ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Signal and Information Processing Laboratory
Institut für Signal- und Informationsverarbeitung

Some Remarks on Factor Graphs

Hans-Andrea Loeliger

Why Factor Graphs?

Factor graphs are a "language" for system models.

Well known: A large variety of algorithms in coding, signal processing, and artificial intelligence can be explained as special cases of summary-product message passing on a factor graph: BCJR forward-backward algorithm, Viterbi algorithm, iterative decoding of turbo codes and low-density parity check codes, belief propagation in Bayesian networks, Kalman filtering and smoothing, ...

More important: Factor graphs are highly useful for the development of new algorithms for a wide variety of real-world detection/estimation problems. In particular, they allow to combine and to extend most prior art, including gradient methods, EM-type algorithms, particle filters, ...

Outline

- Forney-style factor graphs
- Some simple remarks
- Dealing with continuous variables
- On gradient methods

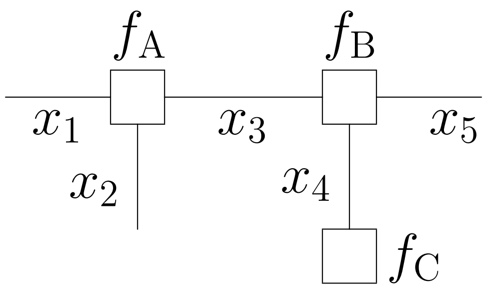
Forney-Style Factor Graphs (FFG)

represent the factorization of a function of several variables.

(D. Forney: "Codes on graphs: normal realizations", 2001)

Example:

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



Rules:

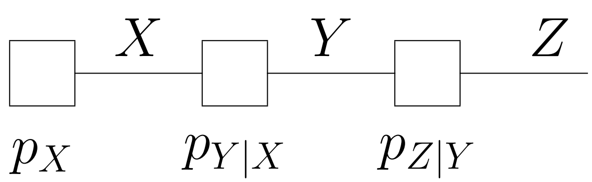
- A node for every factor.
- An edge or half-edge for every variable.
- Node g is connected to edge x iff variable x appears in factor g.

What if some variable appears in more than 2 factors?

Example:

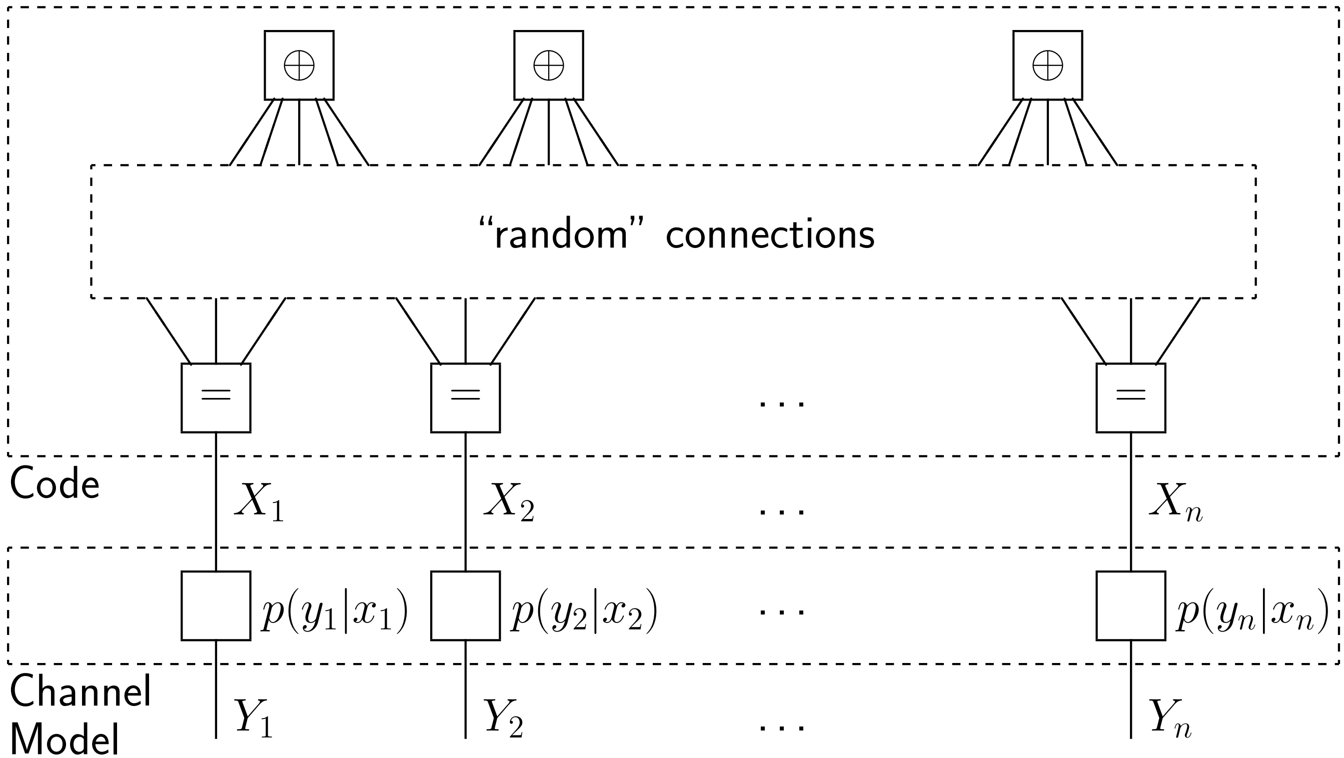
Markov Chain

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



Example:

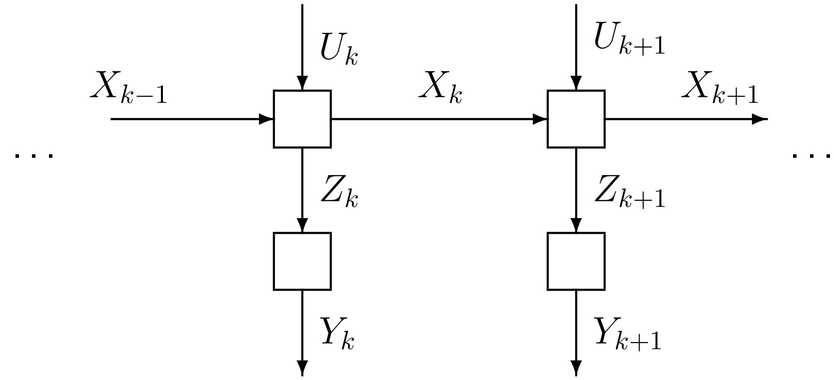
Low-Density Parity Check Code



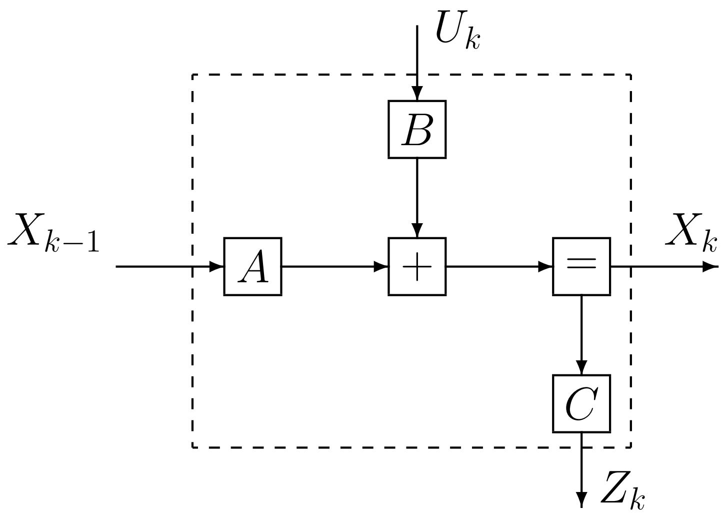
Example:

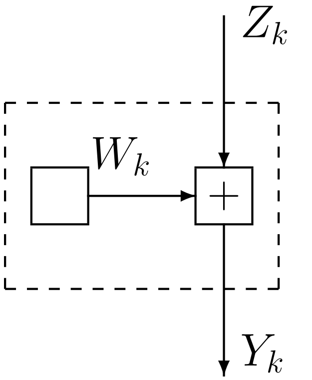
Classical Linear State Space Models

$$\begin{aligned} X_k &= AX_{k-1} + BU_k \\ Z_k &= CX_k \\ Y_k &= Z_k + W_k \end{aligned}$$



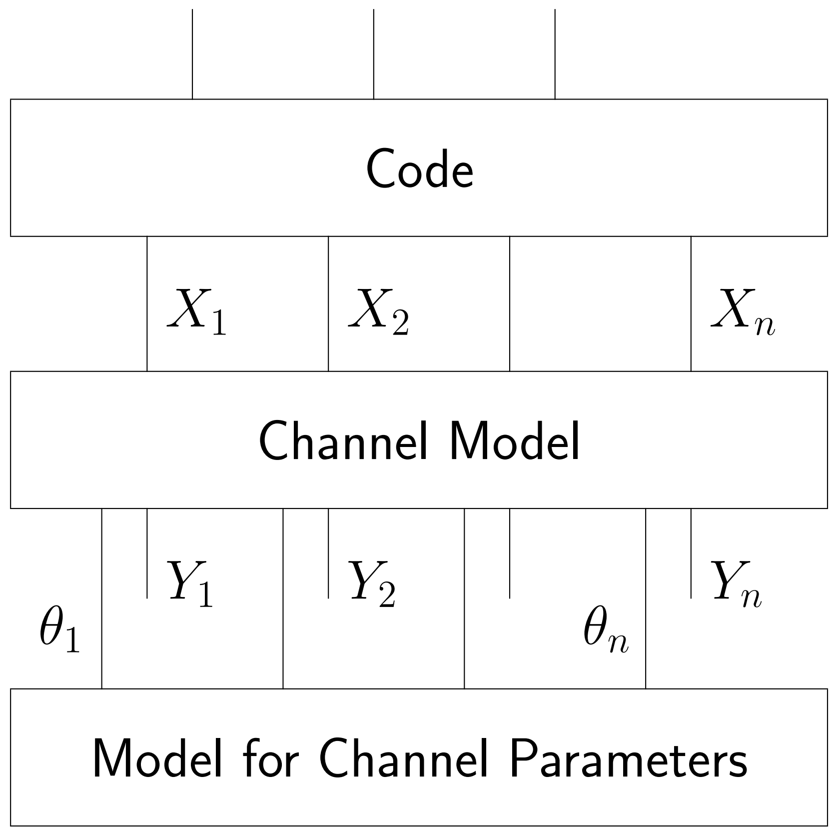
W_k = white Gaussian noise.



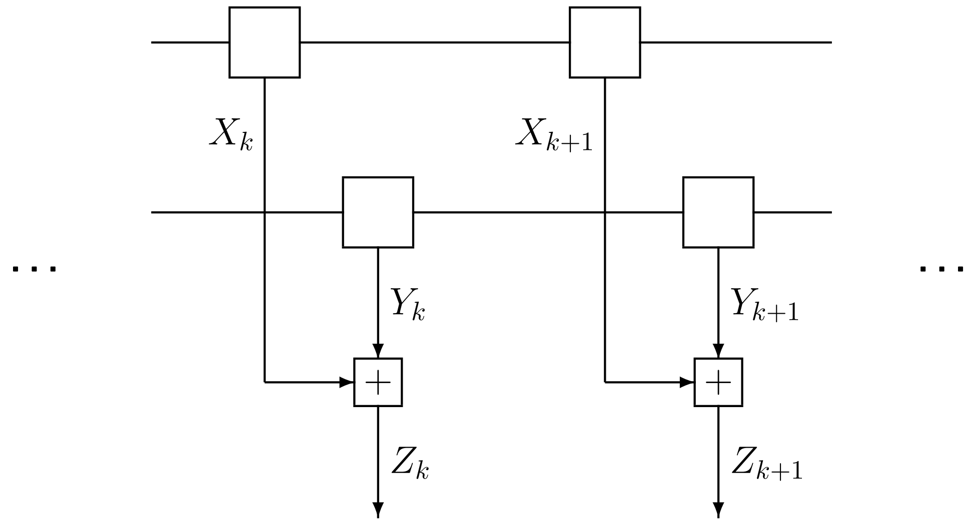


Example:

Channel with Unknown Parameters

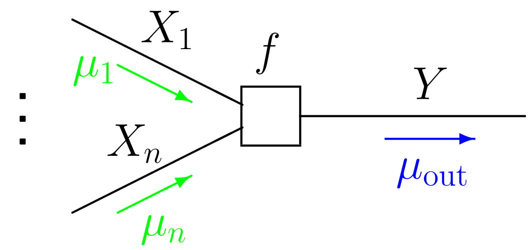


Superimposed Signals



Summary-Product Message Update Rule

Each message is a function (usually a scaled pdf) of the variable associated with the corresponding edge. This function is a summary of everything "behind" this edge.

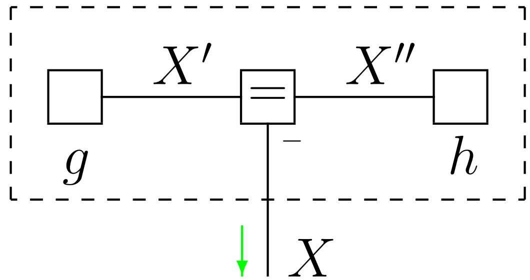


$$\mu_{\mathrm{out}}(y) \stackrel{\triangle}{=} \sum_{x_1} \ldots \sum_{x_n} f(x_1, \ldots, x_n, y)\, \mu_1(x_1) \cdots \mu_n(x_n)$$

or (e.g.)

$$\mu_{\mathrm{out}}(y) \stackrel{\triangle}{=} \max_{x_1} \ldots \max_{x_n} f(x_1, \ldots, x_n, y)\, \mu_1(x_1) \cdots \mu_n(x_n)$$

Example



Open box:

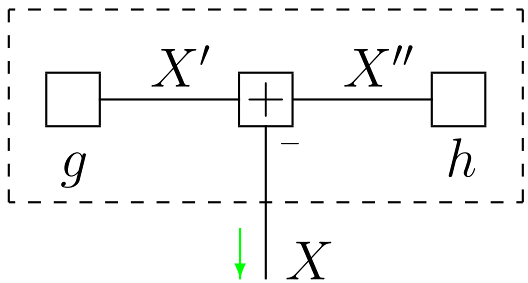
$$f(x, x', x'') = g(x')h(x'')\delta(x' - x)\delta(x'' - x)$$

$$X' = X'' = X \quad \text{for all valid configurations.}$$

Closed box (= message out of the box):

$$f(x) = \sum_{x'} \sum_{x''} g(x')h(x'')\delta(x' - x)\delta(x'' - x) = g(x)h(x)$$

Example: Addition / Convolution



Open box:

$$f(x, x', x'') = g(x')h(x'')\delta(x' + x'' - x)$$

$$X' + X'' = X \quad \text{for all valid configurations.}$$

Closed box (= message out of the box):

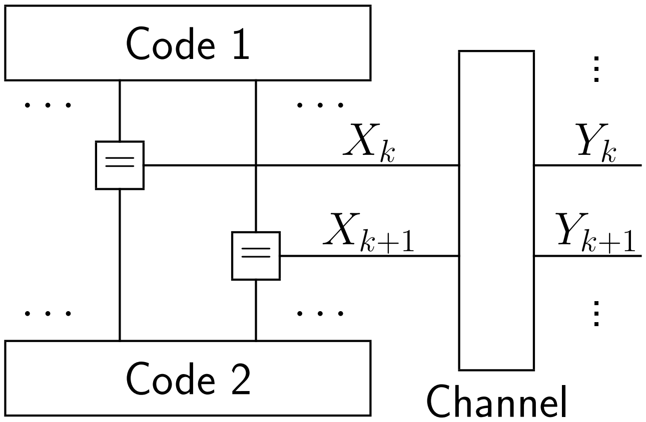
$$f(x) = \sum_{x'} \sum_{x''} g(x')h(x'')\delta(x' + x'' - x) = \sum_{x'} g(x')h(x - x')$$

Outline

- Forney-style factor graphs
- Some simple remarks
- Dealing with continuous variables
- On gradient methods

Simple Remark 1:

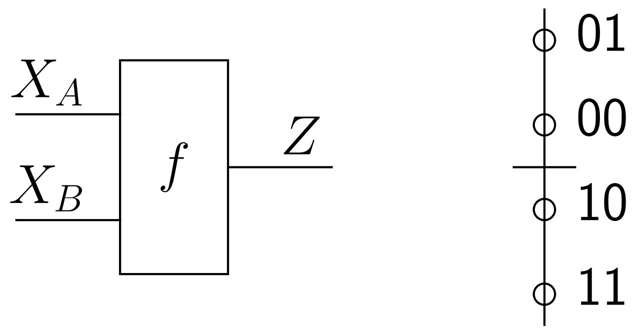
Combining Information



The correct handling of extrinsic/intrinsic information is automatic from the summary-product rule.

Simple Remark 2:

Mappers and Such

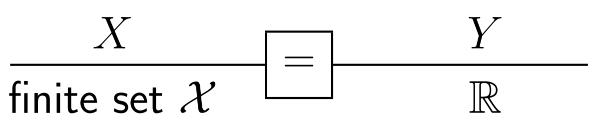


In the FFG, the mapper becomes a factor node with the local function

$$\delta_f(x_A, x_B, z) \triangleq \begin{cases} 1, & \text{if } f(x_A, x_B) = z \\ 0, & \text{else} \end{cases}$$

Simple Remark 3:

Hybrid Equality Node



$\delta(x-y)$ is a Kronecker delta in x and a Dirac delta in y.

Messages:

$\longrightarrow$ $\mu_{\mathrm{out}Y}(y) = \sum_{x\in\mathcal{X}} \delta(y-x)\,\mu_{\mathrm{in}X}(x)$,
a sum of weighted Dirac deltas.

$\longleftarrow$ $\mu_{\mathrm{out}X}(x) = \int_y \delta(x-y)\,\mu_{\mathrm{in}Y}(y) = \mu_{\mathrm{in}Y}(x)$,
sampling the incoming density $\mu_{\mathrm{in}Y}$ at the elements of $\mathcal{X}$.

Signal Processing by Message Passing

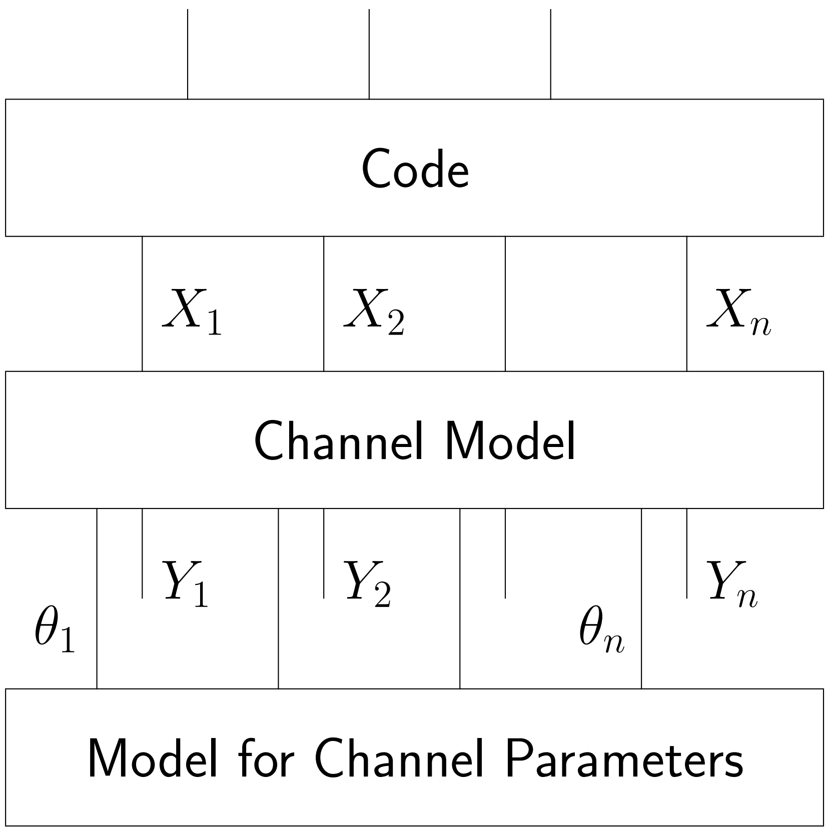
Design choices:

- Modeling = choice of graph.
- Choice of message types and of the corresponding update rules for continuous variables.
- Scheduling of the message computations.

Most good known signal processing techniques can be used, combined, and extended.

Example:

Channel with Unknown Parameters

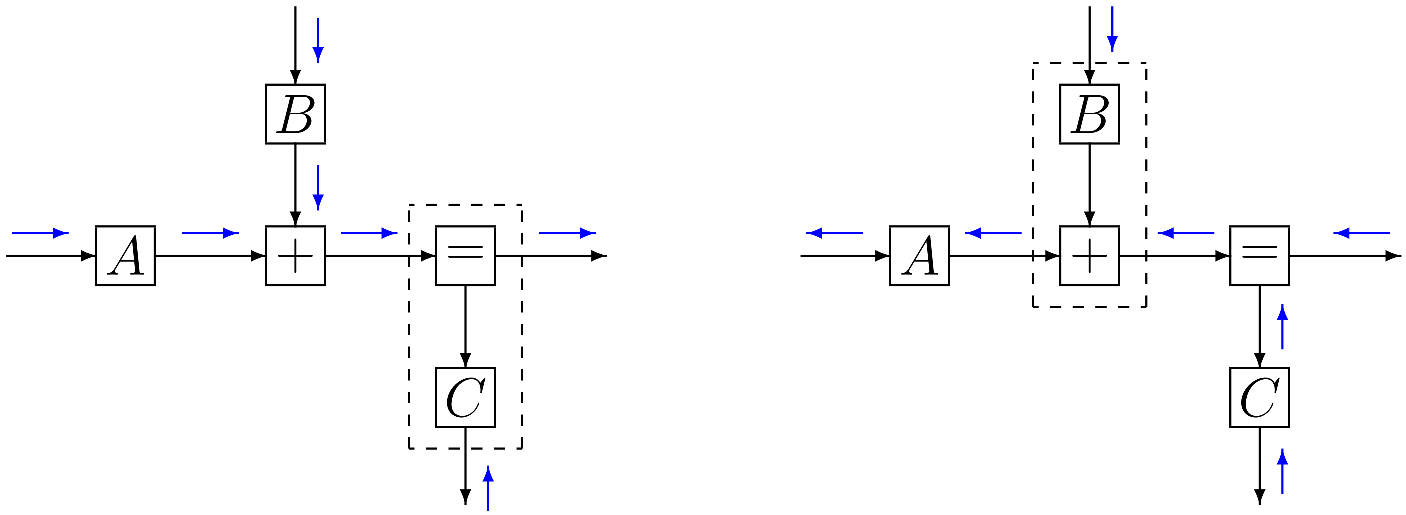


Continuous Variables: Message Types

The following message types are widely applicable.

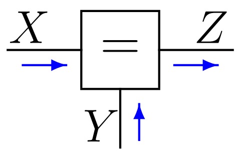
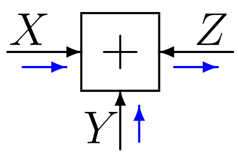
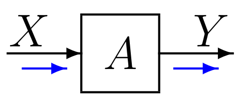
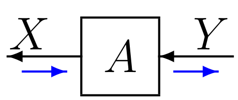
- Hard-decision estimate.
- Quantization of variables.
  Obvious, but infeasible in higher dimensions.
- Function value and derivative (or gradient) at a point selected by the receiving node. This data type allows gradient algorithms (e.g., LMS).
- Mean and variance, often with an underlying assumption of Gaussianity. This is the realm of Kalman filtering.
- List of samples. A pdf can be represented by a list of samples. This data type is the basis of the particle filter, but it can be used as a generic data type in a graphical model.
- Compound messages. The “product” of other message types.

Kalman Filtering and Smoothing



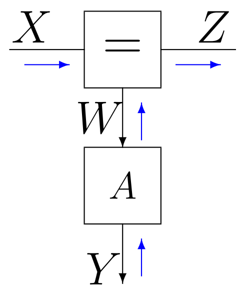
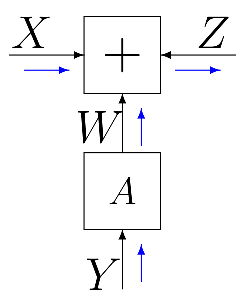
Messages consist of mean vector m and covariance matrix V or $W = V^{-1}$.

Sum-Product Rules for Gaussian Messages

1	 $\delta(x-y)\delta(x-z)$	$m_Z = \left(W_X + W_Y\right)^{\#} (W_X m_X + W_Y m_Y)$ $W_Z = W_X + W_Y$ $V_Z = V_X\left(V_X + V_Y\right)^{\#} V_Y$
2	 $\delta(x+y+z)$	$m_Z = -m_X - m_Y$ $V_Z = V_X + V_Y$ $W_Z = W_X\left(W_X + W_Y\right)^{\#} W_Y$
3	 $\delta(y - Ax)$	$m_Y = A m_X$ $V_Y = A V_X A^H$ Problem with W_Y if A has not full row rank!
4	 $\delta(x - Ay)$	$m_Y = \left(A^H W_X A\right)^{\#} A^H W_X m_X$ $W_Y = A^H W_X A$ If A has full row rank: $m_Y = A^H\left(A A^H\right)^{-1} m_X$

Rules for Composite Blocks

The Heart of Kalman Filtering and RLS

5		$m_Z = m_X + V_X A^H G \left(m_Y - A m_X\right)$ $V_Z = V_X - V_X A^H G A V_X$ with $G \triangleq \left(V_Y + A V_X A^H\right)^{-1}$
6		$m_Z = -m_X - A m_Y$ $W_Z = W_X - W_X A H A^H W_X$ with $H \triangleq \left(W_Y + A^H W_X A\right)^{-1}$

Data Type "List of Samples": Particle Filter

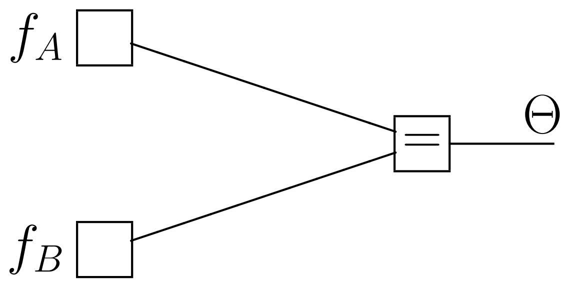
- A probability distribution can be represented by a list of samples from the distribution.
- This data type is the basis of the particle filter, which has recently gained much attention in the statistics literature as a means to overcome the limitations of Kalman filtering.
- Its use for message passing in general factor graphs seems to be largely unexplored, but promising.

Continuous Variables: Message Types

The following message types are widely applicable.

- Hard-decision estimate.
- Quantization of variables.
  Obvious, but infeasible in higher dimensions.
- Function value and derivative (or gradient) at a point selected by the receiving node. This data type allows gradient algorithms (e.g., LMS).
- Mean and variance, often with an underlying assumption of Gaussianity. This is the realm of Kalman filtering.
- List of samples. A pdf can be represented by a list of samples. This data type is the basis of the particle filter, but it can be used as a generic data type in a graphical model.
- Compound messages. The "product" of other message types.

On Gradient Methods



$$f(\theta) = f_A(\theta) f_B(\theta)$$

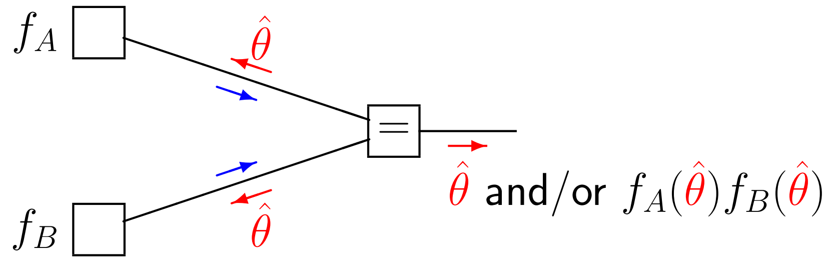
Suppose we wish to find

$$\hat{\theta} \stackrel{\triangle}{=} \operatorname*{argmax}_{\theta} f(\theta)$$

(and/or $f(\hat{\theta})$) by solving

$$\frac{d}{d\theta}\big(\log f(\theta)\big) = 0$$

On Gradient Methods (cont'd)



1. Broadcast some initial estimate $\hat{\theta}$.
2. Node f_A (and similarly f_B) replies by sending $\frac{d}{d\theta}\big(\log f_A(\theta)\big)\Big|_{\theta=\hat{\theta}}$
3. New estimate $\hat{\theta}$ is computed as

$$\begin{aligned}\hat{\theta}_{\text{new}} &= \hat{\theta}_{\text{old}} + s \cdot \frac{d}{d\theta}\big(\log f(\theta)\big)\Big|_{\theta=\hat{\theta}_{\text{old}}} \\ &= \hat{\theta}_{\text{old}} + s \cdot \left(\frac{d}{d\theta}\big(\log f_A(\theta)\big)\Big|_{\theta=\hat{\theta}_{\text{old}}} + \frac{d}{d\theta}\big(\log f_B(\theta)\big)\Big|_{\theta=\hat{\theta}_{\text{old}}} \right)\end{aligned}$$

where $s \in \mathbb{R}$ is a step-size parameter.

Conclusions

- Factor graphs help to develop practical algorithms for complex real-world detection and estimation problems.
- Such algorithms may include combinations and extensions of most prior art, including gradient methods, Kalman filtering, particle filters, ...
- There are parallel developments (with similar graphical models) in statistics, artificial intelligence, and neural networks.
- Forney-style factor graphs are especially elegant.