Introduction to Numerical Bayesian Methods

A. Taylan Cemgil

Signal Processing and Communications Lab.

UNIVERSITY OF CAMBRIDGE
Department of Engineering

Boğaziçi Üniversitesi, Bilgisayar Mühendisliği Bölümü
7-8 Eylül 2006, İstanbul, Türkiye
Thanks to

- Nick Whiteley
- Simon Godsill
- Bill Fitzgerald

(Bu slaytlar muhtemelen değişebilir, en son versyon için aşağıdaki link’e bakın)

Outline, Day 1

- Introduction, Bayes’ Theorem,
- Probability models, Bayesian Networks and Factor graphs
- Applications
- Deterministic Inference Techniques
  - Variational Methods: Variational Bayes, EM, ICM
- Stochastic (Sampling Based) Methods
  - Markov Chain Monte Carlo (MCMC)
    * Gibbs Sampler
    * Simulated Annealing
    * Iterative Improvement
Outline, Day 2

• Time Series models
  – Hidden Markov Models, Kalman Filter Models
  – Switching State Space models, Changepoint models
  – Nonlinear Dynamical Systems

• Applications

• Exact inference in time series models
  – Filtering
  – Smoothing

• Online Approximate Inference
  – Importance Sampling
  – Sequential Monte Carlo, Particle Filtering

• Yetsin artık bu kadar
Bayes’ Theorem [4, 6]

Thomas Bayes (1702-1761)

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

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

Posterior = Likelihood \( \times \) Prior \( \frac{\text{Evidence}}{\text{Evidence}} \)
Bayes’ Theorem

• This rather simple looking formula has surprisingly many applications

\[ p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \]

– Medical Diagnosis (Symptoms/Diseases)
– Computer Vision (Pixels/Object)
– Speech Recognition (Signal/Phoneme)
– Music Transcription (Audio/Score)
– Robotics/Navigation (Sensor Reading/Position)
– Finance (Past Price/Future Price)
– . . .
An application of Bayes’ Theorem: “Parameter Estimation”

Given two fair dice with outcomes $\lambda$ and $y$, 

$$D = \lambda + y$$

What is $\lambda$ when $D = 9$?
An application of Bayes’ Theorem: “Parameter Estimation”

\[ D = \lambda + y = 9 \]

<table>
<thead>
<tr>
<th>(D = \lambda + y)</th>
<th>(y = 1)</th>
<th>(y = 2)</th>
<th>(y = 3)</th>
<th>(y = 4)</th>
<th>(y = 5)</th>
<th>(y = 6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\lambda = 1)</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>(\lambda = 2)</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>(\lambda = 3)</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>(\lambda = 4)</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>(\lambda = 5)</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>(\lambda = 6)</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>

Bayes theorem “upgrades” \(p(\lambda)\) into \(p(\lambda|D)\).

But you have to provide an observation model: \(p(D|\lambda)\)
An application of Bayes’ Theorem: “Parameter Estimation”

Formally we write

\[ p(\lambda) = C(\lambda; [ 1/6 \ 1/6 \ 1/6 \ 1/6 \ 1/6 \ 1/6 ]) \]

\[ p(y) = C(y; [ 1/6 \ 1/6 \ 1/6 \ 1/6 \ 1/6 \ 1/6 ]) \]

\[ p(\mathcal{D}|\lambda, y) = \delta(\mathcal{D} - (\lambda + y)) \]

\[
p(\lambda|\mathcal{D}) = \frac{1}{p(\mathcal{D})} \times p(\mathcal{D}|\lambda, y) \times p(y)p(\lambda)
\]

\[ = \frac{1}{\text{Evidence}} \times \text{Likelihood} \times \text{Prior} \]

Kronecker delta function denoting a degenerate (deterministic) distribution \( \delta(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0 \end{cases} \)
Prior

\[ p(y)p(\lambda) \]

<table>
<thead>
<tr>
<th>( y = 1 )</th>
<th>( y = 2 )</th>
<th>( y = 3 )</th>
<th>( y = 4 )</th>
<th>( y = 5 )</th>
<th>( y = 6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda = 1 )</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
</tr>
<tr>
<td>( \lambda = 2 )</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
</tr>
<tr>
<td>( \lambda = 3 )</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
</tr>
<tr>
<td>( \lambda = 4 )</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
</tr>
<tr>
<td>( \lambda = 5 )</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
</tr>
<tr>
<td>( \lambda = 6 )</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
<td>1/36</td>
</tr>
</tbody>
</table>
Likelihood

\[ p(D = 9|\lambda, y) \]

| \( p(D = 9|\lambda, y) \) | \( y = 1 \) | \( y = 2 \) | \( y = 3 \) | \( y = 4 \) | \( y = 5 \) | \( y = 6 \) |
|-----------------|---------|---------|---------|---------|---------|---------|
| \( \lambda = 1 \) | 0       | 0       | 0       | 0       | 0       | 0       |
| \( \lambda = 2 \) | 0       | 0       | 0       | 0       | 0       | 0       |
| \( \lambda = 3 \) | 0       | 0       | 0       | 0       | 0       | 1       |
| \( \lambda = 4 \) | 0       | 0       | 0       | 0       | 1       | 0       |
| \( \lambda = 5 \) | 0       | 0       | 1       | 0       | 0       | 0       |
| \( \lambda = 6 \) | 0       | 0       | 1       | 0       | 0       | 0       |
\[ \phi_D(\lambda, y) = p(D = 9|\lambda, y)p(\lambda)p(y) \]

| \( p(D = 9|\lambda, y) \) | \( y = 1 \) | \( y = 2 \) | \( y = 3 \) | \( y = 4 \) | \( y = 5 \) | \( y = 6 \) |
|--------------------------|------------|------------|------------|------------|------------|------------|
| \( \lambda = 1 \)      | 0          | 0          | 0          | 0          | 0          | 0          |
| \( \lambda = 2 \)      | 0          | 0          | 0          | 0          | 0          | 0          |
| \( \lambda = 3 \)      | 0          | 0          | 0          | 0          | 0          | \( 1/36 \) |
| \( \lambda = 4 \)      | 0          | 0          | 0          | 0          | 1/36       | 0          |
| \( \lambda = 5 \)      | 0          | 0          | 1/36       | 0          | 0          | 0          |
| \( \lambda = 6 \)      | 0          | 1/36       | 0          | 0          | 0          | 0          |
### Posterior

\[
p(\lambda, y|D = 9) = \frac{1}{p(D)} p(D = 9|\lambda, y)p(\lambda)p(y)
\]

| $p(D = 9|\lambda, y)$ | $y = 1$ | $y = 2$ | $y = 3$ | $y = 4$ | $y = 5$ | $y = 6$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda = 1$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\lambda = 2$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\lambda = 3$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$1/4$</td>
</tr>
<tr>
<td>$\lambda = 4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$1/4$</td>
<td>0</td>
</tr>
<tr>
<td>$\lambda = 5$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$1/4$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\lambda = 6$</td>
<td>0</td>
<td>0</td>
<td>$1/4$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
p(D = 9) = \sum_{\lambda, y} p(D = 9|\lambda, y)p(\lambda)p(y) = 0 + 0 + \ldots + 1/36 + 1/36 + 1/36 + 1/36 + 0 + \ldots + 0 = 1/9
\]

\[
1/4 = (1/36)/(1/9)
\]
Marginal Posterior

\[ p(\lambda|D) = \sum_y \frac{1}{p(D)} p(D|\lambda, y) p(\lambda) p(y) \]

|       | \( p(\lambda|D = 9) \) | \( y = 1 \) | \( y = 2 \) | \( y = 3 \) | \( y = 4 \) | \( y = 5 \) | \( y = 6 \) |
|-------|------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| \( \lambda = 1 \) | 0                      | 0           | 0           | 0           | 0           | 0           | 0           |
| \( \lambda = 2 \) | 0                      | 0           | 0           | 0           | 0           | 0           | 0           |
| \( \lambda = 3 \) | 1/4                    | 0           | 0           | 0           | 0           | 0           | 1/4         |
| \( \lambda = 4 \) | 1/4                    | 0           | 0           | 0           | 0           | 1/4         | 0           |
| \( \lambda = 5 \) | 1/4                    | 0           | 0           | 0           | 1/4         | 0           | 0           |
| \( \lambda = 6 \) | 1/4                    | 0           | 0           | 1/4         | 0           | 0           | 0           |
The “proportional to” notation

\[ p(\lambda|\mathcal{D}) \propto \sum_y p(\mathcal{D}|\lambda, y)p(\lambda)p(y) \]

| \( \lambda \) | \( p(\lambda|\mathcal{D} = 9) \) | \( y = 1 \) | \( y = 2 \) | \( y = 3 \) | \( y = 4 \) | \( y = 5 \) | \( y = 6 \) |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 1/36 | 0 | 0 | 0 | 0 | 0 | 1/36 |
| 4 | 1/36 | 0 | 0 | 0 | 0 | 1/36 | 0 |
| 5 | 1/36 | 0 | 0 | 0 | 1/36 | 0 | 0 |
| 6 | 1/36 | 0 | 0 | 1/36 | 0 | 0 | 0 |
Another application of Bayes’ Theorem: “Model Selection”

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

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

How many dice are there when $D = 9$?

Assume that any number $n$ is equally likely
Another application of Bayes’ Theorem: “Model Selection”

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

\[
p(n|D = 9) = \frac{p(D = 9|n)p(n)}{p(D)} \propto p(D = 9|n)
\]

\[
p(D|n = 1) = \sum_{\lambda_1} p(D|\lambda_1)p(\lambda_1)
\]

\[
p(D|n = 2) = \sum_{\lambda_1} \sum_{\lambda_2} p(D|\lambda_1, \lambda_2)p(\lambda_1)p(\lambda_2)
\]

\[\vdots\]

\[
p(D|n = n') = \sum_{\lambda_1, \ldots, \lambda_{n'}} p(D|\lambda_1, \ldots, \lambda_{n'}) \prod_{i=1}^{n'} p(\lambda_i)
\]
\[ p(D|n) = \sum_{\lambda} p(D|\lambda, n)p(\lambda|n) \]
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 $\lambda$. 
Tutorial’ımız Bitmiştir, İlginize teşekkürler
Probability Models

+ 

Inference Algorithms

= 

Bayesian Numerical Methods
Formal Languages for specification of Probability Models and Inference Algorithms

- Directed Graphical Models, Directed Acyclic Graphs (DAG), Bayesian Networks
- Undirected Graphs, Markov Networks, Random Fields
- Factor Graphs
Conditional Independence
Discrete conditional probability tables

- Assume all $x_i$ are discrete with $|x_i| = k$. If $N$ is large, a naive table representation is HUGE: $k^N$ entries

Example: $p(x_1, x_2, x_3)$ with $|x_i| = 4$

Each cell is a positive number s.t. $\sum_{x_1, x_2, x_3} p(x_1, x_2, x_3) = 1$
Independence Assumption == Complete Factorization

- Assume $p(x_1, x_2, \ldots, x_N) = \prod_k p(x_k)$. 

We need to store $4 \times 3$ numbers instead of $4^3$!

- However, complete independence may be too restrictive.
An alternative Factorization

\[ p(x_1, x_2) \times p(x_3) = p(x_1, x_2, x_3) \]

- We need to store \(4^2 + 4\) numbers instead of \(4^3\).

- Still some variables are independent from rest. It is possible to introduce conditional independence relations to design “richer” distributions.
Conditional Independence

- Two sets of variables $A$ and $B$ are conditionally independent given a third set $C$ if

$$p(A, B|C) = p(A|C)p(B|C)$$

- This is equivalent to

$$p(A|BC) = p(A|C)$$
Exercise

<table>
<thead>
<tr>
<th></th>
<th>$x_2 = 1$</th>
<th>$x_2 = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 = 1$</td>
<td>0.6</td>
<td>0.1</td>
</tr>
<tr>
<td>$x_1 = 2$</td>
<td>0.2</td>
<td>0.1</td>
</tr>
</tbody>
</table>

1. Find the following quantities
   - Marginals: $p(x_1), p(x_2)$
   - Conditionals: $p(x_1|x_2), p(x_2|x_1)$
   - Posterior: $p(x_1, x_2 = 2), p(x_1|x_2 = 2)$
   - Evidence $p(x_2 = 2)$
   - $p(\emptyset)$

2. Are $x_1$ and $x_2$ independent? If not, construct a new probability table where $x_1$ and $x_2$ are independent but still have the same marginals.

3. Construct a new probability table $p(x_1, x_3)$ such that $p(x_2, x_3|x_1) = p(x_2|x_1)p(x_3|x_1)$ but $p(x_3) = [0.5, 0.5]$. Do you have any freedom in choosing the new table?
DAG Example: Two dice

Given two fair dice with outcomes \( \lambda \) and \( y \) where \( D = \lambda + y \),

\[
p(D|\lambda, y) = p(D|\lambda, y)p(\lambda)p(y)
\]
Given two fair dice with outcomes $\lambda$ and $y$ when $D = \lambda + y = 9$

\[
p(\lambda) \quad p(y)
\]

\[
\lambda \quad y
\]

\[
[D]
\]

\[
p(D = 9|\lambda, y)
\]

\[
\phi_D(\lambda, y) = p(D = 9|\lambda, y)p(\lambda)p(y)
\]
Directed Graphical models

- Each random variable is associated with a node in the graph,

- We draw an arrow from $x_j \rightarrow x_i$ each parent node $x_j \in \text{parent}(x_i)$,

$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i | \text{parent}(x_i))$$

- Every joint probability distribution over finite number of variables can be written in this form, but this is not necessarily the minimal representation,

- Describes in a compact way how data is “generated”,

- Technically, missing links denote conditional independence relations between variables. This turns out to be very important in developing efficient inference algorithms.
Graphical Models

- Graphical models represent joint distributions compactly using a set of local variables
- Each variable corresponds to a node in the graph
- The edges tell us qualitatively about the factorization of the joint probability
- There are functions at the nodes that tell us the quantitative details of the factors
Directed Graphical Models

- Consider directed acyclic graphs over N variables
- Each node $x_i$ has a (possibly empty) set of parents denoted by $pa(x_i)$
- Each node has a function $p(x_i|pa(x_i))$.
- The joint probability is given by
  \[
p(x_1, x_2, \ldots, x_N) = \prod_i p(x_i|pa(x_i))
  \]
- Factorization in terms of local functions
Examples

\[ p(x_1) \times p(x_2|x_1) \times p(x_3|x_1) = p(x_1, x_2, x_3) \]
### Examples

<table>
<thead>
<tr>
<th>Model</th>
<th>Structure</th>
<th>factorization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td><img src="image" alt="Full Model Diagram" /></td>
<td>$p(x_1)p(x_2</td>
</tr>
<tr>
<td>Markov(2)</td>
<td><img src="image" alt="Markov(2) Model Diagram" /></td>
<td>$p(x_1)p(x_2</td>
</tr>
<tr>
<td>Markov(1)</td>
<td><img src="image" alt="Markov(1) Model Diagram" /></td>
<td>$p(x_1)p(x_2</td>
</tr>
<tr>
<td>Factorized</td>
<td><img src="image" alt="Factorized Model Diagram" /></td>
<td>$p(x_1)p(x_2</td>
</tr>
</tbody>
</table>

Removing edges eliminates a term from the conditional probability factors.
Examples

Dataset (From Sayood): All four letter English words (2149) of a Sun-Sparc spell checker.
(abbe, abed, abel, abet, able, ... zion, zone, zoom, zorn)

<table>
<thead>
<tr>
<th>Model</th>
<th>Structure</th>
<th>Random Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td><img src="diagram.png" alt="Diagram" /></td>
<td>tabu, else, duly, crib, bohr, seal, tome, free, bern</td>
</tr>
<tr>
<td>Markov(1)</td>
<td><img src="diagram.png" alt="Diagram" /></td>
<td>miro, jaid, saun, trol, bale, liro, pibo, brox, heth</td>
</tr>
<tr>
<td>Factorized</td>
<td><img src="diagram.png" alt="Diagram" /></td>
<td>yiij, ekmy, vguo, addn, ecmi, miui, bhin, hnri, roia, azfa</td>
</tr>
</tbody>
</table>
Estimated model $p(x_k | x_{k-1})$ for the four letter words dataset
Factor graphs
Factor graph for two dice example [5]

\[ p(\lambda) \quad \cdots \quad p(y) \]

\[ \lambda \quad \cdots \quad y \]

\[ p(\mathcal{D} = 9|\lambda, y) \]

\[ \phi_\mathcal{D}(\lambda, y) = p(\mathcal{D} = 9|\lambda, y)p(\lambda)p(y) = \phi_1(\lambda, y)\phi_2(\lambda)\phi_3(y) \]

- A bipartite graph. A powerful graphical representation of the inference problem
  - Factor nodes: Black squares. Factor potentials (local functions) defining the posterior.
  - Variable nodes: White Circles.
  - Edges: denote membership. A variable is connected to a factor if it is an argument of the local function.
Exercise

- For the following Graphical models, write down the factors of the joint distribution and plot the corresponding factor graphs.

**Full**

\[ x_1 ightarrow x_2 ightarrow x_3 ightarrow x_4 \]

**Markov(1)**

\[ x_1 ightarrow x_2 ightarrow x_3 ightarrow x_4 \]

**HMM**

\[ h_1 \rightarrow x_1 \quad h_2 \rightarrow x_2 \quad h_3 \rightarrow x_3 \quad h_4 \rightarrow x_4 \]

**MIX**

\[ h \rightarrow x_1 \quad h \rightarrow x_2 \quad h \rightarrow x_3 \quad h \rightarrow x_4 \]

**IFA**

\[ h_1 \rightarrow h_2 \rightarrow x_1 \quad h_1 \rightarrow h_2 \rightarrow x_2 \quad h_1 \rightarrow h_2 \rightarrow x_3 \quad h_1 \rightarrow h_2 \rightarrow x_4 \]

**Factorized**

\[ x_1 \quad x_2 \quad x_3 \quad x_4 \]
Example: AR(1) model

\[ x_k = Ax_{k-1} + \epsilon_k \quad k = 1 \ldots K \]

\( \epsilon_k \) 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

\[ A \sim \mathcal{N}(A; 0, P) \]
\[ R \sim \mathcal{IG}(R; \nu, \beta/\nu) \]
\[ x_k | x_{k-1}, A, R \sim \mathcal{N}(x_k; Ax_{k-1}, R) \]
\[ x_0 = \hat{x}_0 \]

Gaussian: \( \mathcal{N}(x; \mu, V) \equiv |2\pi V|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x - \mu)^2/V\right) \)
Inverse-Gamma distribution: \( \mathcal{IG}(x; a, b) \equiv \Gamma(a)^{-1}b^{-a}x^{-(a+1)} \exp\left(-1/(bx)\right) \quad x \geq 0 \)

Observed variables are shown with double circles
AR(1) Model. Bayesian Posterior Inference

\[ p(A, R|x_0, x_1, \ldots, x_K) \propto p(x_1, \ldots, x_K|x_0, A, R)p(A, R) \]

Posterior \propto Likelihood \times Prior

Using the Markovian (conditional independence) structure we have

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

Suppose $K = 1$,

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

$$p(A, R | x_0, x_1) \propto p(x_1 | x_0, A, R)p(A)p(R)$$
$$= \mathcal{N}(x_1; Ax_0, R)\mathcal{N}(A; 0, P)\mathcal{IG}(R; \nu, \beta/\nu)$$
Numerical Example, the prior $p(A, R)$

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

\[ A \sim \mathcal{N}(A; 0, 1.2) \quad R \sim \mathcal{IG}(R; 0.4, 250) \]

Suppose: \( x_0 = 1 \quad x_1 = -6 \quad x_1 \sim \mathcal{N}(x_1; Ax_0, R) \)
Numerical Example, the posterior $p(A, R|x)$

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

- $A \approx -6 \iff$ low noise variance $R$.
- $A \approx 0 \iff$ 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**

\[ \langle f(x) \rangle = \int_{\mathcal{X}} dx p(x) f(x) \]

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

\[ x^* = \arg\max_{x \in \mathcal{X}} p(x) f(x) \]

- any “mix” of the above: e.g.,

\[ x^* = \arg\max_{x \in \mathcal{X}} p(x) = \arg\max_{x \in \mathcal{X}} \int_{\mathcal{Z}} dz p(z) p(x|z) \]
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",

Attributes of Probabilistic Inference

- **Exact** ↔ **Approximate**
- **Deterministic** ↔ **Stochastic**
- **Online** ↔ **Offline**
- **Centralized** ↔ **Distributed**

This talk focuses on the bold ones
Deterministic Inference

Mean Field – Variational Bayes
Toy Model: “One sample source separation (OSSS)”

This graph encodes the joint:

\[ p(x, s_1, s_2) = p(x|s_1, s_2)p(s_1)p(s_2) \]

\[ s_1 \sim p(s_1) = \mathcal{N}(s_1; \mu_1, P_1) \]
\[ s_2 \sim p(s_2) = \mathcal{N}(s_2; \mu_2, P_2) \]
\[ x|s_1, s_2 \sim p(x|s_1, s_2) = \mathcal{N}(x; s_1 + s_2, R) \]
The Gaussian Distribution

$\mu$ is the mean and $P$ is the covariance:

$$
\mathcal{N}(s; \mu, P) = |2\pi P|^{-1/2} \exp \left( -\frac{1}{2} (s - \mu)^T P^{-1} (s - \mu) \right)
$$

$$
= \exp \left( -\frac{1}{2} s^T P^{-1} s + \mu^T P^{-1} s - \frac{1}{2} \mu^T P^{-1} \mu - \frac{1}{2} |2\pi P| \right)
$$

$$
\log \mathcal{N}(s; \mu, P) = -\frac{1}{2} s^T P^{-1} s + \mu^T P^{-1} s + \text{const}
$$

$$
= -\frac{1}{2} \text{Tr} P^{-1} ss^T + \mu^T P^{-1} s + \text{const}
$$

$$
=+ \quad -\frac{1}{2} \text{Tr} P^{-1} ss^T + \mu^T P^{-1} s
$$

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 \( x = \hat{x} \).

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.

\[
\log p(s_1) = \mu_1^T P_1^{-1} s_1 - \frac{1}{2} s_1^T P_1^{-1} s_1 + \text{const}
\]

\[
\log p(s_2) = \mu_2^T P_2^{-1} s_2 - \frac{1}{2} s_2^T P_2^{-1} s_2 + \text{const}
\]

\[
\log p(x|s_1, s_2) = \hat{x}^T R^{-1} (s_1 + s_2) - \frac{1}{2} (s_1 + s_2)^T R^{-1} (s_1 + s_2) + \text{const}
\]

\[
\log \phi(s_1, s_2) = \log p(x = \hat{x}|s_1, s_2) + \log p(s_1) + \log p(s_2) \\
=+ \left( \mu_1^T P_1^{-1} + \hat{x}^T R^{-1} \right) s_1 + \left( \mu_2^T P_2^{-1} + \hat{x}^T R^{-1} \right) s_2 \\
-\frac{1}{2} \text{Tr} \left( P_1^{-1} + R^{-1} \right) s_1 s_1^T - s_1^T R^{-1} s_2 - \frac{1}{2} \text{Tr} \left( P_2^{-1} + R^{-1} \right) s_2 s_2^T
\]

- The (*) term is the cross correlation term that makes \(s_1\) and \(s_2\) a-posteriori dependent.
OSSS example, cont.

Completing the square

\[
\log \phi(s_1, s_2) = + \begin{pmatrix} P_1^{-1} \mu_1 + R^{-1} \hat{x} \\ P_2^{-1} \mu_2 + R^{-1} \hat{x} \end{pmatrix}^\top \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} - \frac{1}{2} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}^\top \begin{pmatrix} P_1^{-1} + R^{-1} & R^{-1} \\ R^{-1} & P_2^{-1} + R^{-1} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}
\]

Remember:

\[
\log \mathcal{N}(s; m, \Sigma) = + (\Sigma^{-1} m)^\top s - \frac{1}{2} s^\top \Sigma^{-1} s
\]

\[
\Sigma = \begin{pmatrix} P_1^{-1} + R^{-1} & R^{-1} \\ R^{-1} & P_2^{-1} + R^{-1} \end{pmatrix}^{-1} \quad m = \Sigma \begin{pmatrix} P_1^{-1} \mu_1 + R^{-1} \hat{x} \\ P_2^{-1} \mu_2 + R^{-1} \hat{x} \end{pmatrix}
\]
Variational Bayes (VB), mean field

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

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

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

Here, we choose

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

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 $Q = q(x)$.

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

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

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

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

$$KL(\mathcal{P}||Q) = -\int_X d x p(x) \log \frac{q(x)}{p(x)}$$

$$\geq -\log \int_X d x p(x) \frac{q(x)}{p(x)} = -\log \int_X d x q(x) = -\log 1 = 0$$
OSSS example, cont.

Let the approximating distribution be factorized as

$$Q = q(s_1)q(s_2)$$

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

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

$$KL(Q || P) = \langle \log Q \rangle_Q - \langle \log \frac{1}{Z_x} \phi(s_1, s_2) \rangle_Q$$

(1)
The form of the mean field solution

\[ 0 \leq \langle \log q(s_1)q(s_2) \rangle_{q(s_1)q(s_2)} + \log Z_x - \langle \log \phi(s_1, s_2) \rangle_{q(s_1)q(s_2)} \]
\[ \log Z_x \geq \langle \log \phi(s_1, s_2) \rangle_{q(s_1)q(s_2)} - \langle \log q(s_1)q(s_2) \rangle_{q(s_1)q(s_2)} \]
\[ \equiv -F(p; q) + H(q) \] (2)

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

Evidence \( \geq \) –Energy + Entropy

Note r.h.s. is a lower bound [7]. The mean field equations monotonically increase this bound. Good for assessing convergence and debugging computer code.
Details of derivation

• Define the Lagrangian

\[ \Lambda = \int ds_1 q(s_1) \log q(s_1) + \int ds_2 q(s_2) \log q(s_2) + \log Z_x - \int ds_1 ds_2 q(s_1)q(s_2) \log \phi(s_1, s_2) \]

\[ + \lambda_1 (1 - \int ds_1 q(s_1)) + \lambda_2 (1 - \int ds_2 q(s_2)) \]

(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 \( q(s_1) \),

\[ \log q(s_1) = \lambda_1 - 1 + \langle \log \phi(s_1, s_2) \rangle q(s_2) \]

\[ q(s_1) = \exp(\lambda_1 - 1) \exp(\langle \log \phi(s_1, s_2) \rangle q(s_2)) \]

(4)

• Use the fact that

\[ 1 = \int ds_1 q(s_1) = \exp(\lambda_1 - 1) \int ds_1 \exp(\langle \log \phi(s_1, s_2) \rangle q(s_2)) \]

\[ \lambda_1 = 1 - \log \int ds_1 \exp(\langle \log \phi(s_1, s_2) \rangle q(s_2)) \]
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
OSSS: Factor Graph

- A graphical representation of the inference problem
  - **Factor nodes**: Black squares. Factor potentials (local functions) defining the posterior $\mathcal{P}$.
  - **Variable nodes**: Circles. Think of them as “factors” of the approximating distribution $\mathcal{Q}$. (Caution – non standard interpretation!)
  - **Edges**: denote membership. A variable is connected to a factor if it is a variable of the local function.
Fixed Point Iteration for OSSS

\[ p(s_1) \quad p(s_2) \]

\[ q(s_1) \quad s_1 \quad s_2 \quad q(s_2) \]

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

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

\[
\log q(s_2) \leftarrow \log p(s_2) + \langle \log p(x = \hat{x} | s_1, s_2) \rangle_{q(s_1)}
\]
Fixed Point Iteration for the Gaussian Case

\[
\log q(s_1) \leftarrow -\frac{1}{2} \text{Tr} \left( P_1^{-1} + R^{-1} \right) s_1 s_1^\top - s_1^\top R^{-1} \langle s_2 \rangle_{q(s_2)} + \left( \mu_1^\top P_1^{-1} + \hat{x}^\top R^{-1} \right) s_1 = m_2
\]

\[
\log q(s_2) \leftarrow -\langle s_1 \rangle_{q(s_1)}^\top R^{-1} s_2 - \frac{1}{2} \text{Tr} \left( P_2^{-1} + R^{-1} \right) s_2 s_2^\top + \left( \mu_2^\top P_2^{-1} + \hat{x}^\top R^{-1} \right) s_2 = m_1^\top
\]

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

\[
\log q(s) = + \quad -\frac{1}{2} \text{Tr} \ K ss^\top + h^\top s
\]

\[
\downarrow
\]

\[
S = K^{-1} \quad m = K^{-1} h
\]
Fixed Point Equations for the Gaussian Case

- Covariances are obtained directly

\[
S_1 = \left( P_1^{-1} + R^{-1} \right)^{-1} \quad S_2 = \left( P_2^{-1} + R^{-1} \right)^{-1}
\]

- To compute the means, we should iterate:

\[
m_1 = S_1 \left( P_1^{-1} \mu_1 + R^{-1} (\hat{x} - m_2) \right) \quad m_2 = S_2 \left( P_2^{-1} \mu_2 + R^{-1} (\hat{x} - m_1) \right)
\]

- Intuitive algorithm:
  - Subtract from the observation \( \hat{x} \) the prediction of the other factors of \( Q \).
  - 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) [3]

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

\[ \tilde{q}(s_2) = \delta(s_2 - \tilde{m}) \]

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

\[ \tilde{m} = \arg\min_{\xi} KL(\delta(s_2 - \xi)||q(s_2)) \]

It can be shown that this leads exactly to the EM fixed point iterations.
Iterated Conditional Modes (ICM) [1, 2]

If we choose both distributions degenerate, i.e.

\[
\tilde{q}(s_1) = \delta(s_1 - \tilde{m}_1) \\
\tilde{q}(s_2) = \delta(s_2 - \tilde{m}_2)
\]

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 \( \phi(s_1, s_2) \).

\[
\tilde{m}_1 = \arg\max_{s_1} \phi(s_1, s_2 = \tilde{m}_2) \\
\tilde{m}_2 = \arg\max_{s_2} \phi(s_1 = \tilde{m}_1, s_2)
\]
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.
Structured Mean Field
Main Idea

• Identify tractable substructures to construct richer approximating distributions

• Tradeoff between approximation quality and computation time

The OSSS model is too simple; a richer approximation $Q(s_1, s_2)$ would be equivalent to the exact posterior.
Bayesian Variable Selection

\[ C(r_1; \pi) \quad C(r_W; \pi) \]

\[ r_1 \quad \ldots \quad r_W \]

\[ \mathcal{N}(s_1; \mu(r_1), \Sigma(r_1)) \quad \ldots \quad \mathcal{N}(s_W; \mu(r_W), \Sigma(r_W)) \]

\[ x \]

\[ \mathcal{N}(x; Cs_{1:W}, R) \]

- Generalized Linear Model – Column’s of \( C \) are the basis vectors
- The exact posterior is a mixture of \( 2^W \) Gaussians
- When \( W \) is large, computation of posterior features becomes intractable.
Generative model

\[ r_i \sim \mathcal{C}(r_i; \pi) \]
\[ s_i | r_i \sim \mathcal{N}(s_i; \mu(r_i), \Sigma(r_i)) \]
\[ x | s_{1:W} \sim \mathcal{N}(x; Cs_{1:W}, R) \]
\[ C \equiv \begin{bmatrix} C_1 & \ldots & C_i & \ldots & C_W \end{bmatrix} \]

\[ p(x, s_{1:W}, r_{1:W}) = p(x | s_{1:W}, r_{1:W}) \prod_{i=1}^{W} p(s_i | r_i)p(r_i) \]
Example 1: Variable selection in Polynomial Regression

Given \( \{ t_j, x(t_j) \}_{j=1}^{J} \), what is the order \( N \) of the polynomial?

\[
x(t) = \sum_{i=0}^{N} s_{i+1} t^i + \epsilon(t)
\]
Ex1: Regression

\[ t = (t_1 \ t_2 \ \ldots \ t_J)^\top \]

\[ C \equiv (t^0 \ t^1 \ \ldots \ t^{W-1}) \]

\[ r_i \sim \mathcal{C}(r_i; 0.5, 0.5) \quad \text{for } r_i \in \{\text{on, off}\} \]

\[ s_i|r_i \sim \mathcal{N}(s_i; 0, \Sigma(r_i)) \]

\[ x|s_{1:W} \sim \mathcal{N}(x; Cs_{1:W}, R) \]

\[ \Sigma(r_i = \text{on}) \gg \Sigma(r_i = \text{off}) \]
Ex1: Regression

To find the “active” basis functions we need to calculate

\[
\begin{align*}
    r_{1:W}^* &\equiv \arg\max_{r_{1:W}} p(r_{1:W}|x) = \arg\max_{r_{1:W}} \int ds_{1:W} p(x|s_{1:W}) p(s_{1:W}|r_{1:W}) p(r_{1:W}) \\
    \end{align*}
\]

Then, the reconstruction is given by

\[
\begin{align*}
    \hat{x}(t) &= \left< \sum_{i=0}^{W-1} s_{i+1} t^i \right> p(s_{1:W}|x, r_{1:W}^*) \\
    &= \sum_{i=0}^{W-1} \left< s_{i+1} \right> p(s_{i+1}|x, r_{1:W}^*) t^i \\
\end{align*}
\]
Ex1: Regression

\[ p(x, f_1; W) \]

[Graph showing data points and configurations]

All on Configurations All off
Ex1: Regression
\[
\log \phi(r_{1:W}, s_{1:W}) = \sum_{i=1}^{W} (\log \pi(r_i)) \\
+ \sum_{i=1}^{W} \left( -\frac{1}{2} s_i^\top \Sigma(r_i)^{-1} s_i + \mu(r_i)^\top \Sigma(r_i)^{-1} s_i \\
- \frac{1}{2} \mu(r_i)^\top \Sigma(r_i)^{-1} \mu(r_i) - \frac{1}{2} \log |2\pi \Sigma(r_i)| \right) \\
- \frac{1}{2} x^\top R^{-1} x + s_{1:W} C^\top R^{-1} x - \frac{1}{2} s_{1:W} C^\top R^{-1} C s_{1:W} - \frac{1}{2} \log |2\pi R|
\]
Approximating Distributions

\[ Q_1 = \prod_{i=1}^{W} Q(s_i) Q(r_i) \quad Q_2 = Q(s_{1:W}) \prod_{i=1}^{W} Q(r_i) \quad Q_3 = \prod_{i=1}^{W} Q(s_i, r_i) \]
Update Equations, \( Q_1 = \prod_{i=1}^{W} Q(s_i) Q(r_i) \)

\[
\log Q(r_i) = + \log \pi(r_i) - \frac{1}{2} (\langle s_i \rangle - \mu(r_i))^\top \Sigma(r_i)^{-1} (\langle s_i \rangle - \mu(r_i))
\]

\[
\log Q(s_i) = + \left( \langle \Sigma(r_i)^{-1} \mu(r_i) \rangle + C_i^\top R^{-1} (x - C_{-i} \langle s_{-i} \rangle) \right)^\top s_i - \frac{1}{2} s_i^\top \left( \langle \Sigma(r_i)^{-1} \rangle + C_i^\top R^{-1} C_i \right) s_i
\]

\[
C_{-i} \equiv \left( 
\begin{array}{cccccc}
C_1 & \ldots & C_{i-1} & C_{i+1} & \ldots & C_W
\end{array}
\right)
\]

\[
s_{-i} \equiv \left( 
\begin{array}{cccccc}
s_1^\top & \ldots & s_{i-1}^\top & s_{i+1}^\top & \ldots & s_W^\top
\end{array}
\right)^\top
\]
**Update Equations:** \( Q_2 = Q(s_{1:W}) \prod_{i=1}^{W} Q(r_i) \)

\[
\log Q(r_i) =^+ \log \pi(r_i) - \frac{1}{2} (\langle s_i \rangle - \mu(r_i))^\top \Sigma(r_i)^{-1} (\langle s_i \rangle - \mu(r_i))
\]

\[
\log Q(s_{1:W}) =^+ \left( \langle \Sigma(r)^{-1} \mu(r) \rangle + C^\top R^{-1} x \right)^\top s_{1:W} - \frac{1}{2} s_{1:W}^\top \left( \langle \Sigma(r)^{-1} \rangle + C^\top R^{-1} C \right) s_{1:W}
\]

\[
\Sigma(r)^{-1} \equiv \begin{pmatrix} 
\Sigma(r_1)^{-1} \\
\vdots \\
\Sigma(r_W)^{-1}
\end{pmatrix} \\
\mu(r) \equiv \begin{pmatrix} 
\mu(r_1) \\
\vdots \\
\mu(r_W)
\end{pmatrix}
\]
**Update Equations:** \[ Q_3 = \prod_{i=1}^{W} Q(r_i, s_i) \]

Left as an exercise to the interested reader...
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 $m_{\tau_1}$

• 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

\[ \mathcal{P}_{\tau_i} \propto \mathcal{P}^{\tau_i} \]
OSSS example: Annealing, Bridging, ...

• Remember the cross term (\(\star\)) of the posterior:

\[
\cdots - s_1^\top R^{-1} s_2 \cdots
\]

\(\star\)

• 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 $\theta$ denote the parameter vector of $Q$.

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

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

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

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

This is a deterministic dynamical system in the parameter space.
OSSS: Fixed Point iteration for $m_1$

\[ m_1^{(t)} \leftarrow f(m_1^{(t-1)}) \]

\[ m_1^{(t)} = m_1^{(t-1)} \]
Stochastic Inference

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

- Given a transition kernel $T$ (= a collection of probability distributions conditioned on each $s$) and an initial configuration $s^{(0)}$
  
  $$s^{(t+1)} \sim T(s | s^{(t)}) \quad t = 1, \ldots, \infty$$

- This is a stochastic dynamical system in the configuration space.

- A remarkable fact is that we can estimate any desired expectation by ergodic averages
  
  $$\langle f(s) \rangle_\mathcal{P} \approx \frac{1}{t - t_0} \sum_{n=t_0}^{t} f(s^{(n)})$$

- Consecutive samples $s^{(t)}$ are dependent but we can “pretend” as if they are independent!
Looking ahead...

- For OSSS, the configuration space is \( s = (s_1, s_2)^\top \).

- A possible transition kernel \( T \) is specified by

\[
s_1^{(t+1)} \sim p(s_1|s_2^{(t)}, x = \hat{x}) \propto \phi(s_1, s_2^{(t)})
\]

\[
s_2^{(t+1)} \sim p(s_2|s_1^{(t+1)}, x = \hat{x}) \propto \phi(s_1^{(t+1)}, s_2)
\]

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

$$\pi(s) = T^\infty r(s)$$

(5)

- Sample $s^{(0)} \sim r(s)$

- For $t = 1 \ldots \infty$, Sample $s^{(t)} \sim T(s|s^{(t-1)})$

- Estimate any desired expectation by the average

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

where $t_0$ is a preset burn-in period.

But how to construct $T$ and verify that $\pi(s)$ is indeed its stationary distribution?
Equilibrium condition = Detailed Balance

\[ T(s|s')\pi(s') = T(s'|s)\pi(s) \]

If detailed balance is satisfied then \( \pi(s) \) is a stationary distribution

\[ \pi(s) = \int ds'T(s|s')\pi(s') \]

If the configuration space is discrete, we have

\[ \pi(s) = \sum_{s'} T(s|s')\pi(s') \]

\[ \pi = T\pi \]

\( \pi \) has to be a (right) eigenvector of \( T \).
Conditions on $T$

- Irreducibility (probabilistic 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

\[
a(s \rightarrow s') \equiv \min\{1, \frac{q(s|s')\pi(s')}{q(s'|s)\pi(s)}\}
\]
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: \( u \sim \text{Uniform}[0, 1] \)

   \[
   s^{(t)} := \begin{cases} 
   s' & u < a(s^{(t-1)} \rightarrow s') \rightarrow \text{Accept} \\
   s^{(t-1)} & \text{otherwise} \rightarrow \text{Reject}
   \end{cases}
   \]
Transition Kernel of the Metropolis Algorithm

\[ T(s' \mid s) = \underbrace{q(s' \mid s)a(s \rightarrow s')}_{\text{Accept}} + \underbrace{\delta(s' - s)}_{\text{Reject}} \int ds' q(s' \mid s)(1 - a(s \rightarrow s')) \]
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:

$$T_c = T_1 T_2$$

$$T_m = \nu T_1 + (1 - \nu) T_2 \quad 0 \leq \nu \leq 1$$

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 $\nu$ as a function of rejection rate.
Optimization: Simulated Annealing and Iterative Improvement

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

\[
s^* = \arg \max_{s \in S} \pi(s)
\]

The MCMC sampler may not visit \( s^* \).

Simulated Annealing: We define the target distribution as

\[
\pi(s)^{\tau_i}
\]

where \( \tau_i \) 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 $\tau$
Time series models with latent variables
Online Inference, Terminology

In signal processing and machine learning many phenomena can be modelled by dynamical state space models (SSM)

\[
x_0 \rightarrow x_1 \rightarrow \ldots \rightarrow x_{k-1} \rightarrow x_k \rightarrow \ldots \rightarrow x_K
\]

\[
y_1 \rightarrow \ldots \rightarrow y_{k-1} \rightarrow y_k \rightarrow \ldots \rightarrow y_K
\]

\[
x_k \sim p(x_k|x_{k-1}) \quad \text{Transition Model}
\]

\[
y_k \sim p(y_k|x_k) \quad \text{Observation Model}
\]

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
Goals and uses of Probabilistic Models

- Finding some interesting (hidden) structure
  Clustering
  Dimensionality Reduction

- Finding a compact representation for data = Data Compression

- Outlier Detection

- Prediction

- Classification

- Optimal Decision (given a loss function)
Why are Hidden Variable models Useful?

Example: Highschool grades (inspired by J. Whittaker)
Consider the grades that students get from 7 different subjects: Maths, Physics, Chemistry, History, Sports, Literature, English.
We wish to tell some interesting story about data.

(a) “Visible” Model: Subjects contain related material or require similar abilities.

(b) Hidden Variable Model: Students have some hidden interests, e.g. Science, Languages, Art.
### Mixture Models

<table>
<thead>
<tr>
<th>Hidden</th>
<th>Visible</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete</td>
<td>Discrete</td>
<td>Discrete Mixture</td>
</tr>
<tr>
<td>Discrete</td>
<td>Gaussian</td>
<td>Mixture of Gaussians (MOG)</td>
</tr>
<tr>
<td>Gaussian</td>
<td>Gaussian</td>
<td>Factor Analysis (Constrained Gaussian)</td>
</tr>
<tr>
<td>Gaussian</td>
<td>Discrete</td>
<td>Clipped Gaussian</td>
</tr>
</tbody>
</table>
Factorized (Distributed) Representations

Discrete Factors

- Possible to code $O(2^q)$ states, however intractable for large $q$.

Continuous Factors

- Gaussian Factors $\Rightarrow$ FA, PCA, PPCA ..
- Non Gaussian Factors $\Rightarrow$ ICA, IFA
Some Applications: Audio Restoration

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

\[ p(x_{-\kappa} | x_\kappa) \propto \int d\mathcal{H} p(x_{-\kappa} | \mathcal{H}) p(x_\kappa | \mathcal{H}) p(\mathcal{H}) \]

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

\[ H \equiv \text{(parameters, hidden states)} \]

\[ x_{-\kappa} \quad x_\kappa \]

Missing \quad Observed

Cemgil Introduction to Numerical Bayesian Methods. 7-8 Eylül 2006, İstanbul, Türkiye
Some Applications: Source Separation

Estimate $n$ hidden signals $s_t$ from $m$ observed signals $x_t$.

\[ s^i_t \sim p(s^i_t) \]
\[ x^j_t \sim \mathcal{N}(x; a^j s^1_{1:n_t}, r^j) \]
Time Series models: Introduce Dynamics

\[ S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \]

<table>
<thead>
<tr>
<th>S</th>
<th>O</th>
<th>Static</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete</td>
<td>Discrete</td>
<td>DM</td>
<td>Discrete HMM</td>
</tr>
<tr>
<td>Discrete</td>
<td>Gaussian</td>
<td>MOG</td>
<td>Continuous HMM</td>
</tr>
<tr>
<td>Gaussian</td>
<td>Gaussian</td>
<td>FA</td>
<td>Linear Dynamical System</td>
</tr>
</tbody>
</table>
Inference in HMMs

Compute

\[ p(S|O) = \frac{p(O|S)p(S)}{p(O)} \]

The crux is to compute \( p(O) \).

\[
p(O) = \sum_S p(O|S)p(S)
\]

\[
= \sum_S \prod_t p(S_t|S_{t-1})p(O_t|S_t)
\]

\[
p(S_1|S_0) = p(S_1)
\]
Inference in HMMs: Forward

\[ p(O) = \sum_{S_T} p(O_T|S_T) \sum_{S_{T-1}} p(S_T|S_{T-1}) p(O_{T-1}|S_{T-1}) \cdots \sum_{S_2} p(S_3|S_2) \]

\[ \underbrace{\sum_{S_T} p(O_T|S_T) \sum_{S_{T-1}} p(S_T|S_{T-1}) p(O_{T-1}|S_{T-1}) \cdots \sum_{S_2} p(S_3|S_2)}_{\alpha_T} \]

\[ p(O_2|S_2) \underbrace{\sum_{S_1} p(S_2|S_1) p(O_1|S_1) p(S_1)}_{\alpha_2} \underbrace{\pi_2}_{\pi_1} \]

\[ \underbrace{p(O_2|S_2) \sum_{S_1} p(S_2|S_1) p(O_1|S_1) p(S_1)}_{\alpha_2} \underbrace{\pi_2}_{\pi_1} \]

\[ \pi_1 \alpha_1 \pi_2 \alpha_2 \pi_3 \alpha_3 \pi_4 \alpha_4 \]

\[ S_1 \rightarrow \mathcal{O}_1 \rightarrow S_2 \rightarrow \mathcal{O}_2 \rightarrow S_3 \rightarrow \mathcal{O}_3 \rightarrow S_4 \rightarrow \mathcal{O}_4 \]
Inference in HMMs: Backward

\[ p(O) = \sum_{S_1} p(S_1)p(O_1|S_1) \sum_{S_2} p(S_2|S_1)p(O_2|S_2) \ldots \]

\[ \sum_{S_{T-1}} p(S_{T-1}|S_{T-2})p(O_{T-1}|S_{T-1}) \sum_{S_T} p(S_T|S_{T-1})p(O_T|S_T) \frac{1}{\beta_T} \]

\[ \beta_{T-2} \]

\[ \beta_{T-1} \]

\[ \beta_1 \]

\[ Cemgil \ Introduction \ to \ Numerical \ Bayesian \ Methods. \ 7-8 \ Eylül \ 2006, \ İstanbul, \ Türkiye \]
Forward-Backward as an instance of Belief Propagation on a factor graph
Four Letter Words

Dataset (From Sayood): All four letter English words (2149) of a Sun-Sparc spell checker.
(abbe, abed, abel, abet, able, ... zion, zone, zoom, zorn)

<table>
<thead>
<tr>
<th>Model</th>
<th>loglik</th>
<th>params</th>
<th>random samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>-16488</td>
<td>456975</td>
<td>tabu, else, duly, crib, bohr, seal, tome, free, bern</td>
</tr>
<tr>
<td>Markov(2)</td>
<td>-19216</td>
<td>17575</td>
<td>miro, jaid, saun, trol, bale, liro, pibo, brox, heth</td>
</tr>
<tr>
<td>Markov(1)</td>
<td>-22296</td>
<td>675</td>
<td>sehe, reah, vefa, tlil, hutu, stec, make, otod, pose</td>
</tr>
<tr>
<td>HMM</td>
<td>-24351</td>
<td>53</td>
<td>yiij, ekmy, vguo, addn, ecmi, miui, bhin, hnri, roia, azfa</td>
</tr>
<tr>
<td>Factorized</td>
<td>-25909</td>
<td>25</td>
<td></td>
</tr>
</tbody>
</table>
HMM captures an interesting structure

- For the HMM, the latent states correspond to vowel/non-vowel
- Clustering
Linear Dynamical Systems and Kalman Filter Models
Partitioned Inverse Equations

\[ K = \begin{pmatrix} S & D \\ D^T & F \end{pmatrix} \quad K^{-1} = \begin{pmatrix} \Sigma & \Delta \\ \Delta^T & \Phi \end{pmatrix} \]

\[ \Phi = (F - D^T S^{-1} D)^{-1} \]

\[ \Delta = -S^{-1} D \Phi \]

\[ \Sigma = S^{-1} (I - D \Delta^T) \]

\[ p(t, s) = \mathcal{N}([\mu_t, \mu_s]^T, K) \]

\[ \downarrow \]

\[ p(s|t) = \mathcal{N}(\mu_s + D^T S^{-1} (t - \mu_t), \Phi^{-1}) \]
Factor Analysis

\[ t = Cs + \omega \]
\[ \omega \sim \mathcal{N}(0, R) \quad s \sim \mathcal{N}(\mu_s, P) \]
\[ p(t, s) = \mathcal{N}(\mu, K) \]
\[ \mu = \left( \frac{C\mu_s}{\mu_s} \right) \]
\[ K = \left( \begin{array}{ccc} CPC^T + R & CP \\ PC^T & P \end{array} \right) \]

\[ p(s|t) = \mathcal{N}(\mu_s + PC^T(CPC^T + R)^{-1}(t - C\mu_s), P - PC^T(CPC^T + R)^{-1}CP) \]
Kalman Filtering

\[ t_i = C s_i + \omega \]
\[ s_{i+1} = A s_i + \nu \]
\[ \omega \sim \mathcal{N}(0, R) \quad \nu \sim \mathcal{N}(0, Q) \]
\[ s \sim \mathcal{N}(\mu_s, P) \]
\[ p(s_i, s_{i+1}|t_1:i) = \mathcal{N}(\mu, K) \]
\[ \mu = \left( \frac{\mu_i|i}{AP} \right) \]
\[ K = \left( \frac{P}{AP} \right) \frac{P A^T}{AP A^T + Q} \]

---

Cemgil *Introduction to Numerical Bayesian Methods*. 7-8 Eylül 2006, İstanbul, Türkiye
Kalman Filtering Equations

\[ p(s_i | t_j \ldots, t_1) = \mathcal{N}(\mu_{i|j}, P_{i|j}) \]

(by Partitioned Inverse Equations)

\[
\begin{align*}
\mu_{i|i} &= \mu_{i|i-1} + P_{i|i-1} C^T (C P_{i|i-1} C^T + R)^{-1} (\hat{t} - C \mu_{i|i-1}) \\
P_{i|i} &= P_{i|i-1} - P_{i|i-1} C^T (C P_{i|i-1} C^T + R)^{-1} C P_{i|i-1}
\end{align*}
\]

(by the parametric form of \( p(s_{i+1}, s_i) \))

\[
\begin{align*}
\mu_{i+1|i} &= A \mu_{i|i} \\
P_{i+1|i} &= A P_{i|i} A^T + Q
\end{align*}
\]
Kalman Smoothing

- Computes $p(s_i|t_1, \ldots t_N)$.

- The state estimates are “more smooth” since all observations are available.

- Analog of forward-backward algorithm in HMM’s.
Example: Point moving on the line

\[ s_i \sim \mathcal{N}(s_i; \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} s_{i-1}, Q) \]

\[ t_i \sim \mathcal{N}(t_i; \begin{pmatrix} 1 & 0 \end{pmatrix} s_i, R) \]

run filter-demo.m
Switching State space models - Segmentation - Changepoint detection
Segmentation and Changepoint detection

- Data are modelled by using simple processes with occasional regime switches
- Piecewise constant

![Piecewise constant](image1)

- Piecewise linear

![Piecewise linear](image2)
Bayesian Model Selection by Marginal MAP (MMAP)

Integrating out unknown model parameters:

\[ M^* = \arg \max_M p(D|M)p(M) = \arg \max_M \int d\theta_M p(D|\theta_M)p(\theta_M|M)p(M) \]

where \( M \): Model, \( D \): Data, \( \theta_M \): Model Parameters

- How do we calculate \( \int d\theta_M p(D|\theta_M)p(\theta_M|M)p(M) \) efficiently?
  - When dimensionality of \( \theta_M \) varies with \( M \) the standard choice is reversible Jump Markov Chain Monte Carlo (Green 1995)
  - It is possible to cast the problem to a fixed dimensional problem by introducing indicators that “switch on and off” model parameters (e.g. Godsill 1998)
Conditionally Gaussian Changepoint Model

\[ r_k \sim p(r_k | r_{k-1}) \]

\[ \theta_k = [r_k = \text{reg}] \frac{f(\theta_k | \theta_{k-1})}{\pi(\theta_k)} + [r_k = \text{new}] \frac{\pi(\theta_k)}{\pi(\theta_k)} \]

\[ y_k \sim p(y_k | \theta_k) \]

Changepoint flags \( \in \{ \text{new, reg} \} \)

Latent State

Observations
Sequential Inference Problems

- Filtering

\[ p(\theta_k | y_{1:k}) = \sum_{r_{1:k}} \int d\theta_{0:k-1} p(y_{1:k} | \theta_{0:k}) p(\theta_{0:k} | r_{1:k}) p(r_{1:k}) \]

- Viterbi path (e.g. Raphael 2001)

\[ (r_{1:k}, \theta_{1:k})^* = \arg\max_{r_{1:k}, \theta_{1:k}} p(y_{1:k} | \theta_{0:k}) p(\theta_{0:k} | r_{1:k}) p(r_{1:k}) \]

- Best segmentation (MMAP)

\[ r_{1:k}^* = \arg\max_{r_{1:k}} \int d\theta_{0:k} p(y_{1:k} | \theta_{0:k}) p(\theta_{0:k} | r_{1:k}) p(r_{1:k}) \]

- Each configuration of \( r_{1:K} \) encodes one of the possible \( 2^K \) possible models, \emph{i.e.}, segmentation.

- All problems are similar, but MMAP is usually harder because \( \max \) and \( \int \) do not commute.
Exact Inference in switching state space models

• In general, exact inference is intractable (NP hard)
  – Conditional Gaussians are not closed under marginalization

⇒ Unlike HMM’s or KFM’s, summing over $r_k$ does not simplify the filtering density
⇒ Number of Gaussian kernels to represent exact filtering density $p(r_k, \theta_k | y_{1:k})$ increases exponentially
Exact Inference for Changepoint detection?

- Exact inference is achievable in polynomial time/space
  - Intuition: When a changepoint occurs, the old state vector is reinitialized

- The same structure can be exploited for the MMAP problem
  ⇒ Trajectories $r_{1:k}^{(i)}$ which are dominated in terms of conditional evidence $p(y_{1:k}, r_{1:k}^{(i)})$ can be discarded without destroying optimality
Example 1: Piecewise constant signal

\[ \theta_0 \sim \mathcal{N}(\mu, P) \]

\[ r_k|r_{k-1} \sim p(r_k|r_{k-1}) \]

\[ \theta_k|\theta_{k-1}, r_k \sim \left[ r_k = 0 \right]\delta(\theta_k - \theta_{k-1}) + \left[ r_k = 1 \right]\mathcal{N}(m, V) \]

\[ y_k|\theta_k \sim \mathcal{N}(\theta_k, R) \]
Example 2: Audio Signal Analysis

\[
\begin{align*}
    r_k | r_{k-1} & \sim p(r_k | r_{k-1}) \\
    \theta_k | \theta_{k-1}, r_k & \sim \begin{cases} 
        [r_k = 0] N(A\theta_{k-1}, Q) & \text{reg} \\
        [r_k = 1] N(0, S) & \text{new}
    \end{cases} \\
    y_k | \theta_k & \sim N(C\theta_k, R)
\end{align*}
\]

\[A = \begin{pmatrix}
    G_\omega \\
    G_\omega^2 \\
    \vdots \\
    G_\omega^H
\end{pmatrix}, \quad G_\omega = \rho_k \begin{pmatrix}
    \cos(\omega) & -\sin(\omega) \\
    \sin(\omega) & \cos(\omega)
\end{pmatrix}\]

0 < \rho_k < 1 is a damping factor and \( C = \begin{bmatrix}
    1 & 0 & 1 & 0 & \ldots & 1 & 0
\end{bmatrix} \) is a projection matrix.
Audio Signal Analysis

$r_k$

$x_k$

$y_k$

frequency

$k$

Cemgil *Introduction to Numerical Bayesian Methods*. 7-8 Eylül 2006, İstanbul, Türkiye
Application to music transcription

Cemgil et. al. 2006, IEEE TSALP
Factorial Changepoint model

\[ r_{0,\nu} \sim C(r_{0,\nu}; \pi_{0,\nu}) \]
\[ \theta_{0,\nu} \sim N(\theta_{0,\nu}; \mu_{\nu}, P_{\nu}) \]
\[ r_{k,\nu} | r_{k-1,\nu} \sim C(r_{k,\nu}; \pi_{\nu}(r_{k-1,\nu})) \]  
\[ \text{Changepoint indicator} \]
\[ \theta_{k,\nu} | \theta_{k-1,\nu} \sim N(\theta_{k,\nu}; A_{\nu}(r_{k})\theta_{k-1,\nu}, Q_{\nu}(r_{k})) \]  
\[ \text{Latent state} \]
\[ y_k | \theta_{k,1:W} \sim N(y_k; C_{k}\theta_{k,1:W}, R) \]  
\[ \text{Observation} \]

\[ \nu = 1 \ldots W \]
Polyphonic Pitch tracking

$r_{k,v}$

$x_k$

frequency

$k$

$k$

$k$
Importance Sampling,
Online Inference, Sequential Monte Carlo
Importance Sampling

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

\[
p(\mathbf{x}) = \frac{1}{Z} \phi(\mathbf{x})
\]  

(6)

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

\[
\langle f(\mathbf{x}) \rangle_{p(\mathbf{x})} = \int d\mathbf{x} f(\mathbf{x}) p(\mathbf{x})
\]

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

(7)
Importance Sampling (cont.)

• Change of measure with **weight function** $W(x) \equiv \phi(x)/q(x)$

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

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

$$Z = \int dx \phi(x) = \int dx \frac{\phi(x)}{q(x)} q(x) = \langle W(x) \rangle_q$$

$$\langle f(x) \rangle_{p(x)} = \frac{\left\langle f(x) W(x) \right\rangle_q}{\langle W(x) \rangle_q}$$
Importance Sampling (cont.)

- Draw \( i = 1, \ldots, N \) independent samples from \( q \)
  \[
  x^{(i)} \sim q(x)
  \]

- We calculate the **importance weights**
  \[
  W^{(i)} = W(x^{(i)}) = \frac{\phi(x^{(i)})}{q(x^{(i)})}
  \]

- Approximate the normalizing constant
  \[
  Z = \langle W(x) \rangle_{q(x)} \approx \sum_{i=1}^{N} W^{(i)}
  \]

- Desired expectation is approximated by
  \[
  \langle f(x) \rangle_{p(x)} = \frac{\langle f(x) W(x) \rangle_{q(x)}}{\langle W(x) \rangle_{q(x)}} \approx \frac{\sum_{i=1}^{N} W^{(i)} f(x^{(i)})}{\sum_{i=1}^{N} W^{(i)}} \equiv \sum_{i=1}^{N} \tilde{w}^{(i)} f(x^{(i)})
  \]

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”

\[ x^{(j)}_{\text{new}} \sim \sum_i \tilde{W}^{(i)} \delta(x - x^{(i)}) \]

\[ p(x) \approx \frac{1}{N} \sum_j \delta(x - x^{(j)}_{\text{new}}) \]

- Since we sample from a degenerate distribution, particle locations stay unchanged. We merely duplicate (, 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

\[ x_{\text{new}}^{(j)} \sim \sum_i \tilde{W}^{(i)} \delta(x - x^{(i)}) \]
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. \( q(x) = \frac{p(y|x)}{\int dx p(y|x)} = \frac{p(y|x)}{c(y)} \)

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

- Interesting when sensors are very accurate and \( \text{dim}(y) \gg \text{dim}(x) \). 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

\[
\langle (f(x)W(x) - \langle f(x)W(x) \rangle)^2 \rangle_{q(x)} = \langle f^2(x)W^2(x) \rangle_{q(x)} - \langle f(x)W(x) \rangle^2_{q(x)} \tag{8}
\]
\[
= \langle f^2(x)W^2(x) \rangle_{q(x)} - \langle f(x) \rangle^2_{q(x)} \tag{9}
\]
\[
= \langle f^2(x)W^2(x) \rangle_{q(x)} - I^2(f) \tag{10}
\]

- Minimize the first term since only it depends upon \( q \)
Optimal Proposal Distribution

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

\[
\langle f^2(x)W^2(x) \rangle_{q(x)} \geq \langle |f(x)|W(x) \rangle_{q(x)}^2 = \left( \int |f(x)| p(x|y) dx \right)^2
\]

- 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

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

- We show that $q^*$ attains its lower bound

$$\langle f^2(x)W^2_*(x) \rangle_{q^*(x)} = \left\langle f^2(x)\left(\int |f(x')|p(x'|y)dx'\right)^2 \right\rangle_{q^*(x)}$$

$$= \left(\int |f(x')|p(x'|y)dx'\right)^2 = \langle |f(x)| \rangle^2_{p(x|y)}$$

$$= \langle |f(x)|W_*(x) \rangle_{q^*(x)}^2$$

- ⇒ There are distributions $q^*$ that are even “better” than the exact posterior!
Examples of Proposal Distributions

\[
p(x \mid y) \propto p(y_1 \mid x_1)p(x_1)p(y_2 \mid x_2)p(x_2 \mid x_1)
\]

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

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

\[W(x_1, x_2) = p(y_1 \mid x_1)p(y_2 \mid x_2)\]

- We sample from the prior as follows:

\[
x^{(i)}_1 \sim p(x_1) \quad x^{(i)}_2 \sim p(x_2 \mid x_1 = x^{(i)}_1) \quad W(x^{(i)}) = p(y_1 \mid x^{(i)}_1)p(y_2 \mid x^{(i)}_2)
\]
Examples of Proposal Distributions

\[ p(x \mid y) \propto p(y_1 \mid x_1)p(x_1)p(y_2 \mid x_2)p(x_2 \mid x_1) \]

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

\[ W(x_1, x_2) = \frac{p(y_1 \mid x_1)p(x_1)p(y_2 \mid x_2)p(x_2 \mid x_1)}{p(x_1 \mid y_1)p(x_2 \mid x_1)} = p(y_1)p(y_2 \mid x_2) \]

- Note that this proposal does not depend on \( x_1 \)

- We sample from the proposal and compute the weight

\[ x_1^{(i)} \sim p(x_1 \mid y_1) \quad x_2^{(i)} \sim p(x_2 \mid x_1 = x_1^{(i)}) \quad W(x^{(i)}) = p(y_1)p(y_2 \mid x_2^{(i)}) \]
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

\[
x_1^{(i)} \sim p(x_1|y_1) \quad x_2^{(i)} \sim p(x_2|x_1 = x_1^{(i)}, y_2) \quad W(x^{(i)}) = p(y_1)p(y_2|x_1^{(i)})
\]
Sequential Importance Sampling, Particle Filtering

Apply importance sampling to the SSM to obtain some samples from the posterior

\[ p(x_{0:K}|y_{1:K}). \]

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

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

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

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

(13)

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

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

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

Extend trajectory

$$W_{k}^{(i)} = u_{k|0:k-1}^{(i)} W_{k-1}$$

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

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

• The weight is given by

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

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

---

Cemgil *Introduction to Numerical Bayesian Methods*. 7-8 Eylül 2006, İstanbul, Türkiye
Example (cont.)

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

\[
\begin{align*}
  x_{k}^{(i)} & \sim p(x_k | x_{k-1}^{(i)}) & \text{Extend trajectory} \\
  W_{k}^{(i)} & = p(y_k | x_k^{(i)}) W_{k-1}^{(i)} & \text{Update weight} \\
  \tilde{W}_{k}^{(i)} & = W_{k}^{(i)} / \tilde{Z}_k & \text{Normalize} \ (\tilde{Z}_k \equiv \sum_{i'} W_{k}^{(i')}) \\
  x_{0:k, \text{new}}^{(j)} & \sim \sum_{i=1}^{N} \tilde{W}_{k}^{(i)} \delta(x_{0:k} - x_{0:k}^{(i)}) & \text{Resample} \ j = 1 \ldots N
\end{align*}
\]

• This results in a new representation as

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

\[
\begin{align*}
  x_{0:k}^{(i)} & \leftarrow x_{0:k, \text{new}}^{(j)} & W_{k}^{(i)} & \leftarrow \tilde{Z}_k / N
\end{align*}
\]
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 $x^{(i)}_{1:k-1}$ (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

$$q(x_k|x_{1:k-1}, y_{1:k}) = p(x_k|x_{k-1}, y_k)$$  \hspace{1cm} (14)
Optimal proposal distribution (cont.)

- The weight is given by

\[ x^{(i)}_k \sim p(x_k|x_{k-1}^{(i)}, y_k) \]

\[ W^{(i)}_k = u^{(i)}_{k|0:k-1} W^{(i)}_{k-1} \quad \text{Extend trajectory} \]

\[ u^{(i)}_{k|0:k-1} = \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{p(x_k^{(i)}|x_{k-1}^{(i)}, y_k)} \times \frac{p(y_k|x_{k-1}^{(i)})}{p(y_k|x_{k-1}^{(i)})} \]

\[ = \frac{p(y_k, x_k^{(i)}|x_{k-1}^{(i)})p(y_k|x_{k-1}^{(i)})}{p(x_k^{(i)}, y_k|x_{k-1}^{(i)})} = p(y_k|x_{k-1}^{(i)}) \]

\[ \]
A Generic Particle Filter

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

2. **Evaluate** importance weights

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

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

   Normalize weigths
   $$\tilde{W}_k^{(i)} = W_k^{(i)} / \tilde{Z}_k$$
   $$\tilde{Z}_k \equiv \sum_j W_k^{(j)}$$

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

   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

  

- Frey and Jojic [2]
- Wainwright and Jordan [8]

MCMC and SMC tutorials and overviews

- Andrieu. *Monte Carlo Methods for Absolute beginners*, 2004

The “in Practice” Books

References