# Bayesian Methods for Multimedia Signal Processing

A. Taylan Cemgil

Signal Processing and Communications Lab.

UNIVERSITY OF CAMBRIDGE
Department of Engineering

ACM Multimedia 2007, Augsburg, Germany
Tutorial1a
September 24, 2007



## Goals of this Tutorial

- Provide a basic understanding of underlying principles of probabilistic modeling and Bayesian inference
- Orientation in the broad literature of Bayesian machine learning and statistical signal processing
- Focus on fundamental concepts rather than technical details,

... we avoid heavy use of algebra by a graphical notation
... but there will be some maths



## Goals of this Tutorial

- Model based approach

... rather than description of algorithms for solving specific problems

- Illustrate with examples how certain problems in multimedia signal analysis can be approached using generic tools
- Motivate participants to investigate further

... provide alternative perspective to existing solutions
... and hopefully provide new inspiration



## First Part, Basic Concepts

- Introduction
  - Bayes' Theorem,
  - Trivial toy example to clarify notation
- Graphical Models
  - Bayesian Networks
  - Undirected Graphical models, Markov Random Fields
  - Factor graphs
- Maximum Likelihood, Penalised Likelihood, Bayesian Learning
- Basic Building Blocks in model construction
  - Probability distributions, Exponential family

## Second Part, Models and Applications

- Hidden Markov Models,
  - Tempo tracking, Score-performance matching
  - Inference in Hidden Markov Models
    - Forward Backward Algorithm
    - Viterbi
    - Exact inference by message passing: Belief Propagation
- Linear Dynamical systems, Kalman Filter Models
  - Tracking
  - Computer Accompaniment
  - Kalman Filtering and Smoothing
  - Audio Restoration and Interpolation



- Switching State Space models, Changepoint Models
  - Pitch tracking
  - Particle Filtering
- Nonlinear Dynamical Systems
  - Object tracking in video
  - Particle Filtering, Sequential Monte Carlo
- Markov Random Fields
  - Denoising, Source Separation
  - Markov Chain Monte Carlo, Gibbs sampler
  - Variational Bayes



- Topic-Term Models
  - Latent Semantic indexing
  - Generative aspect model, Latent Dirichlet allocation
- Factorial Models, Sparsity, Model selection
  - Audio Source Separation
  - Polyphonic Pitch Tracking
  - Approximate Inference in Factorial Models
- Final Remarks and Bibliography



## Bayes' Theorem [13, 15]

Thomas Bayes (1702-1761)

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

$$
\begin{aligned}
p(\lambda|\mathcal{D}) &= \frac{p(\mathcal{D}|\lambda)p(\lambda)}{p(\mathcal{D})} \\
\text{Posterior} &= \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}
\end{aligned}
$$

## An application of Bayes' Theorem: "Source Separation"

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

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

What is $\lambda$ when $\mathcal{D} = 9$ ?



## An application of Bayes' Theorem: "Source Separation"

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

| $\mathcal{D} = \lambda + y$ | $y = 1$ | $y = 2$ | $y = 3$ | $y = 4$ | $y = 5$ | $y = 6$ |
|---|---|---|---|---|---|---|
| $\lambda = 1$ | 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: $p(\mathcal{D}|\lambda)$



## "Bureaucratical" derivation

Formally we write

$$\begin{aligned} p(\lambda) &= \mathcal{C}(\lambda; [\ 1/6 \quad 1/6 \quad 1/6 \quad 1/6 \quad 1/6 \quad 1/6\ ]) \\ p(y) &= \mathcal{C}(y; [\ 1/6 \quad 1/6 \quad 1/6 \quad 1/6 \quad 1/6 \quad 1/6\ ]) \\ p(\mathcal{D}|\lambda, y) &= \delta(\mathcal{D} - (\lambda + y)) \end{aligned}$$

$$\begin{aligned} p(\lambda, y|\mathcal{D}) &= \frac{1}{p(\mathcal{D})} \times p(\mathcal{D}|\lambda, y) \times p(y)p(\lambda) \\ \text{Posterior} &= \frac{1}{\text{Evidence}} \times \text{Likelihood} \times \text{Prior} \end{aligned}$$

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



## Prior

$$p(y)p(\lambda)$$

| $p(y) \times p(\lambda)$ | $y = 1$ | $y = 2$ | $y = 3$ | $y = 4$ | $y = 5$ | $y = 6$ |
|---|---|---|---|---|---|---|
| $\lambda = 1$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ |
| $\lambda = 2$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ |
| $\lambda = 3$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ |
| $\lambda = 4$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ |
| $\lambda = 5$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ |
| $\lambda = 6$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ | $1/36$ |

- A table with indicies $\lambda$ and $y$
- Each cell denotes the probability $p(\lambda, y)$

## Likelihood

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

| $p(\mathcal{D}=9\|\lambda,y)$ | $y=1$ | $y=2$ | $y=3$ | $y=4$ | $y=5$ | $y=6$ |
|---|---|---|---|---|---|---|
| $\lambda=1$ | 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 | 0 | **1** | 0 | 0 |
| $\lambda=6$ | 0 | 0 | **1** | 0 | 0 | 0 |

- A table with indicies $\lambda$ and $y$
- The likelihood is **not** a probability distribution, but a positive function.



## Likelihood $\times$ Prior

$$\phi_{\mathcal{D}}(\lambda,y)=p(\mathcal{D}=9|\lambda,y)p(\lambda)p(y)$$

| $p(\mathcal{D}=9\|\lambda,y)$ | $y=1$ | $y=2$ | $y=3$ | $y=4$ | $y=5$ | $y=6$ |
|---|---|---|---|---|---|---|
| $\lambda=1$ | 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 | 0 | **1/36** | 0 | 0 |
| $\lambda=6$ | 0 | 0 | **1/36** | 0 | 0 | 0 |



## Evidence

$$\begin{aligned} p(\mathcal{D}=9) &= \sum_{\lambda,y} p(\mathcal{D}=9|\lambda,y)p(\lambda)p(y) \\ &= 0+0+\cdots+1/36+1/36+1/36+1/36+0+\cdots+0 \\ &= 1/9 \end{aligned}$$

| $p(\mathcal{D}=9\|\lambda,y)$ | $y=1$ | $y=2$ | $y=3$ | $y=4$ | $y=5$ | $y=6$ |
|---|---|---|---|---|---|---|
| $\lambda=1$ | 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 | 0 | **1/36** | 0 | 0 |
| $\lambda=6$ | 0 | 0 | **1/36** | 0 | 0 | 0 |



## Posterior

$$p(\lambda,y|\mathcal{D}=9) = \frac{1}{p(\mathcal{D})}p(\mathcal{D}=9|\lambda,y)p(\lambda)p(y)$$

| $p(\mathcal{D}=9\|\lambda,y)$ | $y=1$ | $y=2$ | $y=3$ | $y=4$ | $y=5$ | $y=6$ |
|---|---|---|---|---|---|---|
| $\lambda=1$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $\lambda=2$ | 0 | 0 | 0 | 0 | 0 | 0 |
| $\lambda=3$ | 0 | 0 | 0 | 0 | 0 | **1/4** |
| $\lambda=4$ | 0 | 0 | 0 | 0 | **1/4** | 0 |
| $\lambda=5$ | 0 | 0 | 0 | **1/4** | 0 | 0 |
| $\lambda=6$ | 0 | 0 | **1/4** | 0 | 0 | 0 |

$$1/4 = (1/36)/(1/9)$$

# Marginal Posterior

$$p(\lambda|\mathcal{D}) \quad = \quad \sum_y \frac{1}{p(\mathcal{D})} p(\mathcal{D}|\lambda, y)p(\lambda)p(y)$$

| | $p(\lambda\|\mathcal{D}=9)$ | $y=1$ | $y=2$ | $y=3$ | $y=4$ | $y=5$ | $y=6$ |
|---|---|---|---|---|---|---|---|
| $\lambda=1$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ |
| $\lambda=2$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ |
| $\lambda=3$ | $\mathbf{1/4}$ | $0$ | $0$ | $0$ | $0$ | $0$ | $1/4$ |
| $\lambda=4$ | $\mathbf{1/4}$ | $0$ | $0$ | $0$ | $0$ | $1/4$ | $0$ |
| $\lambda=5$ | $\mathbf{1/4}$ | $0$ | $0$ | $0$ | $1/4$ | $0$ | $0$ |
| $\lambda=6$ | $\mathbf{1/4}$ | $0$ | $0$ | $1/4$ | $0$ | $0$ | $0$ |



# The "proportional to" notation

$$p(\lambda|\mathcal{D}=9) \quad \propto \quad p(\lambda, \mathcal{D}=9) = \sum_y p(\mathcal{D}=9|\lambda, y)p(\lambda)p(y)$$

| | $p(\lambda,\mathcal{D}=9)$ | $y=1$ | $y=2$ | $y=3$ | $y=4$ | $y=5$ | $y=6$ |
|---|---|---|---|---|---|---|---|
| $\lambda=1$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ |
| $\lambda=2$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ | $0$ |
| $\lambda=3$ | $1/36$ | $0$ | $0$ | $0$ | $0$ | $0$ | $\mathbf{1/36}$ |
| $\lambda=4$ | $1/36$ | $0$ | $0$ | $0$ | $0$ | $\mathbf{1/36}$ | $0$ |
| $\lambda=5$ | $1/36$ | $0$ | $0$ | $0$ | $\mathbf{1/36}$ | $0$ | $0$ |
| $\lambda=6$ | $1/36$ | $0$ | $0$ | $\mathbf{1/36}$ | $0$ | $0$ | $0$ |



# Exercise

| $p(x_1, x_2)$ | $x_2=1$ | $x_2=2$ |
|---|---|---|
| $x_1=1$ | $0.3$ | $0.3$ |
| $x_1=2$ | $0.1$ | $0.3$ |

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: $p(x_2=2)$
   - $p(\{\})$
   - Max: $p(x_1^*) = \max_{x_1} p(x_1|x_2=1)$
   - Mode: $x_1^* = \arg\max_{x_1} p(x_1|x_2=1)$
   - Max-marginal: $\max_{x_1} p(x_1, x_2)$
2. Are $x_1$ and $x_2$ independent ? (i.e., Is $p(x_1, x_2) = p(x_1)p(x_2)$ ?)



# Answers

| $p(x_1, x_2)$ | $x_2=1$ | $x_2=2$ |
|---|---|---|
| $x_1=1$ | $0.3$ | $0.3$ |
| $x_1=2$ | $0.1$ | $0.3$ |

- Marginals:

| $p(x_1)$ | |
|---|---|
| $x_1=1$ | $0.6$ |
| $x_1=2$ | $0.4$ |

| $p(x_2)$ | $x_2=1$ | $x_2=2$ |
|---|---|---|
| | $0.4$ | $0.6$ |

- Conditionals:

| $p(x_1\|x_2)$ | $x_2=1$ | $x_2=2$ |
|---|---|---|
| $x_1=1$ | $0.75$ | $0.5$ |
| $x_1=2$ | $0.25$ | $0.5$ |

| $p(x_2\|x_1)$ | $x_2=1$ | $x_2=2$ |
|---|---|---|
| $x_1=1$ | $0.5$ | $0.5$ |
| $x_1=2$ | $0.25$ | $0.75$ |

## Answers

| $p(x_1, x_2)$ | $x_2 = 1$ | $x_2 = 2$ |
|---|---|---|
| $x_1 = 1$ | 0.3 | 0.3 |
| $x_1 = 2$ | 0.1 | 0.3 |

- Posterior:

| $p(x_1, x_2 = 2)$ | $x_2 = 2$ |
|---|---|
| $x_1 = 1$ | 0.3 |
| $x_1 = 2$ | 0.3 |

| $p(x_1 \vert x_2 = 2)$ | $x_2 = 2$ |
|---|---|
| $x_1 = 1$ | 0.5 |
| $x_1 = 2$ | 0.5 |

- Evidence:

$$p(x_2 = 2) = \sum_{x_1} p(x_1, x_2 = 2) = 0.6$$

- Normalisation constant:

$$p(\{\}) = \sum_{x_1} \sum_{x_2} p(x_1, x_2) = 1$$



## Answers

| $p(x_1, x_2)$ | $x_2 = 1$ | $x_2 = 2$ |
|---|---|---|
| $x_1 = 1$ | 0.3 | 0.3 |
| $x_1 = 2$ | 0.1 | 0.3 |

- Max: (get the value)

$$\max_{x_1} p(x_1 | x_2 = 1) = 0.75$$

- Mode: (get the index)

$$\operatorname*{argmax}_{x_1} p(x_1 | x_2 = 1) = 1$$

- Max-marginal: (get the "skyline") $\max_{x_1} p(x_1, x_2)$

| $\max_{x_1} p(x_1, x_2)$ | $x_2 = 1$ | $x_2 = 2$ |
|---|---|---|
| | 0.3 | 0.3 |



## Another application of Bayes' Theorem: "Model Selection"

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

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

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

Assume that any number $n$ is equally likely



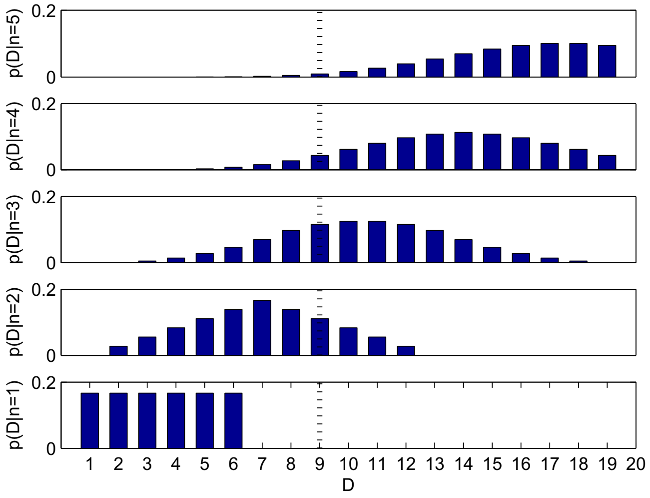
## Another application of Bayes' Theorem: "Model Selection"

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

$$p(n | \mathcal{D} = 9) = \frac{p(\mathcal{D} = 9 | n) p(n)}{p(\mathcal{D})} \propto p(\mathcal{D} = 9 | n)$$

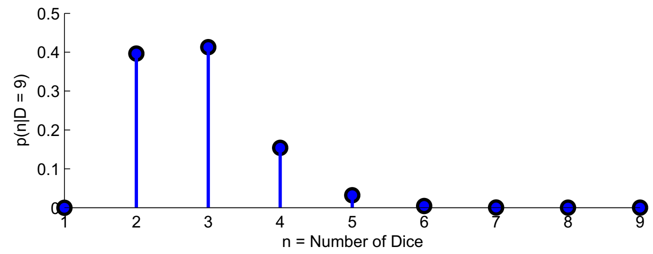
$$\begin{aligned}
p(\mathcal{D} | n = 1) &= \sum_{\lambda_1} p(\mathcal{D} | \lambda_1) p(\lambda_1) \\
p(\mathcal{D} | n = 2) &= \sum_{\lambda_1} \sum_{\lambda_2} p(\mathcal{D} | \lambda_1, \lambda_2) p(\lambda_1) p(\lambda_2) \\
\dots \\
p(\mathcal{D} | n = n') &= \sum_{\lambda_1, \dots, \lambda_{n'}} p(\mathcal{D} | \lambda_1, \dots, \lambda_{n'}) \prod_{i=1}^{n'} p(\lambda_i)
\end{aligned}$$

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






## 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 $\lambda$



## Probabilistic Inference

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

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

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

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

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

- any "mix" of the above: e.g.,

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



## 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
  - Mechanical (In theory! not in practice)
  - Generic

## Applications of Probability Models

- Classification
- Optimal Decision, given a loss function
- Finding interesting (hidden) structure
  - Clustering, Segmentation
  - Dimensionality Reduction
  - Outlier Detection
- Finding a compact representation $=$ Data Compression
- Prediction



## Probability Models

$+$

## Inference Algorithms

$=$

## Bayesian Numerical Methods



## Graphical Models

- formal languages for specification of probability models and associated inference algorithms
- historically, introduced in probabilistic expert systems (Pearl 1988) as a visual guide for representing expert knowledge
- today, a standard tool in machine learning, statistics and signal processing



## Graphical Models

- provide graph based algorithms for derivations and computation
- pedagogical insight/motivation for model/algorithm construction
  - Statistics:
    "Kalman filter models and hidden Markov models (HMM) are equivalent upto parametrisation"
  - Signal processing:
    "Fast Fourier transform is an instance of sum-product algorithm on a factor graph"
  - Computer Science:
    "Backtracking in Prolog is equivalent to inference in Bayesian networks with deterministic tables"
- Automated tools for code generation start to emerge, making the design/implement/test cycle shorter

## Important types of Graphical Models

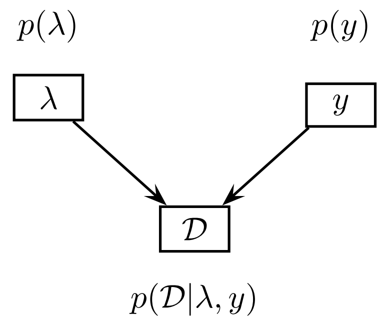
- Useful for Model Construction
  - **Directed Acyclic Graphs (DAG), Bayesian Networks**
  - **Undirected Graphs, Markov Networks, Random Fields**
  - Influence diagrams
  - ...
- Useful for Inference
  - **Factor Graphs**
  - Junction/Clique graphs
  - Region graphs
  - ...



## Directed Graphical models (DAG)



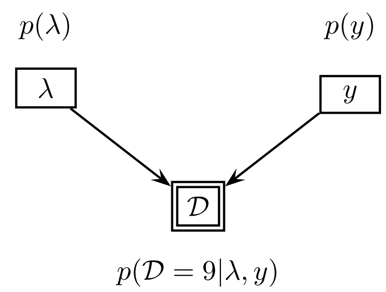
## DAG Example: Two dice




$$p(\mathcal{D}, \lambda, y) = p(\mathcal{D}|\lambda, y)p(\lambda)p(y)$$



## DAG with observations




$$\phi_{\mathcal{D}}(\lambda, y) = p(\mathcal{D} = 9|\lambda, y)p(\lambda)p(y)$$

## Directed Graphical models

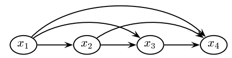
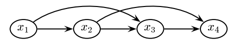
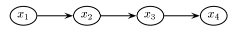
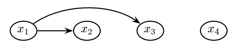
- Each random variable is associated with a node in the graph,
- We draw an arrow from $A \to B$ if $p(B| \dots, A, \dots)$ ($A \in \text{parent}(B)$),
- The edges tell us *qualitatively* about the factorization of the joint probability
- For $N$ random variables $x_1, \dots, x_N$, the distribution admits

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

- Describes in a compact way an algorithm to "generate" the data – "Generative models"



## Examples

| Model | Structure | factorization |
|---|---|---|
| Full |  | $p(x_1)p(x_2\|x_1)p(x_3\|x_1,x_2)p(x_4\|x_1,x_2,x_3)$ |
| Markov(2) |  | $p(x_1)p(x_2\|x_1)p(x_3\|x_1,x_2)p(x_4\|x_2,x_3)$ |
| Markov(1) |  | $p(x_1)p(x_2\|x_1)p(x_3\|x_2)p(x_4\|x_3)$ |
| |  | $p(x_1)p(x_2\|x_1)p(x_3\|x_1)p(x_4)$ |
| Factorized |  | $p(x_1)p(x_2)p(x_3)p(x_4)$ |

Removing edges eliminates a term from the conditional probability factors.



# Undirected Graphical Models



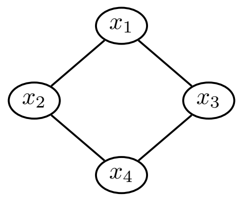
## Undirected Graphical Models

- Define a distribution by non-negative *local compatibility functions* $\phi(x_\alpha)$

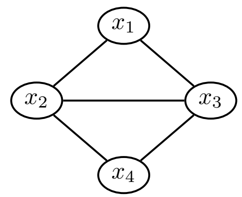
$$p(\mathbf{x}) = \frac{1}{Z} \prod_{\alpha} \phi(x_\alpha)$$

where $\alpha$ runs over **cliques** : fully connected subsets

- Examples

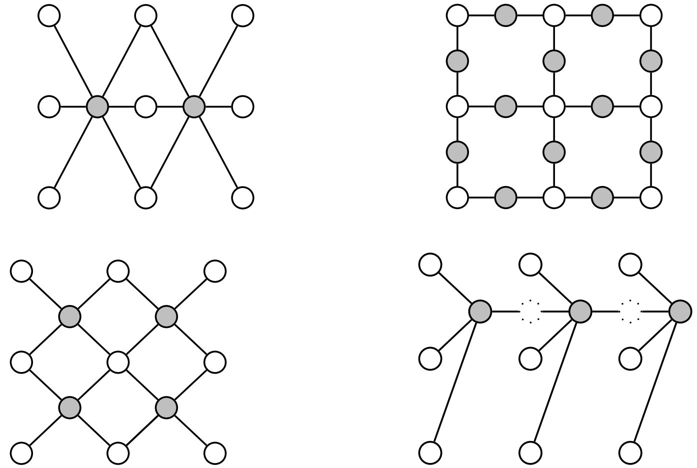


$p(\mathbf{x}) = \frac{1}{Z}\phi(x_1,x_2)\phi(x_1,x_3)\phi(x_2,x_4)\phi(x_3,x_4)$



$p(\mathbf{x}) = \frac{1}{Z}\phi(x_1,x_2,x_3)\phi(x_2,x_3,x_4)$

## Possible Model Topologies



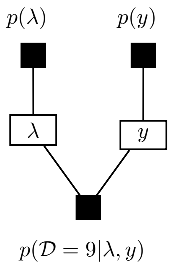


## Factor graphs



## Factor graphs [14]

- 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 Nodes. Define collections of random variables
  - **Edges**: denote membership. A variable node is connected to a factor node if a member variable is an argument of the local function.

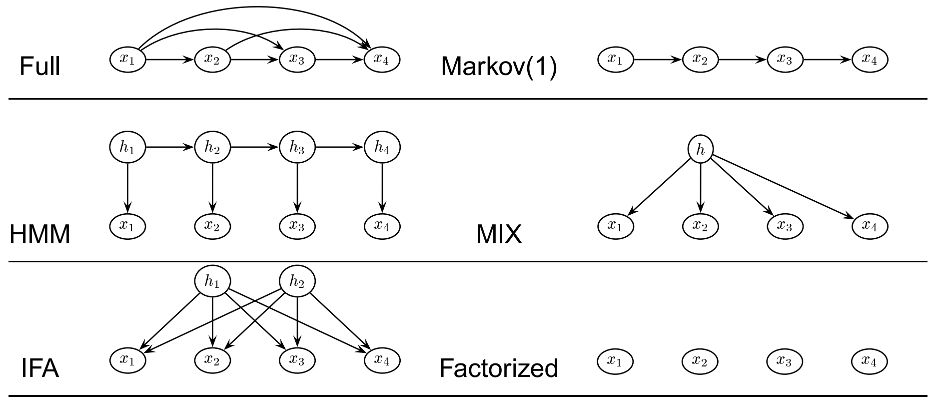



$$\phi_{\mathcal{D}}(\lambda,y) \quad = \quad p(\mathcal{D}=9|\lambda,y)p(\lambda)p(y) = \phi_1(\lambda,y)\phi_2(\lambda)\phi_3(y)$$

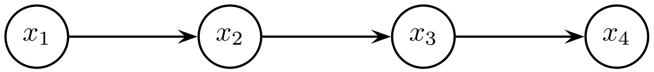


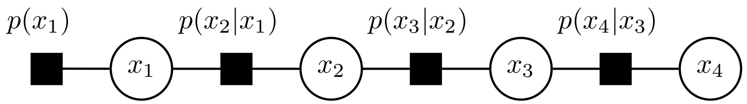
## Exercise

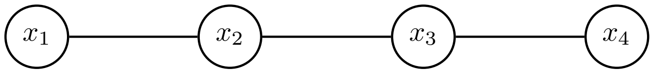
- For the following Graphical models, write down the factors of the joint distribution and plot an equivalent factor graph and an undirected graph.



# Answer (Markov(1))



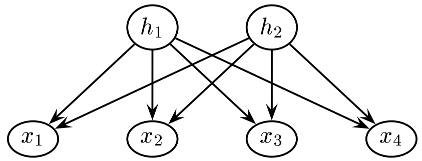



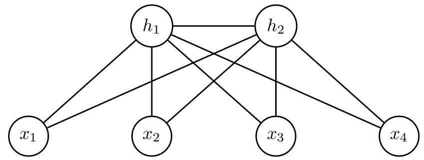

$$\underbrace{p(x_1)p(x_2|x_1)}_{\phi(x_1,x_2)}\underbrace{p(x_3|x_2)}_{\phi(x_2,x_3)}\underbrace{p(x_4|x_3)}_{\phi(x_3,x_4)}$$



# Answer (IFA – Factorial)

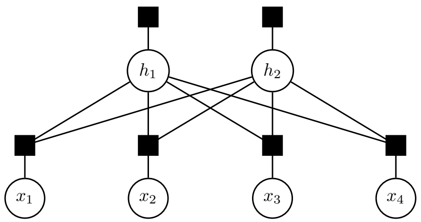



$$p(h_1)p(h_2)\prod_{i=1}^{4} p(x_i|h_1,h_2)$$

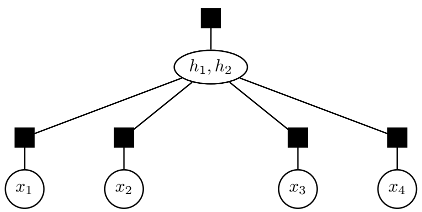





# Answer (IFA – Factorial)




- We can also cluster nodes together






# Inference and Learning

- Data set
$$\mathcal{D} = \{x_1, \dots x_N\}$$
- Model with parameter $\lambda$
$$p(\mathcal{D}|\lambda)$$
- Maximum Likelihood (ML)
$$\lambda^{\text{ML}} = \arg\max_{\lambda} \log p(\mathcal{D}|\lambda)$$
- Predictive distribution
$$p(x_{N+1}|\mathcal{D}) \approx p(x_{N+1}|\lambda^{\text{ML}})$$

## Regularisation

- Prior

$$p(\lambda)$$

- Maximum a-posteriori (MAP) : Regularised Maximum Likelihood

$$\lambda^{\text{MAP}} = \arg\max_{\lambda} \log p(\mathcal{D}|\lambda)p(\lambda)$$

- Predictive distribution

$$p(x_{N+1}|\mathcal{D}) \approx p(x_{N+1}|\lambda^{\text{MAP}})$$



## Bayesian Learning

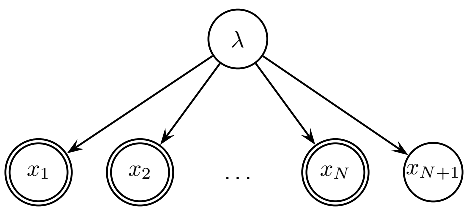
- We treat parameters on the same footing as all other variables
- We integrate over unknown parameters rather than using point estimates (remember the many-dice example)
  - Avoids overfitting
  - Natural setup for online adaptation
  - Model selection
    - (arguably) many problems in music processing are model selection problems



## Bayesian Learning

- Predictive distribution

$$p(x_{N+1}|\mathcal{D}) = \int d\lambda \;\; p(x_{N+1}|\lambda)p(\lambda|\mathcal{D})$$

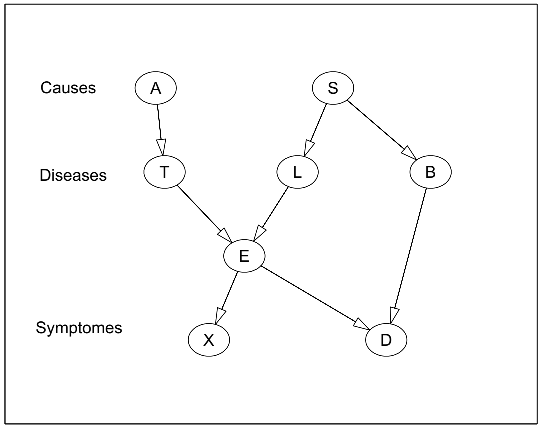



- Bayesian learning is just inference ...



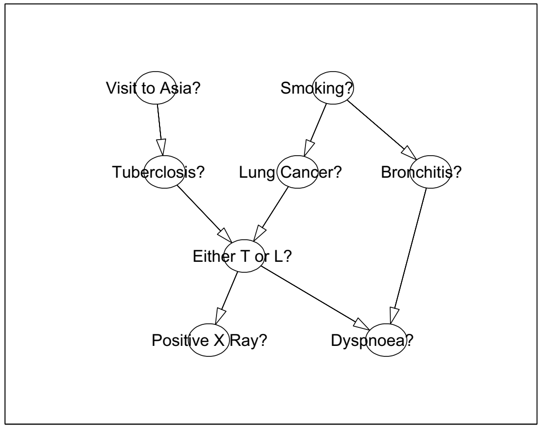
# Example Applications and Models

# Medical Expert Systems



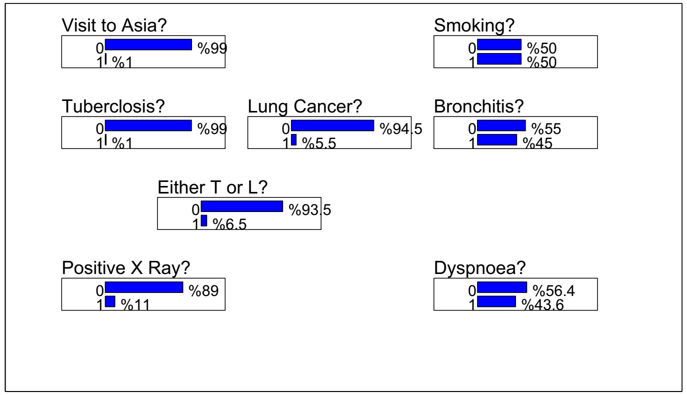



# Medical Expert Systems



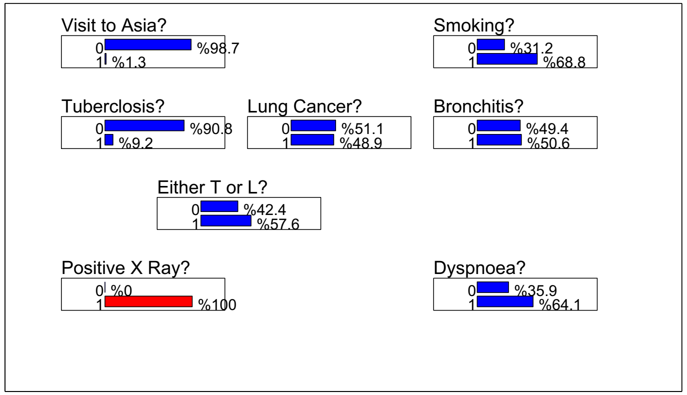



# Medical Expert Systems

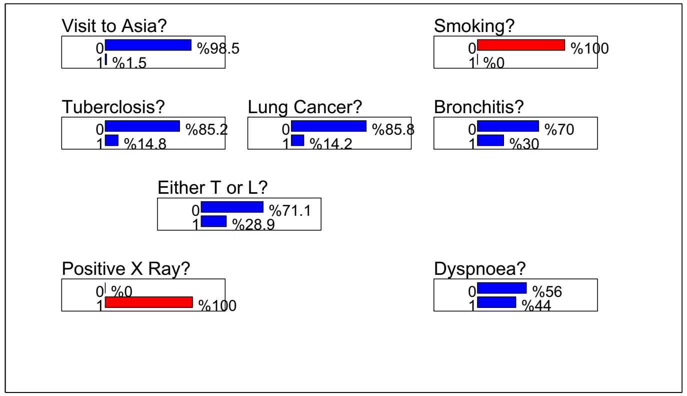





# Medical Expert Systems



# Medical Expert Systems

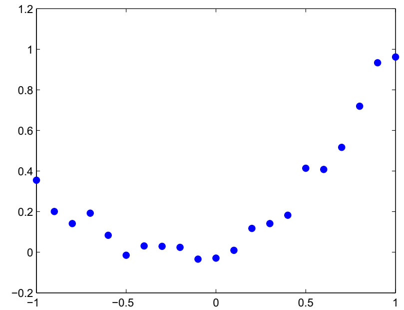





# Model Selection: Variable selection in Polynomial Regression

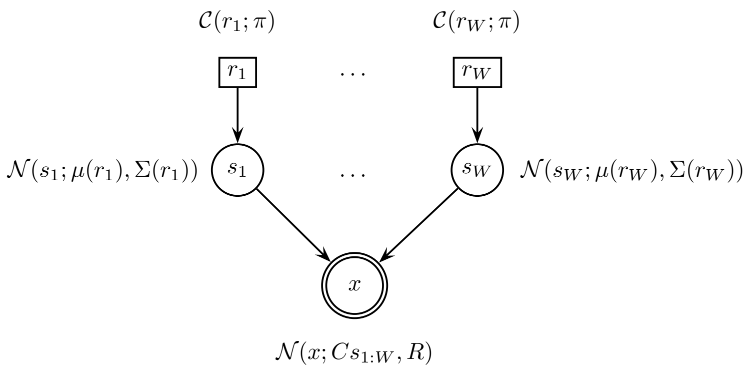
- Given $\mathcal{D} = \{t_j, x(t_j)\}_{j=1\ldots J}$, what is the order $N$ of the polynomial?

$$x(t) = \sum_{i=0}^{N} s_{i+1} t^i + \epsilon(t)$$





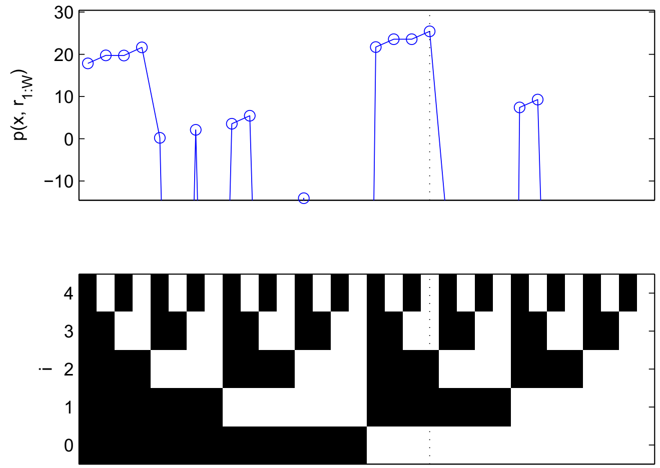
# Bayesian Variable Selection




- 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.

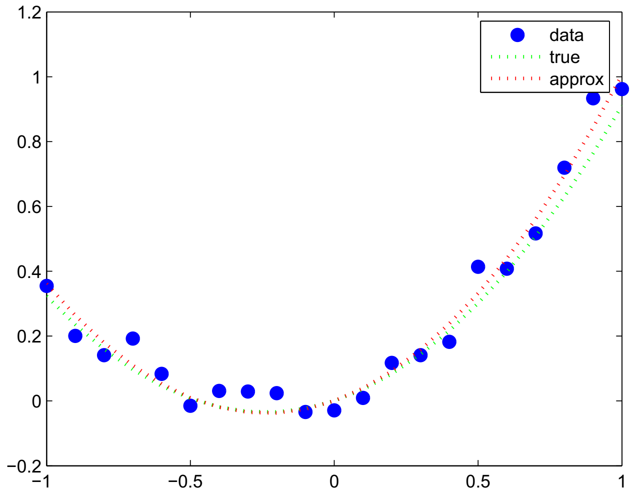


# Regression



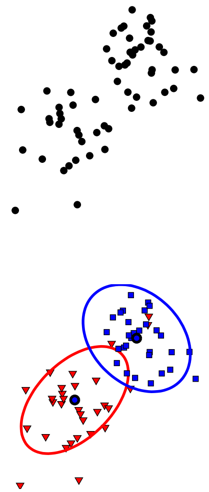

All on Configurations All off

# Regression



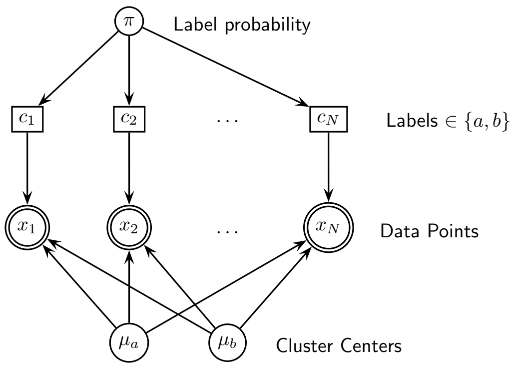


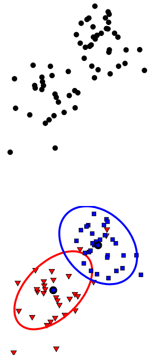

# Clustering





# Clustering



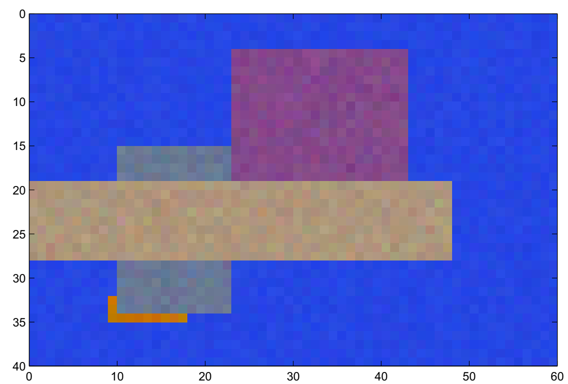



$$(\mu_a^*, \mu_b^*, \pi^*) = \underset{\mu_a,\mu_b,\pi}{\operatorname{argmax}} \sum_{c_{1:N}} \prod_{i=1}^{N} p(x_i|\mu_a, \mu_b, c_i)p(c_i|\pi)$$

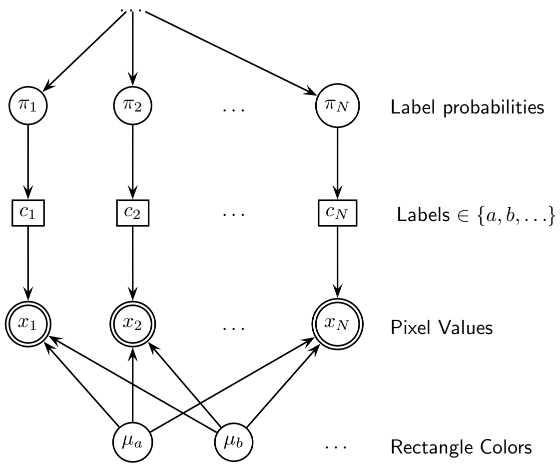


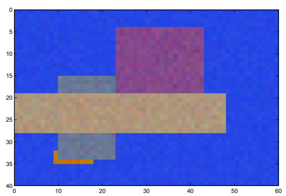
# Computer vision / Cognitive Science

How many rectangles are there in this image?



# Computer vision / Cognitive Science

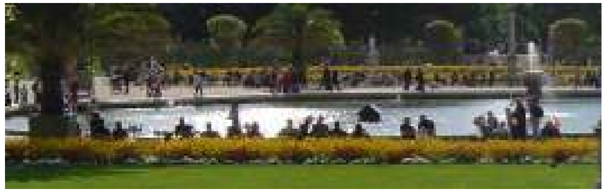


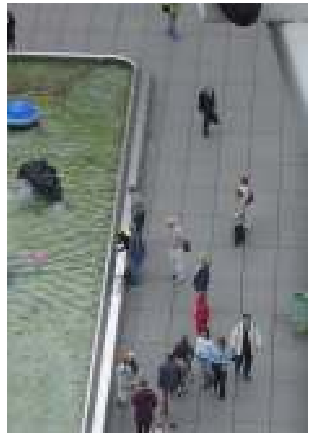





# Computer Vision

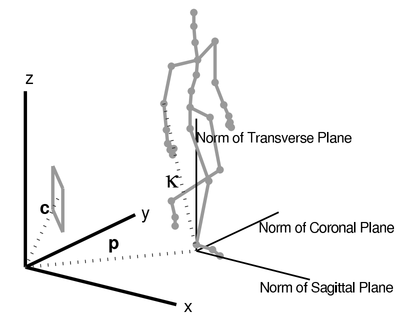
How many people are there in these images?

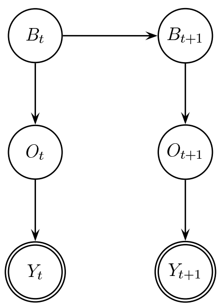


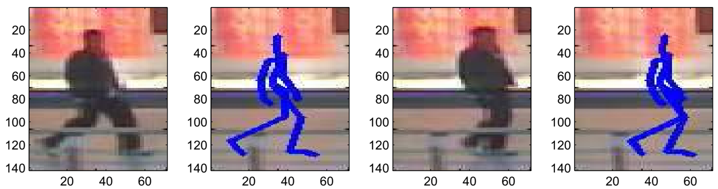




# Visual Tracking

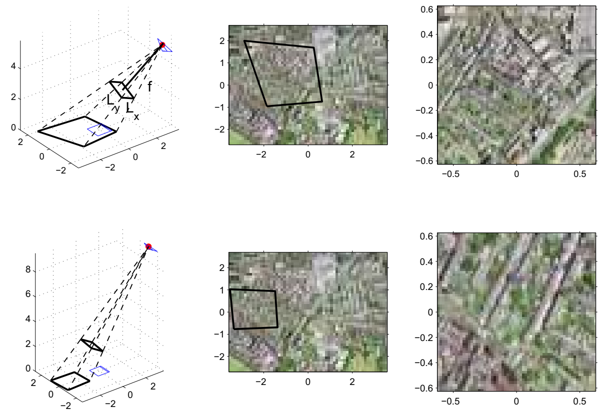


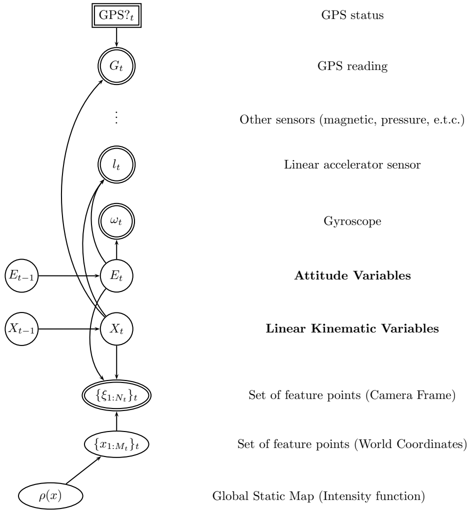




# Navigation, Robotics



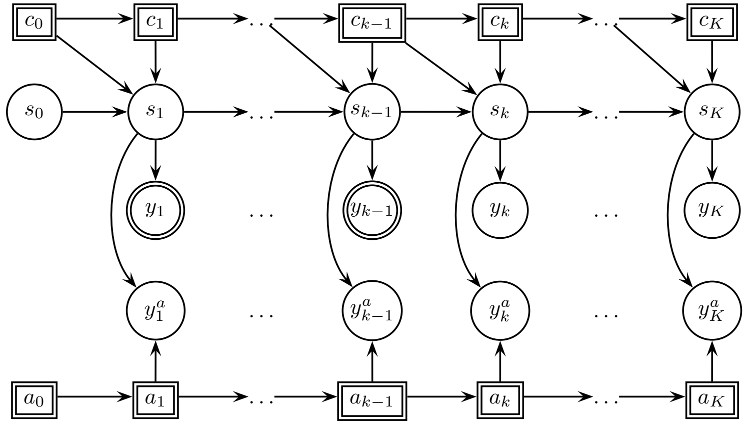
# Navigation, Robotics






# Computer Accompaniment

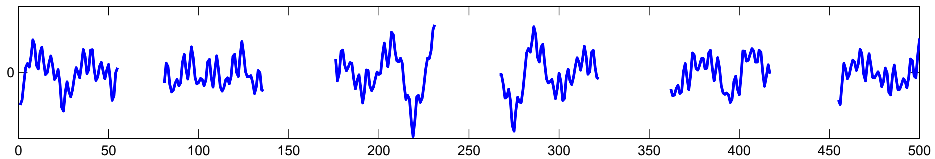
(Music Plus One, Raphael 2000 [18], Dannenberg and Raphael 2006)





# Audio Restoration

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

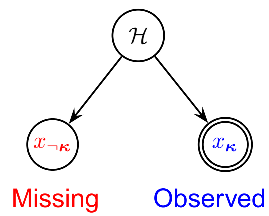


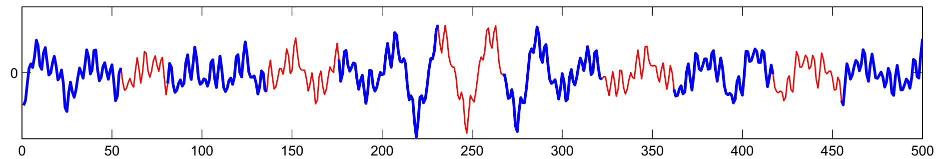


# Examples: Audio Restoration

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

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



## Restoration

(Cemgil and Godsill 2005 [5])

- Piano
  - Signal with missing samples (37%)
  - Reconstruction, 7.68 dB improvement
  - Original
- Trumpet
  - Signal with missing samples (37%)
  - Reconstruction, 7.10 dB improvement
  - Original



## Basic Building Blocks



## Probability Distributions : Exponential Family

- Following distributions are used often as elementary building blocks:
  - Gaussian
  - Gamma, Inverse Gamma, (Exponential, Chi-square, Wishart)
  - Dirichlet
  - Discrete (Categorical), Bernoulli, multinomial
- All of those distributions can be written as

$$p(x|\theta) = \exp\{\theta^\top \psi(x) - A(\theta)\}$$

$$A(\theta) = \log \int_{\mathcal{X}^n} dx \; \exp(\theta^\top \psi(x)) \quad \text{log-partition function}$$

$$\theta \quad \text{canonical parameters}$$

$$\psi(x) \quad \text{sufficient statistics}$$



## Example: Bernoulli

Binary (Bernoulli) random variable $c = \{0, 1\}$ with probability of sucsess $w$

$$p(c = 1|w) = w \qquad p(c = 0|w) = 1 - w$$

We write

$$\begin{aligned} p(c|w) &= w^c(1-w)^{1-c} \\ &= \exp\left(c \log w + (1-c)\log(1-w)\right) \\ &= \exp\left(\log(\frac{w}{1-w})c + \log(1-w)\right) \\ &= \mathcal{C}(c; w) \end{aligned}$$

$\mathcal{C}$ stays for categorical

## Example, Univariate Gaussian

The Gaussian distribution with mean $m$ and covariance $S$ has the form

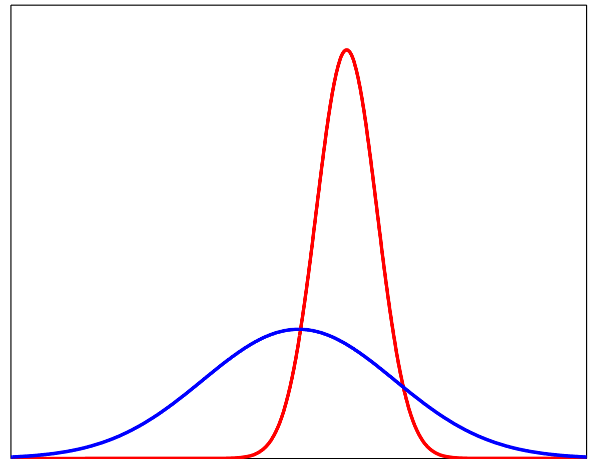
$$\begin{aligned}
\mathcal{N}(x; m, S) &= (2\pi S)^{-1/2} \exp\{-\frac{1}{2}(x-m)^2/S\} \\
&= \exp\{-\frac{1}{2}(x^2 + m^2 - 2xm)/S - \frac{1}{2}\log(2\pi S)\} \\
&= \exp\left\{\frac{m}{S}x - \frac{1}{2S}x^2 - \left(\frac{1}{2}\log(2\pi S) + \frac{1}{2S}m^2\right)\right\} \\
&= \exp\{\underbrace{\begin{pmatrix} m/S \\ -\frac{1}{2}/S \end{pmatrix}^\top}_{\theta} \underbrace{\begin{pmatrix} x \\ x^2 \end{pmatrix}}_{\psi(x)} - A(\theta)\}
\end{aligned}$$

Hence by matching coefficients we have

$$\exp\left\{-\tfrac{1}{2}Kx^2 + hx + g\right\} \Leftrightarrow S = K^{-1} \qquad m = K^{-1}h$$



## Example, Gaussian





## Example, Inverse Gamma

The inverse Gamma distribution with shape $a$ and scale $b$

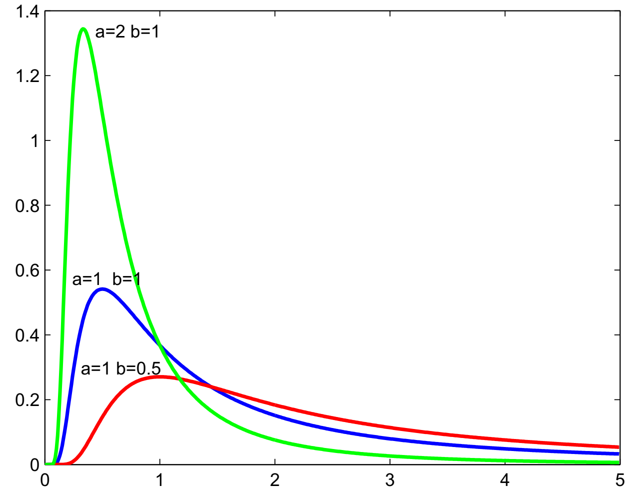
$$\begin{aligned}
\mathcal{IG}(r; a, b) &= \frac{1}{\Gamma(a)} \frac{r^{-(a+1)}}{b^a} \exp(-\frac{1}{br}) \\
&= \exp\left(-(a+1)\log r - \frac{1}{br} - \log\Gamma(a) - a\log b\right) \\
&= \exp\left(\begin{pmatrix} -(a+1) \\ -1/b \end{pmatrix}^\top \begin{pmatrix} \log r \\ 1/r \end{pmatrix} - \log\Gamma(a) - a\log b\right)
\end{aligned}$$

Hence by matching coefficients, we have

$$\exp\left\{\alpha\log r + \beta\frac{1}{r} + c\right\} \Leftrightarrow a = -\alpha - 1 \qquad b = -1/\beta$$



## Example, Inverse Gamma



## Example, Beta

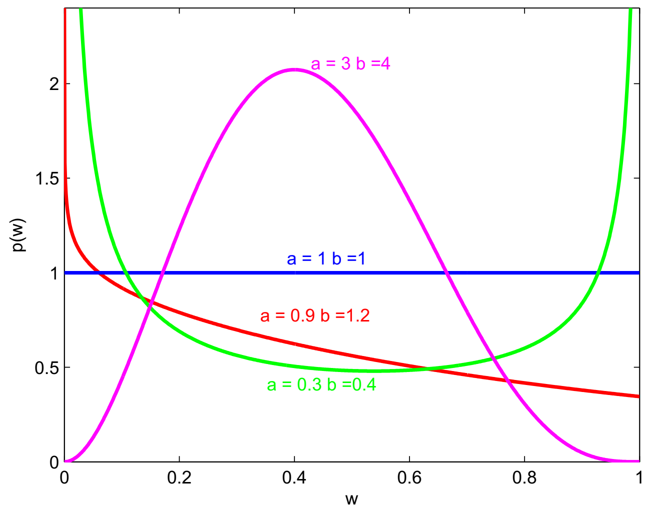
$$\begin{aligned}\mathcal{B}(w;a,b) &\equiv \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} w^{a-1}(1-w)^{b-1} \\ &= \exp\left((a-1)\log w + (b-1)\log(1-w) - A(a,b)\right) \\ &= \exp\left(\begin{pmatrix} a-1 & b-1 \end{pmatrix}\begin{pmatrix} \log w \\ \log(1-w)\end{pmatrix} - A(a,b)\right) \\ A(a,b) &= \log\Gamma(a) + \log\Gamma(b) - \log\Gamma(a+b)\end{aligned}$$

Mean :

$$\langle w \rangle_{\mathcal{B}} = a/(a+b)$$



## Example, Beta






## Conjugate priors: Posterior is in the same family as the prior.

Example: posterior inference for the probability of sucsess $w$ of a binary (Bernoulli) random variable $c$

$$\begin{aligned} p(c|w) &= \mathcal{C}(c;w) = \exp\left(c\log w + (1-c)\log(1-w)\right) \\ p(w) &= \mathcal{B}(w;a,b)\end{aligned}$$

$$\begin{aligned} p(w|c) &\propto p(c|w)p(w) \\ &\propto \exp\left(c\log w + (1-c)\log(1-w)\right) \\ &\quad \times \exp\left((a-1)\log w + (b-1)\log(1-w)\right) \\ &\propto \mathcal{B}(w;a+c,b+(1-c))\end{aligned}$$

$$p(w|c) = \begin{cases} \mathcal{B}(w;a+1,b) & c=1 \\ \mathcal{B}(w;a,b+1) & c=0\end{cases}$$



## Conjugate priors: Posterior is in the same family as the prior.

Example: posterior inference for the variance $R$ of a zero mean Gaussian.

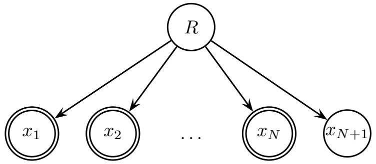
$$\begin{aligned} p(x|R) &= \mathcal{N}(x;0,R) \\ p(R) &= \mathcal{IG}(R;a,b)\end{aligned}$$

$$\begin{aligned} p(R|x) &\propto p(R)p(x|R) \\ &\propto \exp\left(-(a+1)\log R - (1/b)\frac{1}{R}\right)\exp\left(-(x^2/2)\frac{1}{R} - \frac{1}{2}\log R\right) \\ &= \exp\left(\begin{pmatrix} -(a+1+\frac{1}{2}) \\ -(1/b + x^2/2)\end{pmatrix}^\top \begin{pmatrix}\log R \\ 1/R\end{pmatrix}\right) \\ &\propto \mathcal{IG}(R; a+\frac{1}{2}, \frac{2}{x^2+2/b})\end{aligned}$$

Like the prior, this is an inverse-Gamma distribution.

## Conjugate priors: Posterior is in the same family as the prior.

Example: posterior inference of variance $R$ from $x_1, \dots, x_N$.

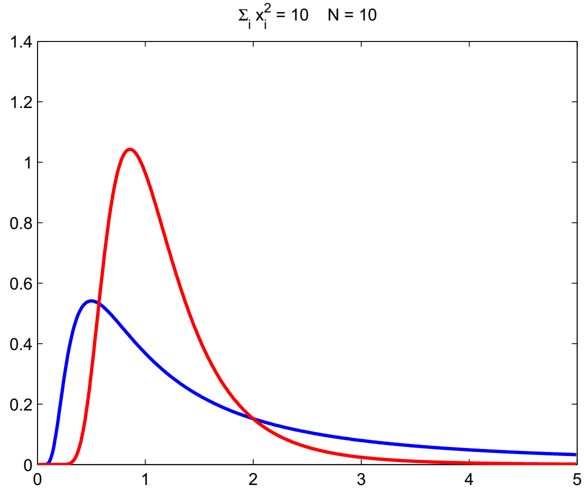



$$
\begin{aligned}
p(R|x) &\propto p(R) \prod_{i=1}^{N} p(x_i|R) \\
&\propto \exp\left(-(a+1)\log R - (1/b)\frac{1}{R}\right) \exp\left(-\left(\frac{1}{2}\sum_i x_i^2\right)\frac{1}{R} - \frac{N}{2}\log R\right) \\
&= \exp\left(\left(\begin{array}{c} -(a+1+\frac{N}{2}) \\ -(1/b+\frac{1}{2}\sum_i x_i^2) \end{array}\right)^\top \left(\begin{array}{c} \log R \\ 1/R \end{array}\right)\right) \propto \mathcal{IG}(R; a+\frac{N}{2}, \frac{2}{\sum_i x_i^2 + 2/b})
\end{aligned}
$$

Sufficient statistics are **additive**

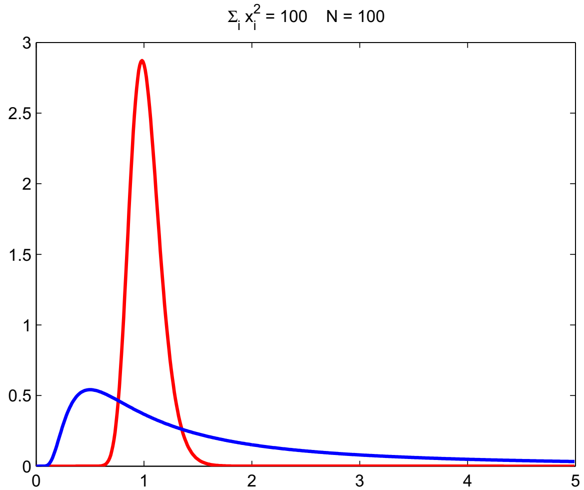


## Inverse Gamma, $\sum_i x_i^2 = 10 \quad N = 10$



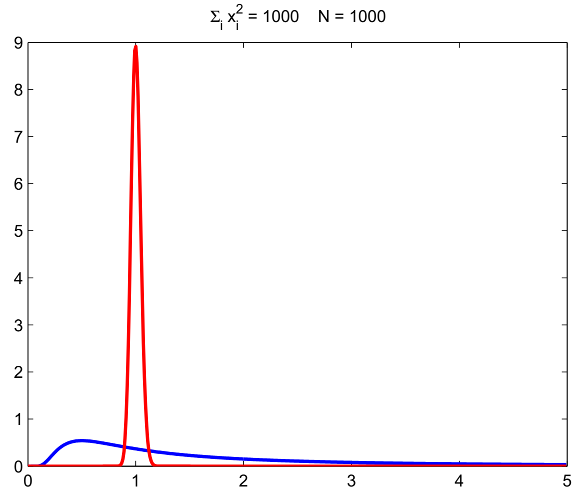



## Inverse Gamma, $\sum_i x_i^2 = 100 \quad N = 100$

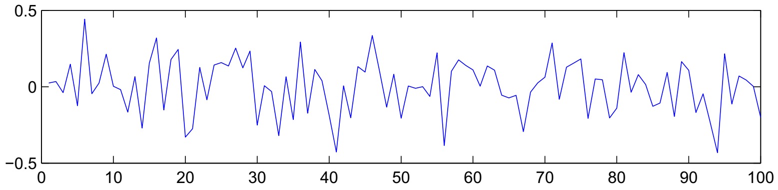





## Inverse Gamma, $\sum_i x_i^2 = 1000 \quad N = 1000$



# Example: AR(1) model




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

$\epsilon_k$ is i.i.d., zero mean and normal with variance $R$.

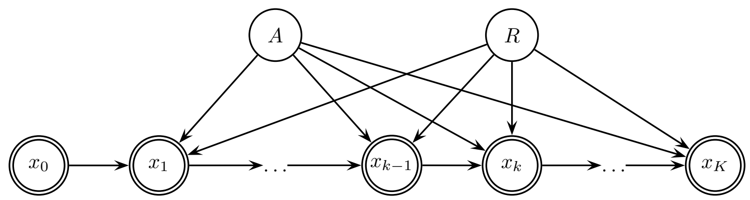
**Estimation problem**:

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



# AR(1) model, Generative Model notation

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




Observed variables are shown with double circles

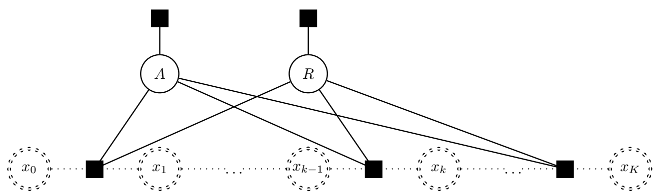


# AR(1) Model. Bayesian Posterior Inference

$$\begin{aligned} p(A, R | x_0, x_1, \dots, x_K) &\propto p(x_1, \dots, x_K | x_0, A, R) p(A, R) \\ \text{Posterior} &\propto \text{Likelihood} \times \text{Prior} \end{aligned}$$

Using the Markovian (conditional independence) structure we have

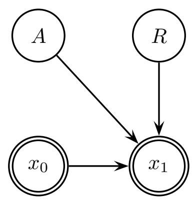
$$p(A, R | x_0, x_1, \dots, x_K) \quad \propto \quad \left( \prod_{k=1}^{K} p(x_k | x_{k-1}, A, R) \right) p(A) p(R)$$

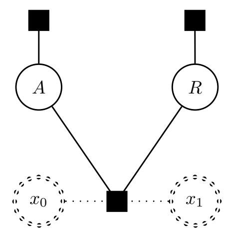





# Numerical Example

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

$$
\begin{aligned}
p(A, R|x_0, x_1) &\propto p(x_1|x_0, A, R)p(A)p(R) \\
&= \mathcal{N}(x_1; Ax_0, R)\mathcal{N}(A; 0, P)\mathcal{IG}(R; \nu, \beta/\nu) \\
&\propto \exp\left(-\frac{1}{2}\frac{x_1^2}{R} + x_0 x_1 \frac{A}{R} - \frac{1}{2}\frac{x_0^2 A^2}{R} - \frac{1}{2}\log 2\pi R\right) \\
&\quad \exp\left(-\frac{1}{2}\frac{A^2}{P}\right)\exp\left(-(\nu+1)\log R - \frac{\nu}{\beta}\frac{1}{R}\right)
\end{aligned}
$$

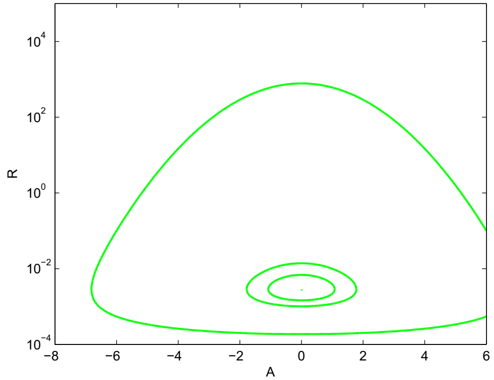
This posterior has a nonstandard form

$$
\exp\left(\alpha_1 \frac{1}{R} + \alpha_2 \frac{A}{R} + \alpha_3 \frac{A^2}{R} + \alpha_4 \log R + \alpha_5 A^2\right)
$$



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

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

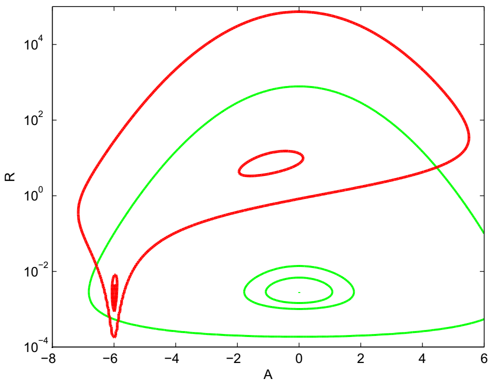



$A \sim \mathcal{N}(A; 0, 1.2) \qquad R \sim \mathcal{IG}(R; 0.4, 250)$

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 point estimates such as ML or MAP are 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
- Ambiguous data usually leads to a multimodal posterior, each mode corresponding to one possible explanation

## Remarks

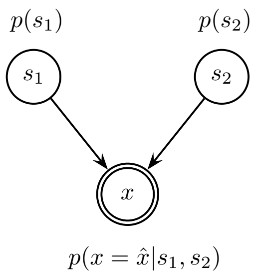
- *A-priori* independent variables often become dependent *a-posteriori* (“Explaining away”)
- The difficulty of an inference problem depends, among others, upon the particular “parameter regime” and observed data sequence

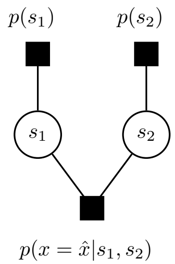


## Approximate Inference



## A Toy Model

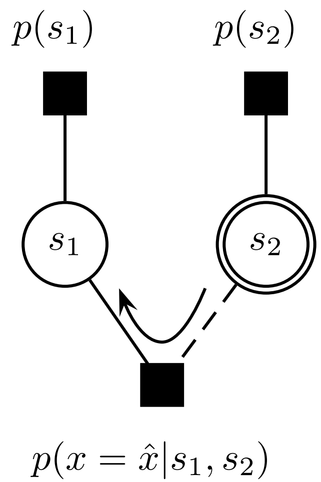


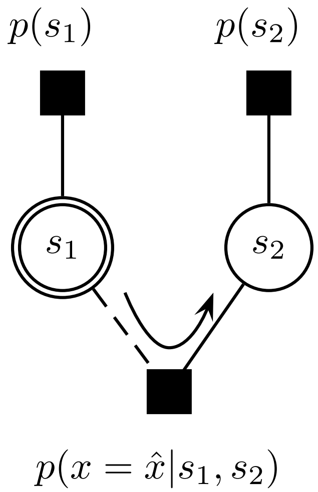

$$
\begin{aligned}
s_1 &\sim p(s_1) = \mathcal{N}(s_1; \mu_1, P_1) \\
s_2 &\sim p(s_2) = \mathcal{N}(s_2; \mu_2, P_2) \\
x|s_1, s_2 &\sim p(x|s_1, s_2) = \mathcal{N}(x; s_1 + s_2, R)
\end{aligned}
$$



## Gibbs Sampling

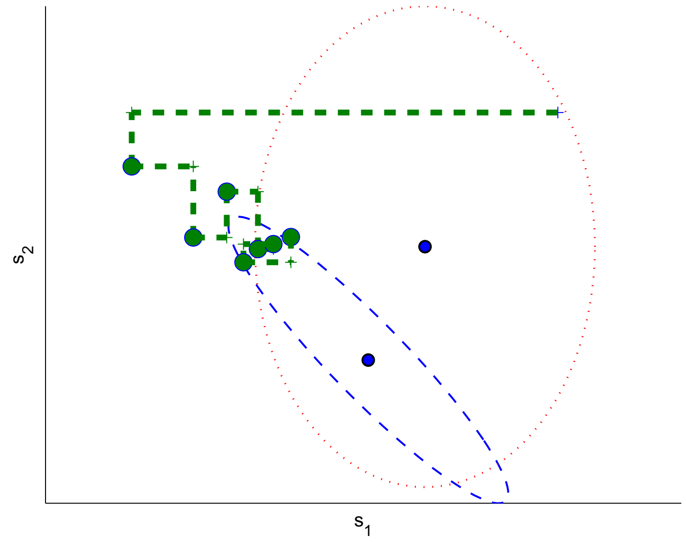


## Gibbs Sampling



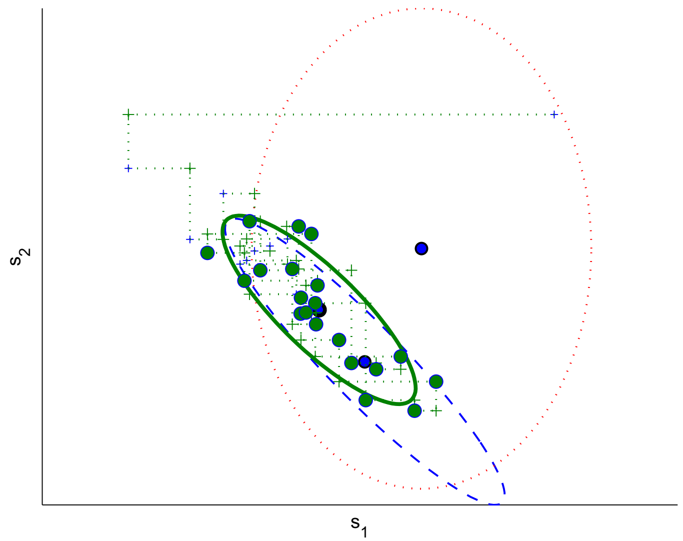



## Gibbs Sampling



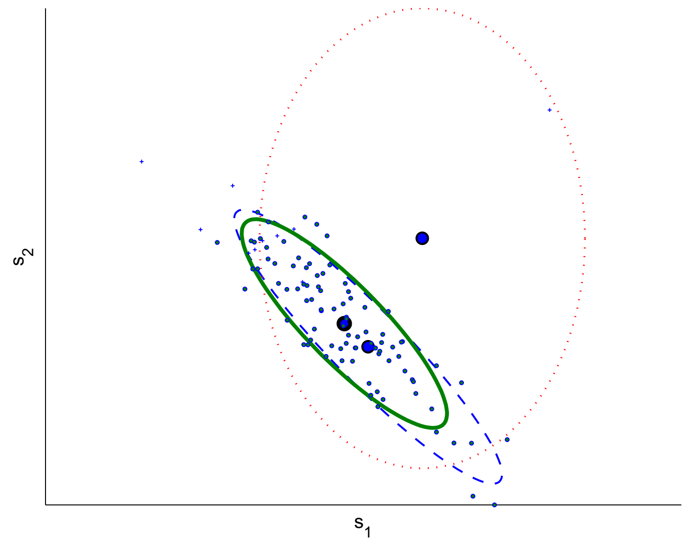



## Gibbs Sampling, $t = 20$

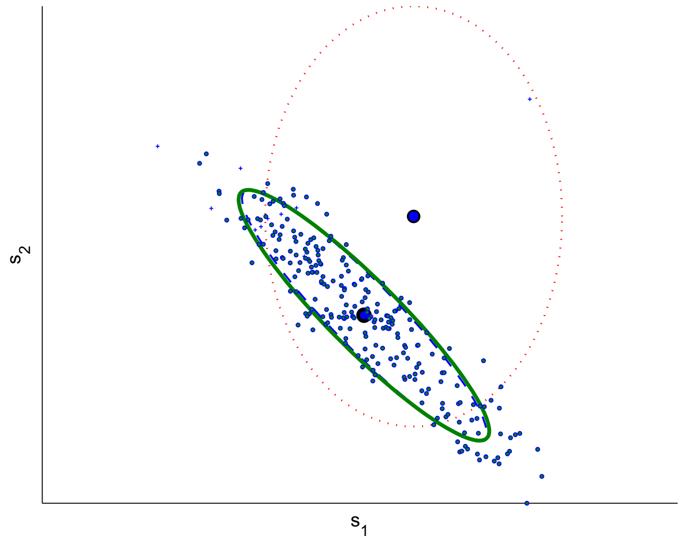





## Gibbs Sampling, $t = 100$



## Gibbs Sampling, $t = 250$






## Gibbs Sampling

- 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!
- The sequence of samples are obtained from a Markov chain, hence the name MCMC



## 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

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

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

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

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

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

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



## The form of the mean field solution

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

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** [16]. The mean field equations **monotonically** increase this bound. Good for assessing convergence and debugging computer code.



## The form of the solution

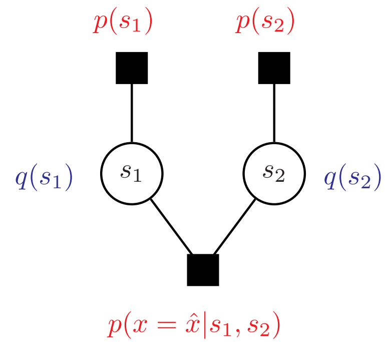
- No direct analytical solution
- We obtain fixed point equations in closed form

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

Note the nice symmetry

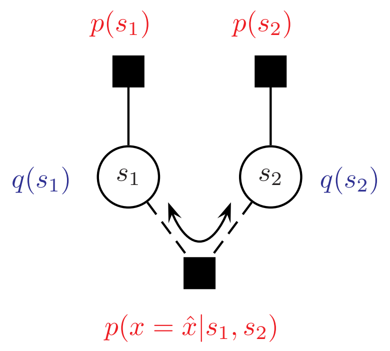


## Variational Message Passing on a Factor Graph




- **Factor nodes**: Factor potentials (local functions) defining the posterior $\mathcal{P}$.
- **Variable nodes**: Now, think of them as “factors” of the approximating distribution $\mathcal{Q}$. (Caution – non standard interpretation!)

## Fixed Point Iteration

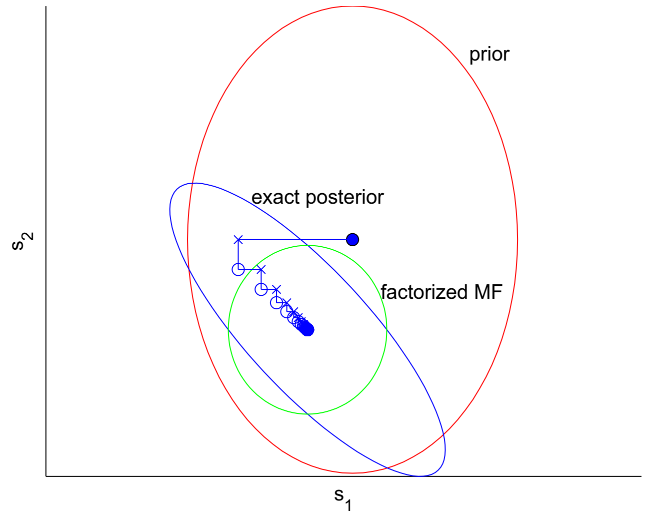



$$\log q(s_1) \quad \leftarrow \quad \log p(s_1) + \langle \log p(x = \hat{x}|s_1, s_2) \rangle_{q(s_2)}$$

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



## VB Convergence






## Direct Link to Expectation-Maximisation (EM) [12]

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

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

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



## Iterated Conditional Modes (ICM) [2, 11]

If we choose both distributions degenerate, i.e.

$$\tilde{q}(s_1) \quad = \quad \delta(s_1 - \tilde{m}_1)$$
$$\tilde{q}(s_2) \quad = \quad \delta(s_2 - \tilde{m}_2)$$

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 $\phi(s_1, s_2)$.

$$\tilde{m}_1 \quad = \quad \operatorname*{argmax}_{s_1} \phi(s_1, s_2 = \tilde{m}_2)$$
$$\tilde{m}_2 \quad = \quad \operatorname*{argmax}_{s_2} \phi(s_1 = \tilde{m}_1, s_2)$$

## 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.

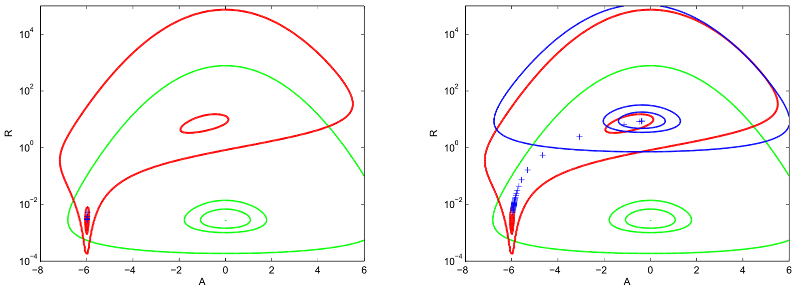



Figure 1: Left, ICM, Right VB. EM is similar to ICM in this AR(1) example.

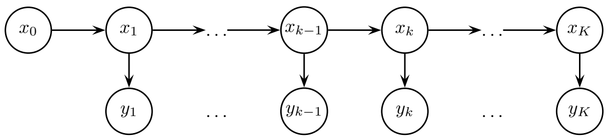


## Models and Applications



## Time series models and Inference, Terminology

In music signal processing and machine learning many phenomena are modelled by dynamical models



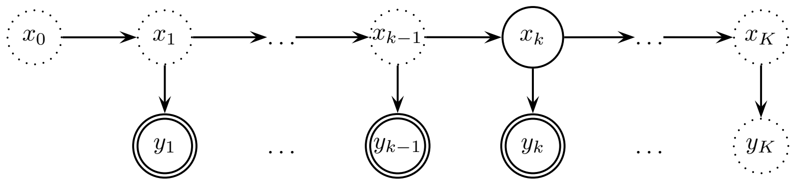

$$
\begin{aligned}
x_k &\sim p(x_k|x_{k-1}) && \text{Transition Model} \\
y_k &\sim p(y_k|x_k) && \text{Observation Model}
\end{aligned}
$$

- $x$ is the latent state (tempo, pitch, velocity, attitude, class label, ...)
- $y$ are observations (samples, onsets, sensor reading, pixels, features, ... )
- In a full Bayesian setting, $x$ includes unknown model parameters



## Online Inference, Terminology

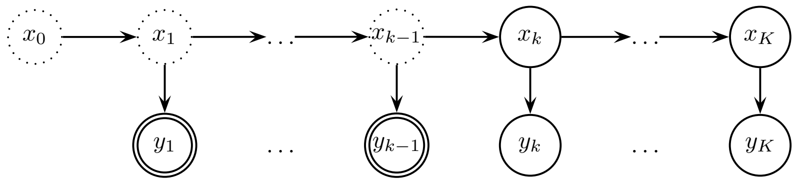
- **Filtering:** $p(x_k|y_{1:k})$
  - Distribution of current state given all past information
  - Realtime/Online/Sequential Processing




- Potentially confusing misnomer:
  - More general than “digital filtering” (convolution) in DSP – but algoritmically related for some models (KFM)

## Online Inference, Terminology

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

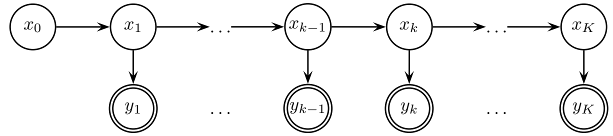


- Accompaniment, Tracking, Restoration

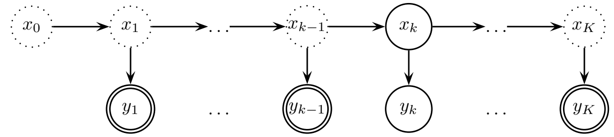


## Offline Inference, Terminology

- **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



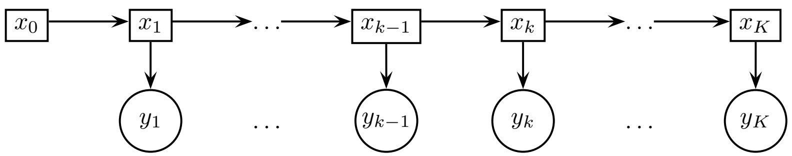
- **Interpolation** $p(y_k, x_k|y_{1:k-1}, y_{k+1:K})$
  fill in lost observations given past and future





## Hidden Markov Model [17]

- Mixture model evolving in time

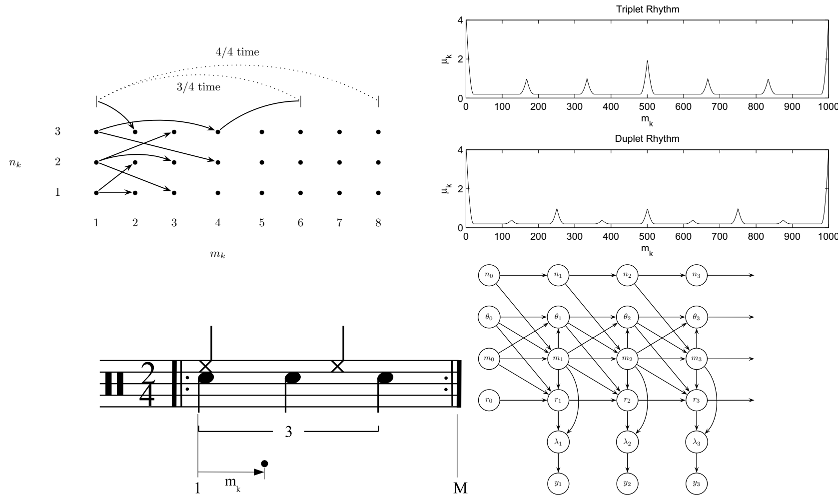


- Observations $y_k$ are continuous or discrete
- Latent variables $x_k$ are discrete
  - Represents the fading memory of the process
- Exact inference possible if $x_k$ has a “small” number of states

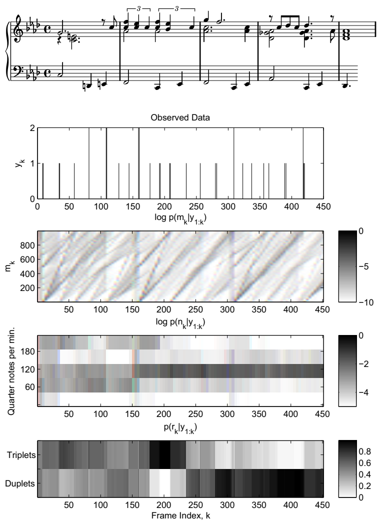


## Tempo, Rhythm, Meter analysis

Bar Pointer Model (Whiteley, Cemgil, Godsill 2006)

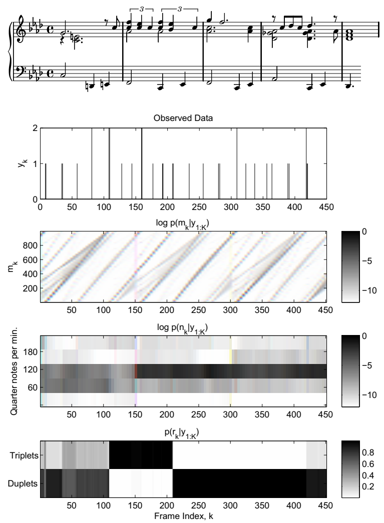


# Filtering





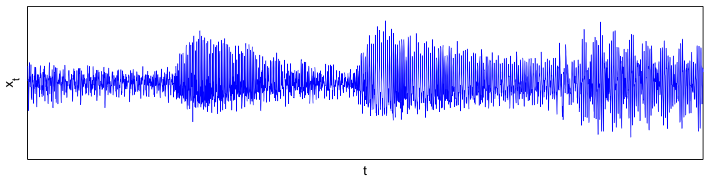

# Smoothing





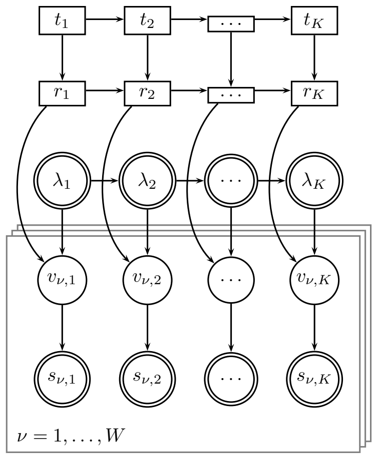

# Score-Performance matching (Peeling, Cemgil, Godsill)

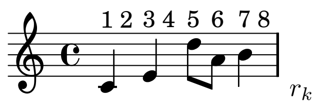
- Given a musical score, associate note events with the audio

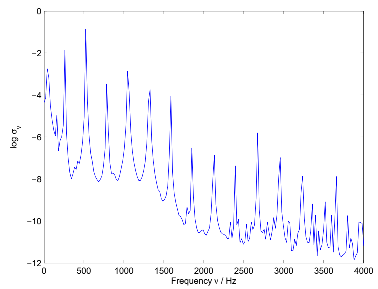





# Score-Performance matching - Graphical Model

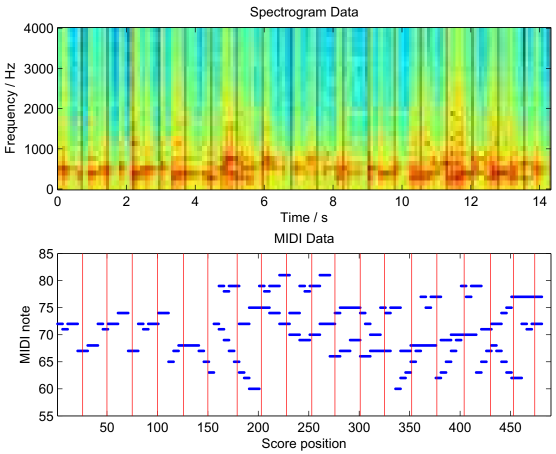


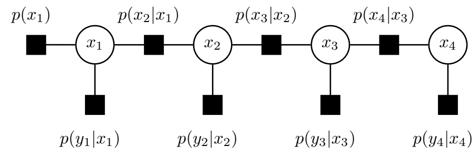

$$v_{\nu,\tau} \sim \mathcal{IG}(v_{\nu,\tau}; a, 1/(a\lambda\sigma_\nu(r_\tau)))$$

# Score-Performance matching






# Exact Inference in HMM, Forward/Backward Algorithm




- Forward Pass

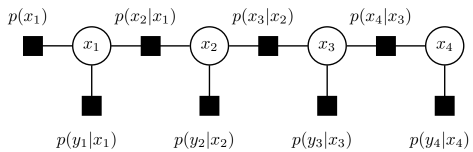
$$\begin{aligned} p(y_{1:K}) &= \sum_{x_{1:K}} p(y_{1:K}|x_{1:K})p(x_{1:K}) \\ &= \underbrace{\sum_{x_K} p(y_T|x_K) \sum_{x_{K-1}} p(x_K|x_{K-1})}_{\alpha_K} \cdots \sum_{x_2} p(x_3|x_2) \underbrace{p(y_2|x_2) \overbrace{\sum_{x_1} p(x_2|x_1)}^{\alpha_{2|1}}}_{\alpha_2} \underbrace{p(y_1|x_1) \overbrace{p(x_1)}^{\alpha_{1|0}}}_{\alpha_1} \end{aligned}$$

- Backward Pass

$$p(y_{1:K}) = \sum_{x_1} p(x_1)p(y_1|x_1) \dots \underbrace{\sum_{x_{K-1}} p(x_{K-1}|x_{K-2})p(y_{K-1}|x_{K-1})}_{\beta_{K-2}} \underbrace{\sum_{x_K} p(x_K|x_{K-1})p(y_K|x_K)}_{\beta_{K-1}} \underbrace{\mathbf{1}}_{\beta_K}$$



# Exact Inference in HMM, Viterbi Algorithm




- Merely replace sum by max, equivalent to dynamic programming
- Forward Pass

$$\begin{aligned} p(y_{1:K}|x^*_{1:K}) &= \max_{x_{1:K}} p(y_{1:K}|x_{1:K})p(x_{1:K}) \\ &= \underbrace{\max_{x_K} p(y_T|x_K) \max_{x_{K-1}} p(x_K|x_{K-1})}_{\alpha_K} \dots \max_{x_2} p(x_3|x_2) \underbrace{p(y_2|x_2) \overbrace{\max_{x_1} p(x_2|x_1)}^{\alpha_{2|1}}}_{\alpha_2} \underbrace{p(y_1|x_1) \overbrace{p(x_1)}^{\alpha_{1|0}}}_{\alpha_1} \end{aligned}$$

- Backward Pass

$$p(y_{1:K}|x^*_{1:K}) = \max_{x_1} p(x_1)p(y_1|x_1) \dots \underbrace{\max_{x_{K-1}} p(x_{K-1}|x_{K-2})p(y_{K-1}|x_{K-1})}_{\beta_{K-2}} \underbrace{\max_{x_K} p(x_K|x_{K-1})p(y_K|x_K)}_{\beta_{K-1}} \underbrace{\mathbf{1}}_{\beta_K}$$

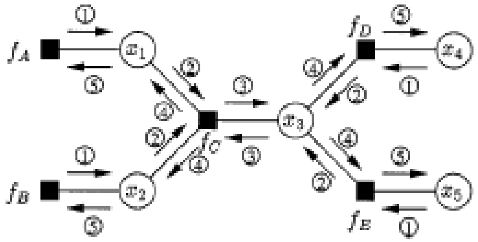


# Exact Inference on general factor graphs

- When the factor graph is a tree, one can define a local message propagation
  - If factor graph is not a tree, one can always do this by clustering nodes together
- Sum-product
  - Generalises Forward/Backward
  - Rule:
    "The message sent from a node $v$ on an edge $e$ is the product of the local function at $v$ (or the unit function if is a variable node) with all messages received at $v$ on edges other than $e$, summarized for the variable associated with $e$."
- Max-product
  - Generalises Viterbi

Look at the seminal tutorial paper by Kschischang, Frey and Loeliger [14] on factor graphs.

## Exact Inference on general factor graphs




**variable to local function:**

$$\mu_{x \to f}(x) = \prod_{h \in n(x) \setminus \{f\}} \mu_{h \to x}(x)$$

**local function to variable:**

$$\mu_{f \to x}(x) = \sum_{\sim\{x\}} \left( f(X) \prod_{y \in n(f) \setminus \{x\}} \mu_{y \to f}(y) \right)$$



## Kalman Filter Models, Linear Dynamical Systems

- The latent variables $s_k$ and observations $y_k$ are continuous
- The transition and observations models are linear
  - Example: a point moving on the real line
  - A deterministic dynamical system with two state variables

$$\mathbf{s}_k = \begin{pmatrix} \text{position} \\ \text{velocity} \end{pmatrix}_k = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \mathbf{s}_{k-1} = \mathbf{A}\mathbf{s}_{k-1}$$

$$y_k = \text{position}_k = \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{s}_k = \mathbf{C}\mathbf{s}_k$$



## Tracking

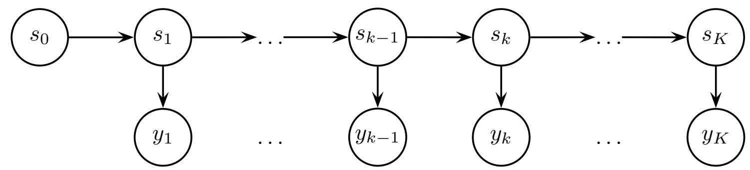
- We allow random (unknown) accelerations

$$\begin{aligned} \mathbf{s}_k &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \mathbf{s}_{k-1} + \epsilon_k \\ &= \mathbf{A}\mathbf{s}_{k-1} + \epsilon_k \end{aligned}$$

$$\begin{aligned} y_k &= \begin{pmatrix} 1 & 0 \end{pmatrix} \mathbf{s}_k + \nu_k \\ &= \mathbf{C}\mathbf{s}_k + \nu_k \end{aligned}$$



## Tracking

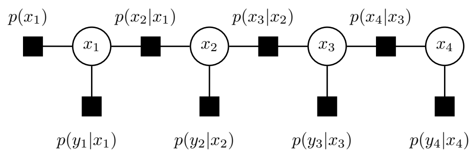



- In generative model notation

$$\begin{aligned} \mathbf{s}_k &\sim \mathcal{N}(\mathbf{s}_k; \mathbf{A}\mathbf{s}_{k-1}, Q) \\ y_k &\sim \mathcal{N}(y_k; \mathbf{C}\mathbf{s}_k, R) \end{aligned}$$

- Tracking = estimating the latent state of the system = Kalman filtering

## Kalman Filtering and Smoothing (two filter formulation)




- Forward Pass

$$p(y_{1:K}) = \underbrace{\int_{x_K} p(y_T|x_K) \int_{x_{K-1}} p(x_K|x_{K-1})}_{\alpha_K} \dots \int_{x_2} p(x_3|x_2) \underbrace{p(y_2|x_2) \overbrace{\int_{x_1} p(x_2|x_1)}^{\alpha_{2|1}}}_{\alpha_2} \underbrace{p(y_1|x_1) \overbrace{p(x_1)}^{\alpha_{1|0}}}_{\alpha_1}$$

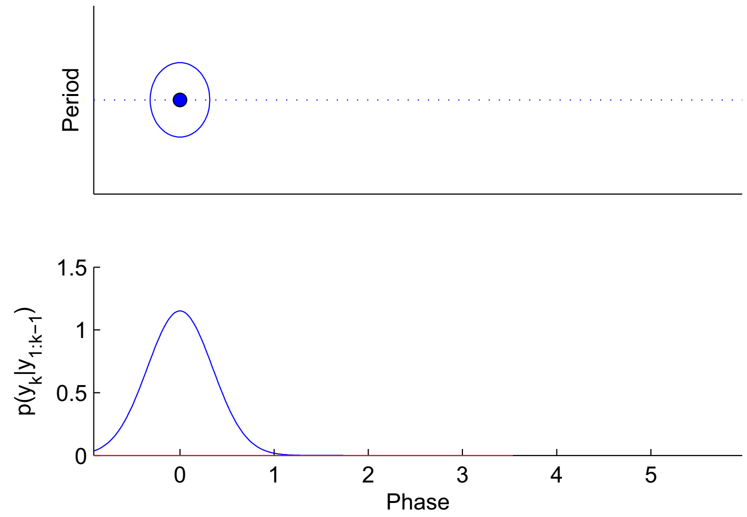
- Backward Pass

$$p(y_{1:K}) = \int_{x_1} p(x_1)p(y_1|x_1) \dots \underbrace{\int_{x_{K-1}} p(x_{K-1}|x_{K-2})p(y_{K-1}|x_{K-1})}_{\beta_{K-2}} \underbrace{\int_{x_K} p(x_K|x_{K-1})p(y_K|x_K)}_{\beta_{K-1}} \underbrace{\mathbf{1}}_{\beta_K}$$

- Replace summation by integration

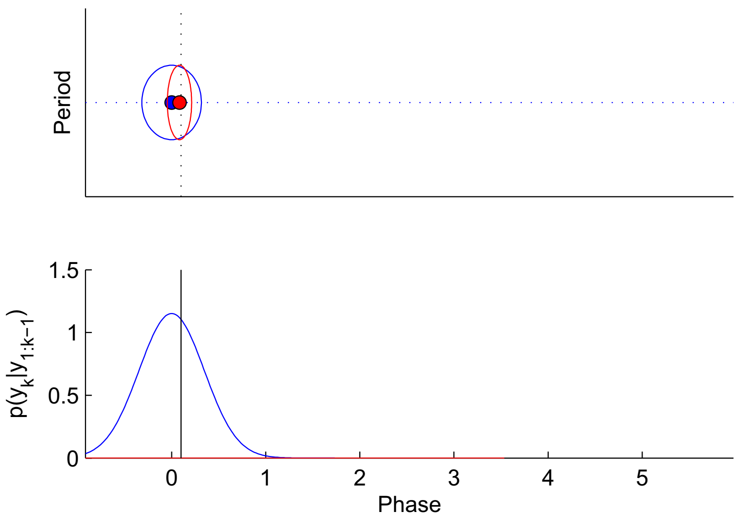


$$p(s_1)$$



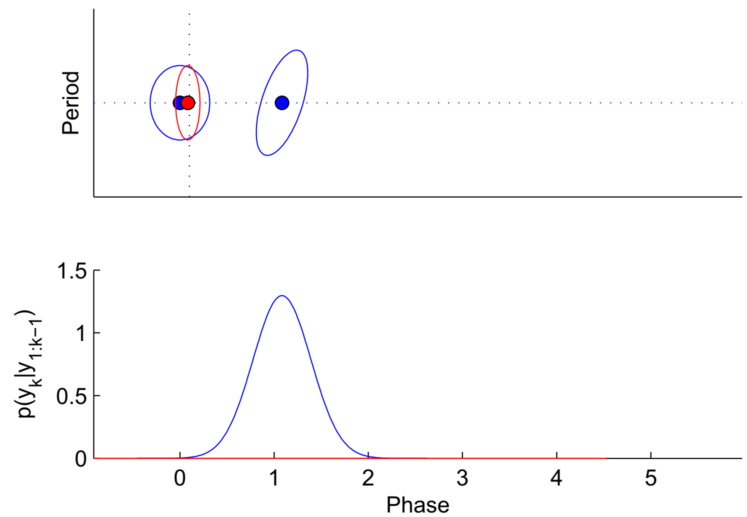



$$p(y_1|s_1)p(s_1)$$

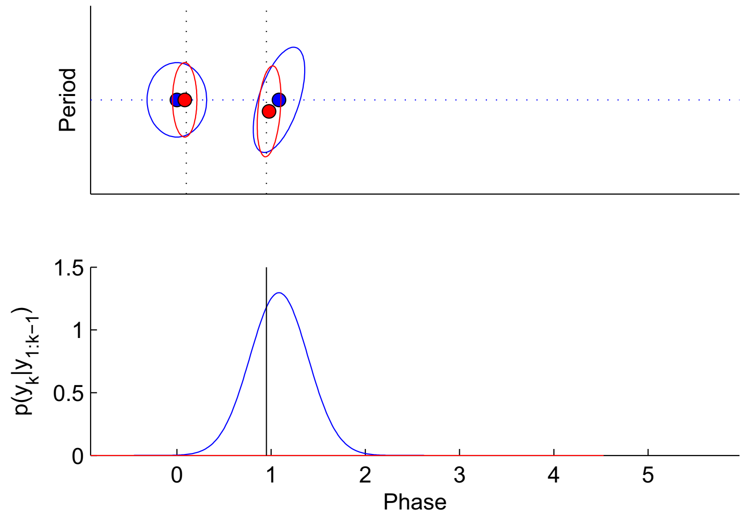





$$p(s_2|y_1) \propto \int ds_1 p(s_2|s_1)p(y_1|s_1)p(s_1)$$

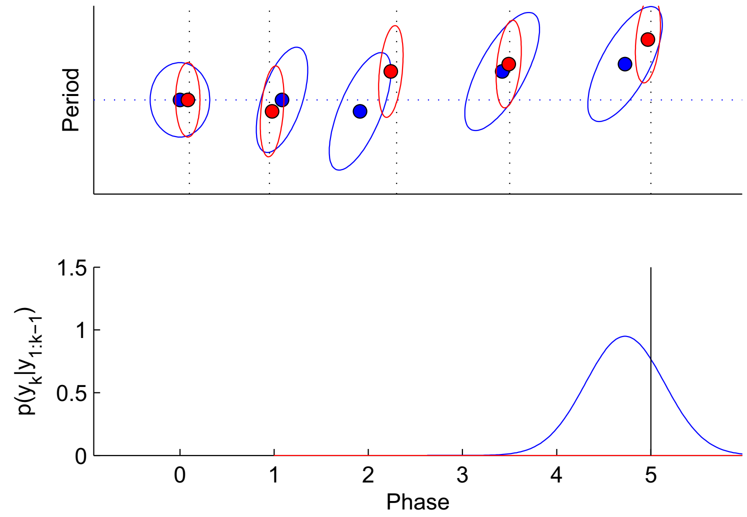


$p(y_2|s_2)p(s_2|y_1)$





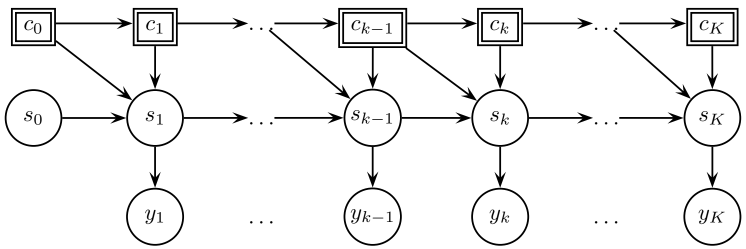

$p(s_5|y_{1:5})$






## Computer Accompaniment

(Music Plus One, Raphael 2000 [18], Dannenberg and Raphael 2006)




- $c_k$ are score positions of notes of the soloist and $l_k = c_k - c_{k-1}$

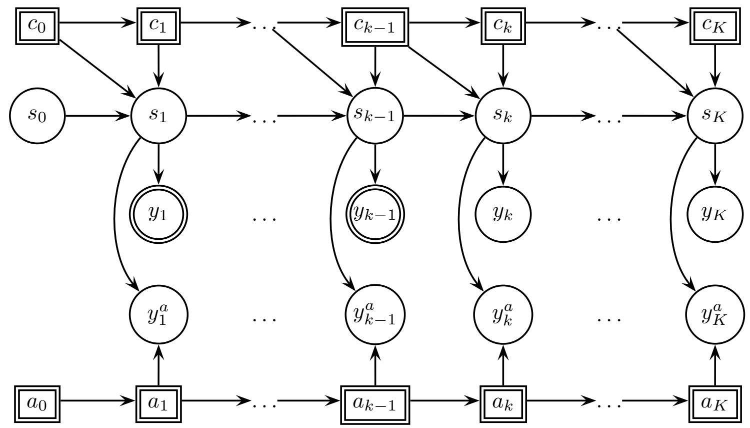
$$\mathbf{s}_k = \begin{pmatrix} 1 & l_k \\ 0 & 1 \end{pmatrix} \mathbf{s}_{k-1} + \epsilon_k = \mathbf{A}_k \mathbf{s}_{k-1} + \epsilon_k \qquad y_k = C\mathbf{s}_k + \nu_k$$

$$\epsilon_k \sim \mathcal{N}(\epsilon; 0, Q_k)$$

$$\nu_k \sim \mathcal{N}(\nu; m_k, R_k)$$ (note $k$ dependent mean and variance!)

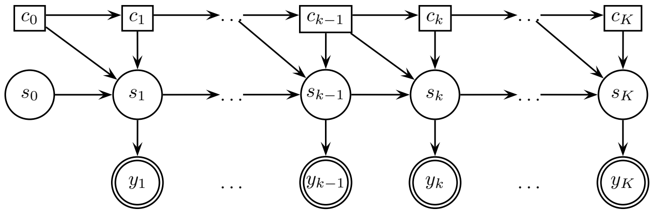


## Music Plus One




- Note that this is ruthless simplification, see Chris Raphaels' papers...

# Switching State Space models

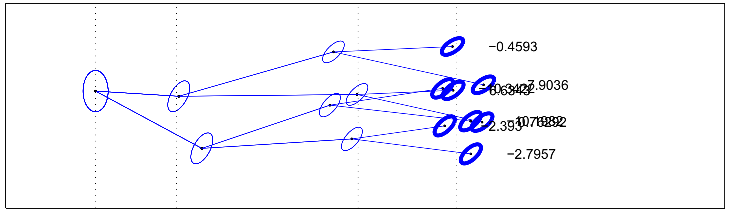


- We introduce latent switch variables to switch between different transition and observation models
- Powerful framework for modelling nonstationary processes and nonlinear dynamical systems



# Inference in Switching State Space models

- Unlike HMM's or KFM's, summing over $c_k$ does not simplify the filtering density.
- Number of Gaussian kernels to represent exact filtering density $p(c_k, s_k|y_{1:k})$ increases exponentially

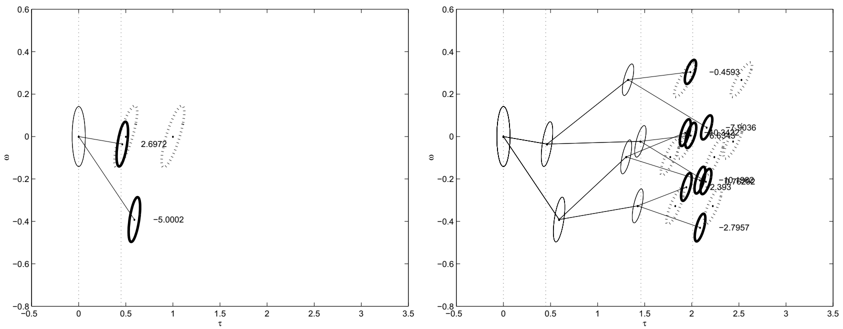



- Bad news: exact inference problem is shown to be NP hard



# Example

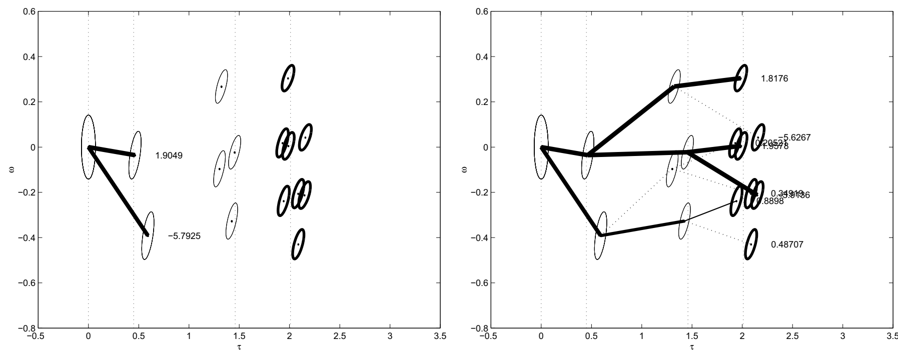
Suppose that a score can consist of only two notes:



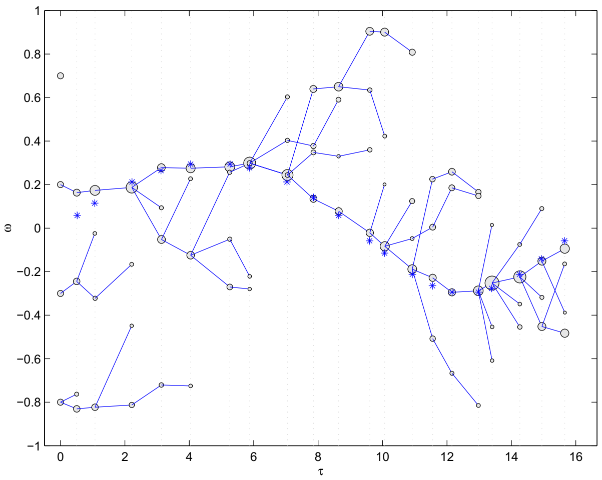



# Sequential Monte Carlo (Particle Filtering)

- Main idea: Select a branch to expand with a probability propotional to the evidence



# Particle Filtering for tracking






# Sequential Monte Carlo

- This variant is known as Mixture Kalman Filter or Rao-Blackwellized Particle filter (Chen and Liu 2001 [9], Cemgil 2002 [6], Hainsworth and MacLeod 2003)
- (For this model) algorithmically similar to Breadth first search/Multi Hypothesis Tracking/Genetic algorithms
- Generic tool for inference with a rich background theory (Doucet, et. al. 2001, Del Moral, "Feynman-Kac Formulae", 2005)
- Many applications in various fields
  - Robotics, Navigation, Econometrics,...

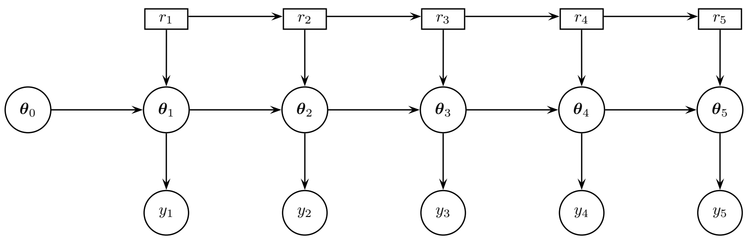


# Changepoint models

$r_k \sim p(r_k|r_{k-1})$ Indicators $\in \{\text{new}, \text{reg}\}$

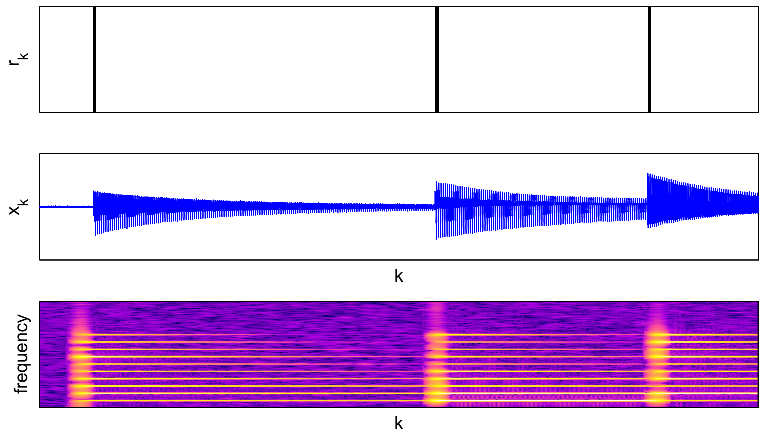
$\theta_k \sim [r_k = \text{reg}] \underbrace{f(\theta_k|\theta_{k-1})}_{\text{Transition}} + [r_k = \text{new}] \underbrace{\pi(\theta_k)}_{\text{Reinitialization}}$ Latent State

$y_k \sim p(y_k|\theta_k)$ Observations





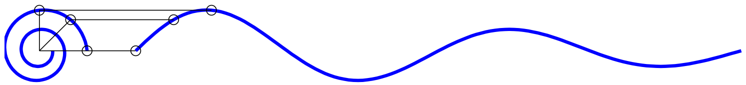

# Example: Single Key, Onsets




- Each changepoint denotes the onset of a new audio event

## Dynamic Harmonic Model (Cemgil et. al. 2005, 2006) [4, 7]

$$
\begin{aligned}
r_k | r_{k-1} &\sim p(r_k | r_{k-1}) \\
s_k | s_{k-1}, r_k &\sim \underbrace{[r_k = 0]\mathcal{N}(A s_{k-1}, Q)}_{\text{reg}} + \underbrace{[r_k = 1]\mathcal{N}(0, S)}_{\text{new}} \\
y_k | s_k &\sim \mathcal{N}(C s_k, R)
\end{aligned}
$$



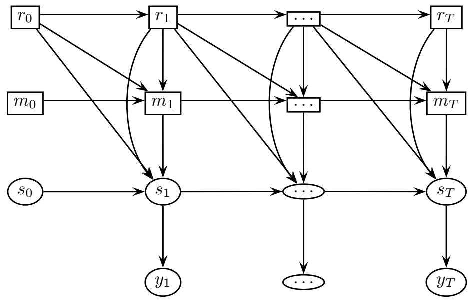
$$
A = \begin{pmatrix} G_\omega & & & \\ & G_\omega^2 & & \\ & & \ddots & \\ & & & G_\omega^H \end{pmatrix}^N \qquad G_\omega = \rho_k \begin{pmatrix} \cos(\omega) & -\sin(\omega) \\ \sin(\omega) & \cos(\omega) \end{pmatrix}
$$

damping factor $0 < \rho_k < 1$, framelength $N$ and damped sinusoidal basis matrix $C$ of size $N \times 2H$



## Monophonic model [7]

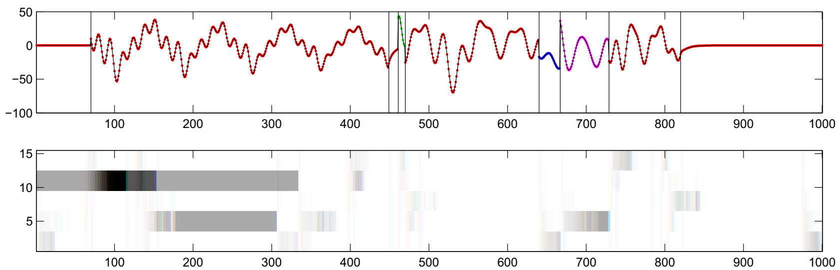
- We introduce a pitch label indicator $m$
- At each time $k$, the process can be in one of the {"mute", "sound"} $\times M$ states.






## Monophonic Pitch Tracking

Monophonic Pitch Tracking = Online estimation (filtering) of $p(r_k, m_k | y_{1:k})$.

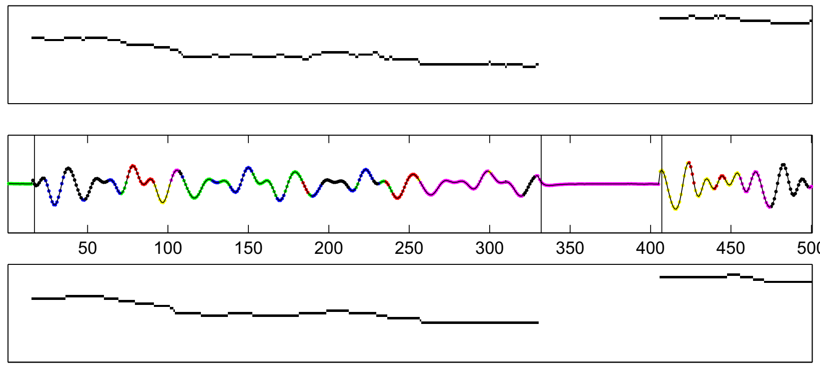



- If pitch is constant exact inference is possible



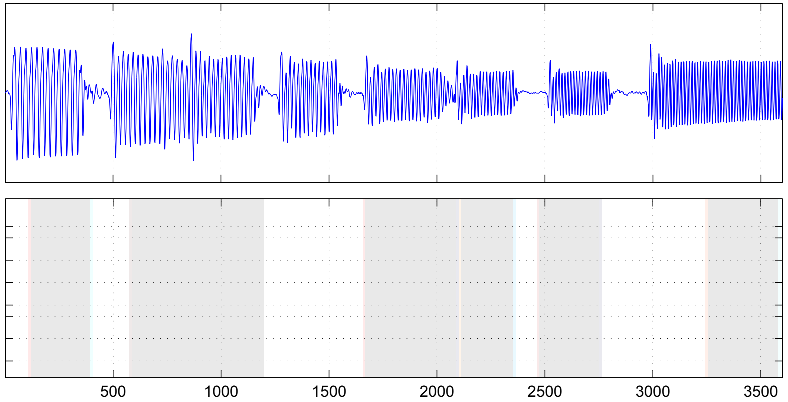
## Tracking Pitch Variations

- Allow $m$ to change with $k$. We take a fine grid Piano-roll, e.g. $\mathcal{Q} = 2^{1/128}$




- Intractable, need to run a particle filter

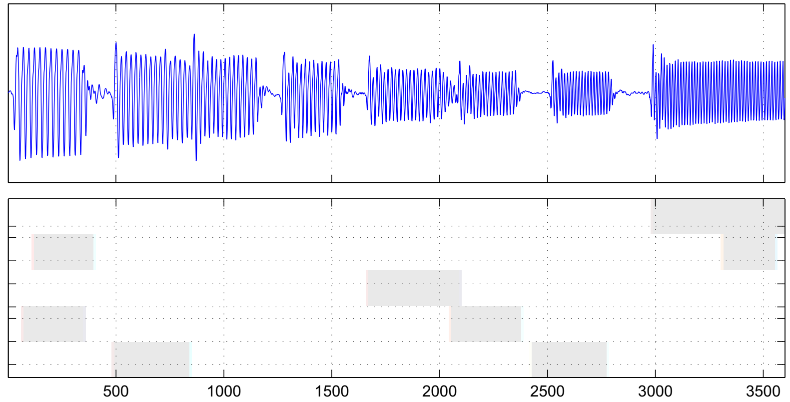
## Real Data Results




Top: F major scale played on an electric bass.
Bottom: Estimated MAP configuration $(r, m)_{1:T}$.



## Real Data Results

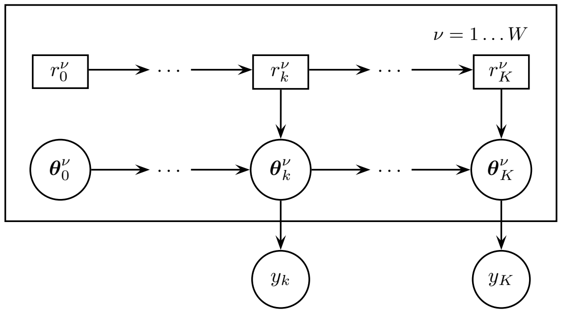



A finer analysis with $\mathcal{Q} = 2^{1/48}$ reveals that the 5'th and 7'th degree of the scale are intonated slightly low.



## Polyphony: Factorial Dynamic Harmonic Model [4]

$$
\begin{aligned}
r_{0,\nu} &\sim \mathcal{C}(r_{0,\nu}; \pi_{0,\nu}) \\
\theta_{0,\nu} &\sim \mathcal{N}(\theta_{0,\nu}; \mu_\nu, P_\nu) \\
r_{k,\nu}|r_{k-1,\nu} &\sim \mathcal{C}(r_{k,\nu}; \pi_\nu(r_{t-1,\nu})) && \text{Changepoint indicator} \\
\theta_{k,\nu}|\theta_{k-1,\nu} &\sim \mathcal{N}(\theta_{k,\nu}; A_\nu(r_k)\theta_{k-1,\nu}, Q_\nu(r_k)) && \text{Latent state} \\
y_k|\theta_{k,1:W} &\sim \mathcal{N}(y_k; C_k\theta_{k,1:W}, R) && \text{Observation}
\end{aligned}
$$

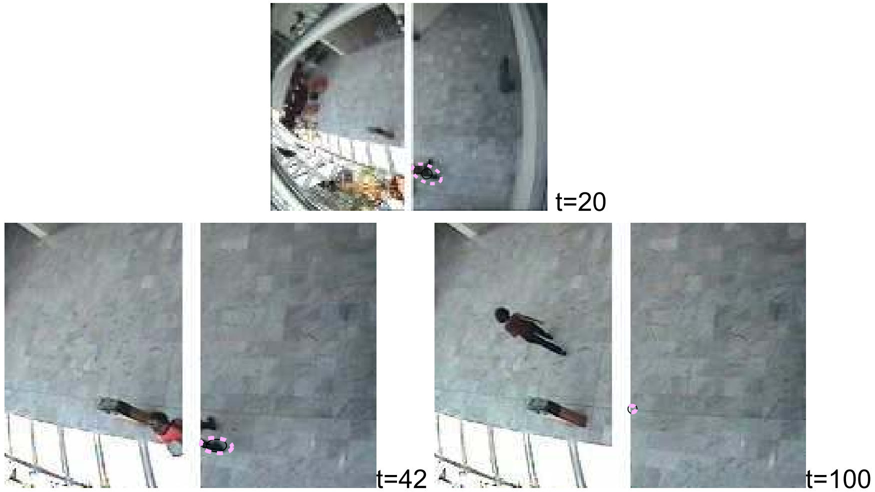





## Visual Tracking

(Video1) (Video2) (Video3)

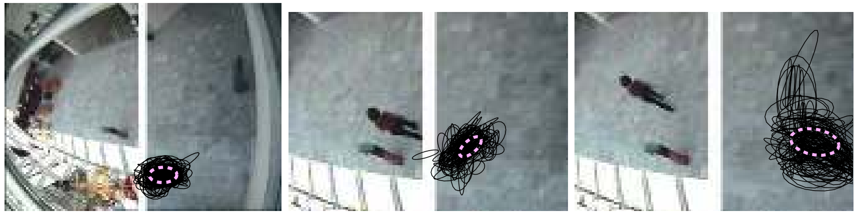
## Visual Tracking – Multimodal Posteriors



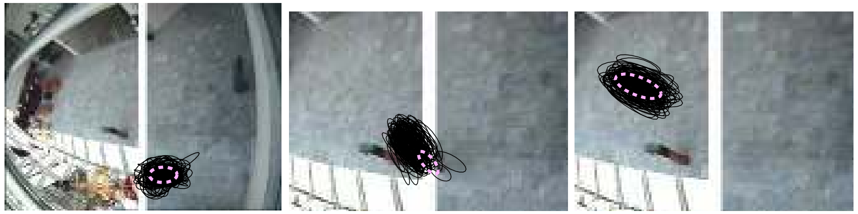

The Kalman Filter looses track due to occlusion



## Visual Tracking – Multimodal Posteriors



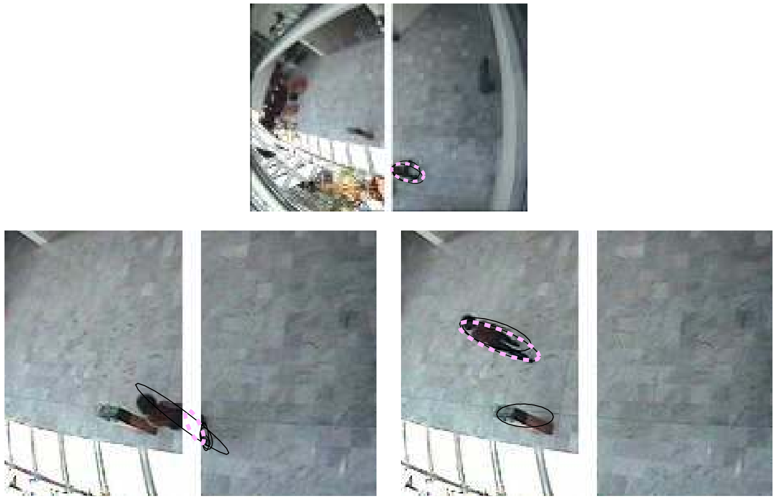
Particle Filter with poorly designed proposal



Particle Filter with better proposal



## Visual Tracking – Multimodal Posteriors

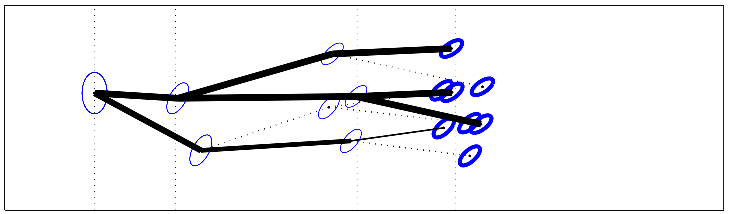


Mixture Kalman Filter



## Sequential Monte Carlo - Particle Filtering

- We try to approximate the so-called filtering density with a set of points/Gaussians $\equiv$ particles
- Algorithms are intuitively similar to randomised search algorithms but are best understood in terms of sequential importance sampling and resampling techniques



## Importance Sampling (IS)

Consider a probability distribution with (possibly unknown) normalisation constant

$$p(\mathbf{x}) = \frac{1}{Z}\phi(\mathbf{x}) \qquad\qquad Z = \int d\mathbf{x}\phi(\mathbf{x}).$$

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

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

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



## 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 $i = 1, \dots N$ independent samples from $q$

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

- We calculate the **importance weights**

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

- Approximate the normalizing constant

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

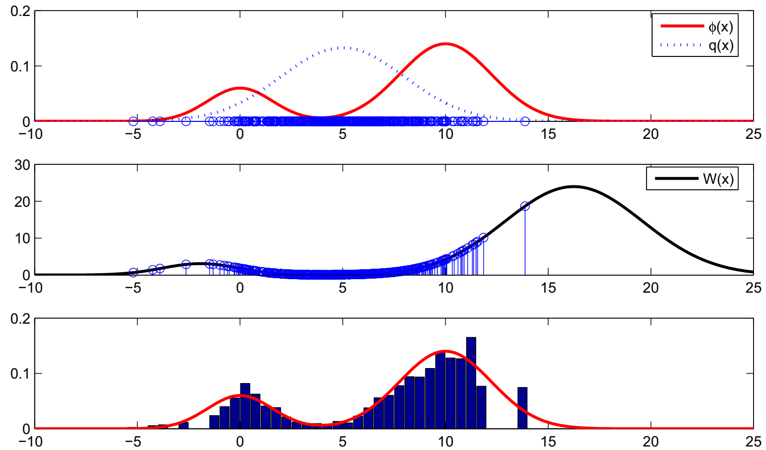
- Desired expectation is approximated by

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

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"

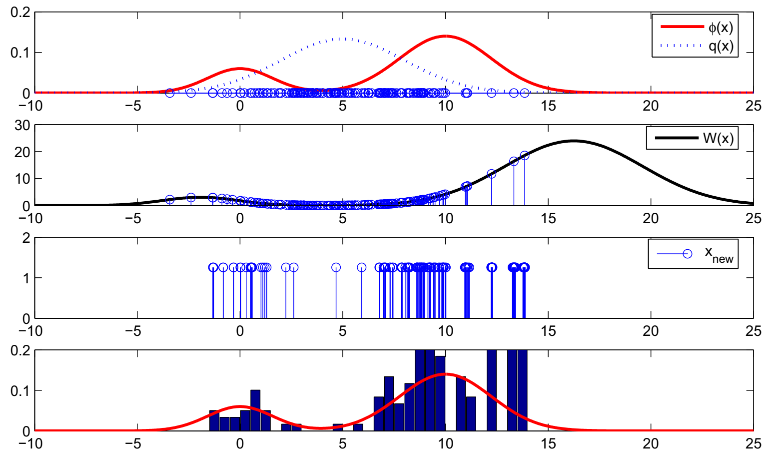
$$x_{\text{new}}^{(j)} \sim \sum_i \tilde{W}^{(i)} \delta(x - x^{(i)})$$

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

- 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




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



# Sequential Importance Sampling, Particle Filtering

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

$$p(x_{0:K}|y_{1:K}) = \frac{1}{p(y_{1:K})} p(y_{1:K}|x_{0:K}) p(x_{0:K}) \equiv \frac{1}{Z_y} \phi(x_{0:K}) \quad (3)$$

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

$$q(x_{0:K}|y_{1:K}) = q(x_0) \prod_{k=1}^{K} q(x_k|x_{1:k-1} y_{1:k})$$



# Markov Random Fields

# Markov Random Fields

- A set of random variables $\boldsymbol{\xi} = \{\xi_i\}_{i \in \mathcal{V}}$, Given
  - an undirected graph with vertex set $\mathcal{V}$ and undirected edge set $\mathcal{E}$
  - A set of local potential functions (with parameters **a**)

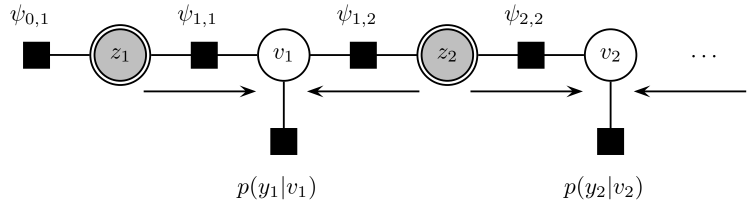
$$p(\boldsymbol{\xi}; \mathbf{a}) = \frac{1}{Z_{\mathbf{a}}} \prod_{i \in \mathcal{V}} \phi_i(\xi_i) \prod_{(i,j) \in \mathcal{E}} \psi_{i,j}(\xi_i, \xi_j)$$

$\phi_i(\xi_i; \mathbf{a})$ : (Singleton)

$\psi_{i,j}(\xi_i, \xi_j; \mathbf{a})$ : (Pairwise)



# VB or Gibbs




- VB

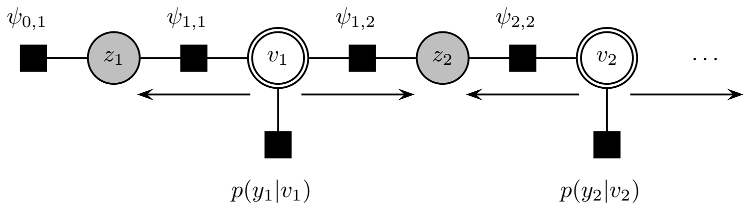
$$q^{(\tau)}(v_k) \quad \leftarrow \quad \exp(\phi_k + \langle \log \psi_{k,k} + \log \psi_{k,k+1} \rangle_{q^{(\tau)}(z_k) q^{(\tau)}(z_{k+1})})$$

- Gibbs

$$v_k^{(\tau)} \quad \sim \quad p(v_k | z_{k-1}, z_k, y_k) \propto p(y_k|v_k) \psi_{k,k}(z_k^{(\tau)}) \psi_{k,k+1}(z_{k+1}^{(\tau)})$$



# VB or Gibbs




- VB

$$q^{(\tau)}(z_k) \quad \leftarrow \quad \exp(\phi_k + \langle \log \psi_{k,k-1} + \log \psi_{k,k} \rangle_{q^{(\tau)}(v_k) q^{(\tau)}(v_{k+1})})$$

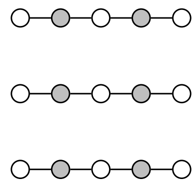
- Gibbs

$$z_k^{(\tau)} \quad \sim \quad p(z_k | v_{k-1}, v_k) \propto \psi_{k,k-1}(v_{k-1}^{(\tau)}) \psi_{k,k}(v_k^{(\tau)})$$

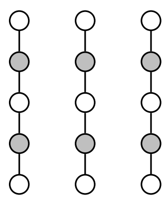


# Harmonic-Transient Decomposition

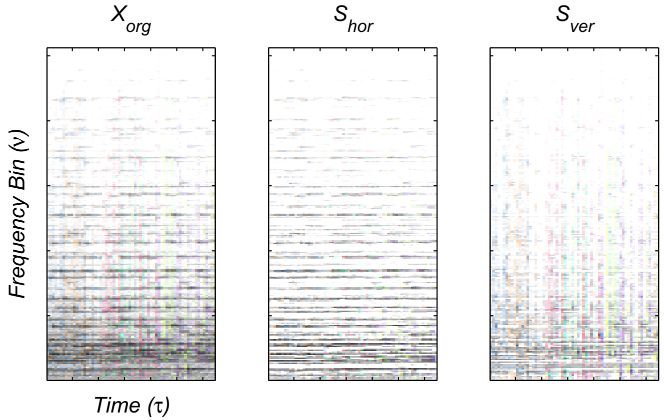
- Source 1: Horizontal : Tie across time : harmonic continuity



- Source 2: Vertical : Tie across frequency : transients, pulse like sounds



## Harmonic-Transient Decomposition

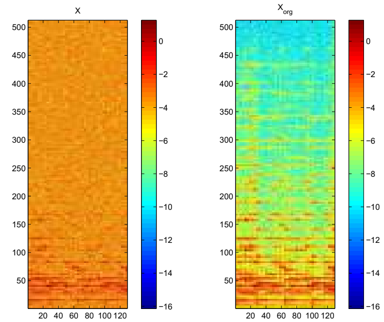



(Original) (Hor) (Vert)

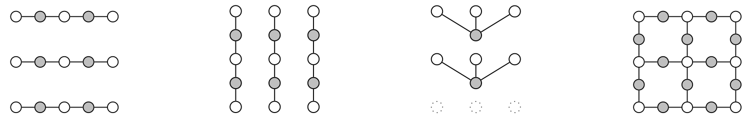
Superstition, (thanks to Tuomas Virtanen)
(Original) (Hor) (Vert)



## Denoising - Piano



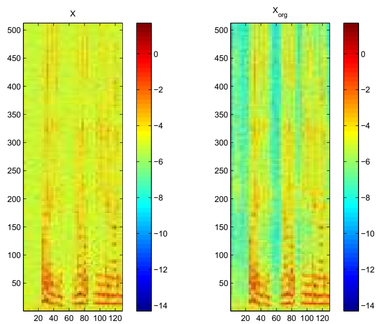

(Noisy) (Original)



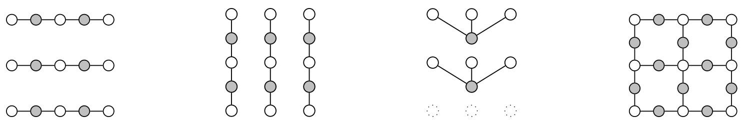
(Hor) (Ver) (Band) (Grid)



## Denoising - Speech




(Noisy) (Original)



(Hor) (Ver) (Band) (Grid)



# Topic-Term-Document Models

# Text Processing, Latent Semantic Indexing

Deerwester et al. (1990), Berry et al. (1995), Manning, Schuetze, Raghavan (2007)

- We are given a database of *documents* $D = \{d_1, \dots, d_j, \dots, d_N\}$
- Each document contains several terms from a codebook of terms $T = \{t_1, \dots, t_i, \dots, t_M\}$
- Retrieval,
  - Given a query $q$ (for example a set of few terms $T_q \subset T$) retrieve a set of documents $D^q_{\text{Retrieved}}$
  - Assume we know the set of relevant documents $D^q_{\text{Relevant}} \subset D$ (with respect to the query q)
  - Quality Measures:

$$\text{Precision}^q = \frac{|D^q_{\text{Relevant}} \cap D^q_{\text{Retrieved}}|}{|D^q_{\text{Retrieved}}|} \qquad \text{Recall}^q = \frac{|D^q_{\text{Relevant}} \cap D^q_{\text{Retrieved}}|}{|D^q_{\text{Relevant}}|}$$



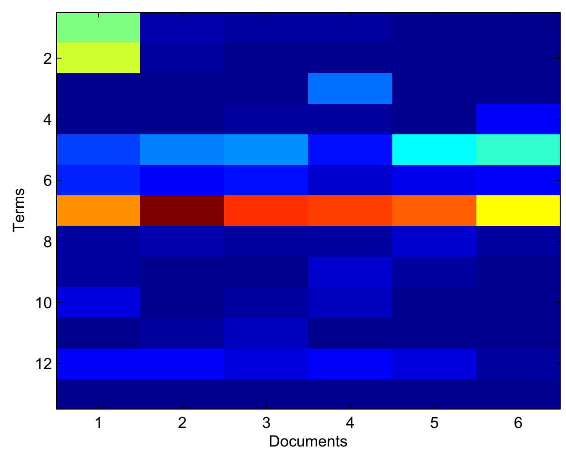
# Representation: Term-Document matrix $A \in \mathbb{R}^{M \times N}$

- Rows : terms $t_i, i = 1 \dots M$
- Columns: documents $d_j = 1 \dots N$

| | j.caesar | hamlet | othello | macbeth | rom&jul | sonnets |
|---|---|---|---|---|---|---|
| caesar | 270 | 2 | 1 | 1 | 0 | 0 |
| brutus | 379 | 1 | 0 | 0 | 0 | 0 |
| malcolm | 0 | 0 | 0 | 60 | 0 | 0 |
| muse | 0 | 0 | 1 | 1 | 0 | 16 |
| ⋮ | | | | | | |
| love | 34 | 68 | 80 | 19 | 150 | 195 |
| friend | 23 | 14 | 18 | 5 | 13 | 16 |
| the | 610 | 1148 | 759 | 733 | 682 | 446 |
| traitor | 1 | 0 | 0 | 5 | 1 | 0 |
| traitors | 9 | 0 | 1 | 3 | 0 | 0 |
| ⋮ | | | | | | |
| napkin | 0 | 1 | 3 | 0 | 0 | 0 |
| sword | 15 | 16 | 10 | 14 | 8 | 1 |
| laptop | 0 | 0 | 0 | 0 | 0 | 0 |



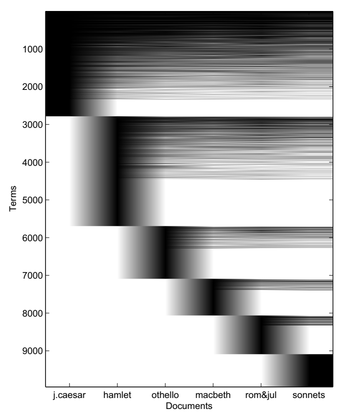
# Term-Document matrix




- Counts



# Term-Document matrix




- Incidence (zero-one) matrix

# Singular Value Decomposition (SVD)

For any $A \in \mathbb{R}^{M \times N}$, there exist **orthogonal** matrices $U \in \mathbb{R}^{M \times M}$ and $V \in \mathbb{R}^{N \times N}$ such that

$$U = [u_1, \dots, u_M] \qquad V = [v_1, \dots, v_N]$$

such that

$$U^\top A V = \mathbf{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{M \times N}$$

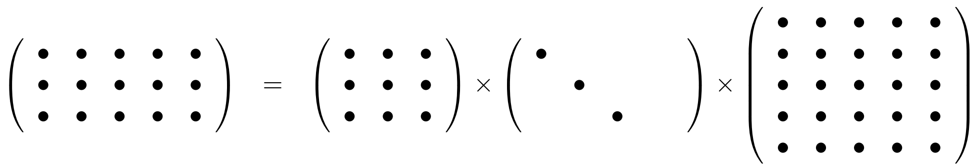
with $p = \min\{M, N\}$. We have

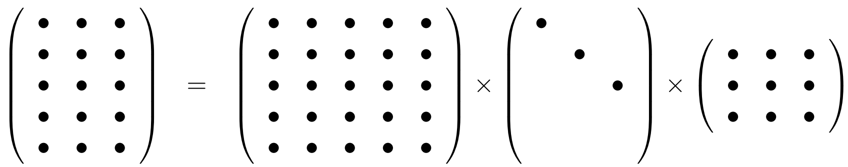
$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$$



# Singular Value Decomposition (SVD)

$$A = U \times \Sigma \times V^\top$$







# Singular Value Decomposition (SVD)

```
>> A =
     1     2
     3     5
     1     0

>> [U S V] = svd(A)          % A == U*S*V'
U =
   -0.3559   -0.2000   -0.9129
   -0.9309   -0.0102    0.3651
   -0.0823    0.9797   -0.1826

S =
    6.2638         0
         0    0.8744
         0         0

V =
   -0.5158    0.8567
   -0.8567   -0.5158
```



# Singular Value Decomposition (SVD)

```
>> U(:,1)*S(1,1)*V(:,1)'
ans =
    1.1498    1.9098
    3.0076    4.9954
    0.2661    0.4419
>> U(:,2)*S(2,2)*V(:,2)'
ans =
   -0.1498    0.0902
   -0.0076    0.0046
    0.7339   -0.4419
>> U(:,1)*S(1,1)*V(:,1)' + U(:,2)*S(2,2)*V(:,2)'   %% ==  U*S*V' == A
ans =
    1.0000    2.0000
    3.0000    5.0000
    1.0000    0.0000
>> A =
     1     2
     3     5
     1     0
```

# Singular Value Decomposition (SVD)

SVD expansion

$$
\begin{aligned}
A &= \sum_{r=1}^{P} \sigma_r u_r v_r^\top \\
&= U\, \mathbf{diag}(\sigma_1, \dots, \sigma_P) V^\top
\end{aligned}
$$

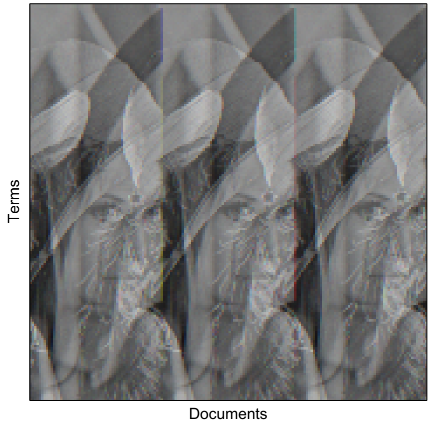
The norm relations for $A \in \mathbb{R}^{M \times N}$, $P = \min\{M, N\}$

$$
\begin{aligned}
\|A\|_F^2 &= \sigma_1^2 + \dots + \sigma_P^2 \\
\|A\|_2^2 &= \sigma_1^2
\end{aligned}
$$



# Singular Value Decomposition of Term-Document Matrices

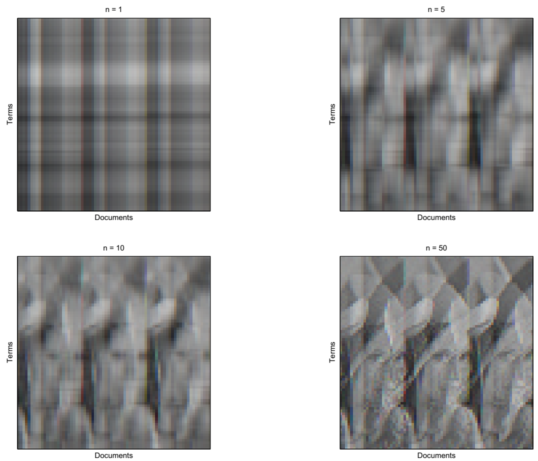
Another "term-document" matrix






# Singular Value Decomposition of Term-Document Matrices

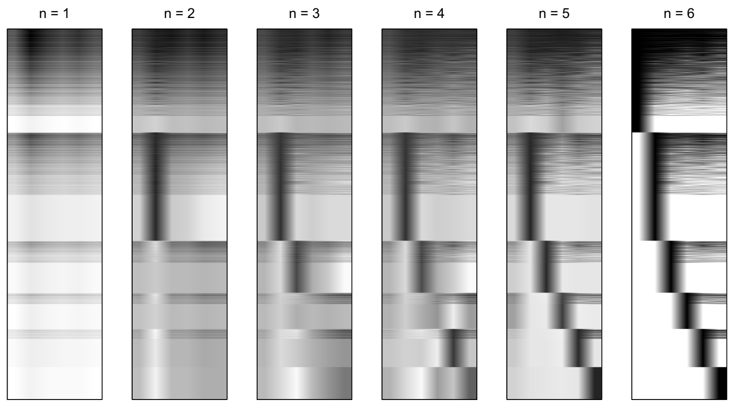
$$A \approx U(:, 1{:}n) S(1{:}n, 1{:}n) V(:, 1{:}n)^\top$$





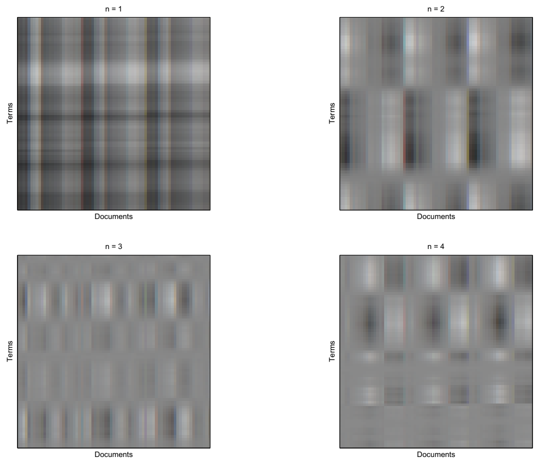

# Singular Value Decomposition of Term-Document Matrices

$$A \approx U(:, 1{:}n) S(1{:}n, 1{:}n) V(:, 1{:}n)^\top$$



# Rank-1 Matrices

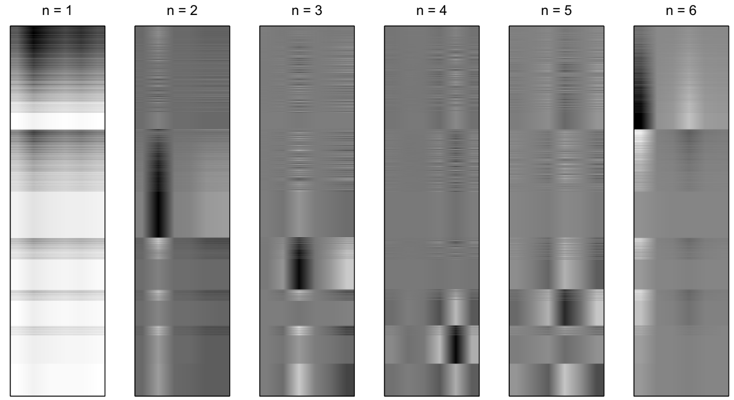
$$U(:, n)V(:, n)^\top$$






# Rank-1 Matrices

$$U(:, n)V(:, n)^\top$$






# LSI: Summary and Remarks

- Low rank approximation to a term-document matrix by keeping the latent dimensions corresponding to $n$ largest singular values of SVD

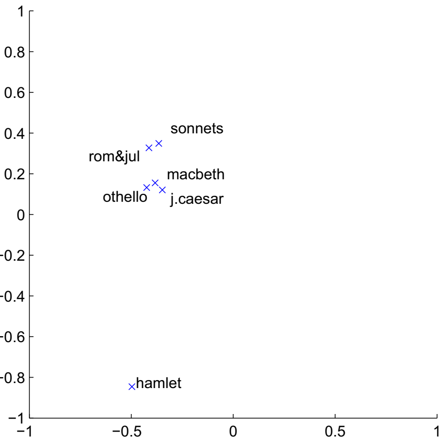
$$A \approx \sum_{r=1}^{n} \sigma_r U(:, r)V(:, r)^\top$$

- No direct statistical interpretation, but loosely
  - Each $r = 1 \dots n$ denotes a *latent topic* ($n$ is the total number of topics)
  - $U(i, r)$ corresponds to *weight* of the $i$'th term given the topic $r$
  - $V(j, r)$ corresponds to *emphasis* of topic $r$ in document $j$
    We can think $V(j, 1{:}n)^\top$ as the coordinates of $j$'th document in an $n$ dimensional *latent topic space*
  - The coordinates of a new document are computed by

$$v_{\text{new}} = \Sigma^{-1} U^\top a_{\text{new}}$$



# Latent Semantic Space



## LSI: Summary and Remarks

- Clustering, assessing similarity, visualisation ...
- Rationale: documents that share frequent co-occurring terms will be close in the latent space
- May deal with synonymy and polysemy
  - different words - same meaning
    `baggage-lugagge`
  - same word - different meaning
    `spider (the animal - the web crawler)`



## Probabilistic Latent Sematic Indexing, the Aspect Model

Hofmann, 1999

| | |
|---|---|
| $d \sim p(d)$ | Document |
| $z \mid d \sim p(z \mid d)$ | Latent Topic |
| $t \mid z \sim p(t \mid z)$ | Term |

More to come ...



# Factorial Models

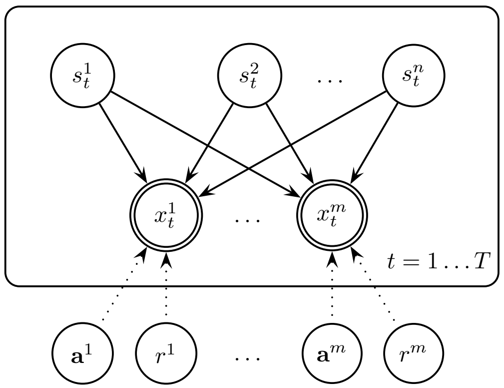
# Source Separation

# Bayesian Model selection



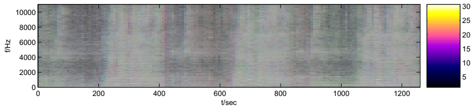
## Audio Source Separation

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

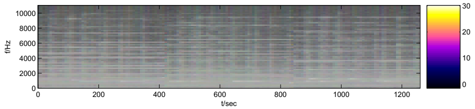



$$
\begin{aligned}
s_t^i &\sim p(s_t^i) \\
x_t^j &\sim \mathcal{N}(x; \mathbf{a}^j s_t^{1:n}, r^j)
\end{aligned}
$$

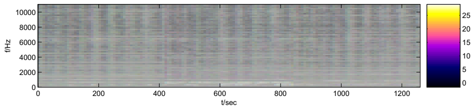
# Audio Source Separation



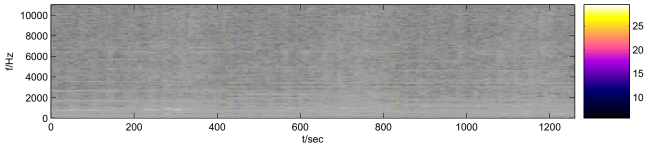

(Speech)




(Piano)




(Guitar)

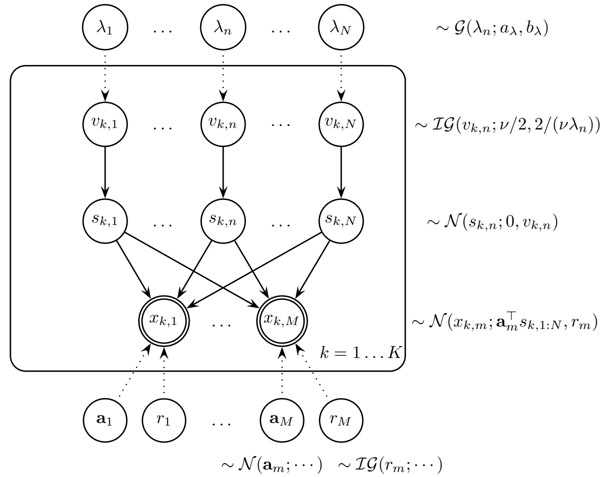



(Mix)



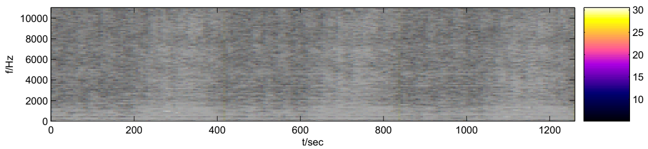
# Audio Source Separation

- Hierarchical Prior Model (Fevotte and Godsill 2005 [10], Cemgil et. al. 2006 [3])

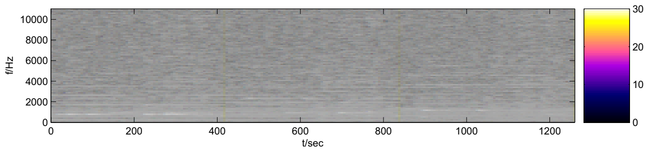




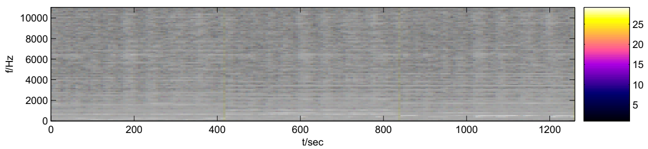

# Reconstructions




(Speech)



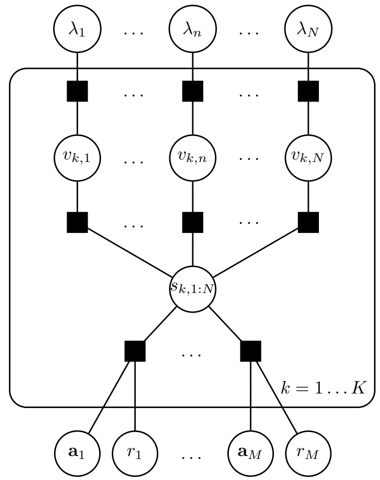

(Piano)




(Guitar)

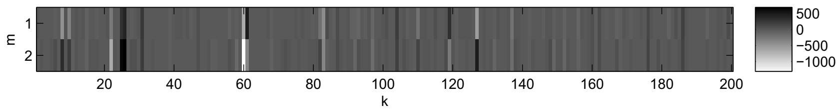


# Audio Source Separation, Inference



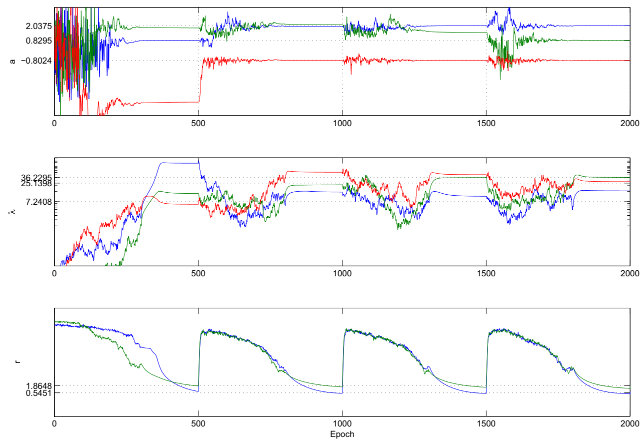

- Exact inference is not possible

# Observations



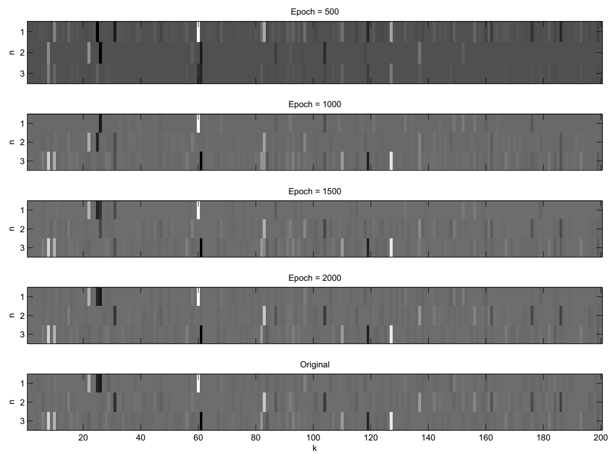



# A typical run, $250/250$ Gibbs/VB with tempering





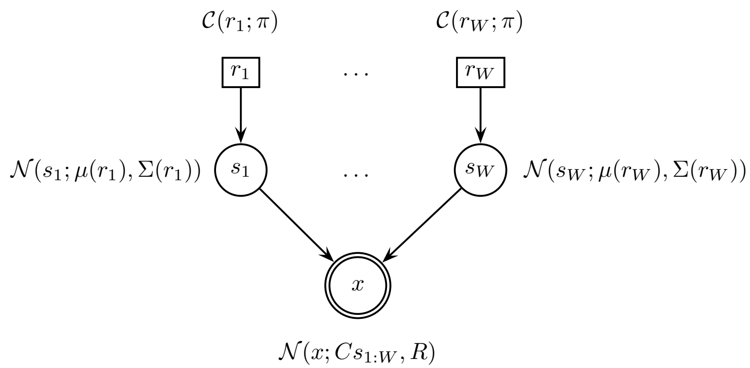

# Reconstructions




Posterior surface is multimodal, each mode corresponding to a viable separation



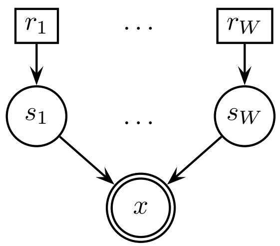
# Bayesian Variable Selection




- 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

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

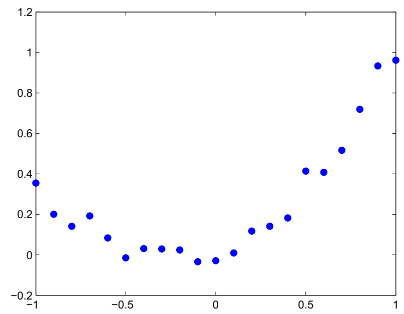



$$p(\mathbf{x},s_{1:W},r_{1:W}) = p(\mathbf{x}|s_{1:W},r_{1:W})\prod_{i=1}^{W} p(s_i|r_i)p(r_i)$$



## Example 1: Variable selection in Polynomial Regression

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




$$x(t) = \sum_{i=0}^{N} s_{i+1}t^i + \epsilon(t)$$



## Ex1: Regression

$$\begin{aligned} \mathbf{t} &= \begin{pmatrix} t_1 & t_2 & \dots & t_J \end{pmatrix}^\top \\ C &\equiv \begin{pmatrix} \mathbf{t}^0 & \mathbf{t}^1 & \dots & \mathbf{t}^{W-1} \end{pmatrix} \end{aligned}$$

```
>> C = fliplr(vander(0:4))  % Van der Monde matrix
     1     0     0     0     0
     1     1     1     1     1
     1     2     4     8    16
     1     3     9    27    81
     1     4    16    64   256
```

$$\begin{aligned} r_i &\sim \mathcal{C}(r_i;0.5,0.5) \qquad r_i \in \{\text{on},\text{off}\} \\ s_i|r_i &\sim \mathcal{N}(s_i;0,\Sigma(r_i)) \\ \mathbf{x}|s_{1:W} &\sim \mathcal{N}(\mathbf{x};Cs_{1:W},R) \end{aligned}$$

$$\Sigma(r_i=\text{on}) \gg \Sigma(r_i=\text{off})$$



## Ex1: Regression

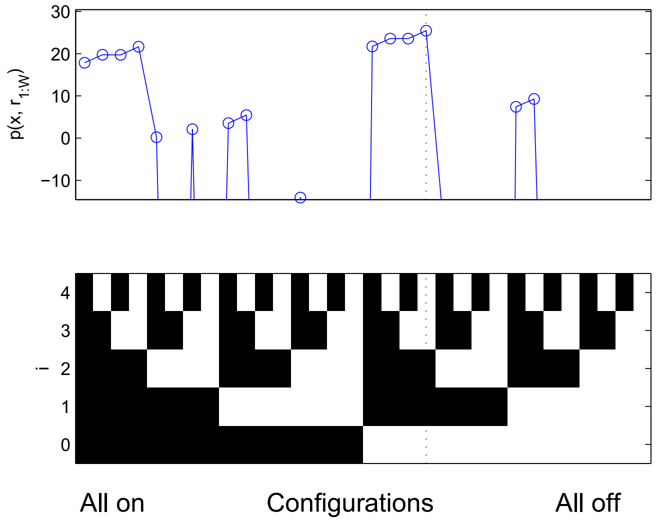
To find the "active" basis functions we need to calculate

$$r^*_{1:W} \equiv \operatorname*{argmax}_{r_{1:W}} p(r_{1:W}|\mathbf{x}) = \operatorname*{argmax}_{r_{1:W}} \int ds_{1:W}\, p(\mathbf{x}|s_{1:W})p(s_{1:W}|r_{1:W})p(r_{1:W})$$

Then, the reconstruction is given by

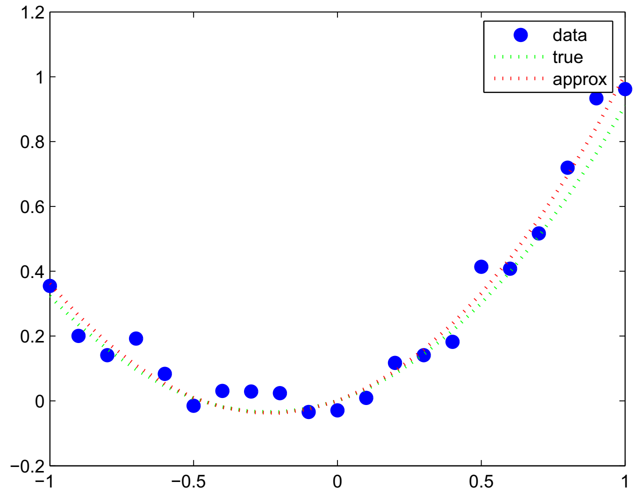
$$\begin{aligned} \hat{x}(t) &= \left\langle \sum_{i=0}^{W-1} s_{i+1}t^i \right\rangle_{p(s_{1:W}|\mathbf{x},r^*_{1:W})} \\ &= \sum_{i=0}^{W-1} \langle s_{i+1}\rangle_{p(s_{i+1}|\mathbf{x},r^*_{1:W})}\, t^i \end{aligned}$$

# Ex1: Regression



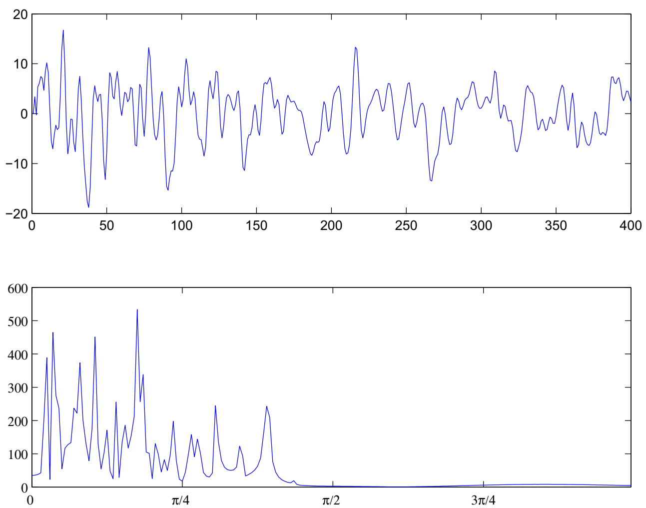



# Ex1: Regression






# Example 2: Chord Recognition





# (Damped) Sinusoidal Basis

- $h = 1 \dots H$, number of harmonics, $t = 0 \dots T-1$, sample index
- $\omega$ : fundamental frequency in rad, $\rho$ damping coefficient

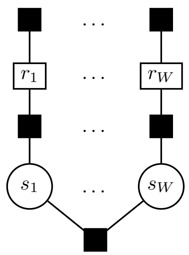
$$C(\omega) \equiv \begin{pmatrix} C_0^1 & \dots & C_0^H \\ \vdots & C_t^h & \vdots \\ C_{T-1}^1 & \dots & C_{T-1}^H \end{pmatrix}$$

$$\begin{aligned} C_t^h &\equiv \rho^t \begin{pmatrix} \cos(th\omega) & \sin(th\omega) \end{pmatrix} \\ \mathbf{C} &= [C(\omega_1) \dots C(\omega_\nu) \dots C(\omega_W)] \end{aligned}$$

- See also Badeau, Boyer, David. Eds parametric modelling and tracking of audio signals. In DAFx 2002

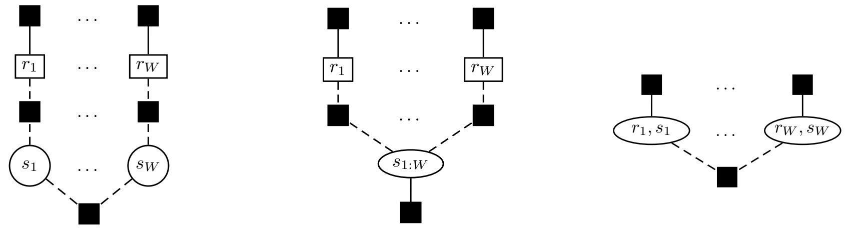
# Factor graph

$$
\begin{aligned}
\log \phi(r_{1:W}, s_{1:W}) &= \sum_{i=1}^{W} (\log \pi(r_i)) \\
&+ \sum_{i=1}^{W} \Big( -\frac{1}{2} s_i^\top \Sigma(r_i)^{-1} s_i + \mu(r_i)^\top \Sigma(r_i)^{-1} s_i \\
&\quad -\frac{1}{2} \mu(r_i)^\top \Sigma(r_i)^{-1} \mu(r_i) - \frac{1}{2} \log |2\pi \Sigma(r_i)| \Big) \\
&-\frac{1}{2} \mathbf{x}^\top R^{-1} \mathbf{x} + s_{1:W}^\top C^\top R^{-1} \mathbf{x} - \frac{1}{2} s_{1:W}^\top C^\top R^{-1} C s_{1:W} - \frac{1}{2} \log |2\pi R|
\end{aligned}
$$






# Approximating Structures

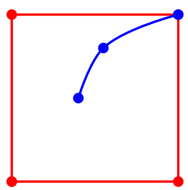



$\mathcal{Q}_1 = \prod_{i=1}^{W} \mathcal{Q}(s_i)\mathcal{Q}(r_i)$ $\quad$ $\mathcal{Q}_2 = \mathcal{Q}(s_{1:W}) \prod_{i=1}^{W} \mathcal{Q}(r_i)$ $\quad$ $\mathcal{Q}_3 = \prod_{i=1}^{W} \mathcal{Q}(s_i, r_i)$



# MCMC versus Variational Bayes (VB)

- Each configuration of $r_{1:W}$ corresponds to a corner of a $W$ dimensional hypercube



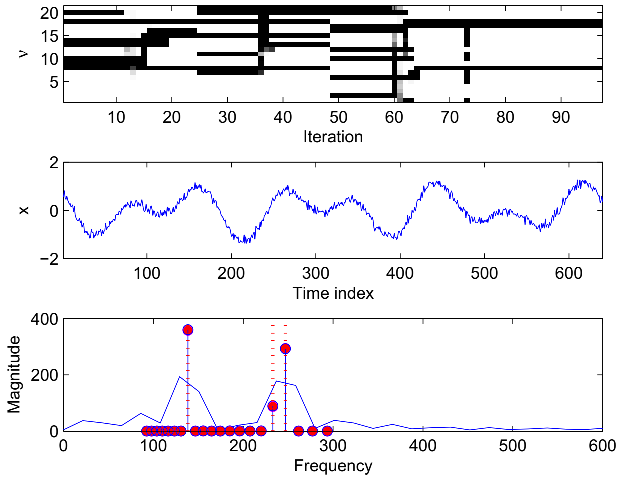
- MCMC moves along the edges stochastically
- Iterative Improvement moves along the edges greedily
- VB moves inside the hypercube deterministically



# Iterative Improvement

| iteration | $r_1$ | | | | | | | | | | | | | | | | | | | | | | | $r_M$ | $\log p(y_{1:T}, r_{1:M})$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | −1220638254 |
| 2 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | −665073975 |
| 3 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | −311983860 |
| 4 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ● | −162334351 |
| 5 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ● | −43419569 |
| 6 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ● | ○ | ● | ○ | ● | −1633593 |
| 7 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ● | ○ | ● | ○ | ● | −14336 |
| 8 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ○ | ● | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ● | ○ | ● | ○ | ● | −5766 |
| 9 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ○ | ○ | ● | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ● | ○ | ● | ○ | ● | −5210 |
| 10 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ● | ○ | ● | ○ | ● | −4664 |
| True | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● | ● | ○ | ○ | ● | ○ | ○ | ○ | ○ | ● | ○ | ● | ○ | ● | −4664 |

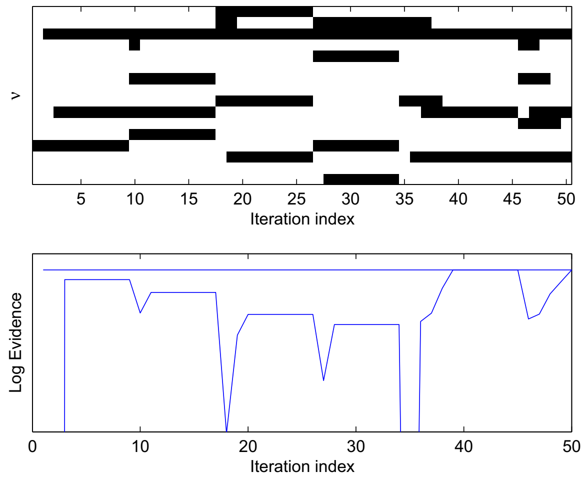
## Results, VB with tempering and reinitialisation




$F_s = 22050$ Hz, $N = 29$ msec, $H = 1$, Midinotes $= 30 \ldots 50$



## Results, MCMC with tempering and reinitialisation




$F_s = 22050$ Hz, $N = 29$ msec, $H = 1$, Midinotes $= 30 \ldots 50$



## Bayesian/Generative/Probabilistic approaches to Polyphonic Transcription

(Walmsley 2000, Davy and Godsill 2002, Raphael 2001, Abdallah 2002, Cemgil et. al. 2003-2006, Vincent 2003, Vincent and Plumbley 2005, Vogel, Jordan and Wessel 2005, Thornburg, Leitsnikov and Berger 2004, Blumensath and Davies 2006, Dubois and Davy 2005)

- Various related but different models
- Inference schemata
  - Reversible Jump MCMC
  - Iterative Improvement
  - Laplace approximation
  - Particle filtering
  - Variational Bayes, MCMC



## Summary

- Bayesian Inference
- Graphical models
- Exact Inference
- Approximate inference

## Summary, Attributes of Probabilistic Inference

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



## Summary of what we have mentioned

- Exact inference, Belief Propagation
- Approximate inference
  - Deterministic
    - Variational Bayes,
    - Expectation/Maximization (EM), Iterative Conditional Modes (ICM)
  - Stochastic
    - Markov Chain Monte Carlo
    - Importance Sampling,
    - Particle filtering



## Summary of what we have not mentioned

- Exact Inference (Junction Tree ...)
- Deterministic Inference
  - 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



## Bibliography

- General background about probability theory
- Graphical models
- Exact inference
- Variational Methods
- Markov Chain Monte Carlo
- Sequential Monte Carlo
- Applications

## General background about probability theory

- Dimitri P. Bertsekas and John N. Tsitsiklis. Introduction to Probability. Athena Scientific, 2002
- Geoffrey Grimmet and David Stirzaker, Probability and Random Processes, (3rd Ed), Oxford, 2006



## "Instant Classics" of Bayesian Machine Learning and Graphical Models

- Michael I. Jordan, Learning in Graphical Models, 1998
- David MacKay Information Theory, Learning and Inference Algorithms, 2003, Cambridge
- Chris Bishop, Machine Learning and Pattern Recognition, 2006, Springer



## Further Reading, Variational Methods

- Jaakkola "Tutorial on variational approximation methods", 2000
  `http://people.csail.mit.edu/tommi/papers/Jaa-var-tutorial.ps`
- Wainwright and Jordan 2003 [19] Berkeley EECS Tech. Rep.
- Frey and Jojic, PAMI 2005 [11]
- Winn and Bishop "Variational Message Passing" 2005 JMLR [20]



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

## Text Processing and Information Retrieval

- *Information Retrieval*, Manning, Schuetze, and Raghavan, Cambridge University Press, 2007 (Draft)
  `http://www-csli.stanford.edu/ schuetze/information-retrieval-book.html`
- *Modeling the Internet and the Web: Probabilistic Methods and Algorithms*. Pierre Baldi, Paolo Frasconi, Padhraic Smyth, Wiley, 2003
  `http://ibook.ics.uci.edu`

Compression, Efficient Data Structures,

- *Managing Gigabytes: Compressing and Indexing Documents and Images*. Witten, Moffat and Bell. Morgan Kaufmann 1999
- *Introduction to Data Compression*. Khalid Sayood, Morgan Kaufmann (3rd Ed), 2005



## Some Generic Software Packages

- Kevin Murphy's Matlab Bayesian Networks toolkit (BNT)
- Gilks, et. al. BUGS, WinBUGS – (Bayesian analysis Using Gibbs Sampling) A powerful program that compiles Gibbs Samplers from
- Winn, et. al, VIBES – Similar to BUGS but for variational inference

For source separation, there are some specialised libraries

- Petersen and Winther (DTU, Kopenhagen)
- Harva, Raiko, Honkela, Valpola "Bayes Blocks" (HUT, Helsinki)



## Music Applications

- Klapuri and Davy (Eds) Signal processing for Music Transcription, Springer, 2006
- Temperley, Probability and Music, MIT Press, 2007



## References

[1] M. Allan and C. K. I. Williams. Harmonising chorales by probabilistic inference. In Advances in Neural Information Processing Systems 17, 2004.

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

[3] A. T. Cemgil, C. Fevotte, and S. J. Godsill. Variational and Stochastic Inference for Bayesian Source Separation. Digital Signal Processing, in Print, 2007.

[4] A. T. Cemgil and S. J. Godsill. Efficient Variational Inference for the Dynamic Harmonic Model. In Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, October 2005.

[5] A. T. Cemgil and S. J. Godsill. Probabilistic Phase Vocoder and its application to Interpolation of Missing Values in Audio Signals. In 13th European Signal Processing Conference, Antalya/Turkey, 2005. EURASIP.

[6] A. T. Cemgil and H. J. Kappen. Monte Carlo methods for Tempo Tracking and Rhythm Quantization. Journal of Artificial Intelligence Research, 18:45–81, 2003.

[7] A. T. Cemgil, H. J. Kappen, and D. Barber. A Generative Model for Music Transcription. IEEE Transactions on Audio, Speech and Language Processing, 14(2):679–694, March 2006.

[8] A.T. Cemgil, H. J. Kappen, P. Desain, and H. Honing. On tempo tracking: Tempogram Representation and Kalman filtering. In Proceedings of the 2000 International Computer Music Conference, pages 352–355, Berlin, 2000. (This paper has received the Swets and Zeitlinger Distinguished Paper Award of the ICMC 2000).

[9] R. Chen and J. S. Liu. Mixture Kalman filters. J. R. Statist. Soc., 10, 2000.

[10] C. Févotte and S. J. Godsill. A Bayesian approach for blind separation of sparse sources. IEEE Trans. Speech and Audio Processing, in press. In press - Preprint available at `http://persos.mist-technologies.com/~cfevotte/`.

[11] 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.

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

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

[14] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory, 47(2):498–519, February 2001.

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

[16] 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.

[17] L. R. Rabiner. A tutorial in hidden Markov models and selected applications in speech recognation. Proc. of the IEEE, 77(2):257–286, 1989.

[18] C. Raphael. A probabilistic expert system for automatic musical accompaniment. Journal of Computational and Graphical Statistics, 10(3):467–512, 2001.

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

[20] J. Winn and C. Bishop. Variational message passing. Journal of Machine Learning Research, 6:661–694, 2005.




## Thank you for your patience and attention!

Slides will be available online

`http://www-sigproc.eng.cam.ac.uk/~atc27/acm-tutorial/`