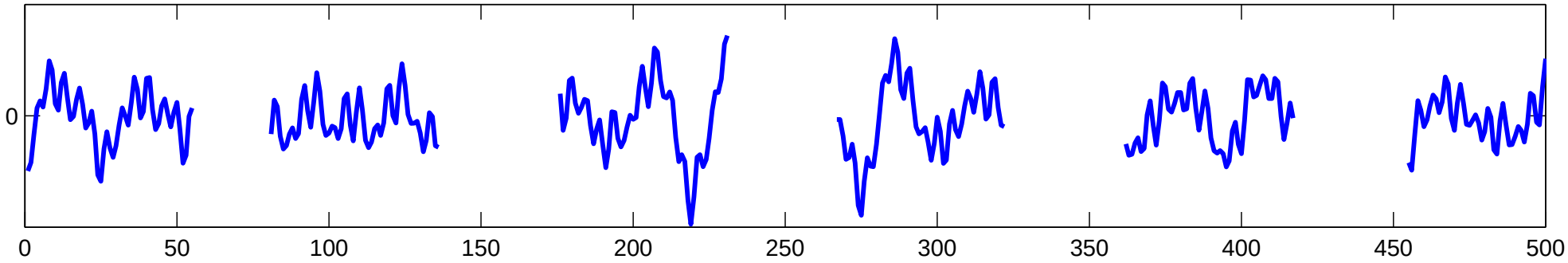
Attributes of Probabilistic Inference

- Exact $\leftrightarrow$ **Approximate**
- **Deterministic** $\leftrightarrow$ **Stochastic**
- **Online** $\leftrightarrow$ **Offline**
- **Centralized** $\leftrightarrow$ Distributed

This talk focuses on the bold ones

Some Applications: Audio Restoration

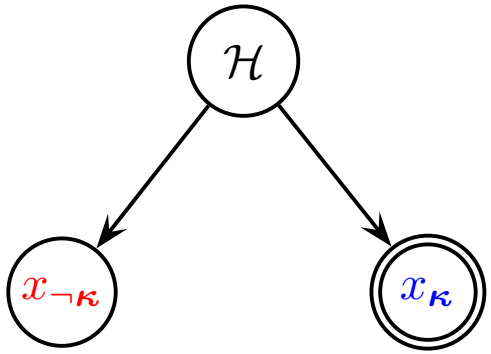
- During download or transmission, some samples of audio are lost
- Estimate missing samples given clean ones



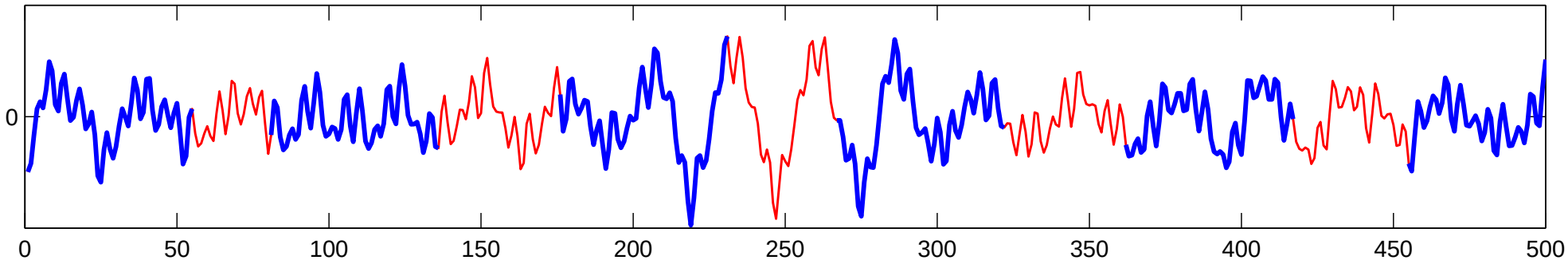
Examples: Audio Restoration

$$p(x_{\neg\boldsymbol{\kappa}}|x_{\boldsymbol{\kappa}}) \propto \int d\mathcal{H} p(x_{\neg\boldsymbol{\kappa}}|\mathcal{H})p(x_{\boldsymbol{\kappa}}|\mathcal{H})p(\mathcal{H})$$

$$\mathcal{H} \equiv (\text{parameters, hidden states})$$

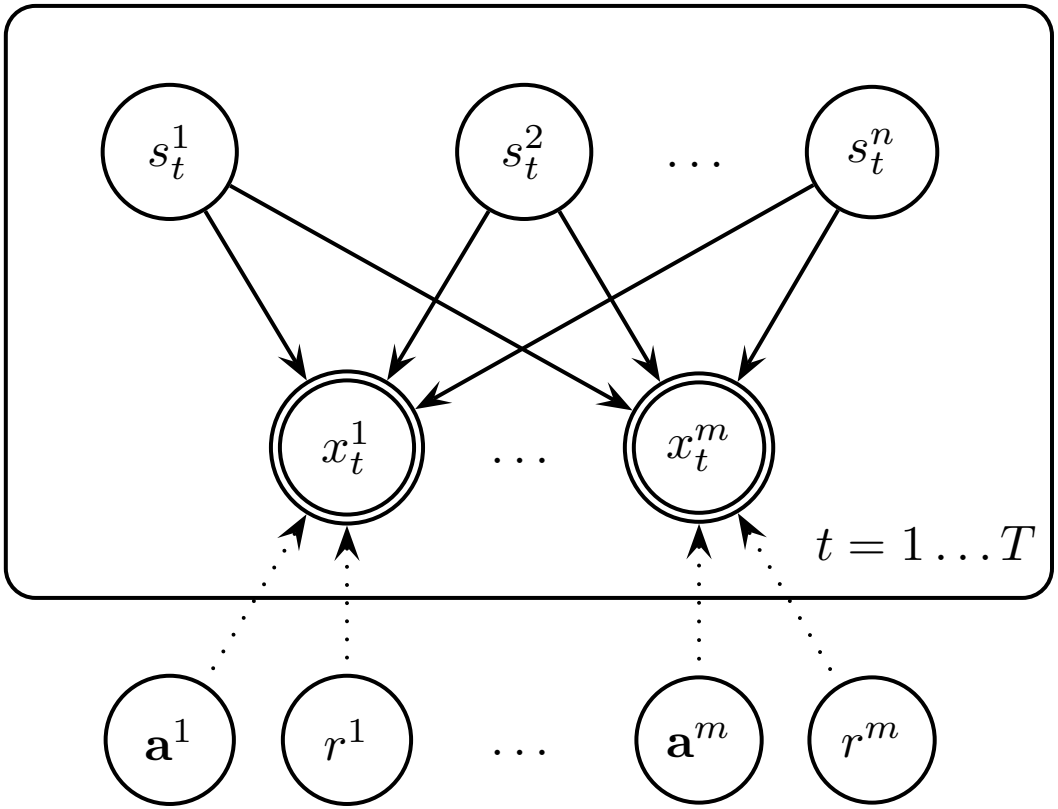


Missing Observed



Some Applications: Source Separation

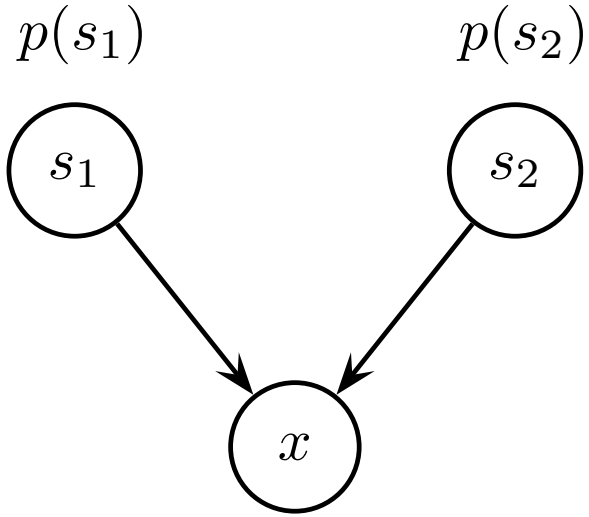
Estimate n hidden signals $\mathbf{s}_t$ from m observed signals $\mathbf{x}_t$.



$$
\begin{aligned}
s_t^i &\sim p(s_t^i) \\
x_t^j &\sim \mathcal{N}(x; \mathbf{a}^j s_t^{1:n}, r^j)
\end{aligned}
$$

Deterministic Inference

Toy Model : “One sample source separation (OSSS)”



This graph encodes the joint: $p(x, s_1, s_2) = p(x|s_1, s_2)p(s_1)p(s_2)$

$$
\begin{aligned}
s_1 &\sim p(s_1) = \mathcal{N}(s_1; \mu_1, P_1) \\
s_2 &\sim p(s_2) = \mathcal{N}(s_2; \mu_2, P_2) \\
x|s_1, s_2 &\sim p(x|s_1, s_2) = \mathcal{N}(x; s_1 + s_2, R)
\end{aligned}
$$

The Gaussian Distribution

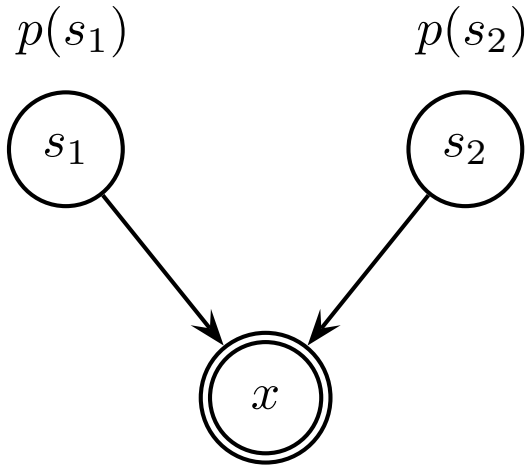
μ is the mean and P is the covariance:

$$
\begin{aligned}
\mathcal{N}(s;\mu,P) &= |2\pi P|^{-1/2}\exp\left(-\frac{1}{2}(s-\mu)^T P^{-1}(s-\mu)\right) \\
&= \exp\left(-\frac{1}{2}s^T P^{-1}s + \mu^T P^{-1}s - \frac{1}{2}\mu^T P^{-1}\mu - \frac{1}{2}|2\pi P|\right) \\
\log\mathcal{N}(s;\mu,P) &= -\frac{1}{2}s^T P^{-1}s + \mu^T P^{-1}s + \text{ const} \\
&= -\frac{1}{2}\mathbf{Tr}\, P^{-1}ss^T + \mu^T P^{-1}s + \text{ const} \\
&=^{+} -\frac{1}{2}\mathbf{Tr}\, P^{-1}ss^T + \mu^T P^{-1}s
\end{aligned}
$$

Notation: $\log f(x) =^{+} g(x) \iff f(x) \propto \exp(g(x)) \iff \exists c \in \mathbb{R} : f(x) = c\exp(g(x))$

OSSS example

Suppose, we observe $x = \hat{x}$.



$$p(x = \hat{x}|s_1, s_2)$$

- By Bayes' theorem, the posterior is given by:

$$\mathcal{P} \equiv p(s_1, s_2|x = \hat{x}) = \frac{1}{Z_{\hat{x}}} p(x = \hat{x}|s_1, s_2) p(s_1) p(s_2) \equiv \frac{1}{Z_{\hat{x}}} \phi(s_1, s_2)$$

- The function $\phi(s_1, s_2)$ is proportional to the exact posterior. ($Z_{\hat{x}} \equiv p(x = \hat{x})$)

OSSS example, cont.

$$\begin{aligned}
\log p(s_1) &= \mu_1^T P_1^{-1} s_1 - \frac{1}{2} s_1^T P_1^{-1} s_1 + \text{const} \\
\log p(s_2) &= \mu_2^T P_2^{-1} s_2 - \frac{1}{2} s_2^T P_2^{-1} s_2 + \text{const} \\
\log p(x|s_1, s_2) &= \hat{x}^T R^{-1} (s_1 + s_2) - \frac{1}{2} (s_1 + s_2)^T R^{-1} (s_1 + s_2) + \text{const}
\end{aligned}$$

$$\begin{aligned}
\log \phi(s_1, s_2) &= \log p(x = \hat{x}|s_1, s_2) + \log p(s_1) + \log p(s_2) \\
&=^+ \left(\mu_1^T P_1^{-1} + \hat{x}^T R^{-1}\right) s_1 + \left(\mu_2^T P_2^{-1} + \hat{x}^T R^{-1}\right) s_2 \\
&\quad -\frac{1}{2} \mathbf{Tr}\left(P_1^{-1} + R^{-1}\right) s_1 s_1^T - \underbrace{s_1^T R^{-1} s_2}_{(*)} - \frac{1}{2} \mathbf{Tr}\left(P_2^{-1} + R^{-1}\right) s_2 s_2^T
\end{aligned}$$

- The (*) term is the cross correlation term that makes s_1 and s_2 a-posteriori dependent.

Variational Bayes (VB), mean field

We will approximate the posterior $\mathcal{P}$ with a simpler distribution $\mathcal{Q}$.

$$\begin{aligned} \mathcal{P} &= \frac{1}{Z_x} p(x = \hat{x}|s_1, s_2) p(s_1) p(s_2) \\ \mathcal{Q} &= q(s_1) q(s_2) \end{aligned}$$

Here, we choose

$$q(s_1) = \mathcal{N}(s_1; m_1, S_1) \qquad q(s_2) = \mathcal{N}(s_2; m_2, S_2)$$

A “measure of fit” between distributions is the KL divergence

Kullback-Leibler (KL) Divergence

- A “quasi-distance” between two distributions $\mathcal{P} = p(x)$ and $\mathcal{Q} = q(x)$.

$$KL(\mathcal{P}||\mathcal{Q}) \equiv \int_{\mathcal{X}} dx p(x) \log \frac{p(x)}{q(x)} = \langle \log \mathcal{P} \rangle_{\mathcal{P}} - \langle \log \mathcal{Q} \rangle_{\mathcal{P}}$$

- Unlike a metric, (in general) it is not symmetric,

$$KL(\mathcal{P}||\mathcal{Q}) \neq KL(\mathcal{Q}||\mathcal{P})$$

- But it is non-negative (by Jensen’s Inequality)

$$\begin{aligned} KL(\mathcal{P}||\mathcal{Q}) &= -\int_{\mathcal{X}} dx p(x) \log \frac{q(x)}{p(x)} \\ &\geq -\log \int_{\mathcal{X}} dx p(x) \frac{q(x)}{p(x)} = -\log \int_{\mathcal{X}} dx q(x) = -\log 1 = 0 \end{aligned}$$

OSSS example, cont.

Let the approximating distribution be factorized as

$$\mathcal{Q} \quad = \quad q(s_1)q(s_2)$$

$$q(s_1) = \mathcal{N}(s_1; m_1, S_1) \qquad q(s_2) = \mathcal{N}(s_2; m_2, S_2)$$

The m_i and S_j are the *variational* parameters to be optimized to minimize

$$KL(\mathcal{Q}||\mathcal{P}) \quad = \quad \langle \log \mathcal{Q} \rangle_{\mathcal{Q}} - \left\langle \log \underbrace{\frac{1}{Z_x}\phi(s_1, s_2)}_{=\mathcal{P}} \right\rangle_{\mathcal{Q}} \tag{1}$$

The form of the mean field solution

$$
\begin{aligned}
0 &\leq \left\langle \log q(s_1)q(s_2) \right\rangle_{q(s_1)q(s_2)} + \log Z_x - \left\langle \log \phi(s_1, s_2) \right\rangle_{q(s_1)q(s_2)} \\
\log Z_x &\geq \left\langle \log \phi(s_1, s_2) \right\rangle_{q(s_1)q(s_2)} - \left\langle \log q(s_1)q(s_2) \right\rangle_{q(s_1)q(s_2)} \\
&\equiv -F(p; q) + H(q) \qquad (2)
\end{aligned}
$$

Here, F is the *energy* and H is the *entropy*. We need to maximize the right hand side.

$$\text{Evidence} \geq -\text{Energy} + \text{Entropy}$$

Note r.h.s. is a **lower bound** [6]. The mean field equations **monotonically** increase this bound. Good for assessing convergence and debugging computer code.

Details of derivation

- Define the Lagrangian

$$\begin{aligned}\Lambda &= \int ds_1 q(s_1)\log q(s_1) + \int ds_2 q(s_2)\log q(s_2) + \log Z_x - \int ds_1 ds_2 q(s_1)q(s_2)\log\phi(s_1,s_2) \\ &\quad +\lambda_1(1-\int ds_1 q(s_1)) + \lambda_2(1-\int ds_2 q(s_2)) \end{aligned} \tag{3}$$

- Calculate the functional derivatives w.r.t. $q(s_1)$ and set to zero

$$\frac{\delta}{\delta q(s_1)}\Lambda = \log q(s_1) + 1 - \langle\log\phi(s_1,s_2)\rangle_{q(s_2)} - \lambda_1$$

- Solve for $q(s_1)$,

$$\begin{aligned}\log q(s_1) &= \lambda_1 - 1 + \langle\log\phi(s_1,s_2)\rangle_{q(s_2)} \\ q(s_1) &= \exp(\lambda_1-1)\exp(\langle\log\phi(s_1,s_2)\rangle_{q(s_2)}) \end{aligned} \tag{4}$$

- Use the fact that

$$\begin{aligned}1 &= \int ds_1 q(s_1) = \exp(\lambda_1-1)\int ds_1 \exp(\langle\log\phi(s_1,s_2)\rangle_{q(s_2)}) \\ \lambda_1 &= 1 - \log\int ds_1 \exp(\langle\log\phi(s_1,s_2)\rangle_{q(s_2)})\end{aligned}$$

The form of the solution

- No direct analytical solution
- We obtain fixed point equations in closed form

$$q(s_1) \propto \exp(\langle \log \phi(s_1, s_2) \rangle_{q(s_2)})$$

$$q(s_2) \propto \exp(\langle \log \phi(s_1, s_2) \rangle_{q(s_1)})$$

Note the nice symmetry

Fixed Point Iteration for OSSS

$$\log q(s_1) \leftarrow \log p(s_1) + \langle \log p(x = \hat{x} | s_1, s_2) \rangle_{q(s_2)}$$

$$\log q(s_2) \leftarrow \log p(s_2) + \langle \log p(x = \hat{x} | s_1, s_2) \rangle_{q(s_1)}$$

We can think of sending messages back and forth.

Fixed Point Iteration for the Gaussian Case

$$\log q(s_1) \quad \leftarrow \quad -\frac{1}{2}\,\mathbf{Tr}\left(P_1^{-1}+R^{-1}\right)s_1 s_1^T - s_1^T R^{-1}\underbrace{\langle s_2\rangle_{q(s_2)}}_{=m_2} + \left(\mu_1^T P_1^{-1}+\hat{x}^T R^{-1}\right)s_1$$

$$\log q(s_2) \quad \leftarrow \quad -\underbrace{\langle s_1\rangle_{q(s_1)}^T}_{=m_1^T} R^{-1}s_2 - \frac{1}{2}\,\mathbf{Tr}\left(P_2^{-1}+R^{-1}\right)s_2 s_2^T + \left(\mu_2^T P_2^{-1}+\hat{x}^T R^{-1}\right)s_2$$

Remember $q(s)=\mathcal{N}(s;m,S)$

$$\log q(s) \quad =^+ \quad -\frac{1}{2}\,\mathbf{Tr}\,K s s^T + h^T s$$

$$\Downarrow$$

$$S=K^{-1} \qquad\qquad m=K^{-1}h$$

Fixed Point Equations for the Gaussian Case

- Covariances are obtained directly

$$S_1 = \left(P_1^{-1} + R^{-1}\right)^{-1} \qquad S_2 = \left(P_2^{-1} + R^{-1}\right)^{-1}$$

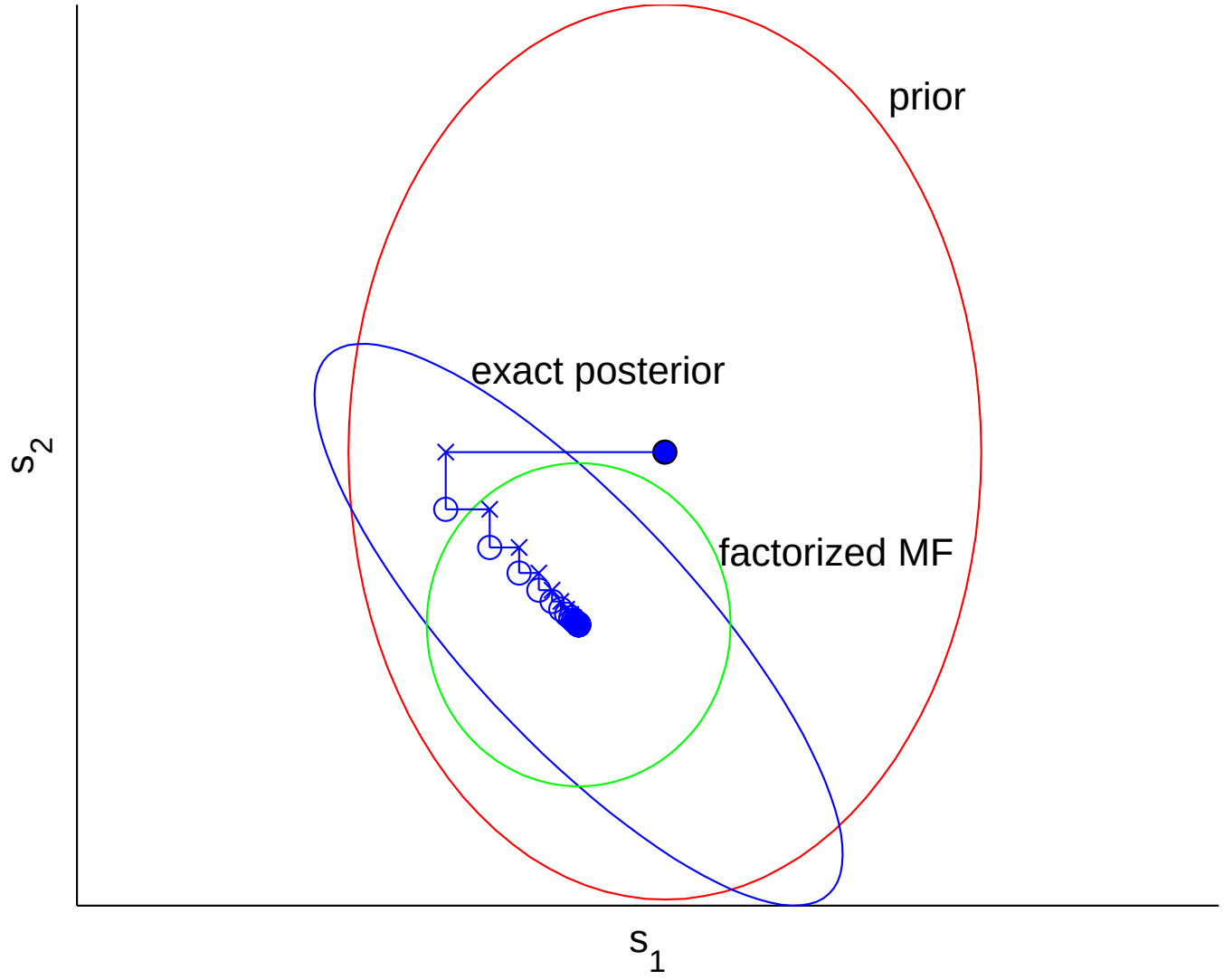
- To compute the means, we should iterate:

$$\begin{aligned} m_1 &= S_1 \left(P_1^{-1}\mu_1 + R^{-1}\left(\hat{x} - m_2\right)\right) \\ m_2 &= S_2 \left(P_2^{-1}\mu_2 + R^{-1}\left(\hat{x} - m_1\right)\right) \end{aligned}$$

- Intuitive algorithm:
  - Substract from the observation $\hat{x}$ the prediction of the other factors of $\mathcal{Q}$.
  - Compute a fit to this residual (e.g. "fit" m_2 to $\hat{x} - m_1$).

- Equivalent to Gauss-Seidel, an iterative method for solving linear systems of equations.

OSSS example, cont.



Direct Link to Expectation-Maximisation (EM) Algorithm [3]

Suppose we choose one of the distributions degenerate, i.e.

$$\tilde{q}(s_2) \quad = \quad \delta(s_2 - \tilde{m})$$

where $\tilde{m}$ corresponds to the "location parameter" of $\tilde{q}(s_2)$. We need to find the closest degenerate distribution to the actual mean field solution $q(s_2)$, hence we take one more KL and minimize

$$\tilde{m} \quad = \quad \underset{\xi}{\operatorname{argmin}} \, KL(\delta(s_2 - \xi)||q(s_2))$$

It can be shown that this leads exactly to the EM fixed point iterations.

Iterated Conditional Modes (ICM) Algorithm [1, 2]

If we choose both distributions degenerate, i.e.

$$\begin{aligned} \tilde{q}(s_1) &= \delta(s_1 - \tilde{m}_1) \\ \tilde{q}(s_2) &= \delta(s_2 - \tilde{m}_2) \end{aligned}$$

It can be shown that this leads exactly to the ICM fixed point iterations. This algorithm is equivalent to coordinate ascent in the original posterior surface $\phi(s_1, s_2)$.

$$\begin{aligned} \tilde{m}_1 &= \operatorname*{argmax}_{s_1} \phi(s_1, s_2 = \tilde{m}_2) \\ \tilde{m}_2 &= \operatorname*{argmax}_{s_2} \phi(s_1 = \tilde{m}_1, s_2) \end{aligned}$$

ICM, EM, VB ...

For OSSS, all algorithms are identical. This is in general not true.

While algorithmic details are very similar, there can be big qualitative differences in terms of fixed points.

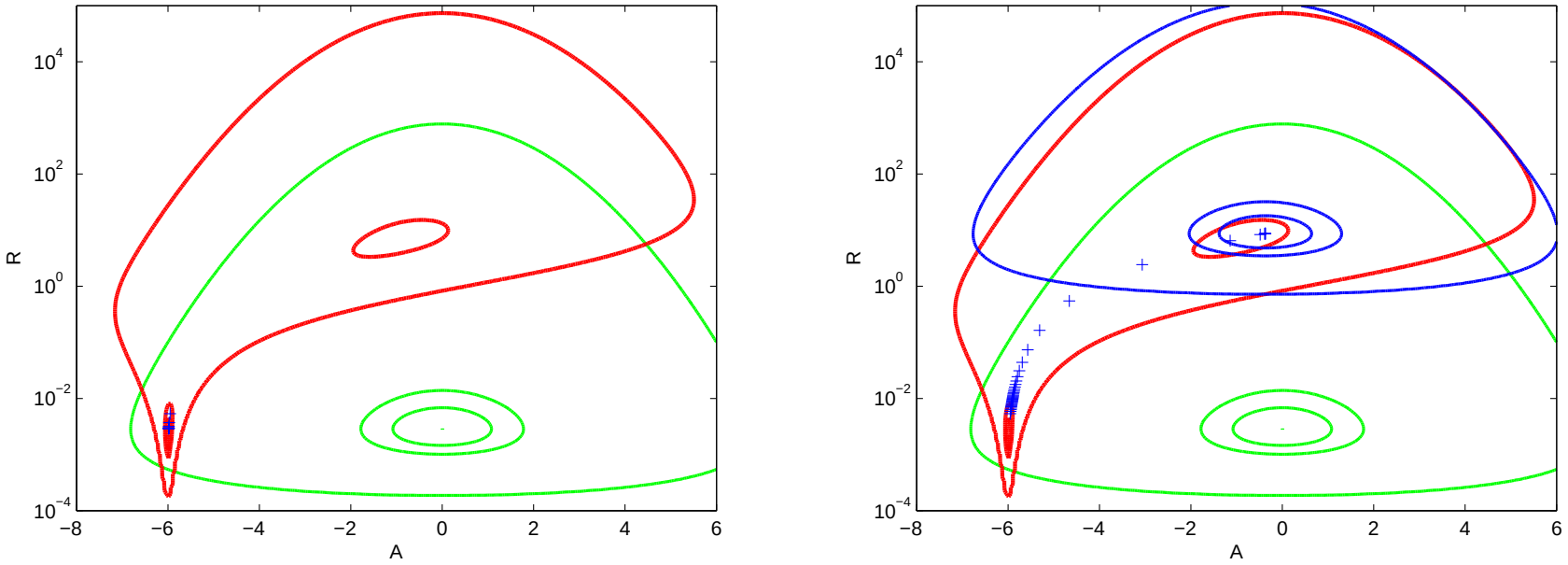
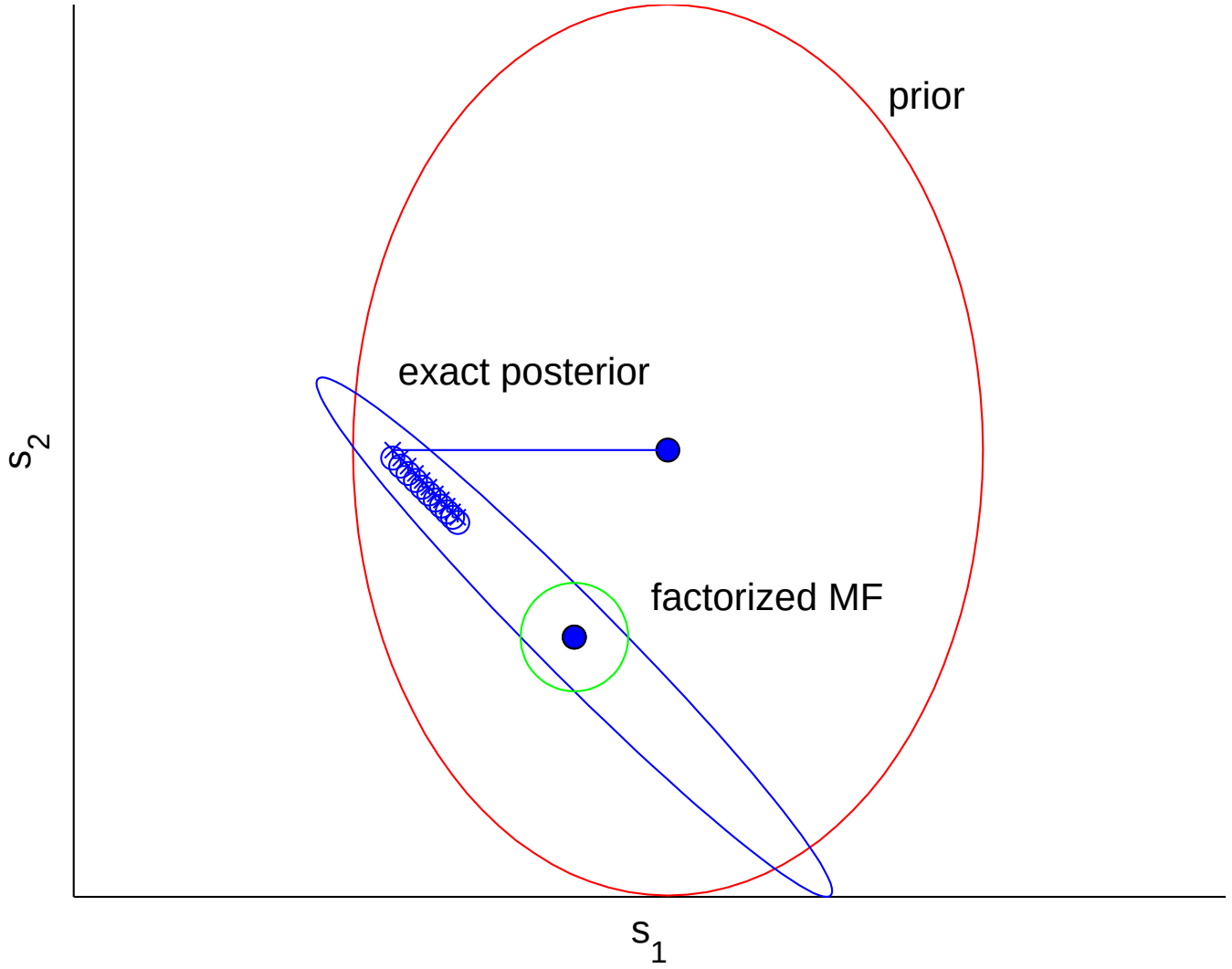


Figure 1: Left, ICM, Right VB. EM is similar to ICM in this AR(1) example.

Convergence Issues

OSSS example, Slow Convergence



Annealing, Bridging, Relaxation, Tempering

Main idea:

- If the original target $\mathcal{P}$ is too complex, relax it.
- First solve a simple version $\mathcal{P}_{\tau_1}$. Call the solution m_{τ_1}
- Make the problem little bit harder $\mathcal{P}_{\tau_1} \rightarrow \mathcal{P}_{\tau_2}$, and improve the solution $m_{\tau_1} \rightarrow m_{\tau_2}$.
- While $\mathcal{P}_{\tau_1} \rightarrow \mathcal{P}_{\tau_2}, \ldots, \rightarrow \mathcal{P}_T = \mathcal{P}$, we hope to get better and better solutions.

The sequence $\tau_1, \tau_2, \ldots, \tau_T$ is called annealing schedule if

$$\mathcal{P}_{\tau_i} \quad \propto \quad \mathcal{P}^{\tau_i}$$

OSSS example: Annealing, Bridging, ...

- Remember the cross term $(*)$ of the posterior:

$$\cdots - \underbrace{s_1^T R^{-1} s_2}_{(*)} \cdots$$

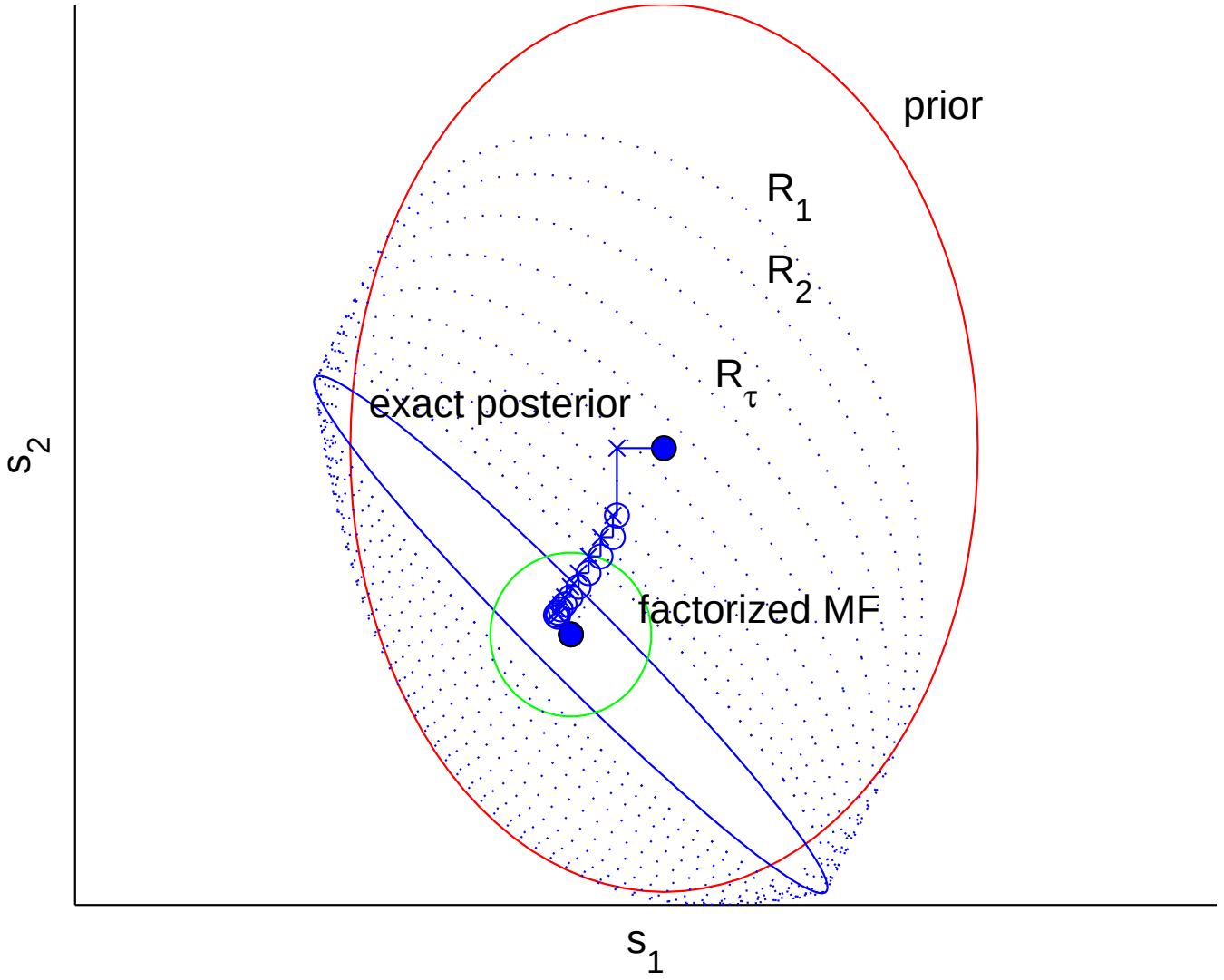
- When the noise variance is low, the coupling is strong.

- If we choose a decreasing sequence of noise covariances

$$R_{\tau_1} > R_{\tau_2} > \cdots > R_{\tau_T} = R$$

  we increase correlations gradually.

OSSS example: Annealing, Bridging, ...



Stochastic Inference

Deterministic versus Stochastic

Let θ denote the parameter vector of $\mathcal{Q}$.

- Given the fixed point equation F and an initial parameter $\theta^{(0)}$, the inference algorithm is simply

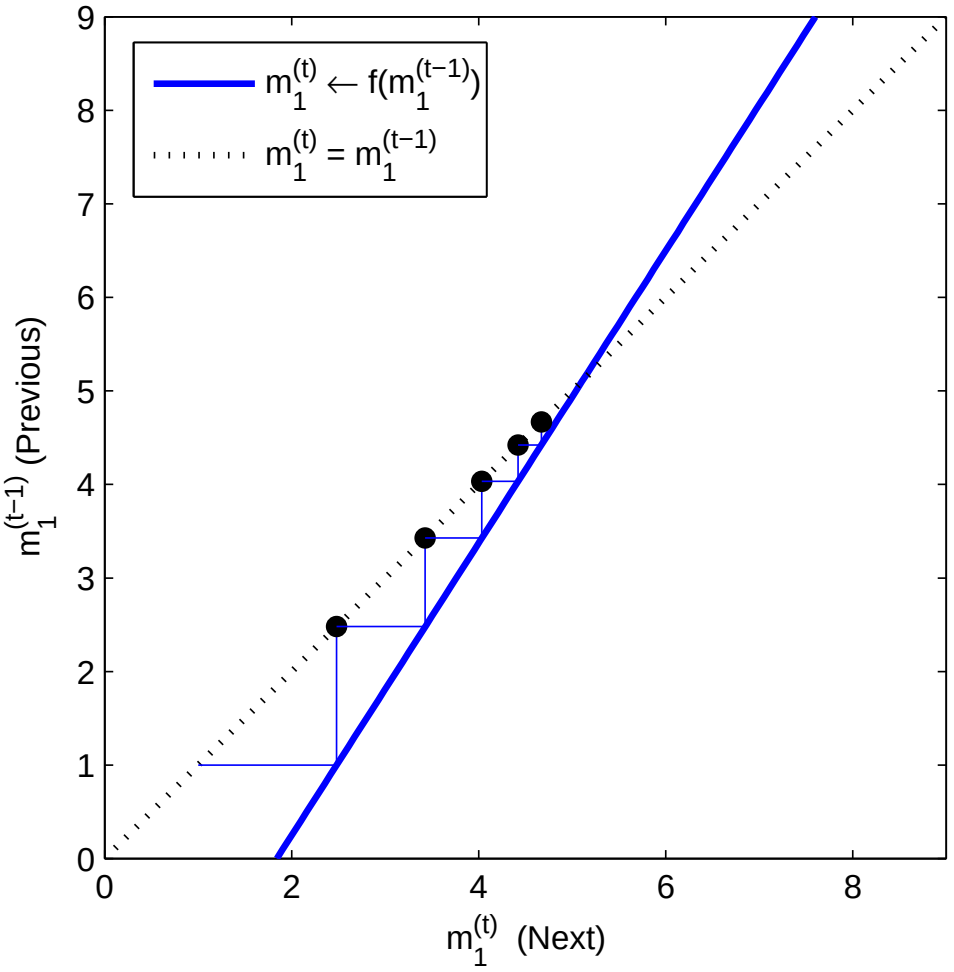
$$\theta^{(t+1)} \quad \leftarrow \quad F(\theta^{(t)})$$

For OSSS $\theta = (m_1, m_2)^T$ (S_1, S_2 were constant, so we exclude them). The update equations were

$$\begin{aligned} m_1^{(t+1)} &\leftarrow F_1(m_2^{(t)}) \\ m_2^{(t+1)} &\leftarrow F_2(m_1^{(t+1)}) \end{aligned}$$

This is a deterministic dynamical system in the parameter space.

Fixed Point iteration for m_1 in the OSS model



- Think of a movement along the $m^{(t)} = m^{(t-1)}$ line

Stochastic Inference

Stochastic inference is similar, but everything happens directly in the configuration space (= domain) of variables $\mathbf{s}$.

- Given a transition kernel T (=a collection of probability distributions conditioned on each $\mathbf{s}$) and an initial configuration $\mathbf{s}^{(0)}$

$$\mathbf{s}^{(t+1)} \quad \sim \quad T(\mathbf{s}|\mathbf{s}^{(t)}) \qquad\qquad t = 1, \ldots, \infty$$

- This is a stochastic dynamical system in the configuration space.
- A remarkable fact is that we can estimate any desired expectation by ergodic averages

$$\langle f(\mathbf{s}) \rangle_{\mathcal{P}} \approx \frac{1}{t - t_0} \sum_{n=t_0}^{t} f(\mathbf{s}^{(n)})$$

- Consecutive samples $\mathbf{s}^{(t)}$ are dependent but we can "pretend" as if they are independent!

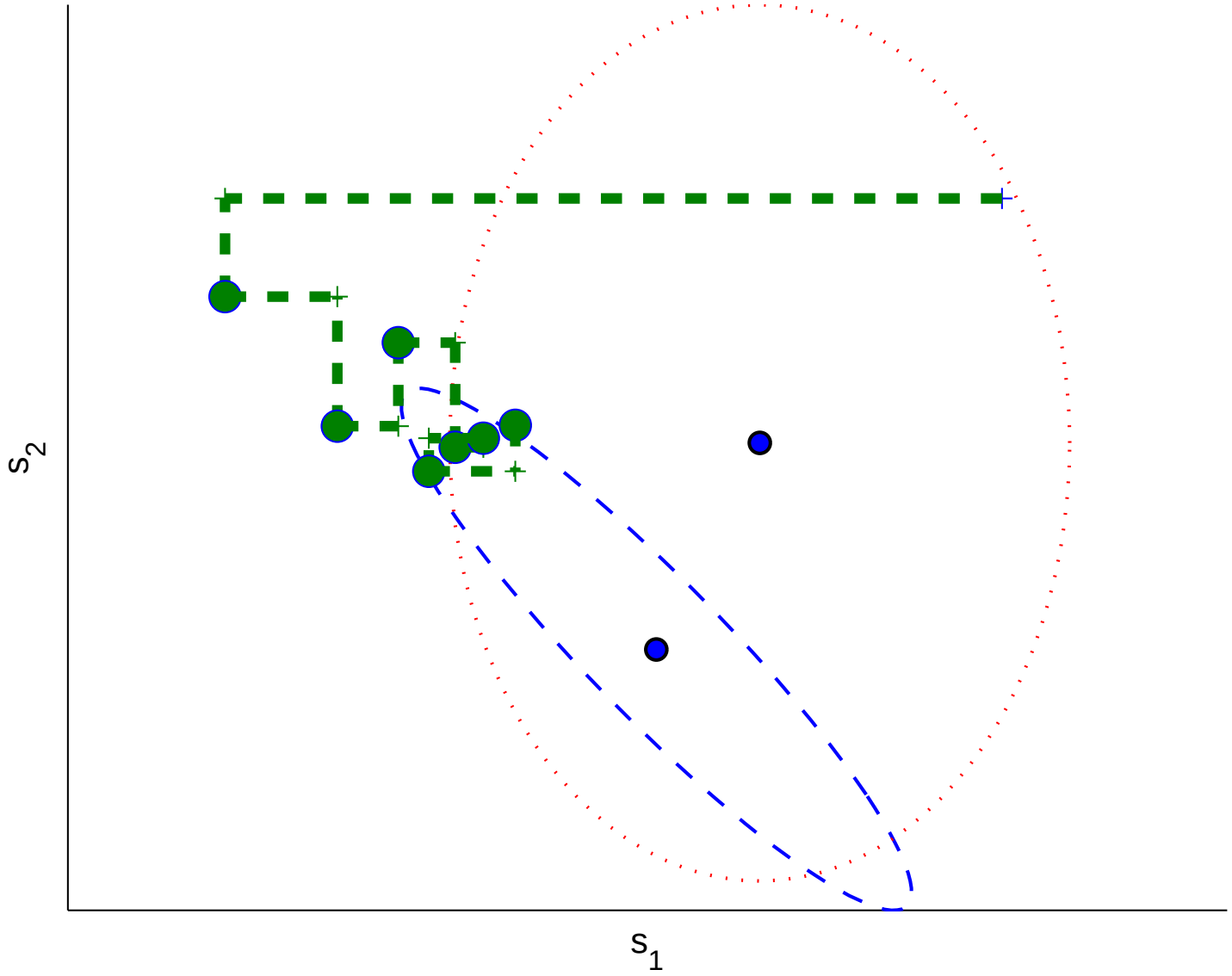
Looking ahead...

- For OSSS, the configuration space is $\mathbf{s} = (s_1, s_2)^T$.
- A possible transition kernel T is specified by

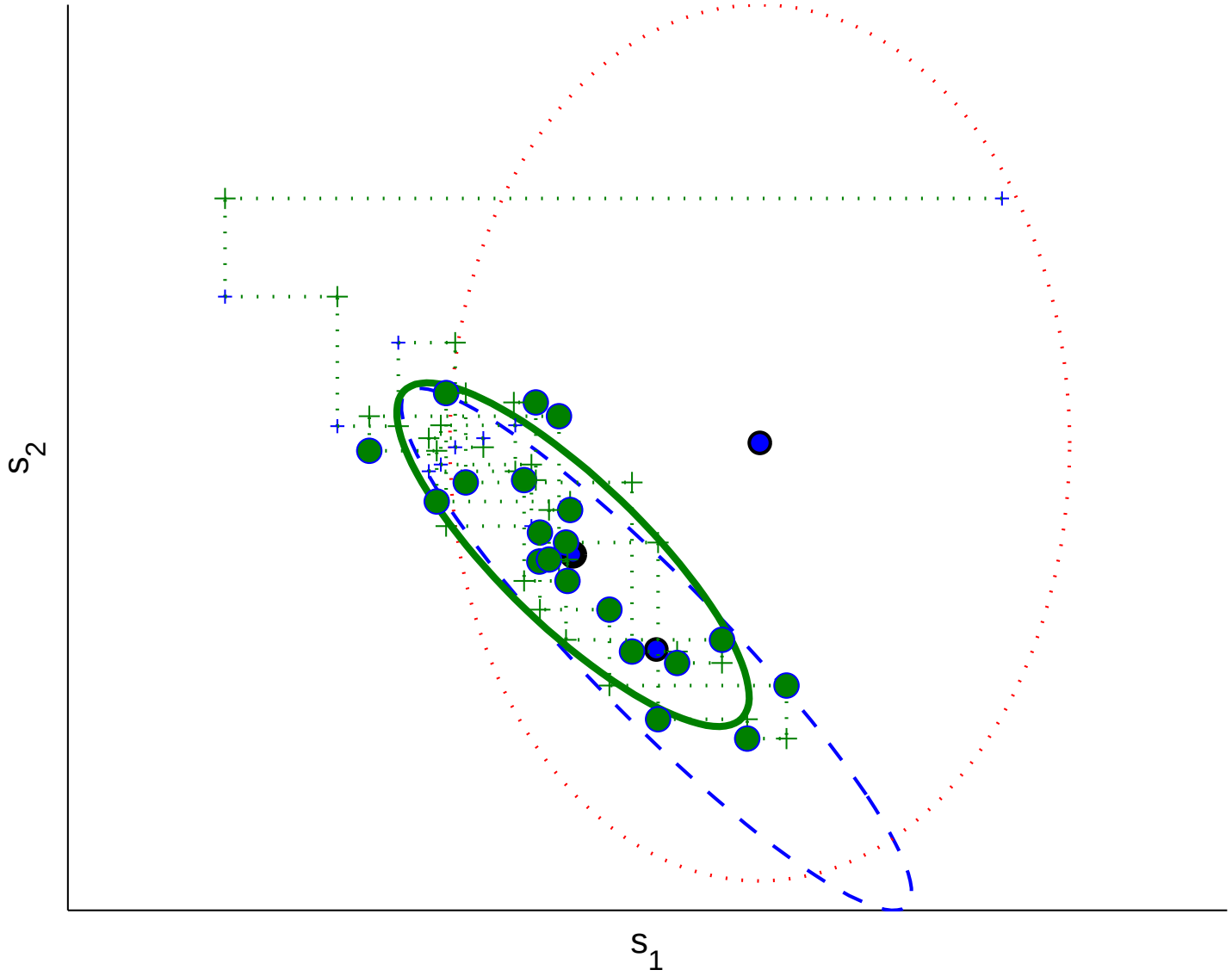
$$\begin{aligned} s_1^{(t+1)} &\sim p(s_1 | s_2^{(t)}, x = \hat{x}) &&\propto \phi(s_1, s_2^{(t)}) \\ s_2^{(t+1)} &\sim p(s_2 | s_1^{(t+1)}, x = \hat{x}) &&\propto \phi(s_1^{(t+1)}, s_2) \end{aligned}$$

- This algorithm, that samples from above conditional marginals is a particular instance of the **Gibbs sampler**.
- The desired posterior $\mathcal{P}$ is the stationary distribution of T (why? – later...).
- Note the algorithmic similarity to ICM. In Gibbs, we make a random move instead of directly going to the conditional mode.

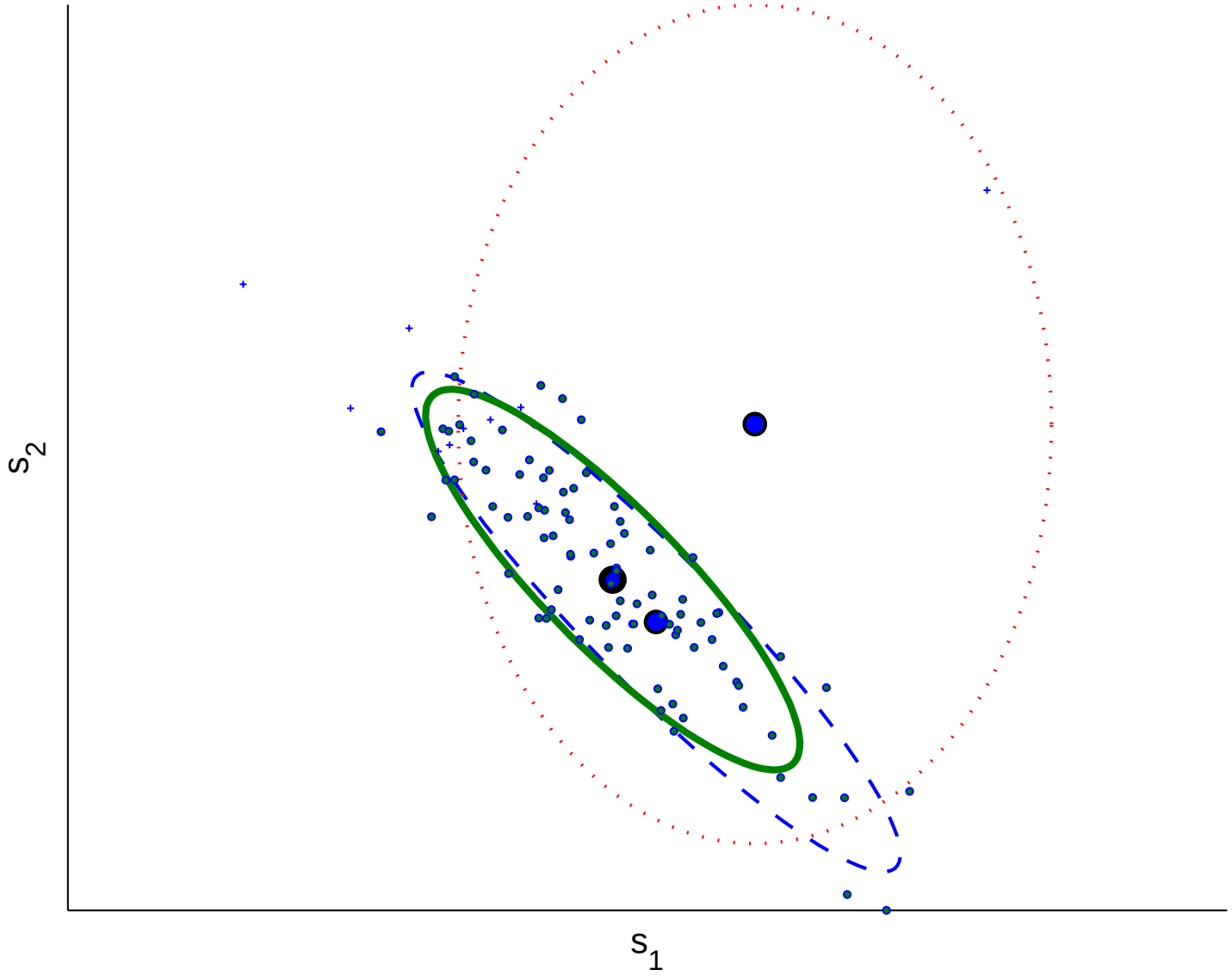
Gibbs Sampling



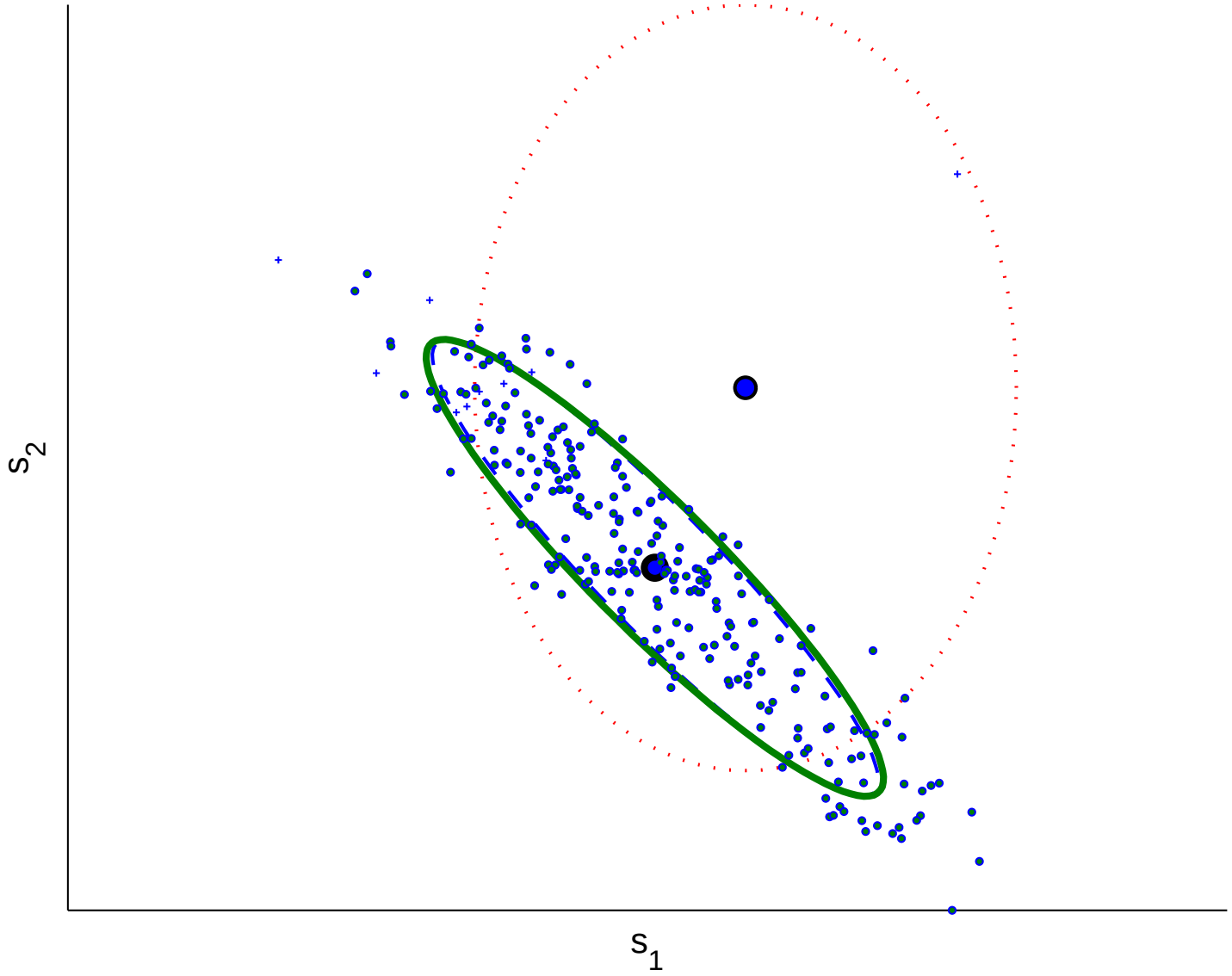
Gibbs Sampling, $t = 20$



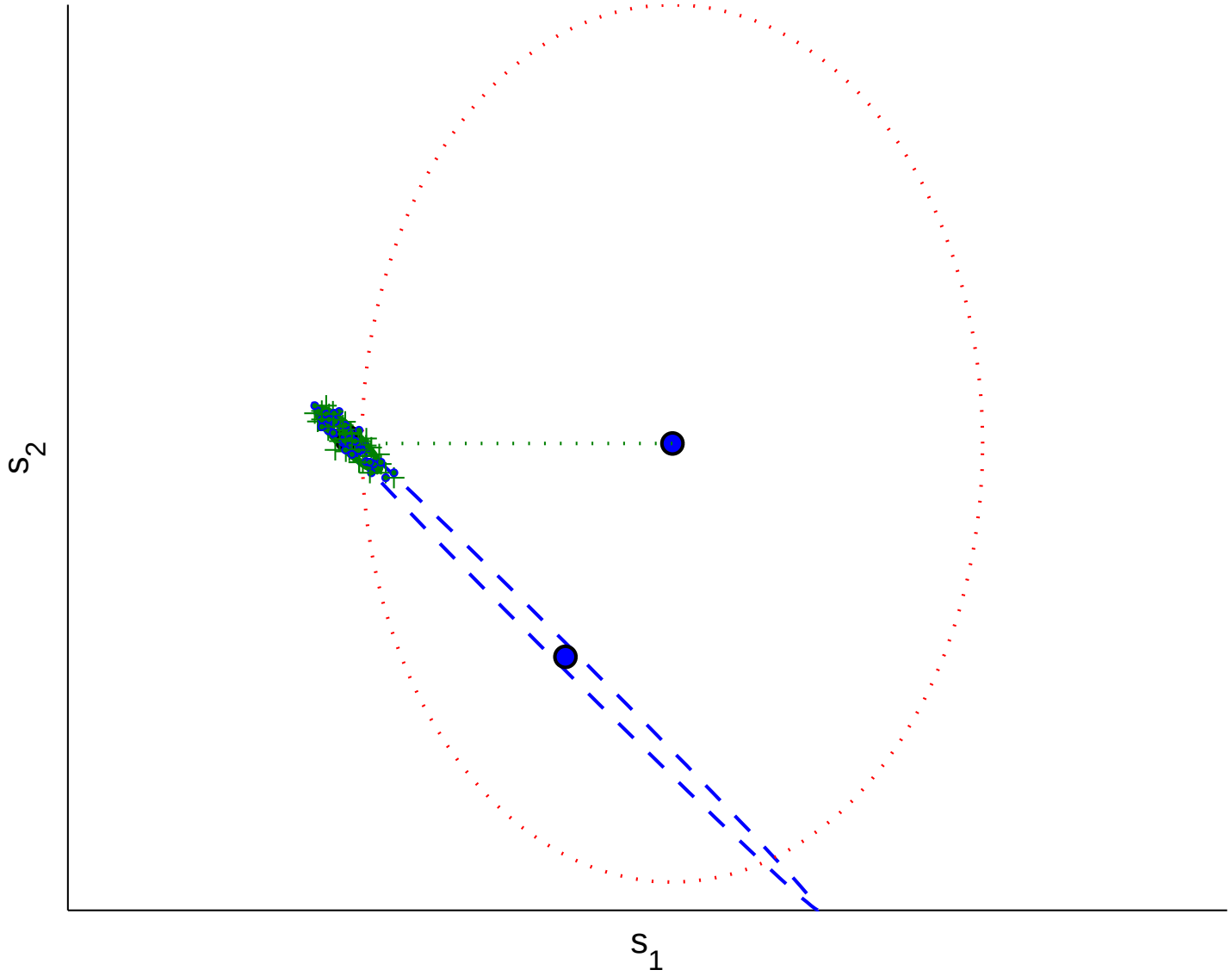
Gibbs Sampling, $t = 100$



Gibbs Sampling, $t = 250$



Gibbs Sampling, Slow convergence



Markov Chain Monte Carlo (MCMC)

- Construct a transition kernel $T(\mathbf{s}'|\mathbf{s})$ with the stationary distribution $\mathcal{P} = \phi(\mathbf{s})/Z_x \equiv \pi(\mathbf{s})$ for any initial distribution $r(\mathbf{s})$.

$$\pi(\mathbf{s}) = T^{\infty} r(\mathbf{s}) \tag{5}$$

- Sample $\mathbf{s}^{(0)} \sim r(\mathbf{s})$
- For $t = 1 \ldots \infty$, Sample $\mathbf{s}^{(t)} \sim T(\mathbf{s}|\mathbf{s}^{(t-1)})$
- Estimate any desired expectation by the average

$$\langle f(\mathbf{s}) \rangle_{\pi(\mathbf{s})} \approx \frac{1}{t - t_0} \sum_{n=t_0}^{t} f(\mathbf{s}^{(n)})$$

  where t_0 is a preset burn-in period.

But how to construct T and verify that $\pi(\mathbf{s})$ is indeed its stationary distribution ?

Equilibrium condition = Detailed Balance

$$T(\mathbf{s}|\mathbf{s}')\pi(\mathbf{s}') \quad = \quad T(\mathbf{s}'|\mathbf{s})\pi(\mathbf{s})$$

If detailed balance is satisfied then $\pi(\mathbf{s})$ is a stationary distribution

$$\pi(\mathbf{s}) \quad = \quad \int d\mathbf{s}' T(\mathbf{s}|\mathbf{s}')\pi(\mathbf{s}')$$

If the configuration space is discrete, we have

$$\begin{aligned} \pi(\mathbf{s}) &= \sum_{\mathbf{s}'} T(\mathbf{s}|\mathbf{s}')\pi(\mathbf{s}') \\ \pi &= T\pi \end{aligned}$$

π has to be a (right) eigenvector of T.

Conditions on T

- Irreducibility (probabilisic connectedness): Every state s' can be reached from every s

$$T(s'|s) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$ is **not** irreducible

- Aperiodicity : Cycling around is not allowed

$$T(s'|s) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$ is **not** aperiodic

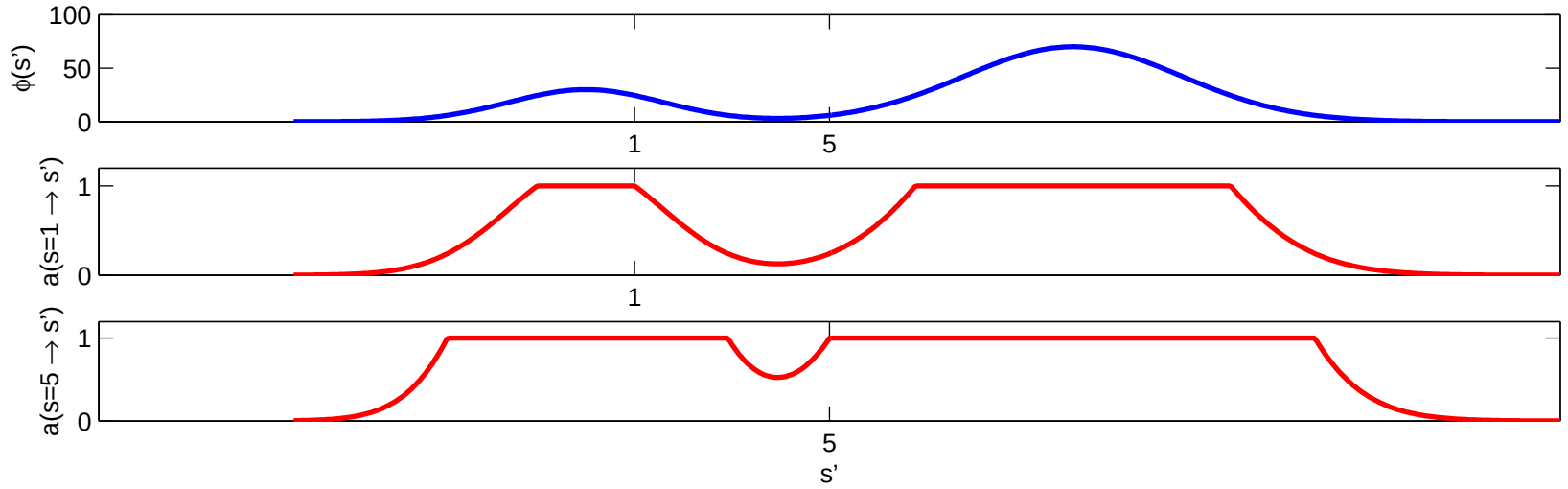
Surprisingly, it is easy to construct a transition kernel with these properties by following the recipe provided by Metropolis (1953) and Hastings (1970).

Metropolis-Hastings Kernel

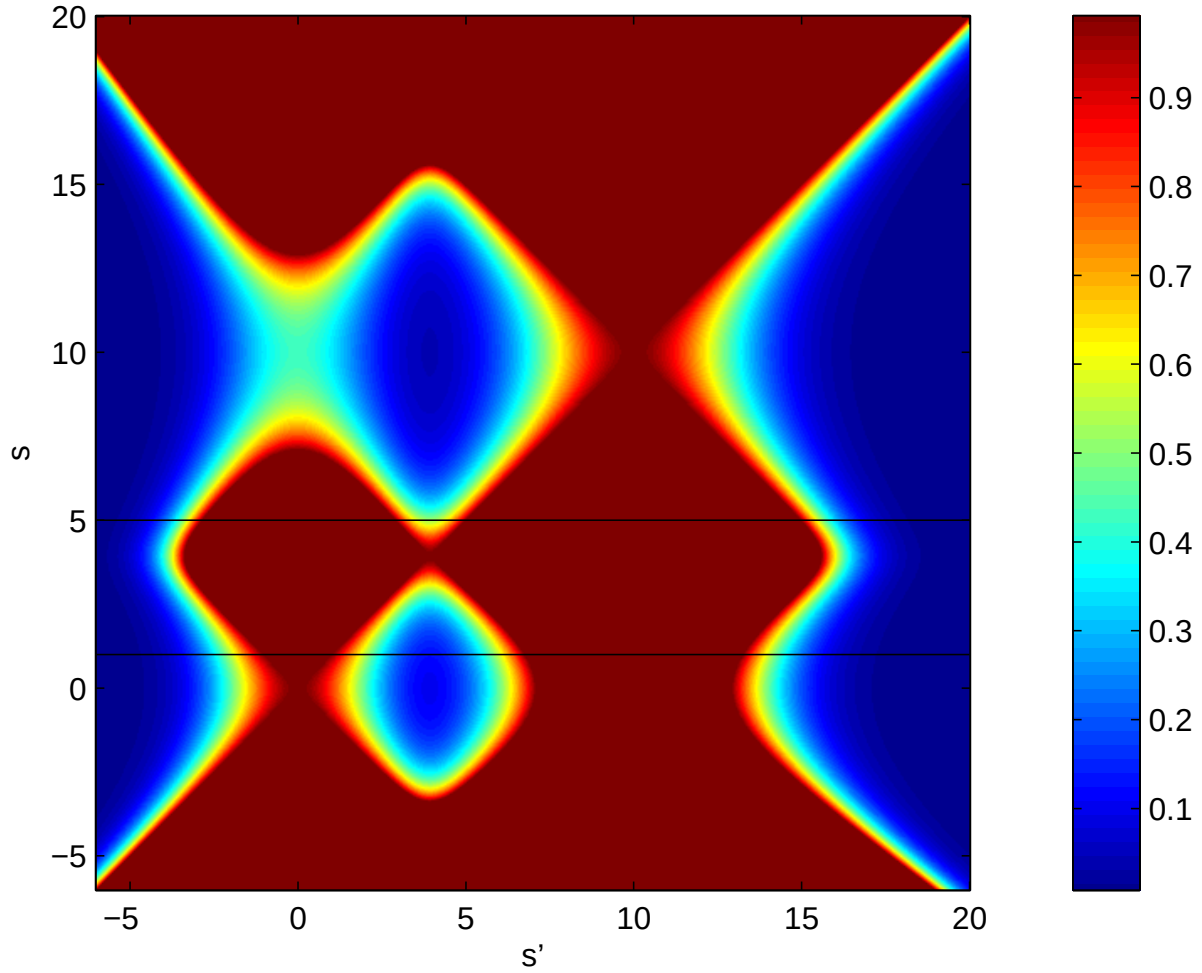
- We choose an arbitrary proposal distribution $q(s'|s)$ (that satisfies mild regularity conditions).
  (When q is symmetric, i.e., $q(s'|s) = q(s|s')$, we have a Metropolis algorithm.)

- We define the *acceptance probability* of a jump from s to s' as

$$a(s \to s') \quad \equiv \quad \min\{1, \frac{q(s|s')\pi(s')}{q(s'|s)\pi(s)}\}$$



Acceptance Probability $a(s \rightarrow s')$



Basic MCMC algorithm: Metropolis-Hastings

1. Initialize: $s^{(0)} \sim r(s)$

2. For $t = 1, 2, \ldots$

   - Propose:

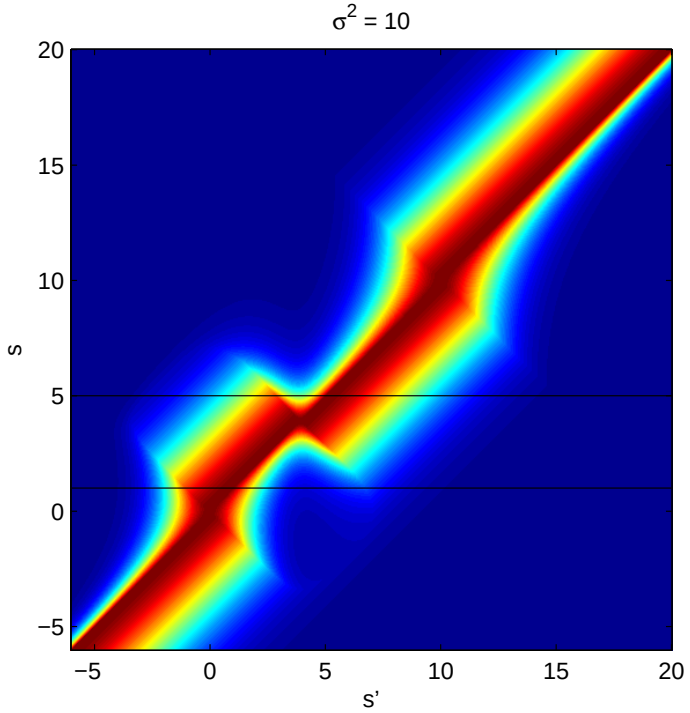
$$s' \sim q(s'|s^{(t-1)})$$

   - Evaluate Proposal: $u \sim \text{Uniform}[0, 1]$

$$s^{(t)} := \begin{cases} s' & u < a(s^{(t-1)} \to s') \quad \text{Accept} \\ s^{(t-1)} & \text{otherwise} \quad \text{Reject} \end{cases}$$

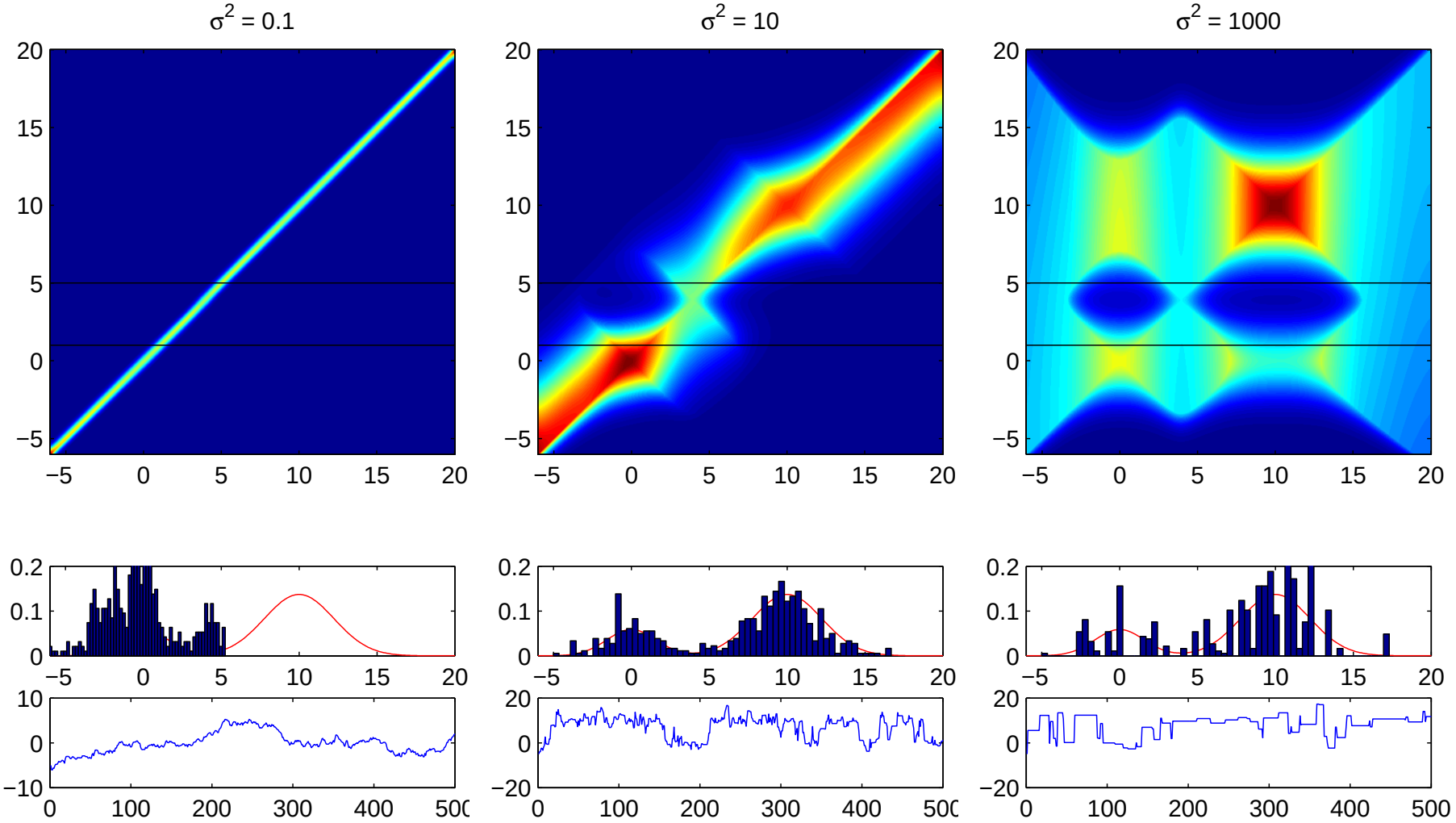
Transition Kernel of the Metropolis Algorithm

$$T(s'|s) = \underbrace{q(s'|s)a(s \to s')}_{\text{Accept}} + \underbrace{\delta(s'-s)\int ds' q(s'|s)(1-a(s \to s'))}_{\text{Reject}}$$



Only Accept part for visual convenience

Various Kernels with the same stationary distribution



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

Cascades and Mixtures of Transition Kernels

Let T_1 and T_2 have the same stationary distribution $p(s)$.

Then:

$$\begin{aligned} T_c &= T_1 T_2 \\ T_m &= \nu T_1 + (1-\nu) T_2 \quad 0 \leq \nu \leq 1 \end{aligned}$$

are also transition kernels with stationary distribution $p(s)$.

This opens up many possibilities to “tailor” application specific algorithms.

For example let

T_1 : global proposal (allows large “jumps”)
T_2 : local proposal (investigates locally)

We can use T_m and adjust ν as a function of rejection rate.

Optimization : Simulated Annealing and Iterative Improvement

For optimization, (e.g. to find a MAP solution)

$$s^* = \arg\max_{s \in \mathcal{S}} \pi(s)$$

The MCMC sampler may not visit s^*.

Simulated Annealing: We define the target distribution as

$$\pi(s)^{\tau_i}$$

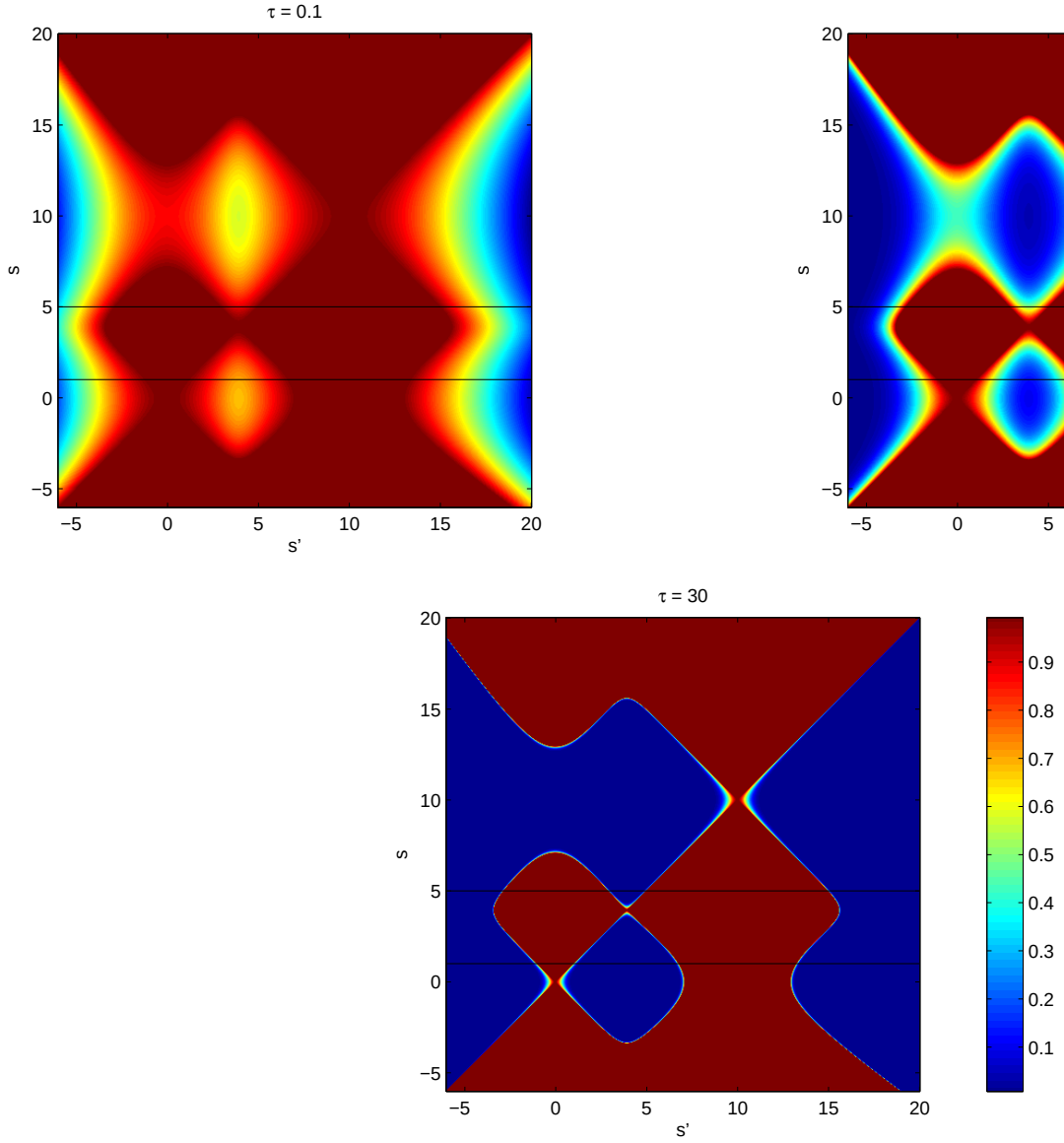
where τ_i is an annealing schedule. For example,

$$\tau_1 = 0.1, \ldots, \tau_N = 10, \tau_{N+1} = \infty \ldots$$

Iterative Improvement (greedy search) is a special case of SA

$$\tau_1 = \tau_2 = \cdots = \tau_N = \infty$$

Acceptance probabilities $a(s \rightarrow s')$ at different τ



Importance Sampling,

Online Inference, Sequential Monte Carlo

Importance Sampling

Consider a probability distribution with $Z = \int d\mathbf{x}\phi(\mathbf{x})$

$$p(\mathbf{x}) = \frac{1}{Z}\phi(\mathbf{x}) \tag{6}$$

Estimate expectations (or features) of $p(\mathbf{x})$ by a weighted sample

$$\langle f(\mathbf{x})\rangle_{p(\mathbf{x})} = \int dx f(\mathbf{x})p(\mathbf{x})$$

$$\langle f(\mathbf{x})\rangle_{p(\mathbf{x})} \approx \sum_{i=1}^{N} \tilde{w}^{(i)} f(\mathbf{x}^{(i)}) \tag{7}$$

Importance Sampling (cont.)

- Change of measure with **weight function** $W(\mathbf{x}) \equiv \phi(x)/q(x)$

$$\langle f(\mathbf{x}) \rangle_{p(\mathbf{x})} = \frac{1}{Z} \int d\mathbf{x} f(\mathbf{x}) \frac{\phi(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) = \frac{1}{Z} \left\langle f(\mathbf{x}) \frac{\phi(\mathbf{x})}{q(\mathbf{x})} \right\rangle_{q(\mathbf{x})} \equiv \frac{1}{Z} \langle f(\mathbf{x}) W(\mathbf{x}) \rangle_{q(\mathbf{x})}$$

- If Z is unknown, as is often the case in Bayesian inference

$$Z = \int d\mathbf{x} \phi(\mathbf{x}) = \int d\mathbf{x} \frac{\phi(\mathbf{x})}{q(\mathbf{x})} q(\mathbf{x}) = \langle W(\mathbf{x}) \rangle_{q(\mathbf{x})}$$

$$\langle f(\mathbf{x}) \rangle_{p(\mathbf{x})} = \frac{\langle f(\mathbf{x}) W(\mathbf{x}) \rangle_{q(\mathbf{x})}}{\langle W(\mathbf{x}) \rangle_{q(\mathbf{x})}}$$

Importance Sampling (cont.)

- Draw $i = 1, \ldots N$ independent samples from q

$$\mathbf{x}^{(i)} \sim q(\mathbf{x})$$

- We calculate the **importance weights**

$$W^{(i)} \quad = \quad W(\mathbf{x}^{(i)}) = \phi(\mathbf{x}^{(i)})/q(\mathbf{x}^{(i)})$$

- Approximate the normalizing constant

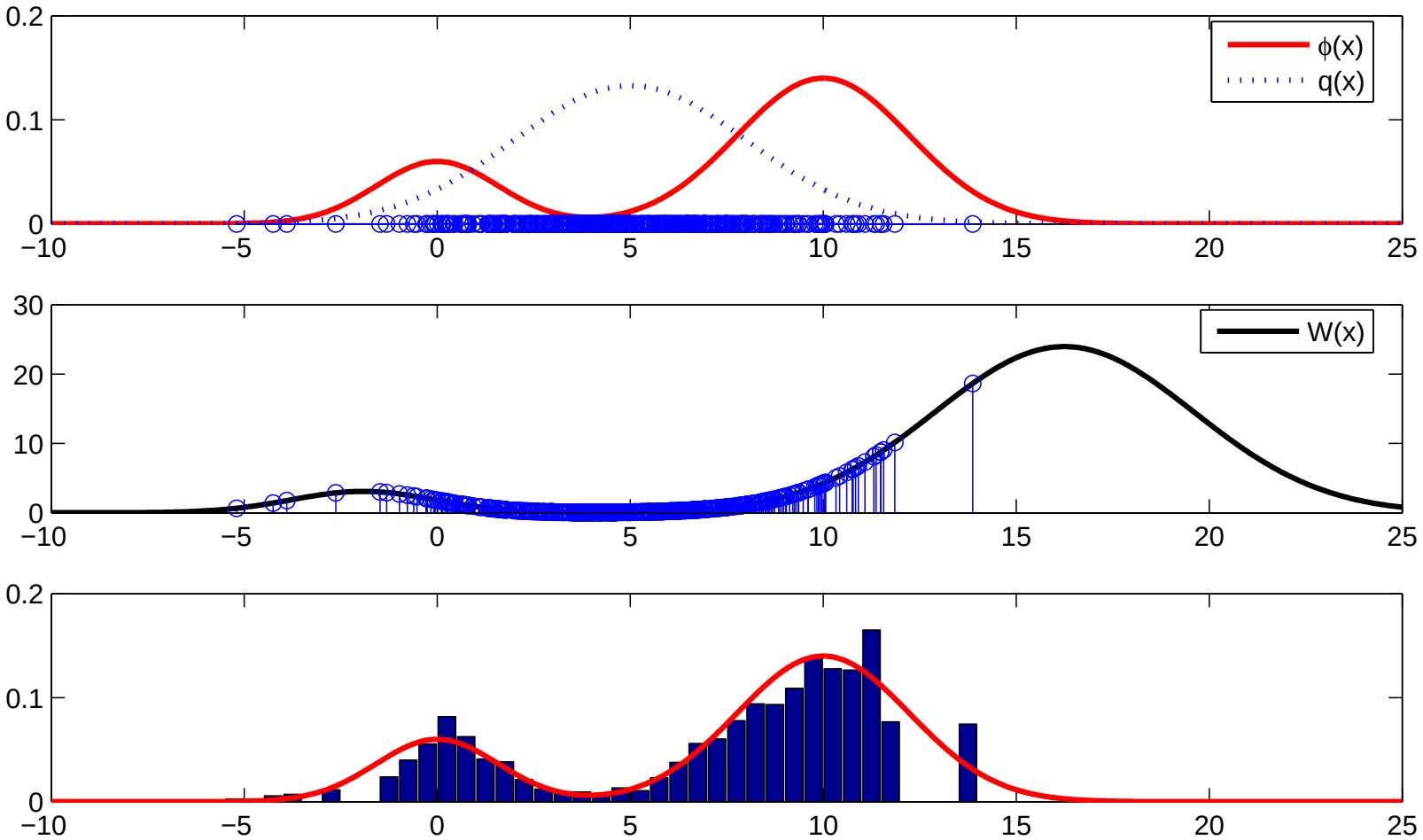
$$Z \quad = \quad \langle W(\mathbf{x}) \rangle_{q(\mathbf{x})} \approx \sum_{i=1}^{N} W^{(i)}$$

- Desired expectation is approximated by

$$\langle f(\mathbf{x}) \rangle_{p(\mathbf{x})} \quad = \quad \frac{\langle f(\mathbf{x}) W(\mathbf{x}) \rangle_{q(\mathbf{x})}}{\langle W(\mathbf{x}) \rangle_{q(\mathbf{x})}} \approx \frac{\sum_{i=1}^{N} W^{(i)} f(\mathbf{x}^{(i)})}{\sum_{i=1}^{N} W^{(i)}} \equiv \sum_{i=1}^{N} \tilde{w}^{(i)} f(\mathbf{x}^{(i)})$$

Here $\tilde{w}^{(i)} = W^{(i)} / \sum_{j=1}^{N} W^{(j)}$ are *normalized importance weights*.

Importance Sampling (cont.)



Resampling

- Importance sampling computes an approximation with weighted delta functions

$$p(x) \approx \sum_i \tilde{W}^{(i)} \delta(x - x^{(i)})$$

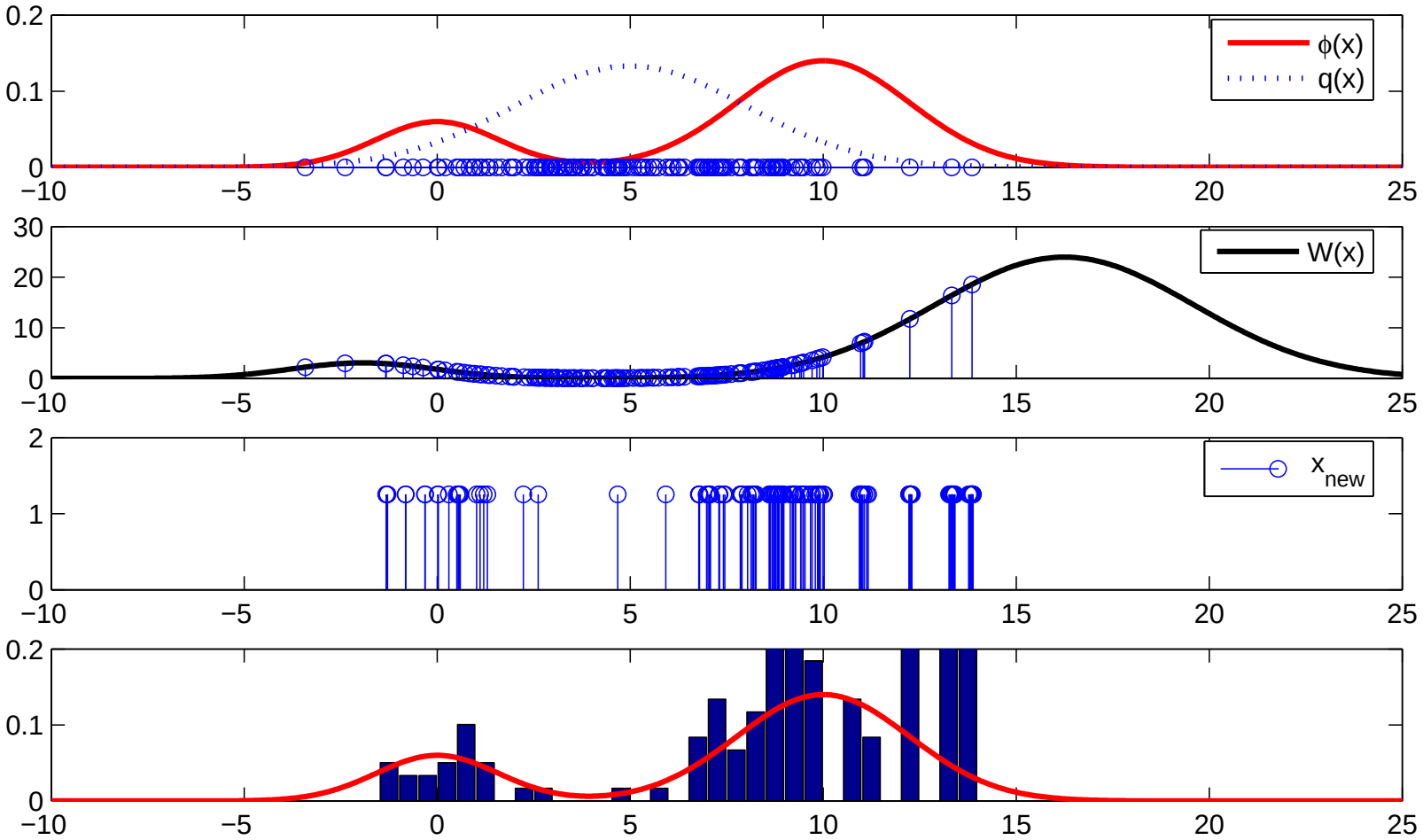
- In this representation, most of $\tilde{W}^{(i)}$ will be very close to zero and the representation may be dominated by few large weights.
- Resampling samples a set of new “particles”

$$x_{\text{new}}^{(j)} \sim \sum_i \tilde{W}^{(i)} \delta(x - x^{(i)})$$

$$p(x) \approx \frac{1}{N} \sum_j \delta(x - x_{\text{new}}^{(j)})$$

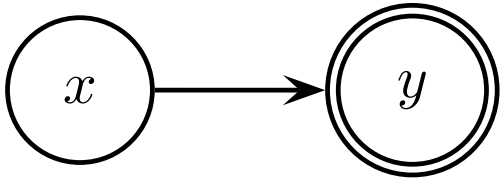
- Since we sample from a degenerate distribution, particle locations stay unchanged. We merely dublicate (, triplicate, ...) or discard particles according to their weight.
- This process is also named “selection”, “survival of the fittest”, e.t.c., in various fields (Genetic algorithms, AI..).

Resampling



$$x_{\text{new}}^{(j)} \sim \sum_i \tilde{W}^{(i)} \delta\left(x - x^{(i)}\right)$$

Examples of Proposal Distributions



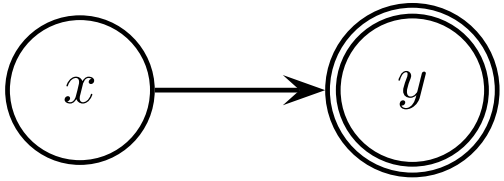
$$p(x|y) \propto p(y|x)p(x)$$

Task: Obtain samples from the posterior $p(x|y)$

- Prior as the proposal. $q(x) = p(x)$

$$W(x) = \frac{p(y|x)p(x)}{p(x)} = p(y|x)$$

Examples of Proposal Distributions



$$p(x|y) \propto p(y|x)p(x)$$

Task: Obtain samples from the posterior $p(x|y)$

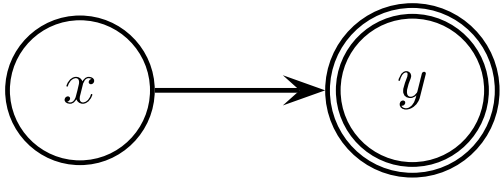
- Likelihood as the proposal. $q(x) = p(y|x)/\int dx p(y|x) = p(y|x)/c(y)$

$$W(x) = \frac{p(y|x)p(x)}{p(y|x)/c(y)} = p(x)c(y) \propto p(x)$$

- Interesting when sensors are very accurate and $\dim(y) \gg \dim(x)$. Idea behind "Dual-PF" (Thrun et.al.. 2000)

Since there are many proposals, is there a "best" proposal distribution?

Optimal Proposal Distribution



$$p(x|y) \propto p(y|x)p(x)$$

Task: Estimate $\langle f(x) \rangle_{p(x|y)}$

- IS constructs the estimator $I(f) = \langle f(x)W(x) \rangle_{q(x)}$ (where $W(x) = p(x|y)/q(x)$)
- Minimize the variance of the estimator

$$\begin{aligned} \left\langle \left(f(x)W(x) - \langle f(x)W(x) \rangle \right)^2 \right\rangle_{q(x)} &= \left\langle f^2(x)W^2(x) \right\rangle_{q(x)} - \langle f(x)W(x) \rangle^2_{q(x)} && (8) \\ &= \left\langle f^2(x)W^2(x) \right\rangle_{q(x)} - \langle f(x) \rangle^2_{p(x)} && (9) \\ &= \left\langle f^2(x)W^2(x) \right\rangle_{q(x)} - I^2(f) && (10) \end{aligned}$$

- Minimize the first term since only it depends upon q

Optimal Proposal Distribution

- (By Jensen's inequality) The first term is lower bounded:

$$\left\langle f^2(x)W^2(x)\right\rangle_{q(x)} \quad \geq \quad \langle |f(x)|W(x)\rangle^2_{q(x)} = \left(\int |f(x)|\, p(x|y)dx\right)^2$$

- We well look for a distribution q^* that attains this lower bound. Take

$$q^*(x) = \frac{|f(x)|p(x|y)}{\int |f(x')|p(x'|y)dx'}$$

Optimal Proposal Distribution (cont.)

- The weight function for this particular proposal q^* is

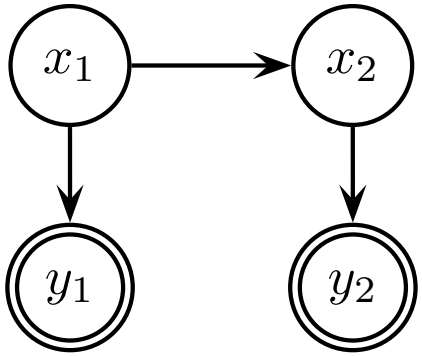
$$W_*(x) = p(x|y)/q^*(x) = \frac{\int |f(x')|p(x'|y)dx'}{|f(x)|}$$

- We show that q^* attains its lower bound

$$\begin{aligned}
\left\langle f^2(x)W_*^2(x)\right\rangle_{q^*(x)} &= \left\langle f^2(x)\frac{\left(\int |f(x')|p(x'|y)dx'\right)^2}{|f(x)|^2}\right\rangle_{q^*(x)} \\
&= \left(\int |f(x')|p(x'|y)dx'\right)^2 = \langle|f(x)|\rangle^2_{p(x|y)} \\
&= \langle|f(x)|W_*(x)\rangle^2_{q^*(x)}
\end{aligned}$$

- $\Rightarrow$ There are distributions q^* that are even "better" than the exact posterior!

Examples of Proposal Distributions



$$p(x|y) \propto p(y_1|x_1)p(x_1)p(y_2|x_2)p(x_2|x_1)$$

Task: Obtain samples from the posterior $p(x_{1:2}|y_{1:2})$

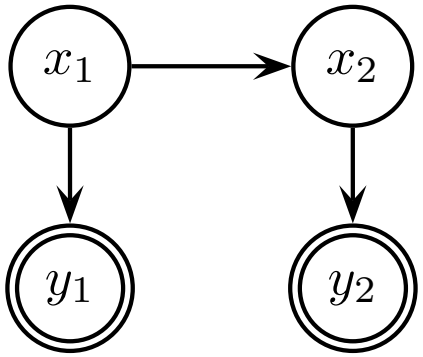
- Prior as the proposal. $q(x_{1:2}) = p(x_1)p(x_2|x_1)$

$$W(x_1, x_2) = p(y_1|x_1)p(y_2|x_2)$$

- We sample from the prior as follows:

$$x_1^{(i)} \sim p(x_1) \qquad x_2^{(i)} \sim p(x_2|x_1 = x_1^{(i)}) \qquad W(\mathbf{x}^{(i)}) = p(y_1|x_1^{(i)})p(y_2|x_2^{(i)})$$

Examples of Proposal Distributions



$$p(x|y) \propto p(y_1|x_1)p(x_1)p(y_2|x_2)p(x_2|x_1)$$

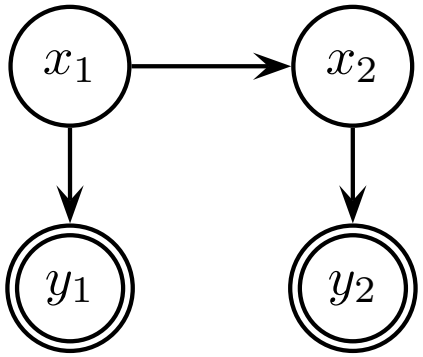
- State prediction as the proposal. $q(x_{1:2}) = p(x_1|y_1)p(x_2|x_1)$

$$W(x_1, x_2) = \frac{p(y_1|x_1)p(x_1)p(y_2|x_2)p(x_2|x_1)}{p(x_1|y_1)p(x_2|x_1)} = p(y_1)p(y_2|x_2)$$

- Note that this proposal does not depend on x_1
- We sample from the proposal and compute the weight

$$x_1^{(i)} \sim p(x_1|y_1) \qquad x_2^{(i)} \sim p(x_2|x_1 = x_1^{(i)}) \qquad W(\mathbf{x}^{(i)}) = p(y_1)p(y_2|x_2^{(i)})$$

Examples of Proposal Distributions



$$p(x|y) \propto p(y_1|x_1)p(x_1)p(y_2|x_2)p(x_2|x_1)$$

- Filtering distribution as the proposal. $q(x_{1:2}) = p(x_1|y_1)p(x_2|x_1, y_2)$

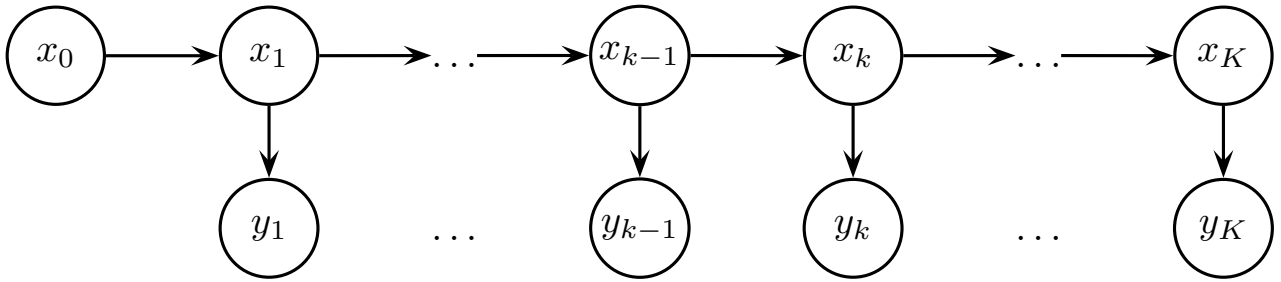
$$W(x_1, x_2) = \frac{p(y_1|x_1)p(x_1)p(y_2|x_2)p(x_2|x_1)}{p(x_1|y_1)p(x_2|x_1, y_2)} = p(y_1)p(y_2|x_1)$$

- Note that this proposal does not depend on x_2
- We sample from the proposal and compute the weight

$$x_1^{(i)} \sim p(x_1|y_1) \qquad x_2^{(i)} \sim p(x_2|x_1 = x_1^{(i)}, y_2) \qquad W(\mathbf{x}^{(i)}) = p(y_1)p(y_2|x_1^{(i)})$$

Online Inference, Terminology

In signal processing we often have dynamical state space models (SSM)



$$x_k \sim p(x_k|x_{k-1}) \qquad \text{Transition Model}$$

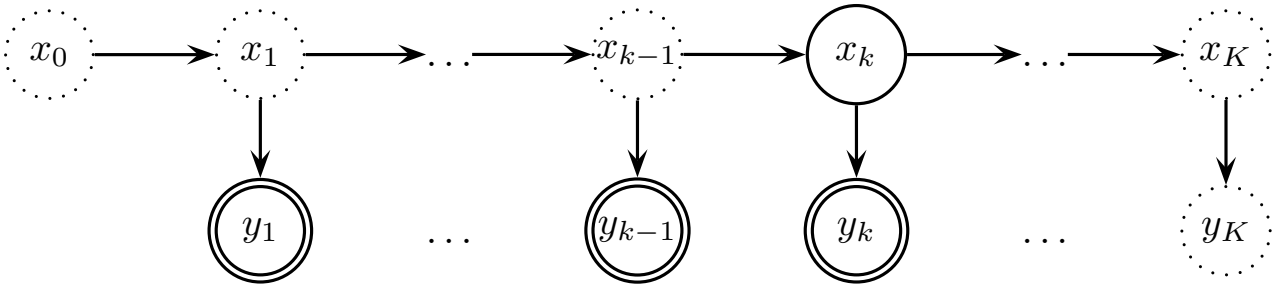
$$y_k \sim p(y_k|x_k) \qquad \text{Observation Model}$$

Here, x is the latent state and y are observations. In a Bayesian setting, x can also include unknown model parameters. This model is very generic and includes as special cases:

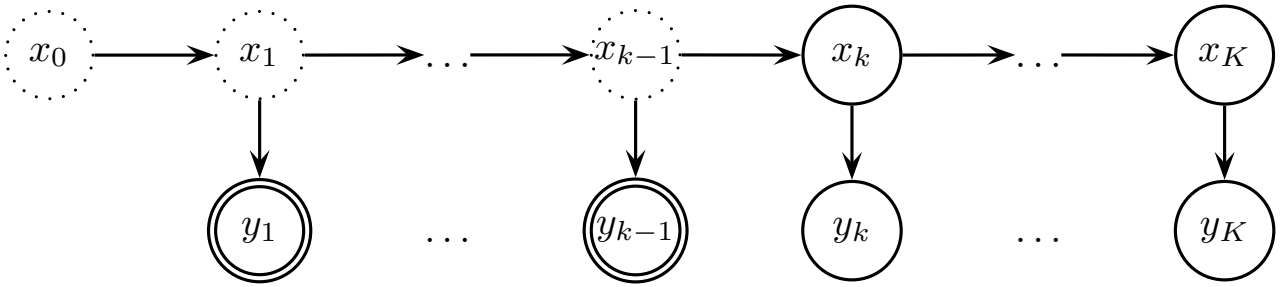
- Linear Dynamical Systems (Kalman Filter models)
- (Time varying) AR, ARMA, MA models
- Hidden Markov Models, Switching state space models
- Dynamic Bayesian networks, Nonlinear Stochastic Dynamical Systems

Online Inference, Terminology

- **Filtering** $p(x_k|y_{1:k})$
  belief state—distribution of current state given all past information

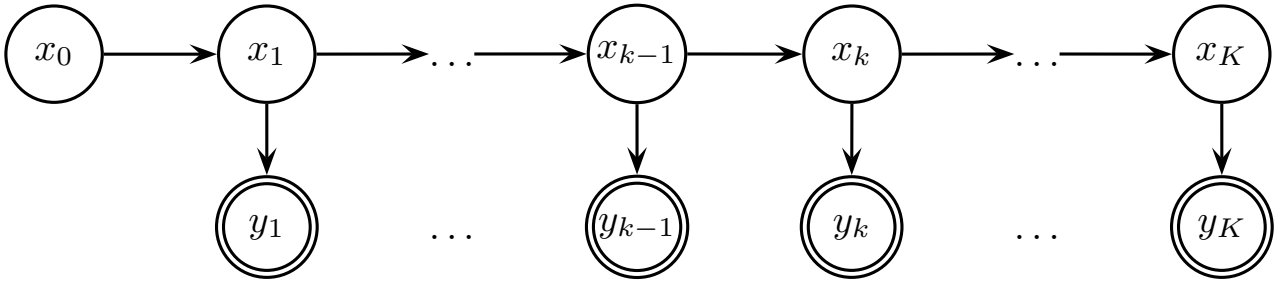


- **Prediction** $p(y_{k:K}, x_{k:K}|y_{1:k-1})$
  evaluation of possible future outcomes; like filtering without observations

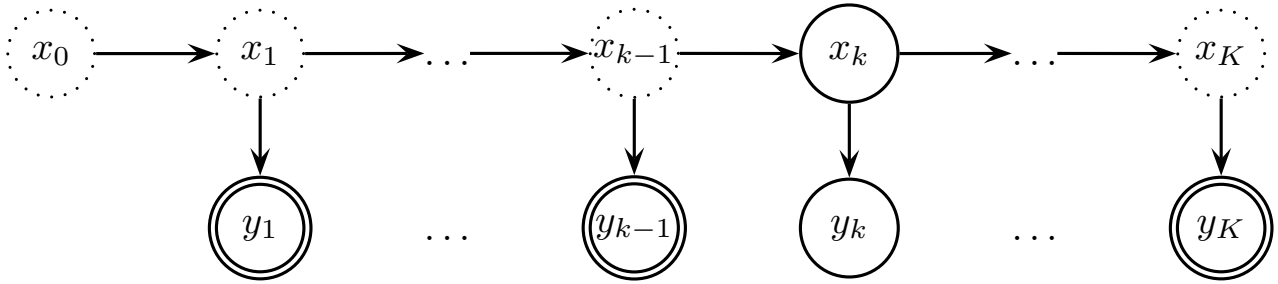


Online Inference, Terminology

- **Smoothing** $p(x_{0:K}|y_{1:K})$,
  Most likely trajectory – Viterbi path $\arg\max_{x_{0:K}} p(x_{0:K}|y_{1:K})$
  better estimate of past states, essential for learning



- **Interpolation** $p(y_k, x_k|y_{1:k-1}, y_{k+1:K})$
  fill in lost observations given past and future



Sequential Importance Sampling, Particle Filtering

Apply importance sampling to the SSM to obtain some samples from the posterior $p(x_{0:K}|y_{1:K})$.

$$p(x_{0:K}|y_{1:K}) \quad = \quad \frac{1}{p(y_{1:K})} p(y_{1:K}|x_{0:K}) p(x_{0:K}) \equiv \frac{1}{Z_y} \phi(x_{0:K}) \tag{11}$$

Key idea: sequential construction of the proposal distribution q, possibly using the available observations $y_{1:k}$, i.e.

$$q(x_{1:K}|y_{1:K}) = q(x_0) \prod_{k=1}^{K} q(x_k|x_{1:k-1} y_{1:k})$$

Sequential Importance Sampling

Due to the sequential nature of the model and the proposal, the importance weight function $W(x_{0:k}) \equiv W_k$ admits *recursive* computation

$$W_k = \frac{\phi(x_{0:k})}{q(x_{0:k}|y_{1:k})} = \frac{p(y_k|x_k)p(x_k|x_{k-1})}{q(x_k|x_{0:k-1}y_{1:k})} \frac{\phi(x_{0:k-1})}{q(x_{0:k-1}|y_{1:k-1})} \tag{12}$$

$$= \frac{p(y_k|x_k)p(x_k|x_{k-1})}{q(x_k|x_{0:k-1}, y_{1:k})} W_{k-1} \equiv u_{k|0:k-1} W_{k-1} \tag{13}$$

Suppose we had an approximation to the posterior (in the sense $\langle f(x) \rangle_\phi \approx \sum_i W_{k-1}^{(i)} f(x_{0:k-1}^{(i)})$)

$$\phi(x_{0:k-1}) \approx \sum_i W_{k-1}^{(i)} \delta(x_{0:k-1} - x_{0:k-1}^{(i)})$$

$$x_k^{(i)} \sim q(x_k|x_{0:k-1}^{(i)}, y_{1:k}) \qquad \text{Extend trajectory}$$

$$W_k^{(i)} = u_{k|0:k-1}^{(i)} W_{k-1} \qquad \text{Update weight}$$

$$\phi(x_{0:k}) \approx \sum_i W_k^{(i)} \delta(x_{0:k} - x_{0:k}^{(i)})$$

Example

- Prior as the proposal density

$$q(x_k|x_{0:k-1}, y_{1:k}) \quad = \quad p(x_k|x_{k-1})$$

- The weight is given by

$$\begin{aligned} x_k^{(i)} &\sim p(x_k|x_{k-1}^{(i)}) && \text{Extend trajectory} \\ W_k^{(i)} &= u_{k|0:k-1}^{(i)} W_{k-1} && \text{Update weight} \\ &= \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{p(x_k^{(i)}|x_{k-1}^{(i)})} W_{k-1}^{(i)} = p(y_k|x_k^{(i)}) W_{k-1}^{(i)} \end{aligned}$$

- However, this schema will **not** work, since we blindly sample from the prior. But ...

Example (cont.)

- Perhaps surprisingly, interleaving importance sampling steps with (occasional) resampling steps makes the approach work quite well !!

$$x_k^{(i)} \sim p(x_k|x_{k-1}^{(i)}) \qquad \text{Extend trajectory}$$

$$W_k^{(i)} = p(y_k|x_k^{(i)})W_{k-1}^{(i)} \qquad \text{Update weight}$$

$$\tilde{W}_k^{(i)} = W_k^{(i)}/\tilde{Z}_k \qquad \text{Normalize } (\tilde{Z}_k \equiv \sum_{i'} W_k^{(i')})$$

$$x_{0:k,\text{new}}^{(j)} \sim \sum_{i=1}^{N} \tilde{W}^{(i)} \delta(x_{0:k} - x_{0:k}^{(i)}) \qquad \text{Resample } \; j = 1 \ldots N$$

- This results in a new representation as

$$\phi(x) \quad \approx \quad \frac{1}{N} \sum_j \tilde{Z}_k \delta(x_{0:k} - x_{0:k,\text{new}}^{(j)})$$

$$x_{0:k}^{(i)} \leftarrow x_{0:k,\text{new}}^{(j)} \qquad W_k^{(i)} \leftarrow \tilde{Z}_k/N$$

Optimal proposal distribution

- The algorithm in the previous example is known as *Bootstrap particle filter* or *Sequential Importance Sampling/Resampling* (SIS/SIR).

- Can we come up with a better proposal in a sequential setting?
  - We are not allowed to move previous sampling points $x_{1:k-1}^{(i)}$ (because in many applications we can't even store them)
  - Better in the sense of minimizing the variance of weight function $W_k(x)$. (remember the optimality story in Eq.(10) and set $f(x) = 1$).

- The answer turns out to be the filtering distribution

$$q(x_k|x_{1:k-1}, y_{1:k}) = p(x_k|x_{k-1}, y_k) \tag{14}$$

Optimal proposal distribution (cont.)

- The weight is given by

$$
\begin{aligned}
x_k^{(i)} &\sim p(x_k|x_{k-1}^{(i)}, y_k) && \text{Extend trajectory} \\
W_k^{(i)} &= u_{k|0:k-1}^{(i)} W_{k-1}^{(i)} && \text{Update weight} \\
u_{k|0:k-1}^{(i)} &= \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{p(x_k^{(i)}|x_{k-1}^{(i)}, y_k)} \times \frac{p(y_k|x_{k-1}^{(i)})}{p(y_k|x_{k-1}^{(i)})} \\
&= \frac{p(y_k, x_k^{(i)}|x_{k-1}^{(i)})p(y_k|x_{k-1}^{(i)})}{p(x_k^{(i)}, y_k|x_{k-1}^{(i)})} = p(y_k|x_{k-1}^{(i)})
\end{aligned}
$$

A Generic Particle Filter

1. **Generation**:
   Compute the proposal distribution $q(x_k|x_{0:k-1}^{(i)}, y_{1:k})$.
   Generate offsprings for $i = 1 \ldots N$

$$\hat{x}_k^{(i)} \quad \sim \quad q(x_k|x_{0:k-1}^{(i)}, y_{1:k})$$

2. **Evaluate** importance weights

$$W_k^{(i)} \quad = \quad \frac{p(y_k|\hat{x}_k^{(i)})p(\hat{x}_k^{(i)}|x_{k-1}^{(i)})}{q(\hat{x}_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})} W_{k-1}^{(i)} \qquad x_{0:k}^{(i)} = (\hat{x}_k^{(i)}, x_{0:k-1}^{(i)})$$

3. **Resampling** (optional but recommended)

Normalize weigts	$\tilde{W}_k^{(i)} = W_k^{(i)} / \tilde{Z}_k$	$\tilde{Z}_k \equiv \sum_j W_k^{(j)}$
Resample	$x_{0:k,\text{new}}^{(j)} \sim \sum_{i=1}^{N} \tilde{W}^{(i)} \delta(x_{0:k} - x_{0:k}^{(i)})$	$j = 1 \ldots N$
Reset	$x_{0:k}^{(i)} \leftarrow x_{0:k,\text{new}}^{(j)}$	$W_k^{(i)} \leftarrow \tilde{Z}_k / N$

Summary of what we have (hopefully) covered

- Deterministic
  - Variational Bayes, Mean field
  - Expectation/Maximization (EM), Iterative Conditional Modes (ICM)
- Stochastic
  - Markov Chain Monte Carlo
  - Importance Sampling,
  - Particle filtering

Summary of what we have not covered

- Exact Inference (Belief Propagation, Junction Tree ...)
- Deterministic
  - Assumed Density Filter (ADF), Extended Kalman Filter (EKF), Unscented Particle Filter
  - Structured Mean field
  - Loopy Belief Propagation, Expectation Propagation, Generalized Belief Propagation
  - Fractional Belief propagation, Bound Propagation, $<$your favorite name$>$ Propagation
  - Graph cuts ...
- Stochastic
  - Unscented Particle Filter, Nonparametric Belief Propagation
  - Annealed Importance Sampling, Adaptive Importance Sampling
  - Hybrid Monte Carlo, Exact sampling, Coupling from the past

Variational or Sampling?

- Possible criteria
  - **How accurate**
  - **How fast**
  - **How easy to learn**
  - **How easy to code/test/maintain**

When all you own is a hammer, every problem looks like a nail

Variational or Sampling?

- Depends upon application domain. My personal impression is:
  - Sampling dominated
    - Bayesian statistics, Scientific data analysis
    - Finance/auditing
    - Operations research
    - Genetics
    - Tracking
  - Variational dominated
    - Communications/error correcting codes
  - Mixed territory
    - Machine Learning, Robotics
    - Computer Vision
    - Human-Computer Interaction
    - Speech/audio/multimedia analysis/information retrieval
    - Statistical Signal processing

Further Reading

Variational tutorials and overviews

- Tommi Jaakkola. *Tutorial on variational approximation methods.* (2000). `http://people.csail.mit.edu/tommi/papers/Jaa-var-tutorial.ps`
- Frey and Jojic [2]
- Wainwright and Jordan [7]

MCMC and SMC tutorials and overviews

- Andrieu, de Freitas, Doucet, Jordan. *An Introduction to MCMC for Machine Learning*, 2001
- Andrieu. *Monte Carlo Methods for Absolute beginners*, 2004
- Doucet, Godsill, Andrieu. ”On Sequential Monte Carlo Sampling Methods for Bayesian Filtering”, Statistics and Computing, vol. 10, no. 3, pp. 197-208, 2000

The “in Practice” Books

- Gilks, Richardson, Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman Hall, 1996
- Doucet, de Freitas, Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001

References

[1] J.E. Besag. On the statistical analysis of dirty pictures (with discussion). Jr. R. Stat. Soc. B, 48:259–302, 1986.

[2] B. J. Frey and N. Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(9), 2005.

[3] Z. Ghahramani and M. Beal. Propagation algorithms for variational Bayesian learning. In Neural Information Processing Systems 13, 2000.

[4] E. T. Jaynes. Probability Theory, The Logic of Science. Cambridge University Press, edited by G. L. Bretthorst, 2003.

[5] D. J. C. MacKay. Information Theory, Inference and Learning Algorithms. Cambridge University Press, 2003.

[6] Radford M. Neal and Geoffrey E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Learning in graphical models, pages 355–368. MIT Press, 1999.

[7] M. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Technical Report 649, Department of Statistics, UC Berkeley, September 2003.