

Factor Graph Based Inference

Justin Dauwels

www.dauwels.com

Amari Research Unit
Brain Science Institute, RIKEN
Saitama, Japan

NAIST, October 12



Outline

- 1 Introduction to **graphical models** (in particular, **factor graphs**).
- 2 **Inference** by means of graphical models.

Joint work with Andrew Eckford (York University), Sascha Korf (Phonak AG) and Andi Loeliger (ETH Zurich).

Inference (2)

Example (Autoregressive model with noisy observations)

Let X_1, X_2, \dots be a real random process defined by:

$$X_k = a_1 X_{k-1} + a_2 X_{k-2} + \dots + a_M X_{k-M} + U_k, \quad U_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_U^2)$$

and let the process $Y = Y_1, Y_2, \dots$ be defined as:

$$Y_k = X_k + W_k, \quad W_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_W^2).$$

Task

Estimate a_1, \dots, a_M from observation Y .

Remark

σ_U^2 , σ_W^2 , and X_k are **nuisance** parameters/random variables

Applications

The problem of **inference** appears in many contexts

- 1 **Digital communications:**
extract transmitted information from received signal.
e.g., (wireless) LAN, 3G, CDMA, etc.
- 2 **Signal processing:**
denoise measured signal, e.g., speech processing.
- 3 **Statistical physics:**
simple stochastic models for materials, e.g., ferromagnets.
- 4 **Machine learning:**
recognize structure in data
e.g., gene expression, weather, stock market, etc.

1–3: true distribution (often) **known** \Rightarrow estimation/detection

4: true distribution **unknown**, but “**guessed**” \Rightarrow learning.

Inference (3)

Estimation/Detection

Given

- Probability density/mass function $p(x, y, z; \theta, \varphi)$.
- Observation $Y = y$.

Task

Estimate the random variable (vector) X and parameter (vector) θ .

Natural solutions

$$(\hat{\theta}, \hat{x}, \hat{\varphi}) = \underset{x, \theta, \varphi}{\operatorname{argmax}} p(x, y; \theta, \varphi) \triangleq \underset{x, \theta, \varphi}{\operatorname{argmax}} \int_{\mathbf{z}} p(x, y, z; \theta, \varphi) dz.$$

$$(\hat{\theta}, \hat{x}) = \underset{x, \theta}{\operatorname{argmax}} p(x, y; \theta) \triangleq \underset{x, \theta}{\operatorname{argmax}} \int_{\mathbf{z}, \varphi} p(x, y, z; \theta, \varphi) p(\varphi) dz d\varphi.$$

Maximization/Summation/Integration often **problematic!**

⇒ choose appropriate **approximations**.

Approximation by Message Passing on Graphical Model

Two Pillars

- **Graphical model** = graphical representation of $p(x, y, z; \theta, \varphi)$.
- **Algorithms** operate on graphical model by sending “messages” between nodes \Rightarrow **local** computations at each node.

Graphical Models

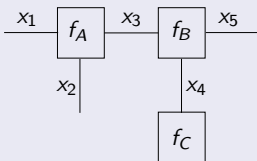
Graphical representation of mathematical model.

- Block diagrams (systems theory)
- Neural networks (e.g., Boltzmann machines/spin glasses)
- Markov random fields (statistics/statistical physics)
- Bayesian networks (machine learning)
- Tanner graphs/factor graphs (coding theory)

Forney-style factor graphs (FFGs)

- Factor graphs represent the factorization of a function.
- Example

$$f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3)f_B(x_3, x_4, x_5)f_C(x_4).$$

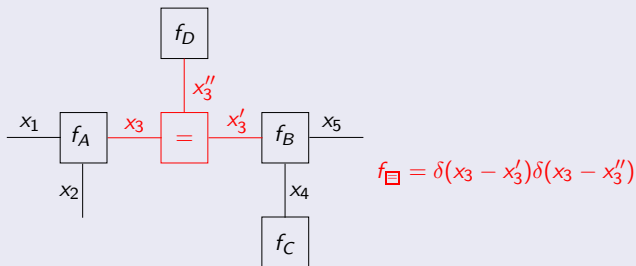


- Rules for drawing a factor graph
 - A node for every factor
 - An edge for every variable
 - Node g is connected to edge x iff variable x appears in factor g

Forney-style factor graphs (FFGs)

- Factor graphs represent the factorization of a function.
- Example

$$f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3) f_B(x_3, x_4, x_5) f_C(x_4) f_D(x_3).$$

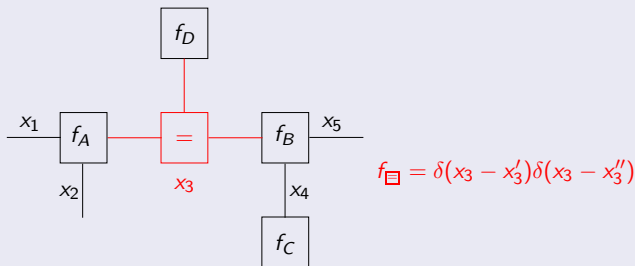


- Rules for drawing a factor graph
 - A node for every factor
 - An edge for every variable
 - Node g is connected to edge x iff variable x appears in factor g

Forney-style factor graphs (FFGs)

- Factor graphs represent the factorization of a function.
- Example

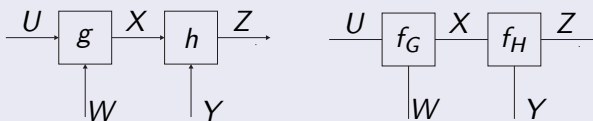
$$f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3) f_B(x_3, x_4, x_5) f_C(x_4) f_D(x_3).$$



- Rules for drawing a factor graph
 - A node for every factor
 - An edge for every variable
 - Node g is connected to edge x iff variable x appears in factor g

Forney-style factor graphs (2)

Factor graphs and block diagrams



- **Block diagram:** $X = g(U, W)$, $Z = h(X, Y)$
- **Factor graph:** $f_G = \delta(x - g(u, w))$, $f_H = \delta(z - h(x, y))$
- **Global function**
 $f(u, w, x, y, z) = \delta(x - g(u, w)) \cdot \delta(z - h(x, y))$

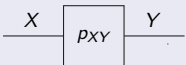
Forney-style factor graphs (3)

Joint Probability Distribution

- Factor graphs can represent **probabilistic models**.
- Given is a **joint probability distribution**

$$p_{XY}(x, y)$$

of the two discrete random variables X and Y .



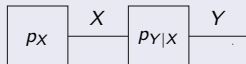
Forney-style factor graphs (4)

Factorization of Joint Probability Distribution

- Chain rule

$$p_{XY}(x, y) = p_X(x)p_{Y|X}(y|x)$$

- Factor graph



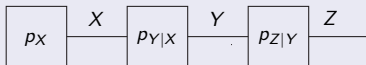
Forney-style factor graphs (6)

Markov Chain

- Markov chain of the form

$$p_{XYZ}(x, y, z) = p_X(x) \cdot p_{Y|X}(y|x) \cdot p_{Z|Y}(z|y)$$

- Factor graph



Summary: Factor Graphs

- Factor graphs represent the factorization of a function.
- Equality constraint node
- Block diagrams and factor graphs
- Probabilistic models represented by factor graph
⇒ statistical properties translated into graph structure

Approximation by Message Passing on Graphical Model

Two Pillars

- **Graphical model** = graphical representation of $p(x, y, z; \theta, \varphi)$.
- **Algorithms** operate on graphical model by sending “messages” between nodes \Rightarrow **local** computations at each node.

Graphical Models

Graphical representation of mathematical model.

- Block diagrams (systems theory)
- Neural networks (e.g., Boltzmann machines/spin glasses)
- Markov random fields (statistics/statistical physics)
- Bayesian networks (machine learning)
- Tanner graphs/factor graphs (coding theory)

Computing marginals

- Given: Probability mass function

$$f(x_1, \dots, x_8) = (f_1(x_1)f_2(x_2)f_3(x_1, x_2, x_3, x_4)) \cdot (f_4(x_4, x_5, x_6)f_5(x_5)(f_6(x_6, x_7, x_8)f_7(x_7)))$$

x_i are discrete variables.

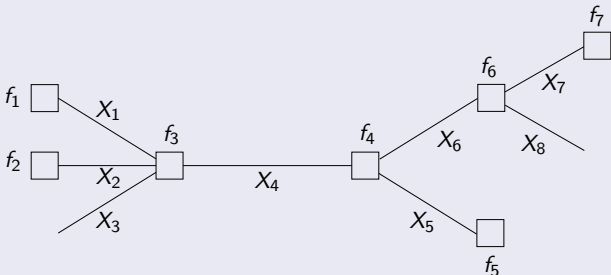
- Wanted: Marginal probability

$$p(x_4) = \sum_{x_1, x_2, x_3, x_5, x_6, x_7, x_8} f(x_1, \dots, x_8).$$

- This factorization can be represented by a factor graph.

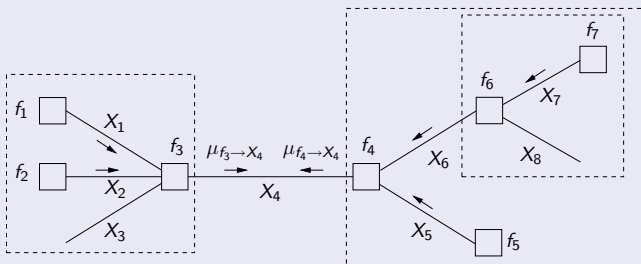
Computing marginals

$$f(x_1, \dots, x_8) = (f_1(x_1)f_2(x_2)f_3(x_1, x_2, x_3, x_4)) \cdot (f_4(x_4, x_5, x_6)f_5(x_5)(f_6(x_6, x_7, x_8)f_7(x_7)))$$



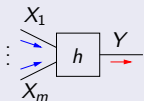
Computing marginals

$$p(X_4) = \underbrace{\left(\sum_{X_1} \sum_{X_2} \sum_{X_3} f_3(X_1, X_2, X_3, X_4) f_1(X_1) f_2(X_2) \right)}_{\mu_{f_3 \rightarrow X_4}} \cdot \underbrace{\left(\sum_{X_5} \sum_{X_6} f_4(X_4, X_5, X_6) f_5(X_5) \left(\sum_{X_7} \sum_{X_8} f_6(X_6, X_7, X_8) f_7(X_7) \right) \right)}_{\mu_{f_6 \rightarrow X_6}}}_{\mu_{f_4 \rightarrow X_4}}$$



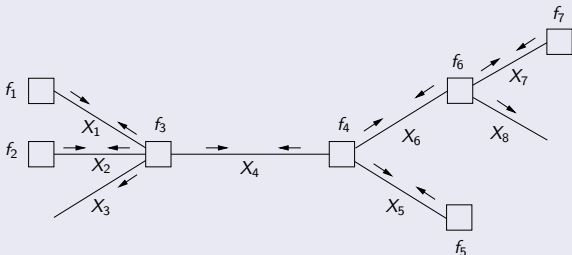
Sum-product algorithm (“belief propagation”)

Step 1: Compute all messages (starting at the leaves)



$$\mu(y) \propto \sum_{x_1, \dots, x_m} h(y, x_1, \dots, x_m) \mu(x_1) \cdots \mu(x_m)$$

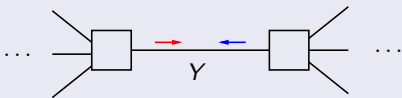
Example



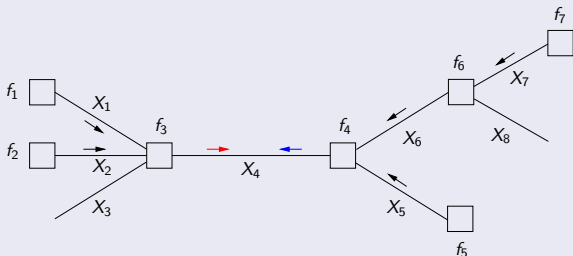
Sum-product algorithm (“belief propagation”)

Step 2: Compute marginals

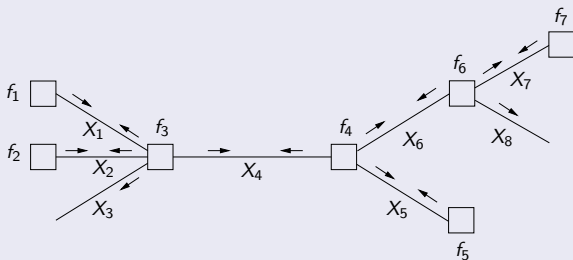
$$p(y) \propto \mu_{\rightarrow}(y) \mu_{\leftarrow}(y)$$



Example

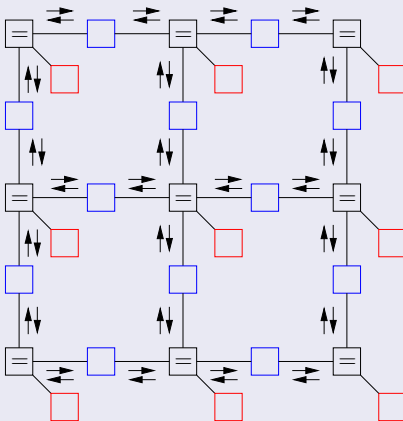


Sum-product algorithm on cycle-free graphs



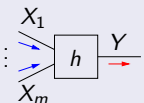
- **Finite** number of computations.
- Leads to **exact** marginals.
- **All** marginals **at once**.

Cyclic graph



Still applicable, but **approximate** marginals; may not **convergence!**

Sum-product rule: continuous variables



$$\mu(y) \propto \int_{x_1, \dots, x_m} h(y, x_1, \dots, x_m) \mu(x_1) \cdots \mu(x_m) dx_1 \dots dx_m$$

May be intractable

- X_k discrete: unwieldy number of terms in sum
- X_k continuous: no closed-form expression for integral.

Assume **integration** over X_1 is intractable.

Approximations vs. message types

Numerical Integration

$$\mu(y) \propto \sum_i \int_{x_2, \dots, x_m} h(y, \hat{x}_1^{(i)}, x_2, \dots, x_m) \mu(x_2) \cdots \mu(x_m) dx_2 \dots dx_m$$



Decision based

$$\mu(y) \propto \int_{x_2, \dots, x_m} h(y, \hat{x}_1, x_2, \dots, x_m) \mu(x_2) \cdots \mu(x_m) dx_2 \dots dx_m$$



Gaussian approximation

$$\mu(y) \propto \int_{x_2, \dots, x_m} h(y, x_1, x_2, \dots, x_m) \mathcal{N}(x_1) \mu(x_2) \cdots \mu(x_m) dx_2 \dots dx_m$$



Particle method

$$\mu(y) \propto \sum_i \int_{x_2, \dots, x_m} h(y, \hat{x}_1^{(i)}, x_2, \dots, x_m) w_1^{(i)} \mu(x_2) \cdots \mu(x_m) dx_2 \dots dx_m$$



Deriving algorithms

Choosing message types

- Each of the messages may be represented **differently**.
- **Combination** of various types of **algorithms**
 - sum/max-product algorithm
 - decision-based algorithms (e.g., gradient methods/EM)
 - Kalman filters
 - particle filters.

Systematic derivation of inference algorithms [Wiberg, 1996]

- ① Draw factor graph of pdf $p(x, y, z; \theta, \varphi)$.
- ② Apply sum-product rule at each node.
- ③ If sum-product rule is **infeasible** at a certain node, then apply an **approximation** = choose appropriate **message types**.
- ④ Choose a message update schedule.

Kalman filter

Linear system perturbed by additive Gaussian noise

Let X_1, X_2, \dots be a real random process defined by:

$$X_k = a_1 X_{k-1} + a_2 X_{k-2} + \dots + a_M X_{k-M} + U_k, \quad U_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_U^2)$$

and let the process $Y = Y_1, Y_2, \dots$ be defined as:

$$Y_k = X_k + W_k, \quad W_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_W^2).$$

Task

Estimate X_1, X_2, \dots from observation Y .

Remark

$a_1, \dots, a_M, \sigma_U^2$, and σ_W^2 assumed to be **known**.

Kalman filter

Linear system perturbed by additive Gaussian noise

Let X_1, X_2, \dots be a real random process defined by:

$$X_k = a_1 X_{k-1} + a_2 X_{k-2} + \dots + a_M X_{k-M} + U_k, \quad U_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_U^2)$$

and let the process $Y = Y_1, Y_2, \dots$ be defined as:

$$Y_k = X_k + W_k, \quad W_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_W^2).$$

State space representation

$$\mathbf{X}_k = \mathbf{A} \mathbf{X}_{k-1} + \mathbf{b} U_k$$

$$Y_k = \mathbf{c}^T \mathbf{X}_k + W_k$$

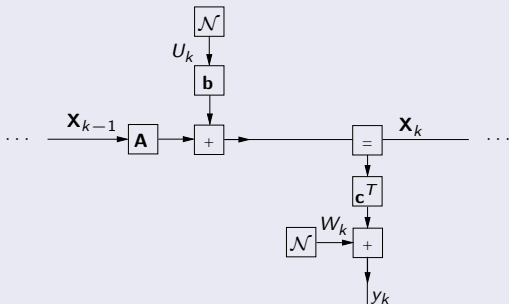
$$\mathbf{X}_k \triangleq [X_k, \dots, X_{k-M+1}]^T \quad \mathbf{A} \triangleq \begin{bmatrix} \mathbf{a}^T \\ \mathbf{I} \quad \mathbf{0} \end{bmatrix}$$

$$\mathbf{b} \triangleq \mathbf{c} \triangleq [1, 0, \dots, 0]^T \quad \mathbf{a} \triangleq [a_1, \dots, a_M]^T$$

Linear model: factor graph

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

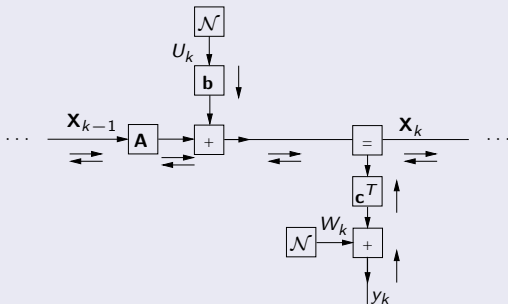
$$Y_k = \mathbf{c}^T\mathbf{X}_k + W_k.$$



Linear model: factor graph

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

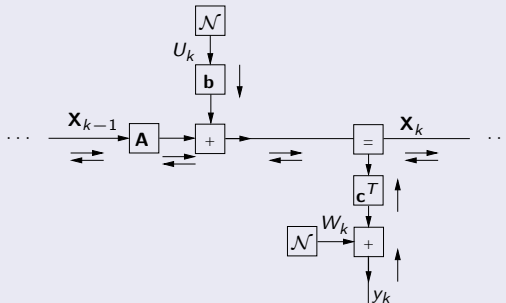
$$Y_k = \mathbf{c}^T\mathbf{X}_k + W_k.$$



Linear model: factor graph

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

$$Y_k = \mathbf{c}^T\mathbf{X}_k + W_k.$$



sum-product message passing = Kalman filtering/smoothing

messages = Gaussian distributions

Gaussian message update rules

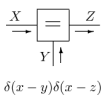
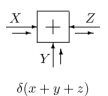
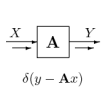
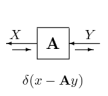
1	 $\delta(x - y)\delta(x - z)$	$m_Z = (\mathbf{W}_X + \mathbf{W}_Y)^\# (\mathbf{W}_X m_X + \mathbf{W}_Y m_Y)$ $\mathbf{V}_Z = \mathbf{V}_X (\mathbf{V}_X + \mathbf{V}_Y)^\# \mathbf{V}_Y$ $\mathbf{W}_Z = \mathbf{W}_X + \mathbf{W}_Y$
2	 $\delta(x + y + z)$	$m_Z = -m_X - m_Y$ $\mathbf{V}_Z = \mathbf{V}_X + \mathbf{V}_Y$ $\mathbf{W}_Z = \mathbf{W}_X (\mathbf{W}_X + \mathbf{W}_Y)^\# \mathbf{W}_Y$
3	 $\delta(y - \mathbf{A}x)$	$m_Y = \mathbf{A} m_X$ $\mathbf{V}_Y = \mathbf{A} \mathbf{V}_X \mathbf{A}^H$ $\mathbf{W}_Y \stackrel{!}{=} \mathbf{A}^{-H} \mathbf{W}_X \mathbf{A}^{-1}$
4	 $\delta(x - \mathbf{A}y)$	$m_Y = (\mathbf{A}^H \mathbf{W}_X \mathbf{A})^\# \mathbf{A}^H \mathbf{W}_X m_X$ $\mathbf{V}_Y \stackrel{!}{=} \mathbf{A}^{-1} \mathbf{V}_X \mathbf{A}^{-H}$ $\mathbf{W}_Y = \mathbf{A}^H \mathbf{W}_X \mathbf{A}$
¹ if \mathbf{A} is invertible		

Table H.2: Computation of multi-dimensional Gaussian messages consisting of mean vector m and covariance matrix \mathbf{V} or $\mathbf{W} = \mathbf{V}^{-1}$. Notation: $(\cdot)^H$ denotes Hermitian transposition and $(\cdot)^\#$ denotes the Moore-Penrose pseudo-inverse.

Gaussian message update rules

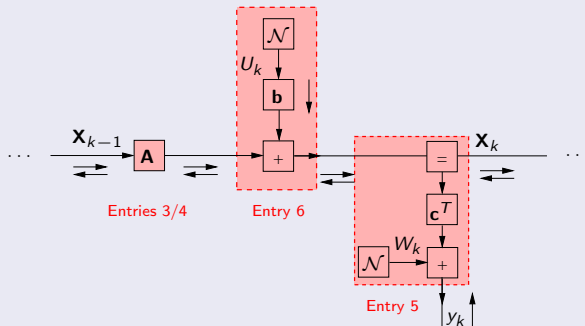
5		$m_Z = m_X + \mathbf{V}_X \mathbf{A}^H \mathbf{G} (m_Y - \mathbf{A} m_X)$ $\mathbf{V}_Z = \mathbf{V}_X - \mathbf{V}_X \mathbf{A}^H \mathbf{G} \mathbf{A} \mathbf{V}_X$ $\mathbf{W}_Z = \mathbf{W}_X + \mathbf{A}^H \mathbf{W}_Y \mathbf{A}$ <p>with $\mathbf{G} \triangleq (\mathbf{V}_Y + \mathbf{A} \mathbf{V}_X \mathbf{A}^H)^{-1}$</p>
6		$m_Z = -m_X - \mathbf{A} m_Y$ $\mathbf{V}_X \stackrel{!}{=} \mathbf{A}^{-1} \mathbf{V}_Y \mathbf{A}^{-H}$ $\mathbf{W}_Z = \mathbf{W}_X - \mathbf{W}_X \mathbf{A} \mathbf{H} \mathbf{A}^H \mathbf{W}_X$ <p>with $\mathbf{H} \triangleq (\mathbf{W}_Y + \mathbf{A}^H \mathbf{W}_X \mathbf{A})^{-1}$</p>
¹ if \mathbf{A} is invertible		

Table H.3: Update rules for composite blocks.

Linear model: factor graph

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

$$Y_k = \mathbf{c}^T\mathbf{X}_k + W_k$$



Example revisited

Example (AR model)

Let X_1, X_2, \dots be a real random process defined by:

$$X_k = a_1 X_{k-1} + a_2 X_{k-2} + \dots + a_M X_{k-M} + U_k, \quad U_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_U^2)$$

and let the process $Y = Y_1, Y_2, \dots$ be defined as:

$$Y_k = X_k + W_k, \quad W_k \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_W^2).$$

Task

Estimate a_1, \dots, a_M from observation Y .

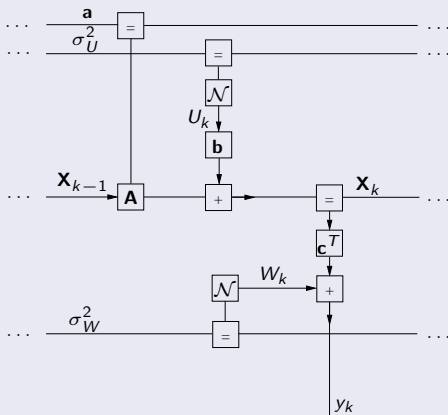
Remark

σ_U^2 , σ_W^2 , and X_k are **nuisance** parameters/random variables

AR model: factor graph

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

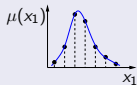
$$Y_k = \mathbf{c}^T \mathbf{X}_k + W_k.$$



Approximations vs. message types

Numerical Integration

$$\mu(y) \propto \sum_i \int_{x_2, \dots, x_m} h(y, \hat{x}_1^{(i)}, x_2, \dots, x_m) \mu(x_2) \cdots \mu(x_m) dx_2 \dots dx_m$$



Decision based

$$\mu(y) \propto \int_{x_2, \dots, x_m} h(y, \hat{x}_1, x_2, \dots, x_m) \mu(x_2) \cdots \mu(x_m) dx_2 \dots dx_m$$



Gaussian approximation

$$\mu(y) \propto \int_{x_2, \dots, x_m} h(y, x_1, x_2, \dots, x_m) \mathcal{N}(x_1) \mu(x_2) \cdots \mu(x_m) dx_2 \dots dx_m$$



Particle method

$$\mu(y) \propto \sum_i \int_{x_2, \dots, x_m} h(y, \hat{x}_1^{(i)}, x_2, \dots, x_m) w_1^{(i)} \mu(x_2) \cdots \mu(x_m) dx_2 \dots dx_m$$



AR model: decision-based inference

Message types

- **Single value** approximation for \mathbf{a} , σ_U^2 , and σ_W^2 .
- Messages in $\mathbf{X}_k =$ **Gaussian distributions**

Decision-based inference algorithms

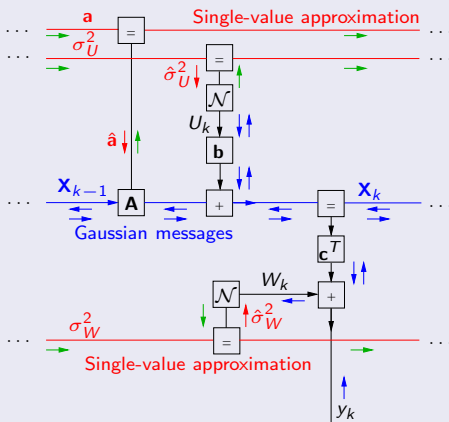
The estimates $\hat{\mathbf{a}}$, $\hat{\sigma}_U^2$, and $\hat{\sigma}_W^2$ determined by

- Gradient methods (e.g., steepest ascent/descent)
- **Expectation Maximization (EM)**
- Gradient EM.

AR model: factor graph

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

$$Y_k = \mathbf{c}^T \mathbf{X}_k + W_k.$$



Expectation Maximization

Task

$$\hat{\theta}_{\max} \triangleq \operatorname{argmax}_{\theta} f(\theta),$$

where

$$f(\theta) \triangleq \sum_x f(x, \theta).$$

e.g., $f(\mathbf{a}, \sigma_U^2, \sigma_W^2) = p(y, x; \mathbf{a}, \sigma_U^2, \sigma_W^2)$

EM

① Make **initial guess** $\hat{\theta}^{(0)}$

② **Expectation** step

$$f^{(\ell)}(\theta) \triangleq \sum_x f(x, \hat{\theta}^{(\ell)}) \log f(x, \theta)$$

③ **Maximization** step

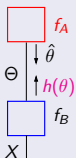
$$\hat{\theta}^{(\ell+1)} \triangleq \operatorname{argmax}_{\theta} f^{(\ell)}(\theta)$$

④ **Repeat** 2–3 until convergence.

Expectation Maximization (2)

Simple factorization

$$f(x, \theta) \triangleq f_A(\theta) f_B(x, \theta)$$



Message passing

Upwards message $h(\theta)$

$$h(\theta) = \frac{\sum_x f_B(x, \hat{\theta}^{(\ell)}) \log f_B(x, \theta)}{\sum_x f_B(x, \hat{\theta}^{(\ell)})}$$

$$\triangleq E_{p_B}[\log f_B(x, \theta)]$$

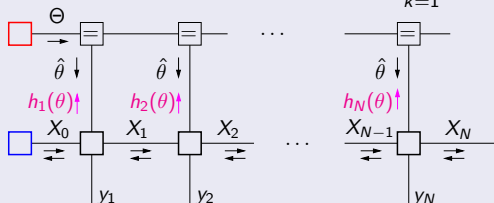
Downwards message $\hat{\theta}^{(\ell+1)}$

$$\hat{\theta}^{(\ell+1)} = \underset{\theta}{\operatorname{argmax}} (\log f_A(\theta) + h(\theta))$$

Expectation Maximization (3)

State space model

$$f(x_0, x_1, \dots, x_N, y_1, y_2, \dots, y_N, \theta) \triangleq f_\theta(\theta) f_0(x_0) \prod_{k=1}^N f(y_k, x_k | x_{k-1}, \theta)$$



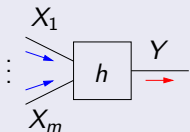
Upwards message $h(\theta)$

$$h(\theta) = \sum_{k=1}^N h_k(\theta_k) = \sum_{k=1}^N \mathbb{E} \left[\log f(y_k, x_k | x_{k-1}, \theta) | \hat{\theta}^{(\ell)} \right]$$

$$p(x_{k-1}, x_k | \hat{\theta}^{(\ell)}) \triangleq \frac{f(y_k, x_k | x_{k-1}, \hat{\theta}^{(\ell)}) \mu_{x_k \rightarrow f}(x_k) \mu_{x_{k-1} \rightarrow f}(x_{k-1})}{\sum_{x_{k-1}, x_k} f(y_k, x_k | x_{k-1}, \hat{\theta}^{(\ell)}) \mu_{x_k \rightarrow f}(x_k) \mu_{x_{k-1} \rightarrow f}(x_{k-1})}$$

Sum-product algorithm (“belief propagation”)

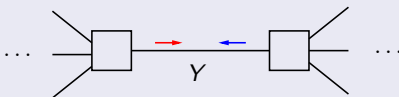
- 1 Compute all messages (starting at the leaves)



$$\mu(y) \propto \sum_{x_1, \dots, x_m} h(y, x_1, \dots, x_m) \mu(x_1) \cdots \mu(x_m)$$

- 2 Compute marginals

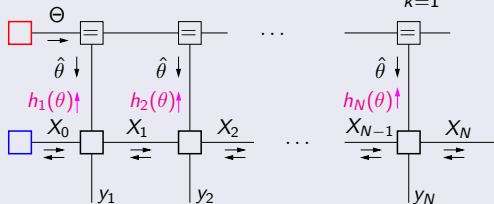
$$p(y) \propto \mu_{\rightarrow}(y) \mu_{\leftarrow}(y)$$



Expectation Maximization (3)

State space model

$$f(x_0, x_1, \dots, x_N, y_1, y_2, \dots, y_N, \theta) \triangleq f_\theta(\theta) f_0(x_0) \prod_{k=1}^N f(y_k, x_k | x_{k-1}, \theta)$$



Upwards message $h(\theta)$

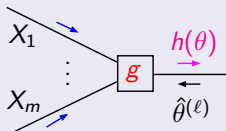
$$h(\theta) = \sum_{k=1}^N h_k(\theta_k) = \sum_{k=1}^N \mathbb{E} \left[\log f(y_k, x_k | x_{k-1}, \theta) | \hat{\theta}^{(\ell)} \right]$$

Downwards message $\hat{\theta}^{(\ell+1)}$

$$\hat{\theta}^{(\ell+1)} = \underset{\theta}{\operatorname{argmax}} \left(\log f_\theta(\theta) + \sum_{k=1}^N h_k(\theta) \right)$$

Expectation Maximization (4)

Generic E-log message



$$h(\theta) = \mathbb{E} \left[\log g(x_1, \dots, x_m, \theta) \mid \hat{\theta}^{(\ell)} \right],$$

where

$$p(x_1, \dots, x_m \mid \hat{\theta}^{(\ell)}) \triangleq \frac{\mu(x_1) \dots \mu(x_m) g(x_1, \dots, x_m, \hat{\theta}^{(\ell)})}{\sum_{x_1, \dots, x_m} \mu(x_1) \dots \mu(x_m) g(x_1, \dots, x_m, \hat{\theta}^{(\ell)})}.$$

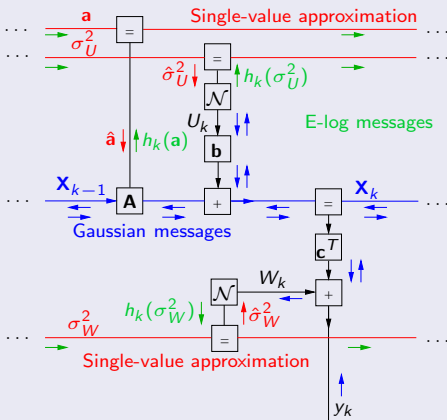
$\mu(x_k)$ are sum-product messages (or approximations, e.g., particle lists, Gaussian distributions)

Expectation Maximization (4)

AR model

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

$$Y_k = \mathbf{c}^T \mathbf{X}_k + W_k.$$



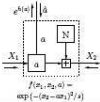
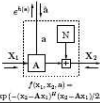
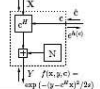
Computation rules for h -messages

Table 4.2: EM Update equations for standard building blocks.

	Graph	Node	EM update rule	
1	$ \begin{array}{c} e^{h(s)} \updownarrow \hat{m} \\ \boxed{\mathcal{N}} \\ \updownarrow X \\ f(x, m) = \mathcal{N}(x m, s) \end{array} $	Gaussian, unknown mean, scalar	$e^{h(m)} \propto \mathcal{N}(m E[X], s)$	$X, m \in \mathbb{R}$ $s \in \mathbb{R}_0^+$
2	$ \begin{array}{c} e^{h(\mathbf{m})} \updownarrow \hat{\mathbf{m}} \\ \boxed{\mathcal{N}} \\ \updownarrow \mathbf{X} \\ f(x, \mathbf{m}) = \mathcal{N}(x \mathbf{m}, \mathbf{V}) \end{array} $	Gaussian, unknown mean	$e^{h(\mathbf{m})} \propto \mathcal{N}(\mathbf{m} E[\mathbf{X}], \mathbf{V})$	$\mathbf{X}, \mathbf{m} \in \mathbb{R}^n$ $\mathbf{V} \in \mathbb{R}^{n \times n}$ $\mathbf{V} \succeq 0$
3	$ \begin{array}{c} e^{h(s)} \updownarrow \hat{s} \\ \boxed{\mathcal{N}} \\ \updownarrow X \\ f(x, s) = \mathcal{N}(x m, s) \end{array} $	Gaussian, unknown variance	$e^{h(s)} \propto \text{Ig}\left(s \mid -\frac{1}{2}, \frac{E[X^2] - 2mE[X] + m^2}{2}\right)$	$X, m \in \mathbb{R}$ $s \in \mathbb{R}_0^+$
4	$ \begin{array}{c} e^{h(\mathbf{V})} \updownarrow \hat{\mathbf{V}} \\ \boxed{\mathcal{N}} \\ \updownarrow \mathbf{X} \\ f(x, \mathbf{V}) = \mathcal{N}(x \mathbf{m}, \mathbf{V}) \end{array} $	Gaussian, unknown variance	$e^{h(\mathbf{V})} \propto \exp(-E[(\mathbf{X} - \mathbf{m})^H \mathbf{V}^{-1} (\mathbf{X} - \mathbf{m})])$	$\mathbf{X}, \mathbf{m} \in \mathbb{R}^n$ $\mathbf{V} \in \mathbb{R}^{n \times n}$ $\mathbf{V} \succeq 0$
5		Identity covariance matrix	$e^{h(s)} \propto \text{Ig}\left(s \mid \frac{n-2}{2}, \frac{1}{2}E[(\mathbf{X} - \mathbf{m})^H (\mathbf{X} - \mathbf{m})]\right)$	$\mathbf{X}, \mathbf{m} \in \mathbb{R}^n$ $\mathbf{V} = \mathbf{I} s$ $s \in \mathbb{R}_0^+$

Computation rules for h -messages

Table 4.2: (continued)

	Graph	Node	EM update rule	
6		Diagonal covariance matrix	$e^{h(s)} \propto \prod_{\ell=1}^n \text{Ig} \left(s_{\ell} \mid -\frac{1}{2}, \frac{1}{2} \mathbb{E}[(X_{\ell} - m_{\ell})^2] \right)$	$\mathbf{X}, \mathbf{m} \in \mathbb{R}^n$ $\mathbf{V} = \text{diag}(\mathbf{s})$ $\mathbf{s} \in \mathbb{R}^{+n}$
7		Scalar multiplication	$e^{h(a)} \propto \mathcal{N}^{-1} \left(a \mid \frac{\mathbb{E}[X_1 X_2], \mathbb{E}[X_1^2]}{\mathbb{E}[X_1^2]}, \frac{\mathbb{E}[X_1^2]}{s} \right)$	$X_1, X_2 \in \mathbb{R}$ $a \in \mathbb{R}$ $s \in \mathbb{R}_0^+$
8		Auto-regression	$e^{h(a)} \propto \mathcal{N}^{-1} \left(a \mid \mathbb{E}[\mathbf{X}_1 \mathbf{X}_1^H]^{-1} \mathbb{E}[\mathbf{X}_1 X_2], \frac{\mathbb{E}[\mathbf{X}_1 \mathbf{X}_1^H]}{s} \right)$	$X_1, X_2 \in \mathbb{R}^n$ $\mathbf{a} \in \mathbb{R}^n, s \in \mathbb{R}_0^+$ $\mathbf{A} = [\mathbf{a}^H; \mathbf{I} \ 0]$ $X_2 = [X_2]_1$
9		Inner vector product	$e^{h(c)} \propto \mathcal{N}^{-1} \left(c \mid \mathbb{E}[\mathbf{X} \mathbf{X}^H]^{-1} \mathbb{E}[\mathbf{X} Y], \frac{\mathbb{E}[\mathbf{X} \mathbf{X}^H]}{s} \right)$	$\mathbf{X}, \mathbf{c} \in \mathbb{R}^n$ $Y \in \mathbb{R}$ $s \in \mathbb{R}_0^+$

Computation rules for h -messages

Table 4.2: (continued)

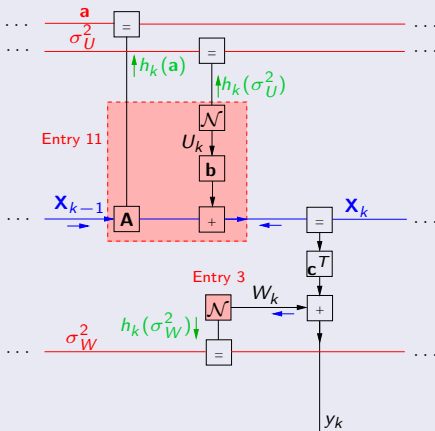
	Graph	Node	EM update rule	
10	<p>$f(x_1, x_2, a, s) = \exp(-[x_2 - ax_1]^2/2s)$</p>	Joint coefficient/ variance estimation scalar	$e^{h(a)} \propto \mathcal{N}^{-1}\left(a \mid \frac{E[X_1 X_2], E[X_1^2]}{E[X_1^2]}, \frac{E[X_1^2]}{\hat{s}}\right)$ $e^{h(s)} \propto \text{Ig}\left(s \mid -\frac{1}{2}, \frac{E[(X_2 - \hat{a}X_1)^2]}{2}\right)$	$X_1, X_2 \in \mathbb{R}$ $a, \hat{a} \in \mathbb{R}$ $s, \hat{s} \in \mathbb{R}_0^+$
11	<p>$f(x_1, x_2, a, s) = \exp(-[x_2 - ax_1]^2/2s)$</p>	Joint coefficient/ variance estimation Auto- regression	$e^{h(a)} \propto \mathcal{N}^{-1}\left(a \mid E[X_1 X_1^H]^{-1} E[X_1 X_2], \frac{E[X_1 X_1^H]}{\hat{s}}\right)$ $e^{h(s)} \propto \text{Ig}\left(s \mid -\frac{1}{2}, \frac{E[(X_2 - \hat{A}X_1)^H (X_2 - \hat{A}X_1)]}{2}\right)$	$X_1, X_2, a \in \mathbb{R}^n$ $s, \hat{s} \in \mathbb{R}_0^+$ $X_2 = [X_2]_1$
12	<p>$f(x_1, x_2, A) = a_{x_1, x_2}$</p>	Finite state machine	$h(\mathbf{A}) = \sum_{x_1, x_2} p(x_1, x_2) \log a_{x_1, x_2}$	$X_1, X_2 \in \mathbf{Z}_n$ $a_{ij} \in [0, 1]$

Expectation Maximization (4)

AR model

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

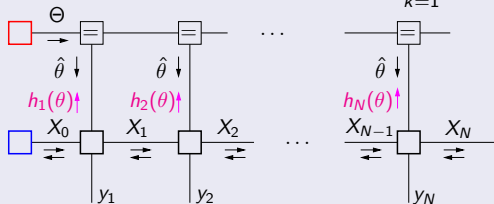
$$Y_k = \mathbf{c}^T \mathbf{X}_k + W_k.$$



Expectation Maximization (3)

State space model

$$f(x_0, x_1, \dots, x_N, y_1, y_2, \dots, y_N, \theta) \triangleq f_\theta(\theta) f_0(x_0) \prod_{k=1}^N f(y_k, x_k | x_{k-1}, \theta)$$



Upwards message $h(\theta)$





$$h(\theta) = \sum_{k=1}^N h_k(\theta_k) = \sum_{k=1}^N \mathbb{E} \left[\log f(y_k, x_k | x_{k-1}, \theta) | \hat{\theta}^{(\ell)} \right]$$

Downwards message $\hat{\theta}^{(\ell+1)}$

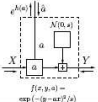
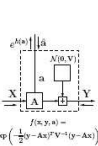
$$\hat{\theta}^{(\ell+1)} = \underset{\theta}{\operatorname{argmax}} \left(\log f_\theta(\theta) + \sum_{k=1}^N h_k(\theta) \right)$$

Computation rules for $\hat{\theta}$

Table 3: $\hat{\theta}$ -message computation rules for standard building blocks with standard prior $\Theta_1 = \Theta_2 = \dots = \Theta_N$.

	Graph	Node	$\hat{\theta}$ -message computation rule	
1		Gaussian, unknown mean, scalar	$\hat{m}^{(k+1)} = \frac{\sum_{\ell=1}^N s_{\ell}^{-1} \mathbb{E}[X_{\ell} \hat{m}^{(k)}]}{\sum_{\ell=1}^N s_{\ell}^{-1}}$	$X_{\ell}, m \in \mathbb{R}$ $s_{\ell} \in \mathbb{R}^+$
2		Gaussian, unknown mean, vector	$\hat{\mathbf{m}}^{(k+1)} = \frac{\sum_{\ell=1}^N \mathbf{V}_{\ell}^{-1} \mathbb{E}[X_{\ell} \hat{\mathbf{m}}^{(k)}]}{\sum_{\ell=1}^N \mathbf{V}_{\ell}^{-1}}$	$\mathbf{X}_{\ell}, \mathbf{m} \in \mathbb{R}^n$ $\mathbf{V}_{\ell} \in \mathbb{R}^{n \times n}$ $\mathbf{V}_{\ell} \succeq 0$
3		Gaussian, unknown (scalar) variance	$\hat{s}^{(k+1)} = \frac{1}{N} \sum_{\ell=1}^N \mathbb{E}[(X_{\ell} - m_{\ell})^2 \hat{s}^{(k)}]$	$X_{\ell}, m_{\ell} \in \mathbb{R}$ $s \in \mathbb{R}^+$
4		Gaussian, unknown covariance matrix	$\hat{\mathbf{V}}^{(k+1)} = \frac{1}{N} \sum_{\ell=1}^N \mathbb{E}[(\mathbf{X}_{\ell} - \mathbf{m}_{\ell})(\mathbf{X}_{\ell} - \mathbf{m}_{\ell})^T \hat{\mathbf{V}}^{(k)}]$	$\mathbf{X}_{\ell}, \mathbf{m}_{\ell} \in \mathbb{R}^n$ $\mathbf{V} \in \mathbb{R}^{n \times n}$ $\mathbf{V} \succeq 0$

Computation rules for $\hat{\theta}$

7		Scalar multiplication	$e^{h(a)} \propto \mathcal{N}^{-1}\left(a \mid \frac{E[XY]}{E[X^2]}, \frac{E[X^2]}{s}\right)$	$X, Y \in \mathbb{R}$ $a \in \mathbb{R}$ $s \in \mathbb{R}^+$
8		Auto-regression General covariance matrix	$e^{h(a)} \propto \mathcal{N}^{-1}\left(\mathbf{a} \mid \mathbf{m}_a, \mathbf{W}_a\right)$ $\mathbf{W}_a = w_1 \mathbf{E}[\mathbf{X}\mathbf{X}^T \mid \hat{\mathbf{a}}^{(k)}]$ $\mathbf{m}_a = \mathbf{W}_a^{-1} \left(\sum_{k=1}^n w_k \mathbf{E}[\mathbf{X}Y_k] - \sum_{k=1}^{n-1} w_{k+1} \mathbf{E}[\mathbf{X}X_k] \right)$	$\mathbf{X}, \mathbf{Y} \in \mathbb{R}^n$ $X_k = [\mathbf{X}]_k$ $Y_k = [\mathbf{Y}]_k$ $\mathbf{A} = [\mathbf{a}^T; \mathbf{I} \ 0]$ $\mathbf{V} \in \mathbb{R}^{n \times n}$ $\mathbf{V} \succ 0, \mathbf{a} \in \mathbb{R}^n$ $w_k = [\mathbf{V}^{-1}]_{1k}$
9		Diagonal covariance matrix	$e^{h(a)} \propto \mathcal{N}^{-1}\left(\mathbf{a} \mid \mathbf{E}[\mathbf{X}\mathbf{X}^T]^{-1} \mathbf{E}[\mathbf{X}\mathbf{Y}], \frac{\mathbf{E}[\mathbf{X}\mathbf{X}^T]}{s_1}\right)$	$\mathbf{X}, \mathbf{Y} \in \mathbb{R}^n$ $Y = [\mathbf{Y}]_1$ $\mathbf{A} = [\mathbf{a}^T; \mathbf{I} \ 0]$ $\mathbf{V} = \text{diag}(s)$ $s \in \mathbb{R}^{+n}, \mathbf{a} \in \mathbb{R}^n$

Computation rules for $\hat{\theta}$

Table 3: (continued)

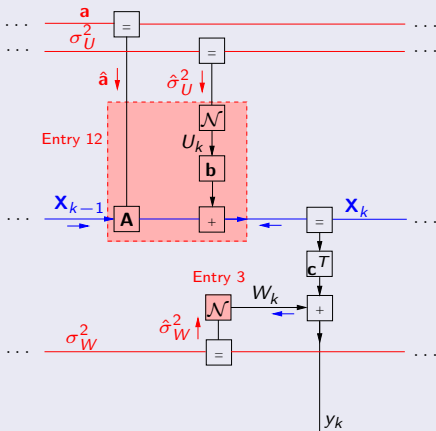
	Graph	Node	$\hat{\theta}$ -message computation rule	
10		Auto-regression Identity covariance matrix	$\hat{\mathbf{a}}^{(k+1)} = \left(\sum_{\ell=1}^N s_{\ell}^{-1} \mathbf{E} \left[\mathbf{X}_{\ell} \mathbf{X}_{\ell}^T \hat{\mathbf{a}}^{(k)} \right] \right)^{-1} \cdot \left(\sum_{\ell=1}^N s_{\ell}^{-1} \mathbf{E} \left[\mathbf{X}_{\ell} Y_{\ell} \hat{\mathbf{a}}^{(k)} \right] \right)$	$\mathbf{X}_{\ell}, Y_{\ell} \in \mathbb{R}^n$ $Y_{\ell} = [Y_{\ell}]_1$ $\mathbf{V}_{\ell} = \mathbf{I}_s$ $\mathbf{A} = [\mathbf{a}^T; \mathbf{I}_0]$ $s_{\ell} \in \mathbb{R}^+, \mathbf{a} \in \mathbb{R}^n$
11		Inner vector product	$\hat{\mathbf{c}}^{(k+1)} = \left(\sum_{\ell=1}^N s_{\ell}^{-1} \mathbf{E} [\mathbf{X}_{\ell} \mathbf{X}_{\ell}^T] \right)^{-1} \left(\sum_{\ell=1}^N s_{\ell}^{-1} \mathbf{E} [\mathbf{X}_{\ell} Y_{\ell}] \right)$	$\mathbf{X}_{\ell}, \mathbf{c} \in \mathbb{R}^n$ $Y_{\ell} \in \mathbb{R}$ $s_{\ell} \in \mathbb{R}^+$
12		Coefficient + variance	$\hat{a}^{(k+1)} = \frac{1}{N} \sum_{\ell=1}^N \mathbf{E} \left[X_{\ell} Y_{\ell} \hat{a}^{(k)}, \hat{s}^{(k)} \right]$ $\hat{s}^{(k+1)} = \frac{1}{N} \sum_{\ell=1}^N \mathbf{E} \left[(Y_{\ell} - \hat{a}^{(k+1)} X_{\ell})^2 \hat{a}^{(k)}, \hat{s}^{(k)} \right]$	$X_{\ell}, Y_{\ell} \in \mathbb{R}$ $a \in \mathbb{R}$ $s \in \mathbb{R}^+$
13		Coefficient + covariance matrix	$\hat{\mathbf{a}}^{(k+1)} = \frac{\sum_{\ell=1}^N \mathbf{E} [\mathbf{X}_{\ell} Y_{\ell} \hat{\mathbf{a}}^{(k)}, \hat{s}^{(k)}]}{\sum_{\ell=1}^N \mathbf{E} [\mathbf{X}_{\ell} \mathbf{X}_{\ell}^T \hat{\mathbf{a}}^{(k)}, \hat{s}^{(k)})]}$ $\hat{s}^{(k+1)} = \frac{1}{nN} \sum_{\ell=1}^N \mathbf{E} \left[(\mathbf{Y} - \mathbf{S} \mathbf{X} - \mathbf{c} \mathbf{X}^T \hat{\mathbf{a}}^{(\ell)})^T (\mathbf{Y} - \mathbf{S} \mathbf{X} - \mathbf{c} \mathbf{X}^T \hat{\mathbf{a}}^{(\ell)}) \right]$	$\mathbf{X}_{\ell}, Y_{\ell}, \mathbf{a} \in \mathbb{R}^n$ $s \in \mathbb{R}^+$ $Y_{\ell} = [Y_{\ell}]_1$
14		Finite state machine	$\hat{A}_{ij}^{(k+1)} = \frac{\sum_{\ell=1}^N p(x_{\ell} = i, y_{\ell} = j \hat{\mathbf{A}}^{(k)})}{\sum_{\ell=1}^N \sum_j p(x_{\ell} = i, y_{\ell} = j \hat{\mathbf{A}}^{(k)})}$	$X_{\ell}, Y_{\ell} \in \mathbb{Z}_N$ $A_{ij} \in [0, 1]$ $(i, j = 1, \dots, n)$

Expectation Maximization (4)

AR model

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

$$Y_k = \mathbf{c}^T \mathbf{X}_k + W_k.$$

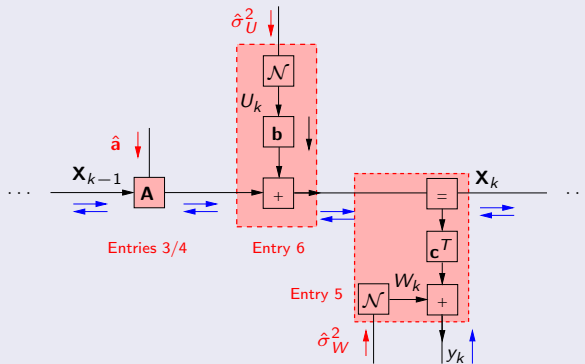


Expectation Maximization (4)

AR model: E-step

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

$$Y_k = \mathbf{c}^T\mathbf{X}_k + W_k.$$

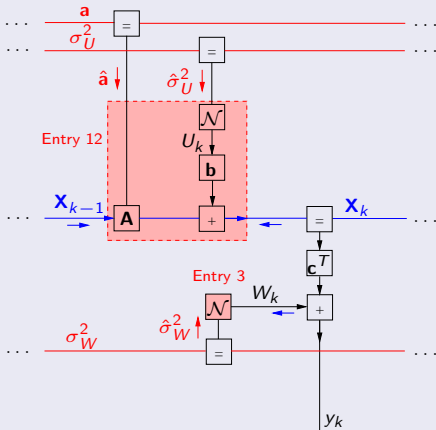


Expectation Maximization (4)

AR model: M-step

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

$$Y_k = \mathbf{c}^T \mathbf{X}_k + W_k.$$

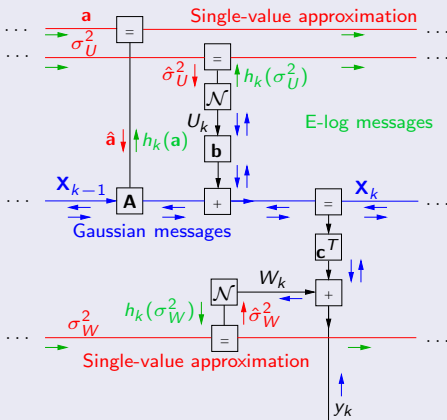


Expectation Maximization (4)

AR model

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

$$Y_k = \mathbf{c}^T\mathbf{X}_k + W_k.$$



Gradient EM

Task

$$\theta_{\max} \triangleq \operatorname{argmax}_{\theta} f(\theta),$$

where

$$f(\theta) \triangleq \sum_x f(x, \theta).$$

e.g., $f(\mathbf{a}, \sigma_U^2, \sigma_W^2) = p(y, x; \mathbf{a}, \sigma_U^2, \sigma_W^2)$

EM

① Make **initial guess** $\hat{\theta}^{(0)}$

② **Expectation** step

$$f^{(\ell)}(\theta) \triangleq \sum_x f(x, \hat{\theta}^{(\ell)}) \log f(x, \theta)$$

③ **Gradient** step

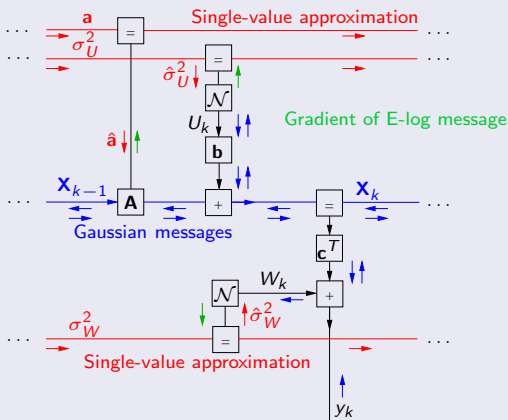
$$\hat{\theta}^{(\ell+1)} \triangleq \hat{\theta}^{(\ell)} + \lambda_{\ell} \nabla_{\theta} f^{(\ell)}(\theta) \big|_{\hat{\theta}^{(\ell)}}.$$

④ **Repeat** 2–3 until convergence.

AR model: gradient EM

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

$$Y_k = \mathbf{c}^T \mathbf{X}_k + W_k.$$



Steepest Ascent

Task

$$\theta_{\max} \triangleq \operatorname{argmax}_{\theta} f(\theta),$$

where

$$f(\theta) \triangleq \sum_x f(x, \theta).$$

e.g., $f(\mathbf{a}, \sigma_U^2, \sigma_W^2) = p(y, x; \mathbf{a}, \sigma_U^2, \sigma_W^2)$

Steepest Ascent

- 1 Make **initial guess** $\hat{\theta}^{(0)}$
- 2 **Gradient** step

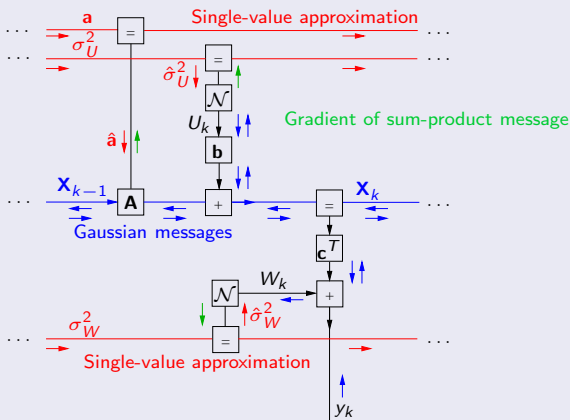
$$\hat{\theta}^{(\ell+1)} \triangleq \hat{\theta}^{(\ell)} + \lambda_{\ell} \nabla_{\theta} \log f(\theta) \Big|_{\hat{\theta}^{(\ell)}}.$$

- 3 **Repeat** 2–3 until convergence.

AR model: steepest ascent

$$\mathbf{X}_k = \mathbf{A}\mathbf{X}_{k-1} + \mathbf{b}U_k$$

$$Y_k = \mathbf{c}^T \mathbf{X}_k + W_k.$$



Conclusion (1)

Inference/Learning by means of graphical models

- 1 Draw factor graph
- 2 Apply sum-product rule to each node
- 3 Choose message types (for continuous variables)
- 4 Choose message update schedule

Conclusion (2)

Divide and conquer

Global estimation/detection problem solved by **simple local** computations.

Disciplined approach

Deriving novel algorithms **systematically** by listing possible message update rules at each node in the graph.

Mix and match

Straightforward to **combine** several approaches, e.g., decision-based, particle-based etc., in a single algorithm.

Plug and play

Novel algorithms by combining **tabulated** message update rules.