Probability-Theoretic Junction Trees

Payam Pakzad,

(with Venkat Anantharam,
EECS Dept, U.C. Berkeley)

EPFL, ALGO/LMA Seminar
2/12/2004

Marginalization Problem

Given an arbitrary function of many variables, find (some of) its 'marginals'. For example

$$\beta_1(x_1) = \sum_{x_2,\cdots,x_n} \beta(x_1,\cdots,x_n)$$

or even

$$\gamma_1(x_1) = \max_{x_2,\cdots,x_n} \beta(x_1,\cdots,x_n)$$

e.g. β (...) could be the joint density on variables, then

β_1 (x_1) = marginal probability of x_1, and

γ_1 (x_1) = probability of most likely conf. of rest of variables

Applications

- Error Correcting Codes
- Image Processing
- Computer Vision
- Networking
- Statistical Physics

Problem: Computations intractable. State space is exponential in number of variables.

Product Functions and Factorization

Suppose β factorizes as product of functions of smaller subsets of variables. e.g.

Markov chain model: $$\beta(x_0,\cdots,x_n) = p(x_0)\prod_{i=1}^{n} p(x_i \mid x_{i-1})$$

Then certain 'conditional independencies' may hold which can simplify the marginalization task.

E.g. $(x_0,\ldots,x_i)$ is cond. indep. of $(x_{i+2},\ldots,x_n)$ given x_{i+1}, hence

$$\sum_{\{x_0,\cdots,x_n\}\setminus x_{i+1}} \beta = \left(\sum_{\{x_0,\cdots,x_i\}} p(x_0)\prod_{j=1}^{i+1} p(x_j \mid x_{j-1}) \right)\left(\sum_{\{x_{i+2},\cdots,x_n\}} \prod_{j=i+2}^{n} p(x_j \mid x_{j-1}) \right)$$

Product Functions - Notation

Given

state vector: $\mathbf{x} = (x_1,\dots, x_N)$

a collection of local domains, *R*: $\forall r \in R,\ r \subset \{1,\cdots,N\}$

and local functions (kernels): $\alpha_r(\mathbf{x}_r)\,,\quad r \in R$

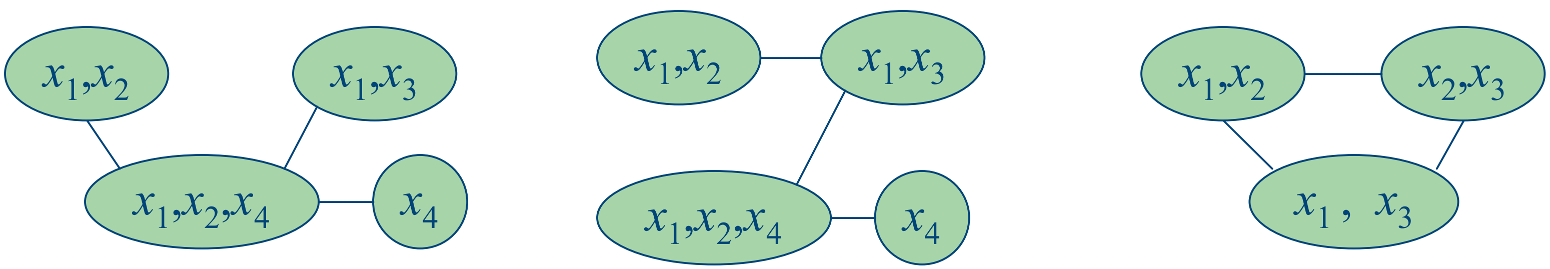
Objective function factorizes: $\beta(\mathbf{x}) \equiv \prod_{r \in R} \alpha_r(\mathbf{x}_r)$

Find marginals: $\beta_r(\mathbf{x}_r) \equiv \sum_{\mathbf{x} \setminus \mathbf{x}_r} \beta(\mathbf{x})$

Independency Graphs

A graph with nodes corresponding to domains $r \in R$, s.t. separation on graph implies (conditional) independence.

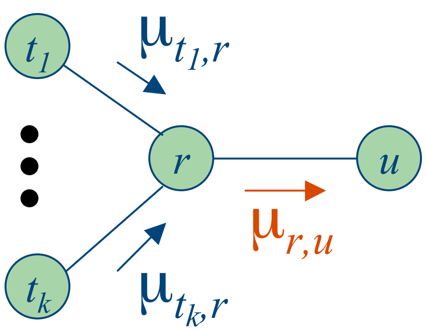
A tree is called a 'Junction Tree' if whenever a node r_0 separates nodes r_1 and r_2 on tree, then $r_1 \cap r_2 \subseteq r_0$



Junction Tree Algorithm

(a.k.a. GDL, Belief Propagation, sum-product, …)

- Define 'messages' $\mu_{r,u}(x_{u\cap r})$ for each edge (r, u) of the independency graph. Initialize to 1.



- Update messages as:

$$\mu_{r,u}(x_{u\cap r}) \equiv \sum_{x_r \setminus x_u} \alpha_r(x_r) \prod_{t\in N(r)\setminus\{u\}} \mu_{t,r}(x_{r\cap t})$$

- Define 'Beliefs' $$b_r(x_r) \equiv \alpha_r(x_r) \prod_{t\in N(r)} \mu_{t,r}(x_{r\cap t})$$

Junction Trees

If graph is junction tree, algorithm converges in finite time, and then beliefs $b_r(x_r)$ are the marginals of the product function.

Given regions, can 'quickly' determine if a junction tree exists. Involves finding a maximal-weight spanning tree of regions.

If none exist, can always expand regions in a way to create a junction tree, but complexity is exponential in the size of regions.

Graphs With Cycles - Perspective

B.P. on graphs with loops is used as approximation to the marginals, e.g. in Turbo codes and LDPC decoding.

Attempts at justification of approximation on loopy graphs:

- One-loop case (Weiss, Horn et. al.)
- Gaussian case (Van Roy, Weiss)
- Fixed points and stability (Richardson)
- LDPC studies (Gallager, McKay & Neal, Luby et. al., Urbanke & Richardson, Shokrollahi).
- Tree-based Reparametrization (Wainwright)

Probability-theoretic View

Motivation:

- Relying on variables can conceal partial structure in state-space; can only recognize/create ‘rectangular decompositions.’

- Generalize the problem to arbitrary state-spaces; State-space need not be represented by ‘variables.’

Example

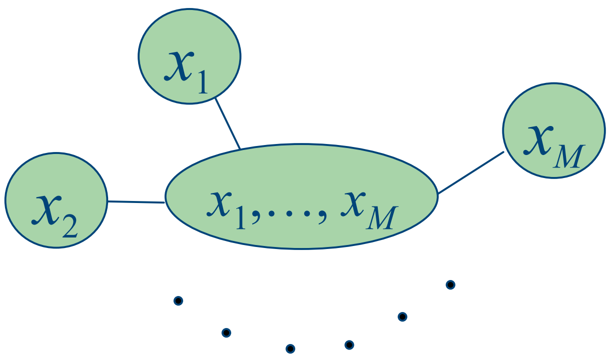
Let $x_1, \ldots, x_M$ take value in (a finite semi-group) A and let $\mu(x_1,\ldots, x_M) = f(x_1+\ldots+ x_M)$, $p_i(x_i)$ be real functions.

We would like to calculate

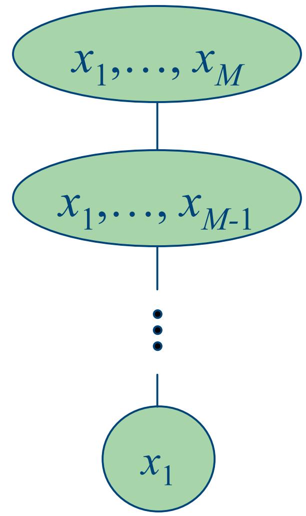
$$E = \sum_{x_1,\cdots,x_M} \mu(x_1,\cdots,x_M)\prod_{i=1}^{M} p_i(x_i)$$

Example, cntd.

GDL Solution:



Or

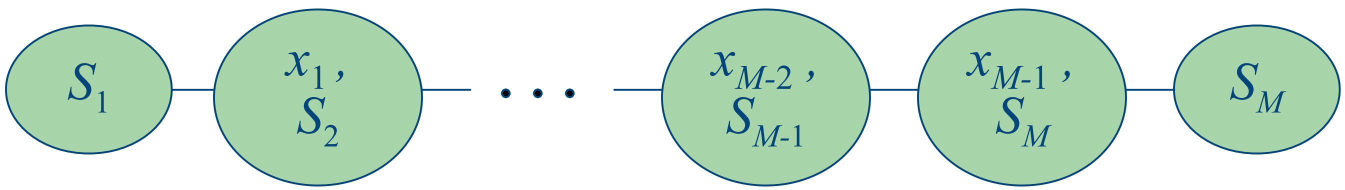


$$E = \sum_{x_1} p_1(x_1) \sum_{x_2} p_2(x_2) \cdots \sum_{x_M} \mu(x_1, \cdots, x_M) p_M(x_M)$$

Requires $O(|A|^M)$ additions and multiplications.

Example, cntd.

In comparison, with $S_i \equiv x_i + \ldots + x_M \in A$



is an independency tree, suggesting

$$E = \sum_{S_1} f(S_1) \sum_{x_1 + S_2 = S_1} p_1(x_1) \cdots \sum_{x_{M-1} + S_M = S_{M-1}} p_M(S_M) p_{M-1}(x_{M-1})$$

This requires only $O(M|A|^2)$ additions and multiplications.

Probability-theoretic GDL

A systematic theory to recognize and exploit such structures.

Reformulate the MPF problem as taking conditional expectations in an arbitrary sample space.

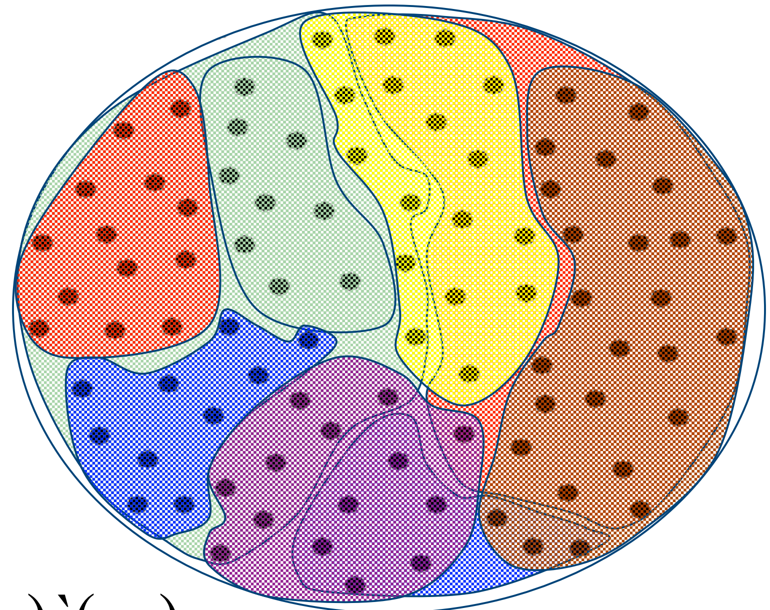
GDL	PGDL
Variables $x_1, \ldots, x_N$	Arbitrary representation
Local domains $r \in R$	σ-fields $\mathcal{F}_1, \ldots, \mathcal{F}_M$
Local functions α_r , $r \in R$	Random variables $X_1, \ldots, X_M$
Marginals $\beta_r = \sum_{x \setminus x_r} \prod X_r$	Cond. Expectations $E[\prod X_i \mid \mathcal{F}_j]$

Preliminaries - Notation

- A discrete state-space Ω
- A measure $\mu(.)$ on Ω

λ σ-fields $\mathcal{F}_i$ on Ω

λ And r.v.'s $X_i \in \mathcal{F}_i$

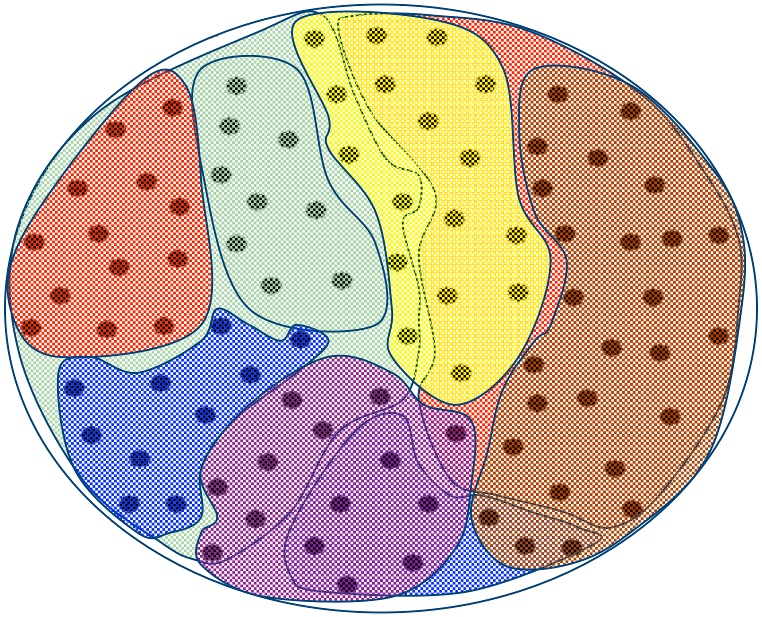


- Want to calculate:

$$E\left[\prod_{i=1}^{M} X_i \middle| \mathcal{F}_j\right](f) = \frac{1}{\mu(f)} \sum_{\omega \in f} \prod_{i=1}^{M} X_i(\omega)\mu(\omega)$$

for each (nonzero measure) atom f of $\mathcal{F}_j$

Preliminaries, cntd.



- Augmentation: $\mathcal{F}_{\{1,2\}} \equiv \mathcal{F}_1 \vee \mathcal{F}_2$

- Conditional Independence:

  Say $\mathcal{F} \perp\!\!\!\perp \mathcal{G} \mid \mathcal{H}$ if

  for each atom h of $\mathcal{H}$:

  - If $\mu(h) \neq 0$ then
    $$\forall f \in \mathcal{F},\ g \in \mathcal{G},\quad \text{ì}\,(f,g,h)\text{ì}\,(h) = \text{ì}\,(f,h)\text{ì}\,(g,h)$$

  - If $\mu(h) = 0$ then $\forall f \in \mathcal{F},\ g \in \mathcal{G},\quad \text{ì}\,(f,g,h) = 0$

Marginalization Problem and Junction Trees

Given a collection $(\Omega, \{\mathcal{F}_1,\ldots, \mathcal{F}_M\}, \mu)$ of meas. spaces,

and r.v.'s $X_i \in \mathcal{F}_i$:

MPF Problem: For one or more $i \in \{1, \ldots, M\}$ find $E[\prod_j X_j | \mathcal{F}_i]$

A tree with nodes $\{1,\ldots, M\}$ is called a junction tree if:

$\forall\, A, B \subset \{1, \ldots, M\}$ and $i \in \{1, \ldots, M\}$ s.t.

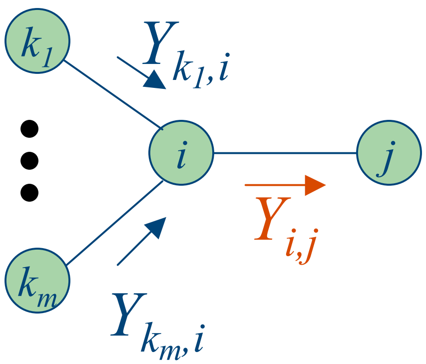
i separates A and B on tree, we have $\mathcal{F}_A \perp\!\!\!\perp \mathcal{F}_B \,|\, \mathcal{F}_i$

As before, a junction tree captures the independencies

Probabilistic Junction Tree Algorithm

The following algorithm solves the MPF problem (c.f. GDL)

- Define 'messages' $Y_{i,j} \in \mathcal{F}_j$ for each edge (i, j) of the junction tree. Initialize to 1.



- Update messages as:

$$Y_{i,j} = \mathrm{E}\left[X_i \prod_{k \in N(i) \setminus \{j\}} Y_{k,i} \,\middle|\, \mathcal{F}_j\right]$$

- Define 'Beliefs'

$$B_i \equiv X_i \prod_{k \in N(i)} Y_{k,i}$$

- At termination $B_i = \mathrm{E}\left[\prod_{j=1}^{M} X_j \,\middle|\, \mathcal{F}_i\right]$

Existence of Junction Trees

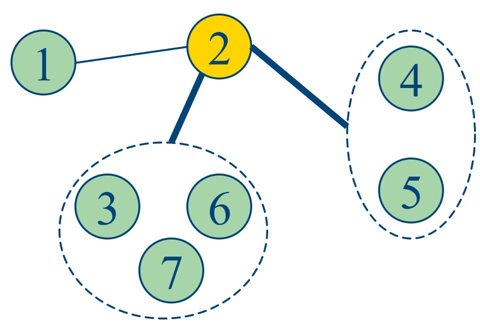
Main results:

Given a collection $(\Omega, \{\mathcal{F}_1,\ldots,\mathcal{F}_M\}, \mu)$ of meas. spaces:

- $\forall i = 1,\ldots, M$, there exists a partition P_i of $\{1,\ldots, M\}\backslash\{i\}$ (called Finest Valid Partition w.r.t. i), and
- If a junction tree exists, then for each $i = 1,\ldots, M$ there exists a J.T. compatible with P_i.

Example: Suppose M=7 and

$P_2=\{\{1\}\ \{3,6,7\}\ \{4,5\}\}$ then:

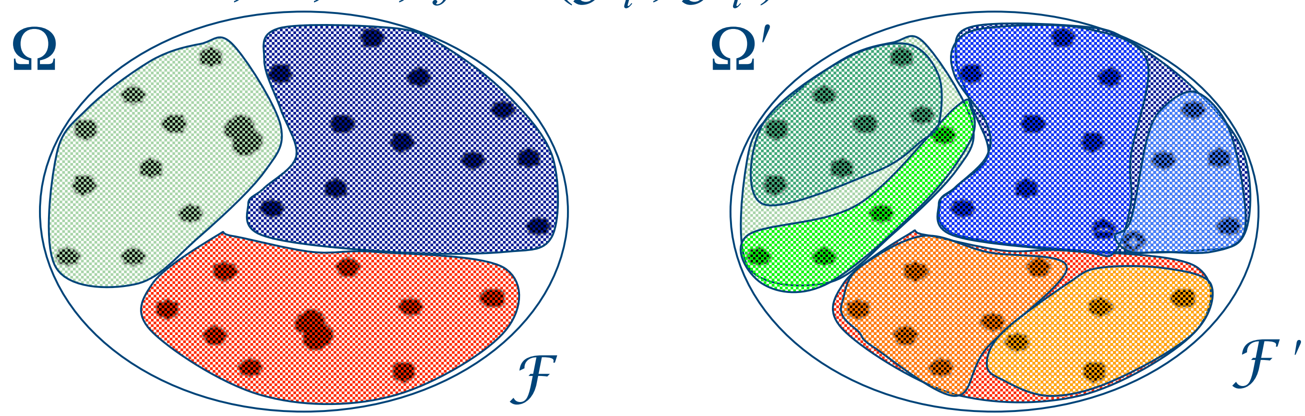


Lifting

Let $(\Omega, \{\mathcal{F}_1,\ldots, \mathcal{F}_M\}, \mu)$ be a collection of meas. spaces.

Another collection $(\Omega', \{\mathcal{F}_1',\ldots, \mathcal{F}_M'\}, \mu')$ is a *lifting* of $(\Omega, \{\mathcal{F}_1,\ldots, \mathcal{F}_M\}, \mu)$ if there is a map $f : \Omega' \rightarrow \Omega$ s.t.:

- μ' is consistent with μ under f, and
- for all $i = 1,\ldots, M$, f is $(\mathcal{F}_i', \mathcal{F}_i)$-measurable.



Lifting to Create Conditional Independency

Given $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$ we like to lift in a way so to have:

$$\mathcal{F}_1' \perp\!\!\!\perp \mathcal{F}_3' \mid \mathcal{F}_2' .$$

Corresponds to certain matrices of joint measures being rank one.

If some of these matrices are not rank-one, find a rank-one decomposition.

$\Rightarrow$ 'lifting' obtained by splitting the atom corresponding to the matrix.

Algorithm to Create a Junction Tree

This algorithm will find a J.T. on $\{\mathcal{F}_1,\ldots,\mathcal{F}_M\}$ if one exists:

- Pick any node $i \in \{1,\ldots,M\}$ as the root,
- Find *any* valid partition w.r.t. i, say $\{c_1,\ldots,c_l\}$
- For each $j = 1$ to l
  - Pick a node $t \in c_j$ and split atoms of $\mathcal{F}_t$ s.t. $\mathcal{F}_i \perp\!\!\!\perp \mathcal{F}_{c_j} \mid \mathcal{F}_t$.
  - Find a J.T. on c_j, with t as the root. Attach this tree by adding the edge (i,t).
- End

Example 2: Exact Decoding of LDPC Codes

$\mathbf{H} \in \mathrm{GF}(2)^{m \times n}$: Parity check matrix of an LDPC code with m ‘checks’ and block size n

- Codewords satisfy $\mathbf{H}\,\mathbf{x}=0$, with *a posteriori* probabilities

$$P^*(\mathbf{x}) = \frac{1}{Z}\prod_{i=1}^{n} P(x_i)P(y_i^* \mid x_i)\prod_{j=1}^{m} 1(H_j \cdot \mathbf{x} = 0)$$

- Objective: find marginals of P^*:

$$P^*(x_i) = \sum_{\mathbf{x} \setminus x_i} P^*(\mathbf{x})$$

Example 2, cntd.

- Exact naive GDL solution: triangulate the graph, and run J.T. algorithm on the tree of cliques.
- Problem: graph highly connected, so cliques are large, algorithm inefficient.
- PGDL solution:

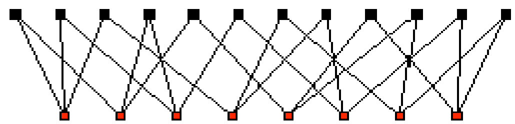
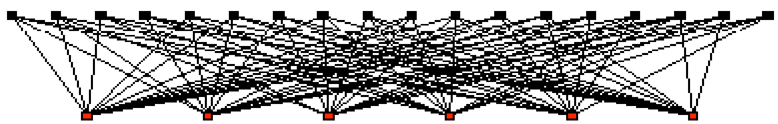
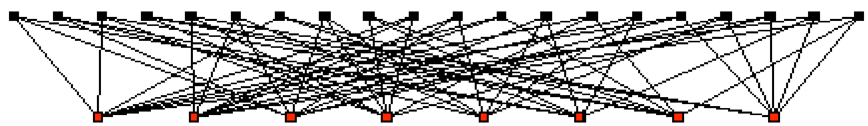
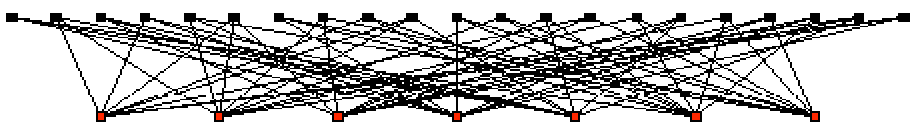
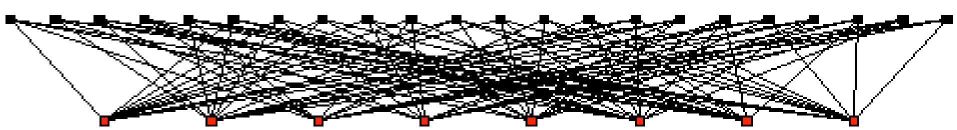
  state-space = the codebook; uniform measure;

  random variables = $P(x_i = j)P(y_i^* \mid x_i = j)$

  original σ-fields = directions of variables x_i's

  ⇒Lift using the automatic Matlab algorithm to form a chain.

Example 2, Results:

n	m	c	rate		Complexity GDL	Complexity PGDL
12	8	2	5/12		1.8×10^{3}	360
18	6	4	2/3		1.0×10^{6}	5.95×10^{3}
20	8	3	3/5		1.4×10^{6}	2.02×10^{4}
21	7	3	2/3		1.8×10^{6}	2.14×10^{4}
22	8	4	7/11		8.4×10^{6}	2.40×10^{4}

Concluding Remarks

- As with GDL, our algorithm generalizes to any ‘semi-field’ (e.g. max-product).
- Cost of lifting is high, but a junction tree can be used over all sets of observations.
- The complete sample space Ω is very large, but the resulting algorithm only requires storage of the matrices of joint densities of atoms of neighboring σ-fields.
- The conventional method corresponds to the case of a product space, with a uniform measure.
- “Measure theory” is transparent to the end user.