THE ESSENCE OF PROBABILITY PROPAGATION IN BAYESIAN NETWORKS.

FRANCESCO A.N. PALMIERI

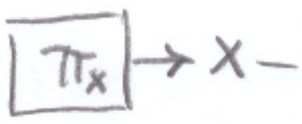
Nov. 2019

Let us begin considering a single random variable X. For simplicity at the beginning we will assume that variables are discrete and take values in a finite space. Continuous variables will be considered later.

X : r.v. ; $x \in \mathcal{X} = \{x^1, x^2, \dots, x^N\}$; x value of the random variable ; $\mathcal{X}$ alphabet ; $X \sim \pi_X(x)$ distributed as

Graphically

Bayesian graph

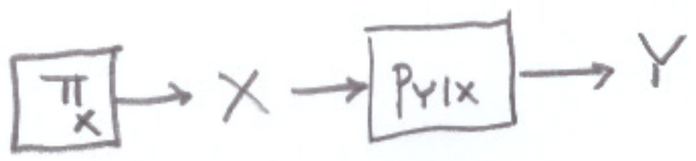


Factor graph

Bayesian graphs associate random variables to nodes, while in (Bayesian) factor graphs, the variables are associated to branches, and blocks describe distributions as shown in the following where we consider two random variables

$$X : x \in \mathcal{X} = \{x^1, x^2, \dots, x^N\} \qquad Y : y \in \mathcal{Y} = \{y^1, y^2, \dots, y^M\}$$

Bayesian graph



Factor graph

The graph is a representation of the joint distribution $P_{XY}(x,y)$ in the factorization

$$P_{XY}(x,y) = P_{Y|X}(y|x)\,\pi_X(x)$$

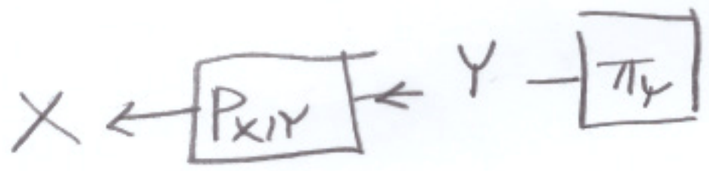
This is why we talk about "Factor" graph, because each block in the diagram represents a factor.

It should be pointed out that on our graphical 2 notation we have shown directionality from X to Y. Alternative representations are well known in the literature with non oriented graphs

where the arc between the two variables denotes that there is a dependence and that the two variables must be considered jointly with a joint distribution $P_{XY}(x,y)$.

We prefer to use oriented graphs because in many cases the directions of the arrows is more directly connected to physical interpretation of the phenomenon under study (causality). Also directionality brings to unambiguous definition of "backward" and "forward"
Note that the joint pdf can also be factorized as

$$P_{XY}(x,y) = P_{X|Y}(x|y)\,\pi_y(y)$$



In this case we have different factors, $P_{X|Y}(x|y)$ and $\pi_y(y)$. The choice of the graph and its directionality depends on the problem under study and computational convenience as we will see in greater detail in the following examples.

Now what are the relevant problems that we would like to solve using the Bayesian model? Even in this very simple scenario?

(P1) Suppose $p_{Y|X}(y|x)$ and $\pi_X(x)$ are known and we want to compute the marginal distribution $p_Y(y)$.

Clearly $$p_Y(y) = \sum_x p_{XY}(x,y) = \sum_x p_{Y|X}(y|x)\,\pi_X(x)$$

Here we can see a first example of possible probability propagation

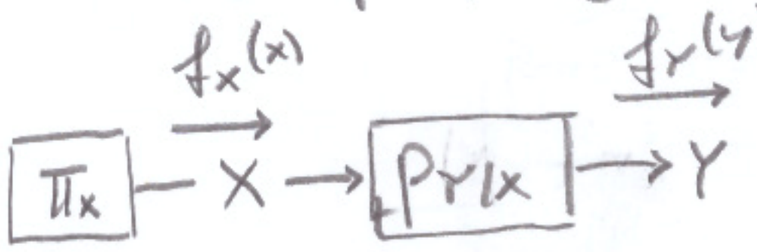


Fig 1

if we associate a "forward" message to X, $f_X(x) = \pi_X(x)$, that propagated (integrated) in the block $p_{Y|X}$ gives a forward message for Y

$$f_Y(y) = \sum_x p_{Y|X}(y|x)\, f_X(x)$$

We recognize that $f_Y(y) = p_Y(y)$.

(P2) Use the same hypotheses of P1 and compute $p_X(x)$. Clearly

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

$$= \pi_X(x) \underbrace{\sum_y p_{Y|X}(y|x)}_{1} = \pi_X(x)$$

as expected since $\pi_X(x)$ was already available. Probability propagation here is not so evident, but if we define a "backward"

message on Y $\quad b_y(y) = U(y)$ (Uniform distr.)
and a backward message on X as

$$b_x(x) = \sum_y p_{Y|X}(y|x)\, b_y(y)$$

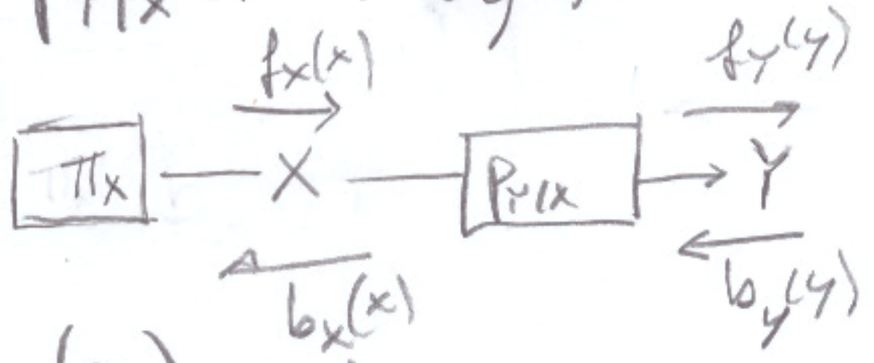


Fig 2

we have

$$p_x(x) \propto f_x(x)\, b_x(x), \text{ where}$$

$\propto$ means "proportional to". Note that the exact $p_x(x)$ can be obtained through normalization

$$p_x(x) = \frac{f_x(x)\, b_x(x)}{\sum_{\xi \in \mathcal{X}} p_x(\xi)\, b_x(\xi)}.$$

(P3) In the same framework given above suppose we have "evidence" on Y, i.e. we know that $Y = \bar{y}$ (perhaps after an observation) and we want to find $p_x(x|Y=\bar{y})$. Clearly we need Bayes' theorem (or just the definition of conditional probability) to write

$$p_x(x|Y=\bar{y}) = \frac{p(x, Y=\bar{y})}{p(Y=\bar{y})} = \frac{p(Y=\bar{y}|x)\,\pi_x(x)}{p(Y=\bar{y})}$$

But $p(Y=\bar{y})$ is just a constant (that can be recuperated via normalization), therefore

$$p_x(x|Y=\bar{y}) \propto \underbrace{\pi_x(x)}_{\text{prior contribution}}\ \underbrace{p(Y=\bar{y}|x)}_{\text{evidence contribution}}$$

$f_x(x)$

Probability propagation can be used if we define a backward message on Y (see Fig. 2)

$$b_y(y) = \delta(y - \bar{y}) .$$

the backward message on X is computed as

$$b_x(x) = \sum_y p_{Y|X}(y|x) \underbrace{\delta(y-\bar{y})}_{b_y(y)}$$

therefore the answer is again the product

$$p_x(x|Y=\bar{y}) \propto f_x(x)\, b_x(x)$$

—o—

In these three examples we have used two simple rules:

SUM-PRODUCT RULE

1. SUM: Propagate the message through a block summing over the input variable (i.e. the variable that we eliminate)
2. PRODUCT: The posterior on a variable is proportional to the product of forward and backward distributions

(PG) To complete the analysis let us consider the case in which we have evidence on X, i.e. $X=\bar{x}$ and we want to compute $p(y|X=\bar{x})$. Obviously since we have $p_{Y|X}(y|x)$ we simply evaluate it for $X=\bar{x}$.

However it is interesting to look at this also from the point of view of probability propagation.

Since we have no information about y we take $b_y(y) = U(y)$. Evidence on X exits the branch into two variables X_u X_d and the graph becomes

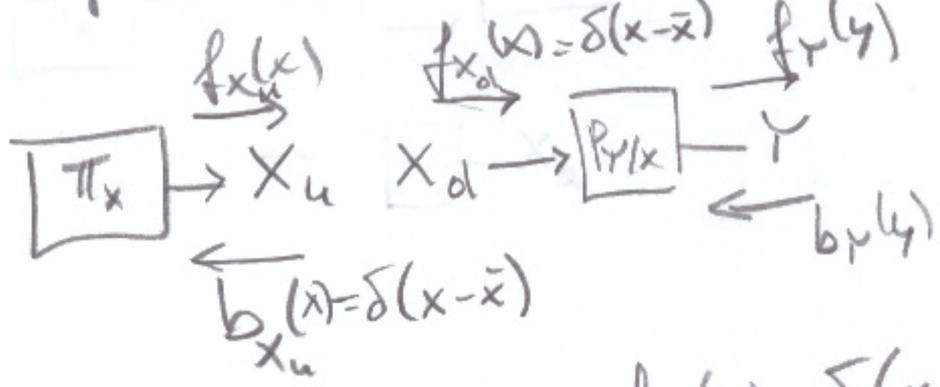


fig 3

A forward message $f_{X_d}(x) = \delta(x-\bar{x})$ is sent towards the block $P_{Y|X}$ and a backward message is sent toward the block π_X. Final knowledge is collected at Y with

$$f_Y(y) = \sum_x \underbrace{f_{X_d}(x)}_{\delta(x-\bar{x})} P_{Y|X}(y|x) = P_Y(\bar{y}|x)$$

this is already the answer, but to show that this is coherent with the general product rule, since $b_y(y)$ is uniform, we have here too that

$$p(y|X=\bar{x}) \propto f_Y(y)\, b_Y(y).$$

Also on the X_u branch, the product rule applies trivially:

$$p(X|X=x) \propto \underbrace{f_{X_u}(x)}_{\pi_X(x)} \underbrace{b_{X_u}(x)}_{\delta(x-\bar{x})} = \pi_X(\bar{x})\delta(x-\bar{x}) \propto \delta(x-\bar{x})$$

In the Bayesian graph notation evidence on each variable is depicted as a "shaded" node

Fig 4

THE DIVERTER

To maintain total description of our system as a probability pipeline, we also define the "diverter" or "equality constraint" block shown in the following figure

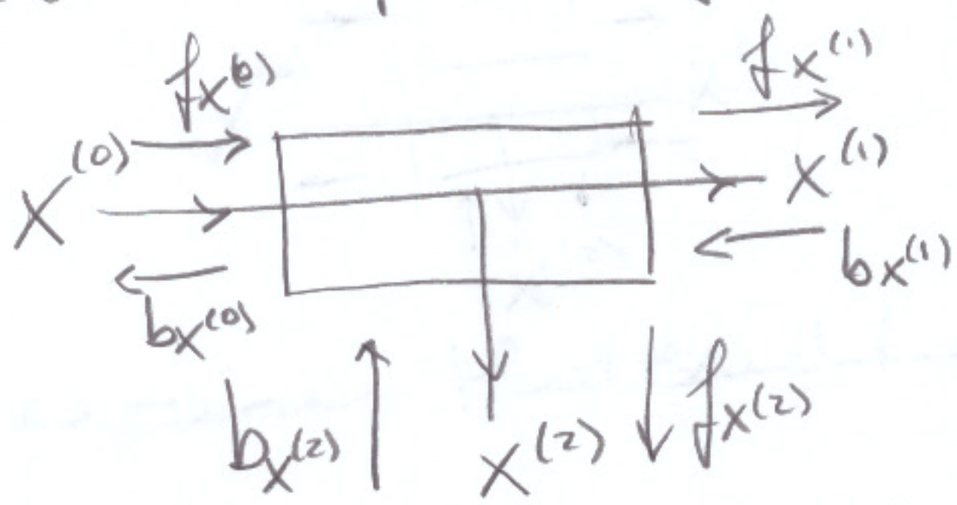


Figure 5

This element expresses the condition

$$X^{(1)} = X^{(2)} = X^{(3)}$$

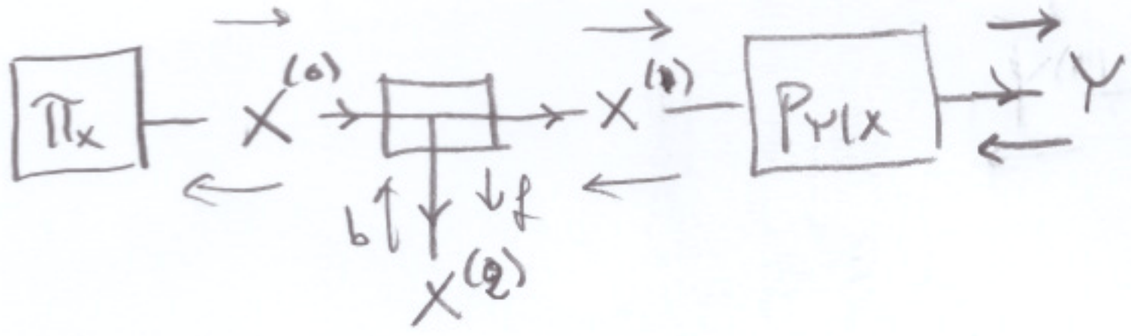
Messages can be propagated using a simple product rule: outgoing messages are proportional to the product of the incoming ones

More specifically with reference to the Figure 5

$$\begin{cases} f_{X^{(1)}}^{(x)} \propto f_{X^{(0)}}(x)\, b_{X^{(2)}}^{(x)} \\ b_{X^{(0)}}^{(x)} \propto b_{X^{(2)}}(x)\, b_{X^{(1)}}^{(x)} \\ f_{X^{(2)}}(x) \propto b_{X^{(1)}}(x)\, f_{X^{(0)}}^{(x)} \end{cases}$$

but we now see how to use the diverter.

With reference to problems P1 P2 P3 and P4, we can insert out of these junctions as in the following figure where variable X is triplicated into three variables $X^{(1)} X^{(2)} X^{(3)}$ constrained to be equal.



Now information can be <u>injected-in</u> and/or <u>spilled-out</u> the system.

going back to the review problems

P1 P2

No evidence anywhere

$b_{X^{(2)}}(x) = U(x)$; $b_Y(y) = U(y)$

no question effect.

$$f_{X^{(0)}}(x) = \pi_x(x) \; ; \; f_{X^{(1)}}(x) = \underbrace{f_{X^{(0)}}(x)}_{\pi_x(x)} \underbrace{b_{X^{(2)}}(x)}_{U(x)} = \pi_x(x)$$

$$f_Y(y) = \sum_{x \in \mathcal{X}} f_{X^{(1)}}(x) P_{Y|X}(y|x)$$

$$P_Y(y) \propto f_Y(y) \underbrace{b_Y(y)}_{U(y)}$$

$$b_{X^{(1)}}(x) \propto \sum_{y \in \mathcal{Y}} P_{Y|X}(y|x) \underbrace{b_Y(y)}_{U(y)} \propto U(x)$$

$$f_{X^{(2)}}(x) \propto \underbrace{b_{X^{(1)}}(x)}_{U(x)} \underbrace{f_{X^{(0)}}(x)}_{\pi_x(x)} \propto \pi_x(x) \quad \left(\begin{array}{c}\text{spilled out}\\ \text{information}\\ \text{on } x\end{array}\right)$$

$$P_{X^{(2)}}(x) \propto \underbrace{f_{X^{(2)}}(x)}_{\pi_x(x)} \underbrace{b_{X^{(2)}}(x)}_{U(x)} \propto \pi_x(x)$$

$$b_{X^{(0)}}(x) \propto \underbrace{b_{X^{(1)}}(x)}_{U(x)} \underbrace{b_{X^{(2)}}(x)}_{U(x)} \propto U(x)$$

$$P_{X^{(0)}}(x) \propto \underbrace{f_{X^{(0)}}(x)}_{\pi_x(x)} \underbrace{b_{X^{(0)}}(x)}_{U(x)} \propto \pi_x(x)$$

(P3) **Evidence on Y**

$$b_Y(y) = \delta(y-\bar{y}) \quad ; \quad b_{X^{(2)}}(x) = U(x)$$

Spill-out information on X using

$$f_{X^{(2)}}(x) \propto \underbrace{f_{X^{(0)}}(x)}_{\pi_X(x)} \underbrace{b_{X^{(1)}}(x)}_{\propto \sum_{y\in\mathcal{Y}} P_{Y|X}(y|x)\, \underbrace{b_Y(y)}_{\delta(y-\bar{y})}} = P_{Y|X}(\bar{y}|x)$$

$$\propto \pi_X(x)\, P_{Y|X}(\bar{y}|x)$$

(P4) **Inject information on X**

$$b_{X^{(2)}}(x) = \delta(x-\bar{x}) \quad , \quad b_Y(y) = U(y)$$

Spill-out information on Y as

$$f_Y(y) \propto \sum_{x\in\mathcal{X}} P_{Y|X}(y|x)\, f_{X^{(1)}}(x)$$

But $$f_{X^{(1)}}(x) \propto \underbrace{f_{X^{(0)}}(x)}_{\pi_X(x)} \underbrace{b_{X^{(2)}}(x)}_{\delta(x-\bar{x})} \propto \delta(x-\bar{x})$$

Therefore

$$f_Y(y) \propto P_{Y|X}(y|\bar{x})$$

Note that a delta distribution (evidence) injected at a diverter breaks the connection because the product rule. Evidence is loaded in all directions.

MARKOV CHAIN

Consider now a bit more complicated system with T variables $X_1 X_2 \cdots X_T$.
The joint distribution can always be factorized using the chain rule

$$P_{X_1 X_2 \cdots X_T}(x_1 x_2 \cdots x_T) = P_{X_T|X_{T-1}\cdots X_1}(x_T|x_{T-1},\cdots x_1)\, P_{X_{T-1}|X_{T-2}\cdots X_1}(x_{T-1}|x_{T-2},\cdots x_1) \cdots$$
$$\cdots P_{X_3|X_2 X_1}(x_3|x_2 x_1)\, P_{X_2|X_1}(x_2|x_1)\, P_{X_1}(x_1)$$

where we have chosen (purposely) just one of the many conditioning orders.

A Markov chain is a system in which we are given a time order for our variables $X_1 \to X_2 \to X_3 \to \cdots \to X_T$ and conditioning can be limited to the previous variable, i.e. the factorization is

$$P_{X_1 X_2 \cdots X_T}(x_1 x_2 \cdots x_T) = P_{X_T|X_{T-1}}(x_T|x_{T-1})\, P_{X_{T-1}|X_{T-2}}(x_{T-1}|x_{T-2}) \cdots P_{X_3|X_2}(x_3|x_2)\, P_{X_2|X_1}(x_2|x_1)\, P_{X_1}(x_1)$$

The following figures show the corresponding Bayesian graph and the factor graph

Total characterization requires knowledge of the distributions

$\{\pi_{X_1}, P_{X_2|X_1}, P_{X_3|X_2}, \ldots, P_{X_T|X_{T-1}}\}$.

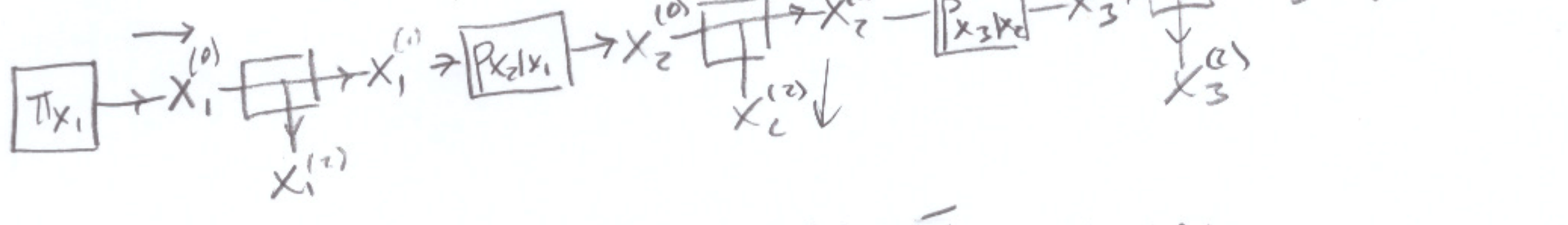
If the chain is time-invariant

$$P_{X_2|X_1}(\xi|\eta) = P_{X_3|X_2}(\xi|\eta) = \cdots = P_{X_T|X_{T-1}}(\xi|\eta)$$

with a unique $P_{X_t|X_{t-1}}(x_t|x_{t-1})$ and $\pi_{X_1}(x_1)$ sufficient to characterize the model.

We are interested in manipulating the chain using probability propagation. As we have seen in our previous two-variable example, we can gain inference and inject information at any section of the system.

Take $T=4$ for simplicity.



If we have evidence on $X_4 = \bar{x}$, we can inject a backward message at X_4

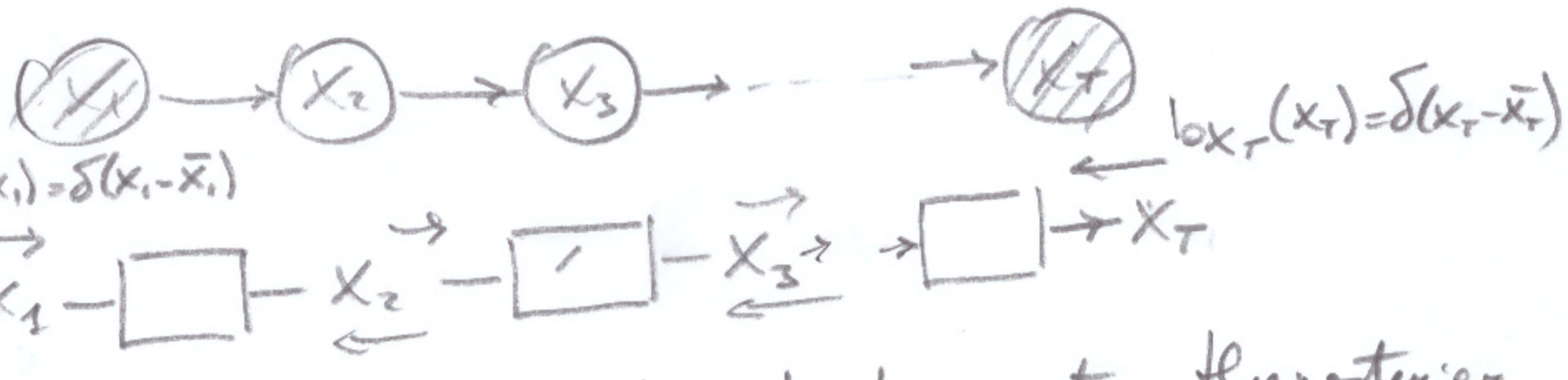
$b_{X_4}(x_4) = \delta(x_4 - \bar{x})$. Then propagate and collect inference with the usual rules.

For example the posterior on X_2 could be spelled out as $f_{X^{(2)}}(x)$; it will be the summary of information coming from the right (evidence) and (prior) information coming from the left.

Similarly evidence on any other variable can be injected anywhere in the graph.

In general messages have to propagate anywhere in the graph before we can collect results. In our simple example this is just a chain and messages will need at most T steps to reach their steady configuration.

Interesting to study the case in which both X_1 and X_T are known: $X_1 = \bar{X}_1$ $X_T = \bar{X}_T$



If X_t represents the state at time t, the posterior distributions

$$p_{X_2}(x_2), p_{X_3}(x_3), \dots, p_{X_{T-1}}(x_{T-1})$$

describe the state distributions at time 2, 3, ..T-1 compatible with $X_1 = \bar{x}_1$ and $X_T = \bar{x}_T$.

To see more explicitely how probability propagation is just nested marginalization, let us explode the case for $T = 4$, with evidence on X_1 and X_4. Joint posterior of the two unknown variables x_2 and x_3 is proportional to

$$p_{X_1 X_2 X_3 X_4}(\bar{x}_1, x_2, x_3, \bar{x}_4) = p_{X_4|X_3}(\bar{x}_4|x_3)\, p_{X_3|X_2}(x_3|x_2)\, p_{X_2|X_1}(x_2|\bar{x}_1)$$

Suppose we are interested in the posterior for x_3, i.e. we have to marginalize with respect to x_2:

$$p_{X_3}(x_3) \propto \sum_{x_2 \in \mathcal{X}_2} p_{X_4|X_3}(\bar{x}_4|x_3)\, p_{X_3|X_2}(x_3|x_2)\, p_{X_2|X_1}(x_2|\bar{x}_1)$$

$$\propto \sum_{x_2} \underbrace{\sum_{x_1} p_{x_2|x_1}(x_2|x_1)\underbrace{\delta(x_1-\bar{x}_1)}_{f_{x_1}(x_1)}}_{f_{x_2}(x_2)} p_{x_3|x_2}(x_3|x_2) \underbrace{\sum_{x_4 \in \mathcal{X}_4} p_{x_4|x_3}(x_4|x_3)\underbrace{\delta(x_4-\bar{x}_4)}_{b_{x_4}(x_4)}}_{b_{x_3}(x_3)}$$

$f_{x_3}(x_3)$

A backward message is injected at x_4 with the evidence $x_4=\bar{x}_4$ $\quad b_{x_4}(x_4)=\delta(x_4-\bar{x}_4)$

and similarly a forward message is injected at x_1

$$f_{x_1}(x_1)=\delta(x_1-\bar{x}_1).$$

Nested marginalization is exactly what the sum rules of probability propagation prescribe.

SMOOTH EVIDENCE

In many applications we may have available only approximate inference about a variable. For example in our object recognition system suppose that Y represents the object class and there are N_Y classes

$$Y = \{y^1, y^2, y^3, \ldots, y^{N_Y}\} = \{\text{dog, tea cup, horse, building, chair, } \ldots\}.$$

Suppose now that the system returns only one inference about the classes. For example it may tell us that the object is a "dog" with 70% accuracy, a "horse" 20% accuracy and else with the remaining 10% spread equally. Our knowledge is described by the distribution $P_Y(y) = \{0.7, 0.2, \frac{0.1}{N_Y-2}, \frac{0.1}{N_Y-2}, \ldots\}$

Now imagine that we have also a context variable X

$$X = \{x^1, x^2, x^3, \ldots, x^{N_X}\} = \{\text{street view, kitchen, office, } \ldots\}$$

that we want to connect probabilistically to Y.

If we have available enough statistics we can define a conditional distribution

$P_{Y|X}(y|x)$ and a prior distribution $\pi_X(x)$ for the context.

$Y|X$: object | context

The partial evidence on the object needs to be used to gain inference on the context X.

How do we pose the problem?

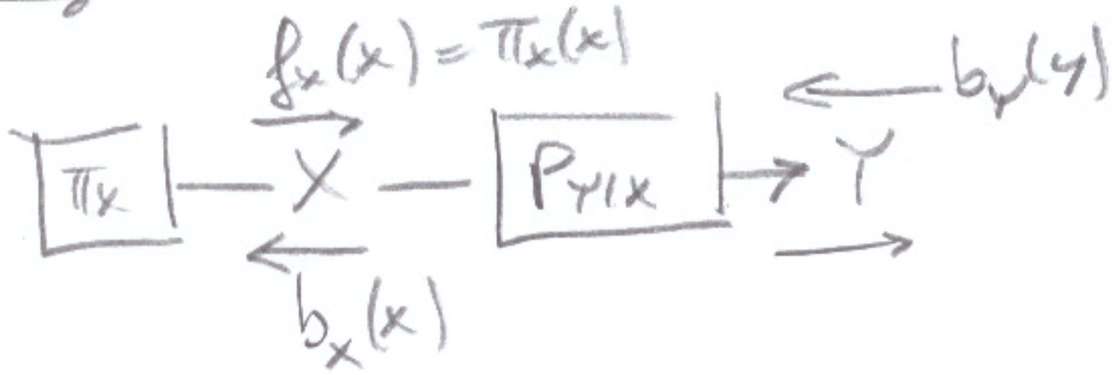
If we had exact knowledge about $Y=\bar{y}$, we have already shown how to compute the posterior $P_{X}(x \mid Y=\bar{y})$. To use at best our partial knowledge about Y, we can compute the

AVERAGE POSTERIOR

$$\sum_{y \in \mathcal{Y}} P_X(x \mid Y=y)\, b_Y(y) = E_{Y \sim b_Y(y)}\left[P_{X}(x \mid Y=y)\right]$$

where $b_Y(y)$ is the distribution that we have available.

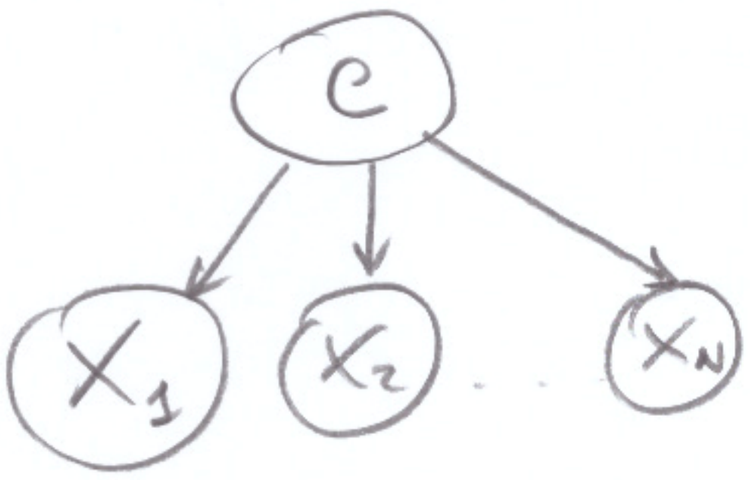
The scheme is the same as above, with the difference that we inject a smooth distribution at Y, and that the result will be an average posterior.



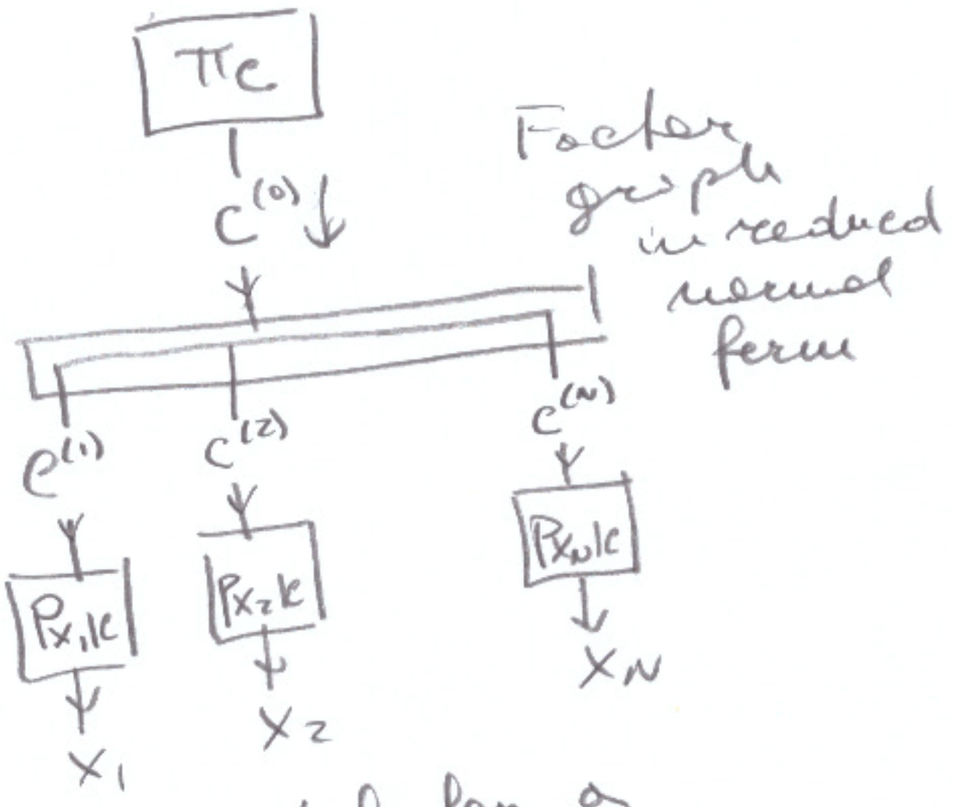
$$b_X(x) \propto \sum_{y \in \mathcal{Y}} P_{Y|X}(y \mid x)\, \underbrace{b_Y(y)}_{\text{evidence}}$$

$$P_X(x) \propto \underbrace{f_x(x)}_{\pi_x(x)}\, \underbrace{b_x(x)}_{\text{summary of evidence}}$$

LATENT VARIABLE MODEL (LVM)



Bayesian graph



The LVM is one of the most used models for a set of dependent random variables $X_1 X_2 \dots X_N$. It is based on the assumption that there exist another variable C, that if known, makes $X_1 X_2 \dots X_N$ independent.

More specifically the factorization is

$$p(X_1 X_2 \dots X_N C) = p(X_1|C)\, p(X_2|C) \dots p(X_N|C)\, p(C)$$

and we say that $X_1 X_2 \dots X_N$ are conditionally independent given C.

The factor graph representation is particularly convenient because it allows a very flexible use of the model.

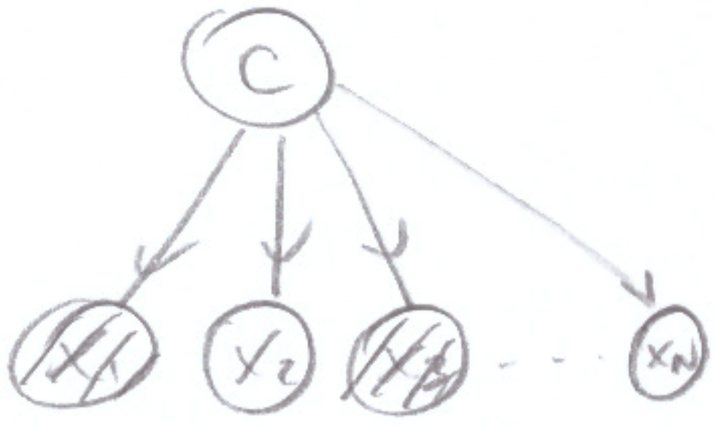
To fix ideas, suppose that C represents a context variable for $X_1 X_2 \dots X_N$. If C is given, $C = \bar{c}$, we can inject a message at $c^{(0)}$ $\quad f_{c^{(0)}}(c) = \delta(c - \bar{c})$

and collect the distributions

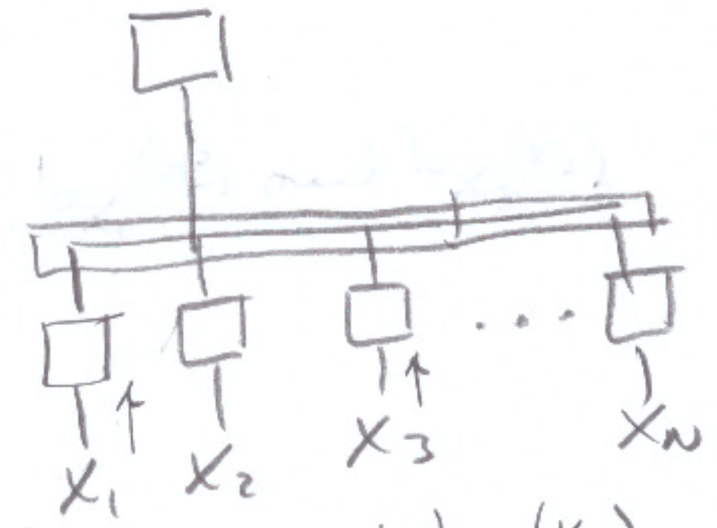
$$p(X_1|C=\bar{c})\quad p(X_2|C=\bar{c}) \dots p(X_N|C=\bar{c})$$

at the bottom as $f_{X_1}(x_1) \dots f_{X_N}(x_N)$.

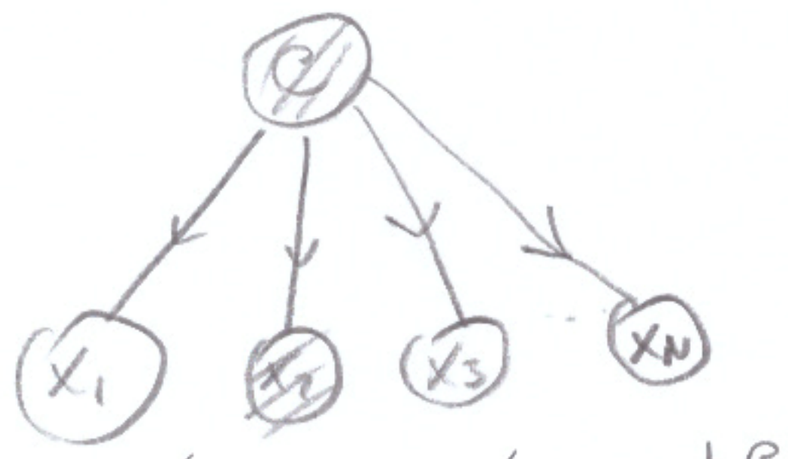
More generally we may have evidence (or smooth evidence) on a set of variable and we want to gain inference on the others



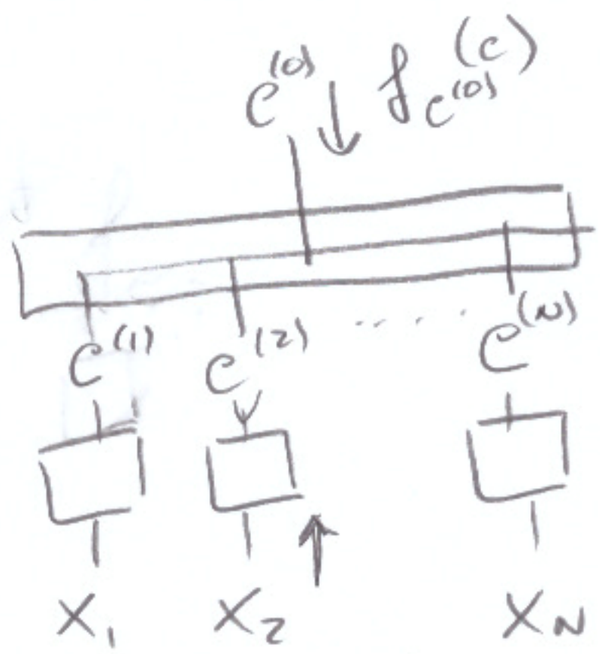
Evidence on X_1 and X_3



set $b_{X_1}(x_1)$ and $b_{X_3}(x_3)$

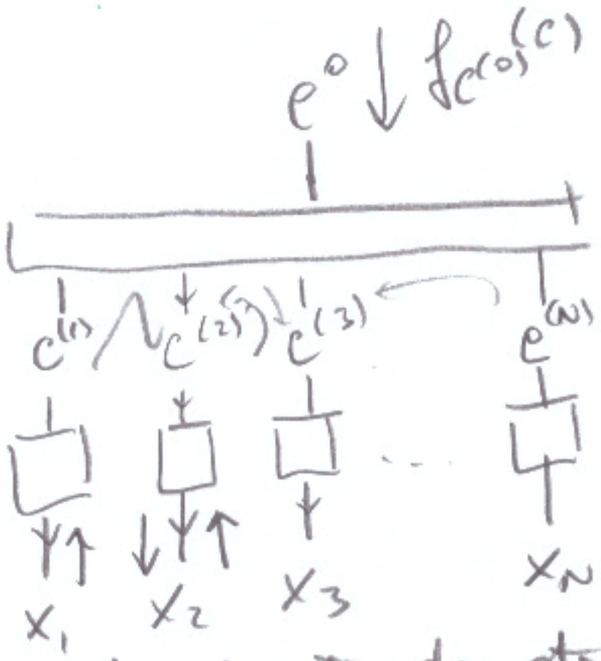


Evidence on X_2 and C



set $b_{X_2}^{(x_2)}$ and $f_{c^{(0)}}^{(c)}$ and $b_{X_2}(x_2)$

Interesting the case in which we may have smooth evidence on a variable, say X_2, but some information about the others, say X_1 and C.



inject information on C as $f_{c^{(0)}}(c)$.

Inject information on X_2 as a smooth $b_{X_2}(x_2)$

Inject information on X_1 as $b_{X_1}(x_1)$

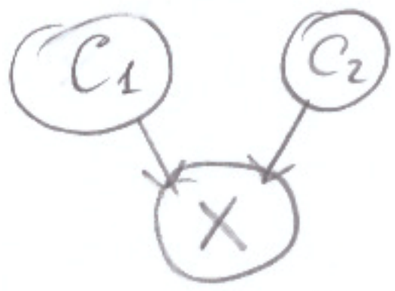
After propagation to steady state, we could gain "improved" information on X_2 using its forward $f_{X_2}(x_2)$ in the product

$$P_{X_2}(x_2) \propto f_{X_2}(x_2)\, b_{X_2}(x_2)$$

Information coming from other variables has been utilized
out "best" on X_2.
Note that information on any of the variables $X_1, X_3, \ldots X_N$
injected as a backward message, influences only
the forward message for the others and not its own.
This is due to the tree structure of the graph.
Similarly information injected as a forward at $C^{(2)}$
influences only forward messages on $X_1, X_3, \ldots X_N$ and not
its own backward message.

VARIABLES WITH MORE THAN ONE PARENT

Let us start with the simple example shown in the following figure.



The probabilistic model corresponds to the factorization

$$p(X C_1 C_2) = p(X | C_1 C_2)\, p(C_1)\, p(C_2) \quad (*)$$

Where X is "jointly" dependent on the two variables (parents) C_1 and C_2 assumed to be mutually independent. Even if C_1 and C_2 are independent they condition jointly X and cannot be separated in the function $p(X|C_1 C_2)$. This is sometimes referred to as "marginal dependence of parents" [ref]

Note that variables C_1 and C_2 are independent, but not conditionally independent given X ! The first factor in the factorization (*) does not imply that the first factor in

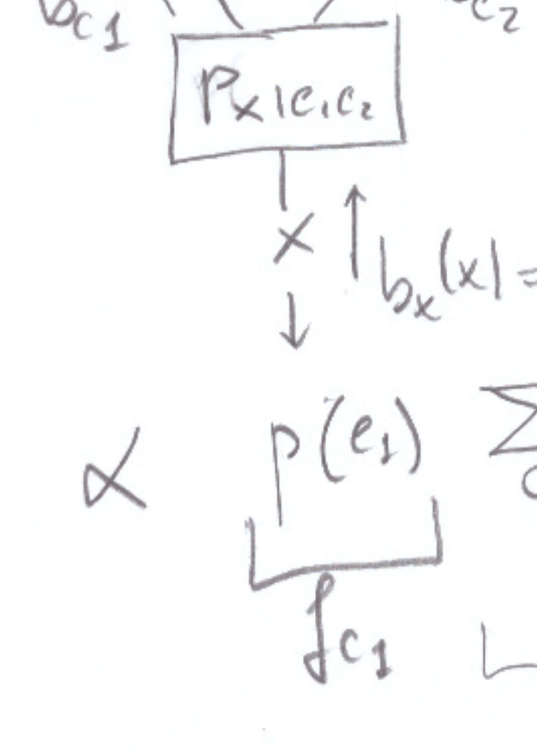
$$p(X C_1 C_2) = p(C_1 C_2 | X)\, p(X)$$

can be split!.

Now if for example we have evidence on $X = \bar{x}$ and we want information on C_1 and C_2 respectively we have to compute

$$p(c_1|X=\bar{x}) \alpha \sum_{c_2 \in C_2} p(X=\bar{x}|c_1 c_2)\pi_{c_1}(c_1)\pi_{c_2}(c_2) = \underbrace{\pi_{c_1}(c_1)}_{f_{c_1}} \underbrace{\sum_{c_2} p(X=\bar{x}|c_1,c_2)\pi_{c_2}(c_2)}_{b_{c_1}(c_1)}$$

$$p(c_2|X=\bar{x}) \alpha \sum_{c_1 \in C_1} p(X=\bar{x}|c_1 c_2)p(c_1)p(c_2) = \underbrace{\pi_{c_2}(c_2)}_{f_{c_2}(c_2)} \underbrace{\sum_{c_1} p(X=\bar{x}|c_1,c_2)\pi_{c_1}(c_1)}_{b_{c_2}(c_2)}$$



$$p_{C_1}(c_1) \alpha \underbrace{p(c_1)}_{f_{c_1}} \sum_{c_2}\sum_{x} \underbrace{\pi_{c_2}(c_2)}_{f_{c_2}(c_2)} p(x|c_1,c_2) \underbrace{\delta(x-\bar{x})}_{b_x(x)}$$

$b_{c_1}(c_1)$

$$p_{C_2}(c_2) \alpha \underbrace{\pi_{c_2}(c_2)}_{f_{c_2}(c_2)} \sum_{c_1}\sum_{x} \underbrace{\pi_{c_1}(c_1)}_{f_{c_1}(c_1)} p(x|c_1,c_2) \underbrace{\delta(x-\bar{x})}_{b_x(x)}$$

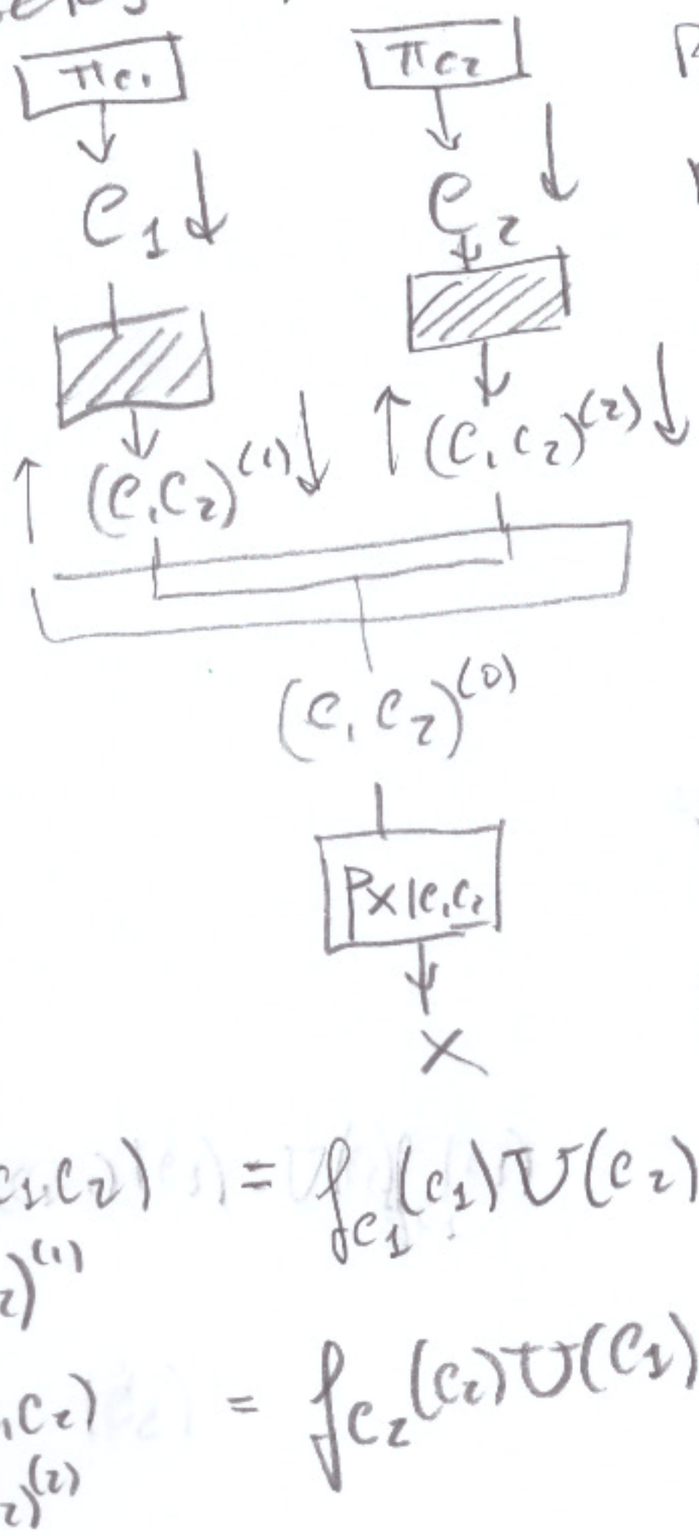
$b_{c_2}(c_2)$

A scenario with evidence on X and, say C_1, can be easily handled to gain information on C_2.

An alternative graphical representation for variables with more than one parent has been introduced to avoid the fact that some blocks in the factor graph have more than two ports. Note that the block $P_{X|C_1C_2}$ has 3 ports and messages on each one of them follows a different rule.

Factor graphs in <u>reduced normal form</u>

require only one-input one-output, or single output ports [ref]. To handle a situation like the one shown above, we have to introduce the so called "shaded blocks" shown in the following picture



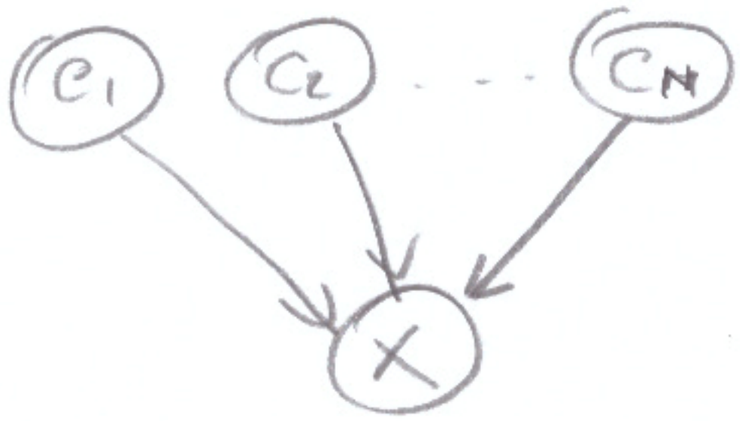
Basically a shaded block maps a variable to a product space with another variable that needs to be considered jointly (marriage)

In the figure forward and backward messages are

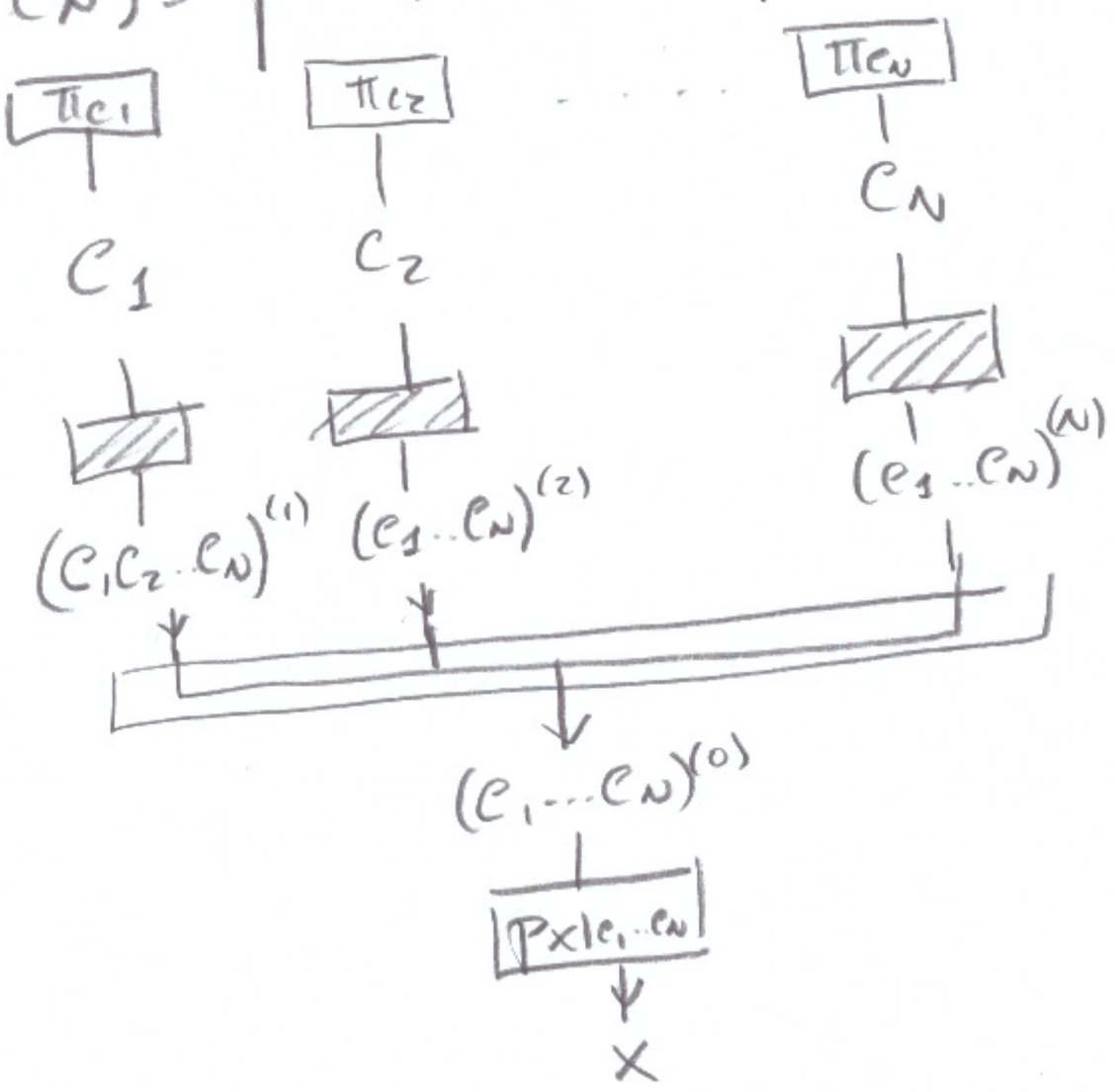
$$\times$$

$$\begin{cases} f^{(C_1C_2)}_{(C_1C_2)^{(1)}} = f_{C_1}(C_1)\,U(C_2) \\ f^{(C_1C_2)}_{(C_1C_2)^{(2)}} = f_{C_2}(C_2)\,U(C_1) \end{cases} \qquad \begin{cases} b_{C_1}(C_1) \propto \sum_{C_2} b^{(C_1,C_2)}_{(C_1C_2)^{(1)}} \\ b_{C_2}(C_2) \propto \sum_{C_1} b^{(C_1,C_2)}_{(C_1C_2)^{(2)}} \end{cases}$$

The situation is easily generalized to more than two parents



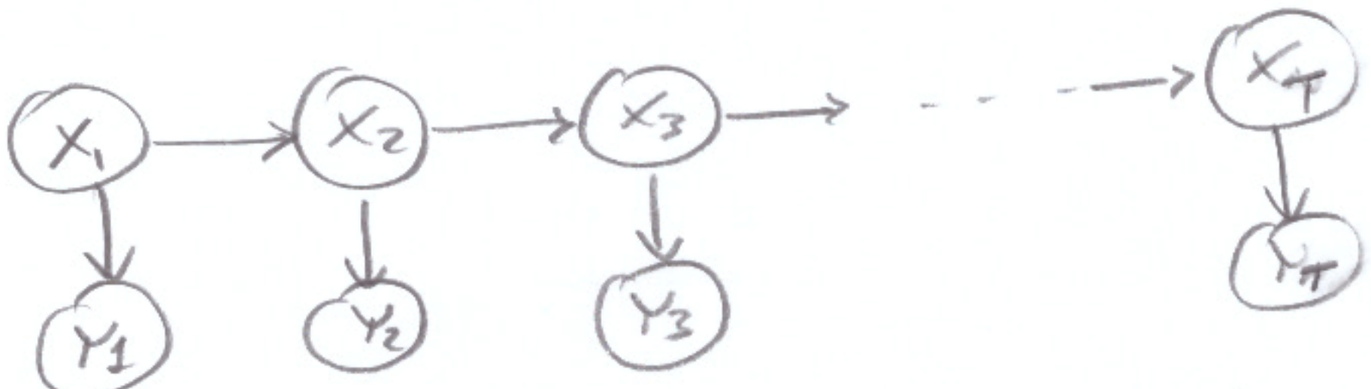
$$p(x, c_1 c_2 \ldots c_N) = p(x \mid c_1 c_2 \ldots c_N)\, p(c_1) p(c_2) \ldots p(c_N)$$



The difficulties encountered to handle this model are clearly related to the size of the product space $N_{C_1} N_{C_2} \ldots N_{C_N}$

HIDDEN MARKOV MODEL

In a Hidden Markov Model (HMM) we have a sequence of states each one connected to another variable.



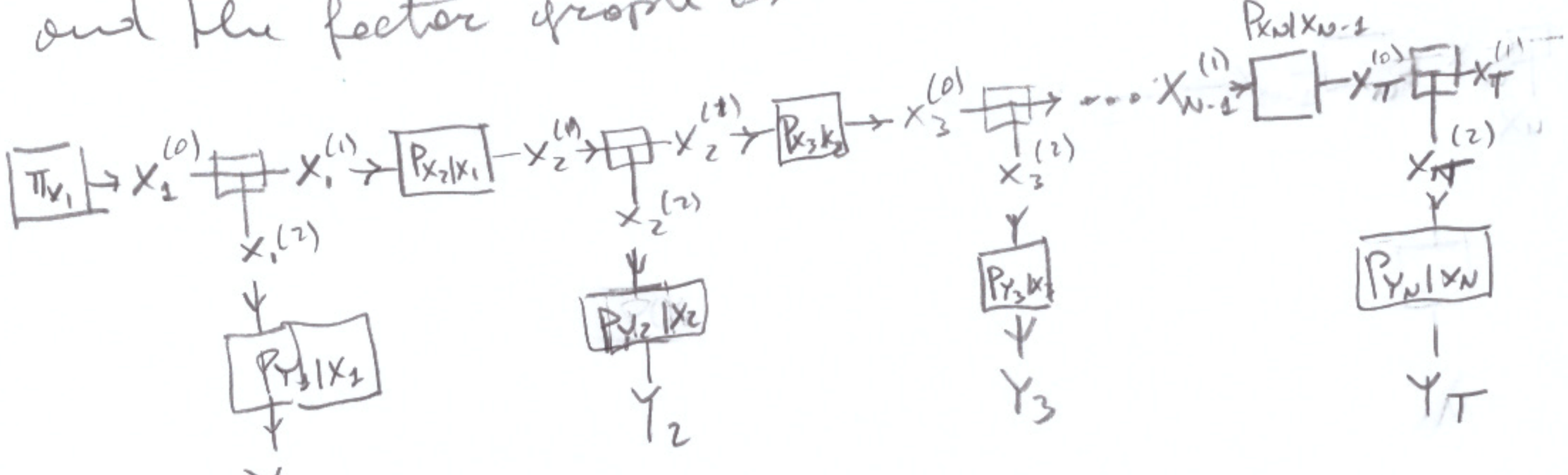
This model is often use to represent non-observable states, assuming that the variables $Y_1 \cdots Y_T$ are the observables.

HMM are very popular in a very large number of applications that go from text classification to tracking.

The system factorization is

$$p(X_1 X_2 \cdots X_T Y_1 \cdots Y_N) = p(X_T|X_{T-1})\,p(X_{T-1}|X_{T-2}) \cdots p(X_3|X_2)\,p(X_2|X_1)$$
$$p(X_1)\,p(Y_1|X_1)\,p(Y_2|X_2) \cdots p(Y_T|X_T)$$

and the factor graph is



In a time-invariant HMM all the state transition functions $P_{X_t|X_{t-1}}$ and the output functions $P_{Y_t|X_t}$ are the same $\forall t = 1,T$

An HMM is the stochastic version of the system theory representation

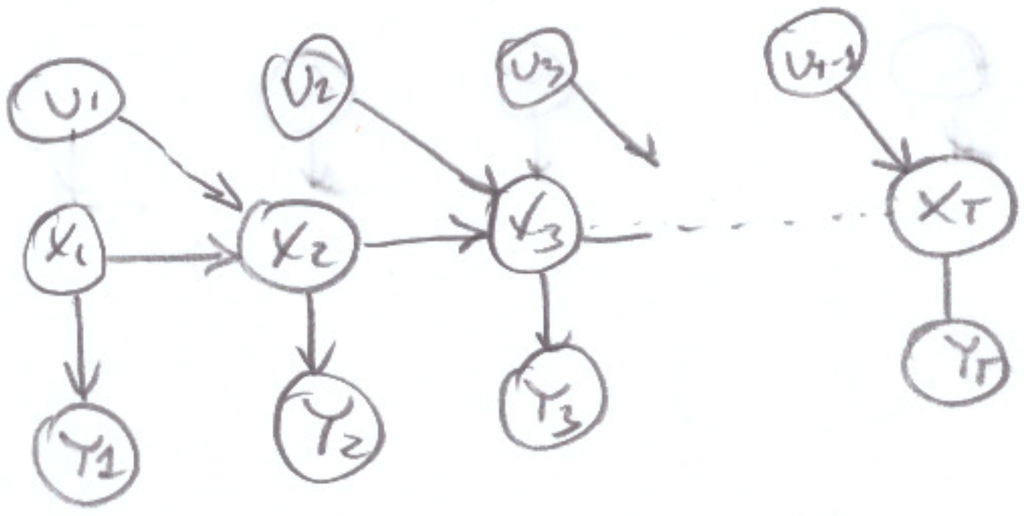
$$\begin{cases} x_{t+1} = \varphi(x_t) \\ y_t = \gamma(x_t) \end{cases}$$

where φ and γ are stochastic functions.

Often the stochasticity is associated to state noise v_t and measurement noise w_t

$$\begin{cases} x_{t+1} = \varphi(x_t + v_t) \\ y_t = \gamma(x_t + w_t) \end{cases}$$

Also an HMM can be augmented to include "input" variables $U_1 U_2 \dots U_T$

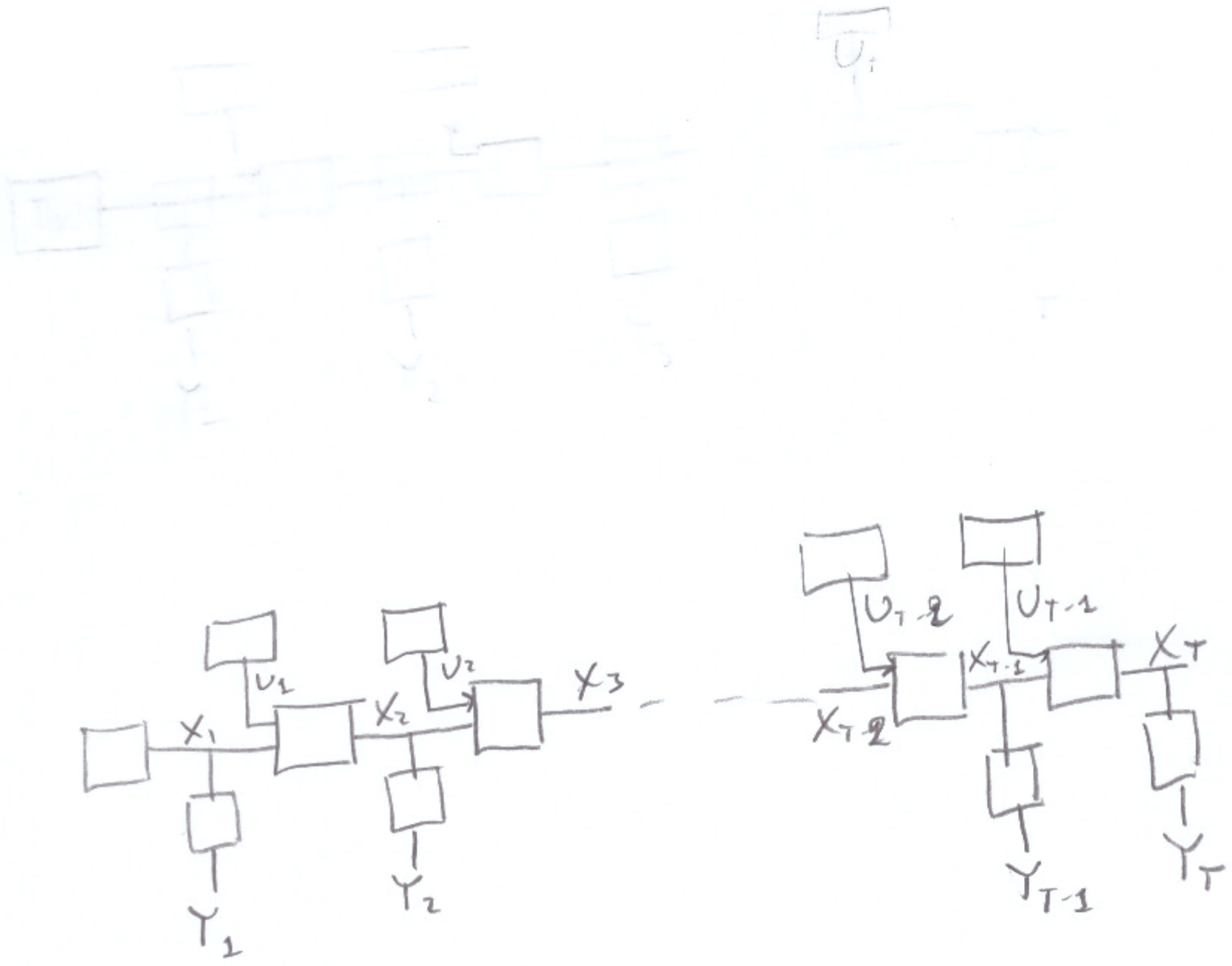


the factorization is then

$$p(X_1 \dots X_T Y_1 \dots Y_T U_1 \dots U_T) = p(U_1) \dots p(U_T)\, p(X_1|U_1)\, p(X_2|X_1 U_1) \dots$$
$$\dots p(X_T|X_{T-1} U_{T-1})\, p(Y_1|X_1) \dots p(Y_T|X_T)$$

State transition is conditioned jointly by input and previous state.

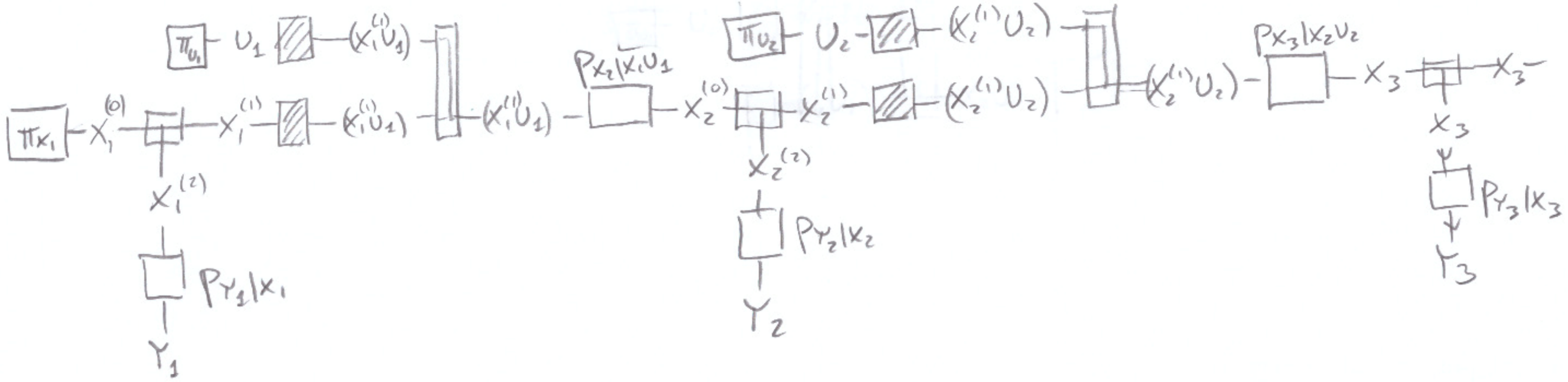
The factor graph is



to represent the stochastic system

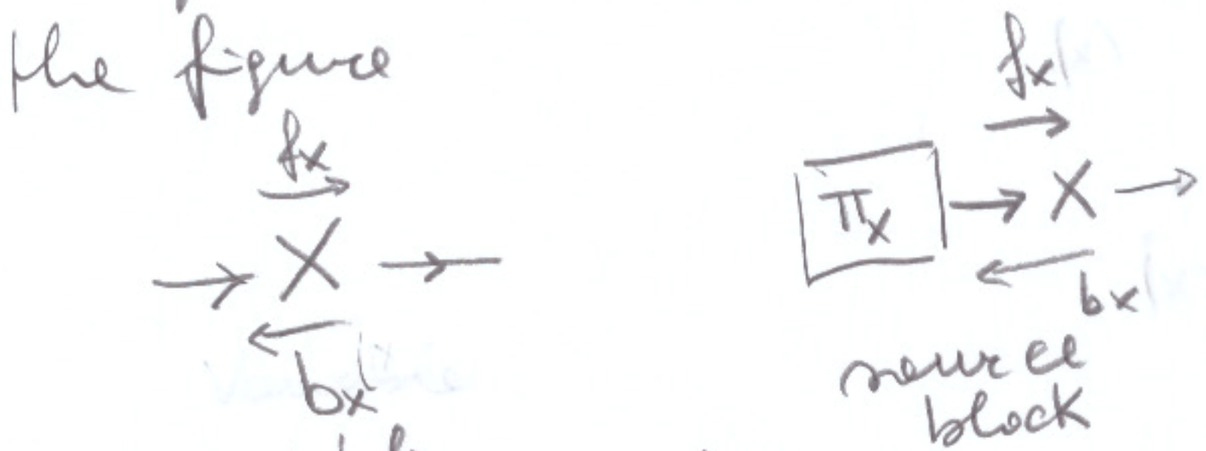
$$\begin{cases} X_{t+1} = \varphi(x_t, u_t) \\ y_t = \gamma(x_t) \end{cases}$$

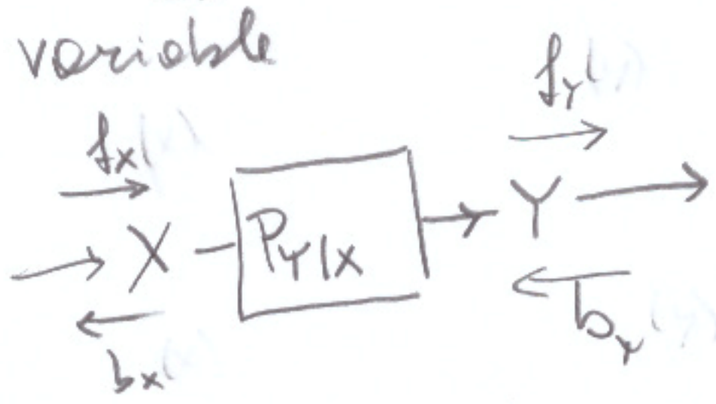
A fully blown factor graph in
reduced normal form is shown
in the following for $T = 3$


π_{U_1}
U_1
$(X_1^{(1)} U_1)$
π_{X_1}
$X_1^{(0)}$
$X_1^{(1)}$
$(X_1^{(1)} U_1)$
$X_1^{(2)}$
$P_{Y_1|X_1}$
Y_1
$(X_1^{(1)} U_1)$
$P_{X_2|X_1 U_1}$
$X_2^{(0)}$
π_{U_2}
U_2
$(X_2^{(1)} U_2)$
$X_2^{(1)}$
$(X_2^{(1)} U_2)$
$X_2^{(2)}$
$P_{Y_2|X_2}$
Y_2
$(X_2^{(1)} U_2)$
$P_{X_3|X_2 U_2}$
X_3
X_3
X_3
$P_{Y_3|X_3}$
Y_3

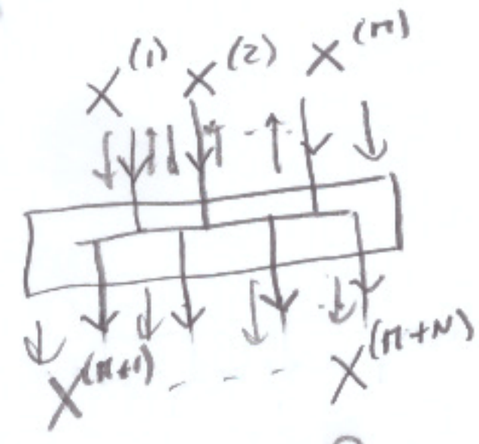
BASIC BLOCKS AND THE VECTOR-MATRIX REPRESENTATION

From the examples we have seen before, a factor graph in reduced normal form (FGrn) is composed only by the elements shown in the figure

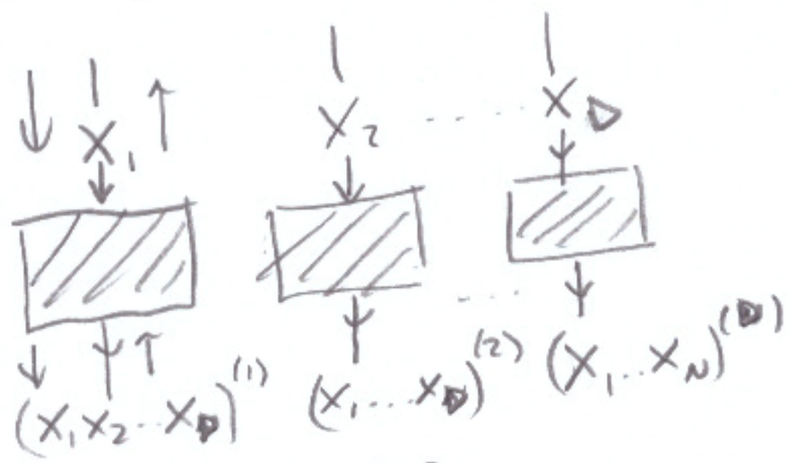




SISO block
(Single-Input Single-Output)



DIVERTER
OR EQUAL-CONSTRAINT
block



SHADED BLOCKS

DISCRETE VARIABLES (FINITE)

For Discrete variable that take values in a finite discrete sample space

$$X: x \in \mathcal{X} = \{x^1, x^2, \ldots, x^{N_X}\}$$

to describe probability distributions we can use functional notation or a vector notation. More specifically

for forward, backward and posterior distributions

$f_x(x)$, $b_x(x)$, $p_x(x)$ functional notation

$\underline{f}_x = \begin{bmatrix} f_x(x^1) \\ f_x(x^2) \\ \vdots \\ f_x(x^{N_x}) \end{bmatrix}$ $\underline{b}_x = \begin{bmatrix} b_x(x^1) \\ b_x(x^2) \\ \vdots \\ b_x(x^{N_x}) \end{bmatrix}$ $\underline{p}_x = \begin{bmatrix} p_x(x^1) \\ p_x(x^2) \\ \vdots \\ p_x(x^{N_x}) \end{bmatrix}$ vector notation

Vector notations often provide a more compact description of sum and product transformations
We describe now each element of FGrn in both notations

SOURCE BLOCK

A source block represents information on a variable

$\pi_x(x)$ $\underline{\pi}_x$

$f_x(x) = \pi_x(x)$ $\underline{f}_x = \underline{\pi}_x$

SISO BLOCK

The SISO block represents the core of the network as it represents the conditional distribution of Y given X

$X: x \in \mathcal{X} = \{x^1, x^2, \dots, x^{N_x}\}$ $Y: y \in \mathcal{Y} = \{y^1, y^2, \dots y^{N_y}\}$

$P_{Y|X}(y|x)$ $\quad$ $\underline{\underline{P}}_{Y|X}$ (Matrix)

$N_x \times N_y$

$$\underline{\underline{P}}_{Y|X} = \begin{bmatrix} Pr\{Y=y^1|X=x^1\} & Pr\{Y=y^2|X=x^1\} & \cdots & Pr\{Y=y^{N_y}|X=x^1\} \\ Pr\{Y=y^1|X=x^2\} & Pr\{Y=y^2|X=x^2\} & \cdots & Pr\{Y=y^{N_y}|X=x^2\} \\ \vdots & & & \\ Pr\{Y=y^1|X=x^{N_x}\} & Pr\{Y=y^2|X=x^{N_x}\} & \cdots & Pr\{Y=y^{N_y}|X=x^{N_x}\} \end{bmatrix}$$

$\underline{\underline{P}}_{Y|X}$ is row stochastic because (sum of elements of each row is 1)

$$\sum_{y\in\mathcal{Y}} P_{Y|X}(y|x) = 1 \quad \forall x \qquad \text{or} \qquad \sum_{y\in\mathcal{Y}} Pr\{Y=y|X=x^i\} = 1 \quad \forall i$$

FORWARD FLOW:

$$f_Y(y) = \sum_{x\in\mathcal{X}} P_{Y|X}(y|x) f_X(x)$$

$$\underline{1}_N \triangleq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \Big\} N$$

$$\underline{f}_Y = \underline{\underline{P}}_{Y|X}^T \underline{f}_X$$

If f_X is normalized $\left(\sum_x f_X(x) = 1 \;;\; \underline{f}_X^T \underline{1}_{N_x} = 1\right)$

also f_Y is normalized $\left(\sum_y f_Y(y) = 1 \;;\; \underline{f}_Y^T \underline{1}_{N_y} = 1\right)$

(law of total probability)

BACKWARD FLOW

$$b_X(x) = \sum_{y\in\mathcal{Y}} P_{Y|X}(y|x) b_Y(y)$$

$$\underline{b}_X = \underline{\underline{P}}_{Y|X} \underline{b}_Y$$

Here even if b_Y is normalized, b_X is not necessarily normalized. It would happen only if $\underline{\underline{P}}_{Y|X}$ were doubly-stochastic. (sum of both rows and columns are = 1)

Normalization is usually not necessary in probability BG
propagation because an un-normalized distribution
can recover its proper scale dividing by the sum
of its elements. This is why in probability
propagation, when we are not sure, or we do not
care, about scale we use the symbol
" $\propto$ " prop-to.

However
In practical distribution manipulations it is
often good practice to keep messages normalized
or at least properly scaled to avoid underflow.
Recall that multiplying probabilities (values
between 0 and 1) we may have very small numbers.

DIVERTOR

The diverter of equal-constraint block describes
the condition that all variables around it
are equal $X^{(1)} = X^{(2)} = \ldots = X^{(M)} = X^{(M+1)} = \ldots = X^{(M+N)}$

The way the arrows are oriented depends on
the graph in which it is inserted.
It acts as a function on the probability
pipelines and is such that all outgoing
messages are the product of the incoming ones.
With reference to the figure in which we
have divided the incoming variables $X^{(1)}, X^{(2)}, \ldots X^{(M)}$
and the outgoing variables $X^{(M+1)}, X^{(M+2)}, \ldots, X^{(M+N)}$

outgoing messages are

$$f_{x^{(i)}}(x) \propto \prod_{j=1}^{M} f_{x^{(j)}}(x) \prod_{\substack{l=M+1 \\ l \neq i}}^{M+N} b_{x^{(l)}}(x) \qquad i = M+1, \ldots, N+M$$

$$b_{x^{(i)}}(x) \propto \prod_{j=M+1}^{M+N} b_{x^{(j)}}(x) \prod_{\substack{l=1 \\ l \neq i}}^{M} f_{x^{(l)}}(x) \qquad i = 1, \ldots, M$$

$$\underline{f}_{x^{(i)}} \propto \bigodot_{j=1}^{M} \underline{f}_{x^{(j)}} \bigodot_{\substack{l=M+1 \\ l \neq i}}^{M+N} \underline{b}_{x^{(l)}} \qquad i = M+1, \ldots, N+M$$

$$\underline{b}_{x^{(i)}} \propto \bigodot_{j=M+1}^{N+M} \underline{b}_{x^{(j)}} \bigodot_{\substack{l=1 \\ l \neq i}}^{M} \underline{f}_{x^{(l)}} \qquad i = 1, \ldots, M$$

where $\odot$ denotes the element-by-element product (Hadamard product) (.* in Matlab).

SHADED BLOCKS

Shaded block represent mapping of a variable into a product space with other variables

The outgoing messages in functional form are

$$\oint_{(X_1 X_i \cdots X_N)^{(i)}} (x_1 x_2 \cdots x_N) \propto \oint_{x_i} (x_i) \prod_{\substack{j=1 \\ j \neq i}}^{D} \upsilon(x_j) \qquad i = 1, \cdots, N$$

$$b_{x_i}(x_i) \propto \sum_{x_1 \cdots x_{i-1} x_{i+1} \cdots x_D} \oint_{(X_1 X_2 \cdots X_D)^i} (x_1 x_2 \cdots x_N) \qquad i = 1, \cdots, N$$

In vector-matrix notation, the block can be represented by a $N_i \times N_1 N_2 \cdots N_N$ stochastic matrix

$$N_i \left[\overbrace{\quad P_{(X_1 X_2 \cdots X_N)^{(i)} | X_i} \quad}^{N_1 N_2 \cdots N_N} \right]$$

$$P_{(X_1 X_2 \cdots X_D)^{(i)} | X_i} = \frac{N_i}{\prod_{i=1}^{D} N_i} \underline{1}_{N_1}^T \otimes \cdots \otimes \underline{1}_{N_{i-1}}^T \otimes \underline{\underline{I}}_{N_i} \otimes \underline{1}_{N_{i+1}}^T \otimes \cdots \otimes \underline{1}_{N_D}^T$$

$\underline{\underline{I}}_{N_i}$: $N_i \times N_i$ identity matrix

$i = 1, \cdots, D$

To see where this definition comes from see the outlined part of the paper.

The other blocks are learned from examples. We have used the architecture first in the generative mode with hand-picked random parameters to generate $N = 100$ examples as delta distributions at X_1, X_2, and X_3. These examples are then used as backward delta messages at the same terminations. The variables' cardinalities in the learning graph are kept at the same values as in the generative model. We have compared the performances in terms of aggregated log-likelihood (28) for Ne $= 600$ EM learning cycles (epoques). The number of iterations per epoque Nit for the ML and KL algorithms have been fixed to 3. Fig. 16 shows the typical results of a simulation. The log-likelihood (top left) converges quite quickly, while the coefficients (other plots) take longer to reach a stable value. The graphs show the convergence for both visible and hidden-variable blocks. The prior blocks tend all to become uniform and are not shown for brevity. It is clear how the superiority of the ML and KL algorithms also holds for this architecture with the smoothest convergence for the coefficients and a better final log-likelihood. The LD algorithm often produces wide oscillations in the coefficients. We have run many more simulations to test for generalization and with mismatched values between the generative model and the learning graph. The results are very similar and are not reported here for brevity.

3) Other Architectures: We have performed other simulations on larger architectures with various topologies and variables' cardinalities. We have also applied the same algorithms to the experimental data with hierachical architectures [24] also using our Simulink library [25]. The results are very consistent with the behavior shown in the above experiments and seem to hold for large classes of cycle-free graphs. We believe that the results of these simulations may provide a useful guidance to the designers of learning graphs for the applications.

VI. Conclusion

We have discussed the application of four different learning algorithms to Bayesian discrete-value acyclic directed graphs in their FGrn form. The advantage of the FGrn is the great modularity in building adaptive architectures. Only SISO and source blocks' parameters must be learned as they can all run the same algorithm in a totally distributed fashion. The updating rules are based solely on the locally available backward and forward messages.

The superiority of the recursive EM multiplicative updates of the ML and KL algorithms, derived from local regularized cost functions, has been demonstrated with simulations on synthetic data. We believe that the comparisons presented in this paper, that to our knowledge have never been reported in the literature, may be relevant to the user that wishes to apply the belief propagation framework to his applications. Future papers will report more on the experimental data and deeper architectures.

Appendix A
Product Space Mappings

Suppose that we have D discrete variables $X_1, X_2, \ldots, X_D$, belonging to the spaces $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_D$, having cardinalities $M_1, M_2, \ldots, M_D$, respectively. The maps from the product space $\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_D$ to each variable are described by the matrices

$$\begin{aligned} P(X_1|X_1X_2\ldots X_D) &= I_{M_1} \otimes 1_{M_2} \otimes 1_{M_3} \otimes \cdots \otimes 1_{M_D} \\ P(X_2|X_1X_2\ldots X_D) &= 1_{M_1} \otimes I_{M_2} \otimes 1_{M_3} \otimes \cdots \otimes 1_{M_D} \\ &\cdots \\ P(X_D|X_1X_2\ldots X_D) &= 1_{M_1} \otimes 1_{M_2} \otimes 1_{M_3} \otimes \cdots \otimes I_{M_D} \end{aligned} \quad (30)$$

where I_Q denotes the $Q \times Q$ identity matrix, 1_Q is the Q-dimensional column vector with all ones, and $\otimes$ is the Kronecker product. All the matrices contain mostly zeros, but they are row stochastic, since only one element per row is equal to one.

For example, matrices (30) define the blocks at the bottom of Fig. 2, if $D = 3$ and if S is the whole product space: $S = (X_1X_2X_3)$.

Vice versa, the maps from each variable to the product space are the transposed normalized of (30)

$$\begin{aligned} &P((X_1X_2\ldots X_D)^{(1)}|X_1) \\ &\quad = \frac{M_1}{\prod_{i=1}^{D} M_i} I_{M_1} \otimes 1_{M_2}^T \otimes 1_{M_3}^T \otimes \cdots \otimes 1_{M_D}^T \\ &P((X_1X_2\ldots X_D)^{(2)}|X_2) \\ &\quad = \frac{M_2}{\prod_{i=1}^{D} M_i} 1_{M_1}^T \otimes I_{M_2} \otimes 1_{M_3}^T \otimes \cdots \otimes 1_{M_D}^T \\ &\quad \cdots \\ &P((X_1X_2\ldots X_D)^{(D)}|X_D) \\ &\quad = \frac{M_D}{\prod_{i=1}^{D} M_i} 1_{M_1}^T \otimes 1_{M_2}^T \otimes 1_{M_3}^T \otimes \cdots \otimes I_{M_D}. \end{aligned} \quad (31)$$

The matrices are row stochastic and have $(\prod_{i=1}^{D} M_i/M_j)$ nonzero equal elements in each row, reflecting the uniform ambiguity that, when given a single variable X_j, is left on the product space. In the graph, the ambiguity is resolved around the replicator block by the intersection (product rule) with messages coming from the other variables.

Matrices (31), for example, can be applied to define the blocks that, in Fig. 3(c), go from X_1, X_2, and X_3 to $(X_1X_2X_3)^{(1)}$, $(X_1X_2X_3)^{(2)}$, and $(X_1X_2X_3)^{(3)}$. Similarly matrices (31) can be used to define the blocks in Fig. 6 that go from $S_1^{(2)}$ to $(S_1^{(2)}S_2)^{(1)}$, from S_2 to $(S_1^{(2)}S_2)^{(2)}$, from Y_2 to $(Y_2S_3)^{(1)}$, and from S_3 to $(Y_2S_3)^{(2)}$.

As an example, the structure of matrices (30) and (31) for X_1, X_2, and X_3 having sizes $M_1 = 2$, $M_2 = 3$, and $M_3 = 2$, respectively, are

$$P(X_1|X_1X_2X_3) = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

$$P(X_2|X_1X_2X_3) = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$P(X_3|X_1X_2X_3) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$P((X_1X_2X_3)^{(1)}|X_1) = \frac{1}{6}P(X_1|X_1X_2X_3)^T$$

$$P((X_1X_2X_3)^{(2)}|X_2) = \frac{1}{4}P(X_2|X_1X_2X_3)^T$$

$$P((X_1X_2X_3)^{(3)}|X_3) = \frac{1}{6}P(X_3|X_1X_2X_3)^T. \quad (32)$$

APPENDIX B
DERIVATION OF THE ML ALGORITHM

To solve the ML problem (9), we seek to minimize the following cost function:

$$C(\theta) = -\sum_{n=1}^{N} L[n]\left[\log\left(\mathbf{f}_{X[n]}^T\theta\mathbf{b}_{Y[n]}\right) - \mathbf{1}_{M_Y}^T\theta^T\mathbf{f}_{X[n]}\right] \quad (33)$$

where we have added an extra term that is just equal to one $\forall\, n$, when θ is row stochastic and $\mathbf{f}_{X[n]}$ is a distribution. This term acts as a regularizer and leads to a very stable algorithm. The constrained problem is

$$\begin{cases} \min_\theta C(\theta) \\ -\theta_{ij} \le 0, \qquad i = 1 : M_X;\ \ j = 1 : M_Y \\ \sum_{j=1}^{M_Y}\theta_{ij} - 1 = 0, \quad i = 1 : M_X. \end{cases} \quad (34)$$

Note that the last two conditions automatically imply that $\theta_{ij} \le 1$. There are $M_X \times M_Y$ inequality and M_X equality constraints. Applying KKT conditions [22], we get

$$\begin{cases} \dfrac{\partial C(\theta)}{\partial\theta_{lm}} + \displaystyle\sum_{i=1}^{M_X}\sum_{j=1}^{M_Y}\lambda_{ij}\frac{\partial(-\theta_{ij})}{\partial\theta_{lm}} + \sum_{i=1}^{M_X}\beta_i\frac{\partial}{\partial\theta_{lm}}\left(\sum_{j=1}^{M_Y}\theta_{ij} - 1\right) = 0 \\ -\theta_{lj} \le 0; \quad \displaystyle\sum_{j=1}^{M_Y}\theta_{lj} - 1 = 0 \\ \lambda_{lm} \ge 0; \quad \lambda_{lm}(-\theta_{lm}) = 0;\ \ l = 1 : M_X;\ \ m = 1 : M_Y. \end{cases} \quad (35)$$

Taking the derivatives

$$\begin{cases} -\displaystyle\sum_{n=1}^{N} L[n]\left(\frac{f_{X[n]}(l)b_{Y[n]}(m)}{\mathbf{f}_{X[n]}^T\theta\mathbf{b}_{Y[n]}} - f_{X[n]}(l)\right) - \lambda_{lm} = 0 \\ -\theta_{lj} \le 0; \quad \displaystyle\sum_{j=1}^{M_Y}\theta_{lj} - 1 = 0 \\ \lambda_{lm} \ge 0; \quad \lambda_{lm}(-\theta_{lm}) = 0;\ \ l = 1 : M_X;\ \ m = 1 : M_Y. \end{cases} \quad (36)$$

Therefore, λ_{lm} and the complementary slackness conditions become

$$\lambda_{lm} = -\sum_{n=1}^{N} L[n]\left(\frac{f_{X[n]}(l)b_{Y[n]}(m)}{\mathbf{f}_{X[n]}^T\theta\mathbf{b}_{Y[n]}} - f_{X[n]}(l)\right) \times \sum_{n=1}^{N} L[n]\left(\theta_{lm}\frac{f_{X[n]}(l)b_{Y[n]}(m)}{\mathbf{f}_{X[n]}^T\theta\mathbf{b}_{Y[n]}} - \theta_{lm}f_{X[n]}(l)\right) = 0$$
$$l = 1 : M_X; \quad m = 1 : M_Y. \quad (37)$$

It is easy to verify that $\lambda_{lm} \ge 0$ for all l, m. From the complementary slackness condition and the constraints, the iterations for the ML algorithm follow immediately.

REFERENCES

[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann, 1988.

[2] M. I. Jordan, E. B. Sudderth, M. Wainwright, and A. S. Willsky, "Major advances and emerging developments of graphical models," *IEEE Signal Process. Mag.*, vol. 27, no. 6, p. 17 and 138, Nov. 2010.

[3] J. A. Costa and A. O. Hero, "Geodesic entropic graphs for dimension and entropy estimation in manifold learning," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2210–2221, Aug. 2004.

[4] S. J. Hwang, "Geometric representations of high dimensional random data," Ph.D. dissertation, Electrical Engineering: Systems, Univ. Michigan, Ann Arbor, MI, USA, 2012.

[5] M. I. Jordan, *Learning in Graphical Models*. Norwell, MA, USA: Kluwer, 1998.

[6] R. E. Neapolitan, *Learning Bayesian Networks*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2004.

[7] K. B. Korb and A. E. Nicholson, *Bayesian Artificial Intelligence*. London, U.K.: Chapman & Hall, 2004.

[8] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[9] D. Koller and N. Friedman, *Probabilistic Graphical Models. Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2010.

[10] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[11] G. D. Forney, Jr., "Codes on graphs: Normal realizations," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 520–548, Feb. 2001.

[12] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[13] J. Binder, D. Koller, S. Russell, and K. Kanazawa, "Adaptive probabilistic networks with hidden variables," *Mach. Learn.*, vol. 29, no. 2, pp. 213–244, 1997.

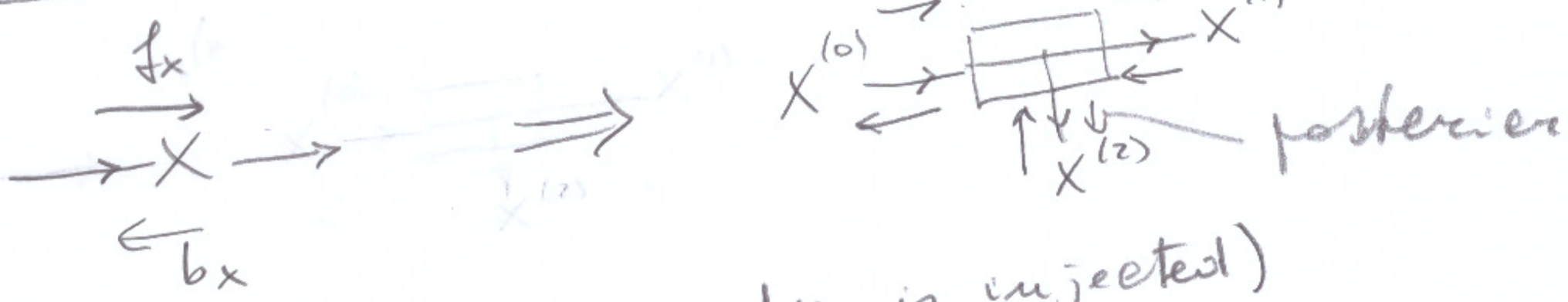
POSTERIOR CALCULATION

Posterior distributions can be obtained on any branch of the Fbrm as the product of forward and backward

$$P_x(x) \propto f_x(x)\, b_x(x)$$

$$\underline{P}_x \propto \underline{f}_x \odot \underline{b}_x$$

Posterior distribution can also be <u>spilled out</u> using a divertor



$\underline{b}_{x^{(2)}} = \text{Uniform}$ (No information is injected)

Set $f_{x^{(0)}} = f_x$ and $b_{x^{(1)}} = b_x$

Then $b_{x^{(0)}} \propto b_x$, $f_{x^{(1)}} \propto f_x$

and $f_{x^{(2)}} \propto f_{x^{(0)}}\, b_{x^{(1)}} \propto f_x\, b_x$