Introduction to Graphical Models and Inference for Communications

Pablo Martinez Olmos
olmos@tsc.uc3m.es

uc3m

Index

1. Course overview

2. Representing probability distributions via Factor graphs
   - Directed graphical models
   - Undirected graphical models
   - Factor graphs

3. Factor graphs in digital communications scenarios
   - Transmission of linear block codes over a memoryless channel
   - Symbol transmission and detection over ISI channels

Graphical Models and approximate inference

- Exact inference over discrete trees: the Belief Propagation algorithm.
- Approximate Inference over graphs with cycles using BP. .
- Inference over Gaussian distributions using BP.
- Beyond BP: Mean field approximations, Expectation propagation and Monte Carlo approximations.

Graphical Models and approximate inference for communications

- Exact inference over discrete trees: the Belief Propagation algorithm.
  - The forward/backward BCJR algorithm is nothing but the BP algorithm.
- Approximate Inference over graphs with cycles using BP. The Bethe approximation and beyond.
  - Codes on graphs and iterative algorithms: achieving channel capacity at low-cost.
- Inference over Gaussian distributions using BP.
  - The Kalman Filter is nothing but the BP algorithm.
- Beyond BP: Mean field approximations, Expectation propagation and Monte Carlo approximations.
  - Approximate inference in digital-communication receivers.

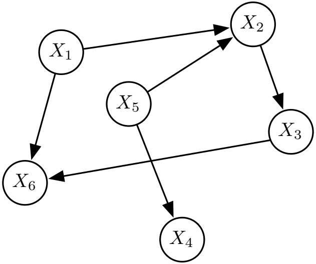
Graphical Models and approximate inference for communications

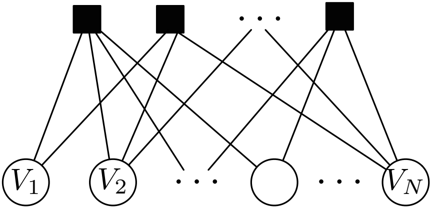
Learning by programming and simulating algorithms ...

- Exact inference over discrete trees: the Belief Propagation algorithm.
  - The forward/backward algorithm is nothing but the BP algorithm.
  - Bit-MAP decoding of convolutional codes using BP.
- Approximate Inference over graphs with cycles using BP. The Bethe approximation and beyond.
  - Codes on graphs and iterative algorithms: achieving channel capacity at low-cost.
  - BP decoding of Turbo codes.
  - BP decoding of binary LDPC codes.
- Inference over Gaussian distributions using BP.
  - The Kalman Filter is nothing but the BP algorithm.
  - Approximate message passing for Compressed Sensing.
  - Gaussian message-passing on linear models.
- Beyond BP: Mean field approximations, Expectation propagation and Monte Carlo approximations.
  - Approximate inference in digital-communication receivers.
  - Iterative receivers for digital communications.

Graphical Models

- Graphical Models (GMs) bringht together graph theory and probability theory in a powerful formalism for multivariate statistical modeling.
- **In a nutshell:** GMs use graphs to represent and manipulate joint probability distributions.





You already know something about inference over GMs

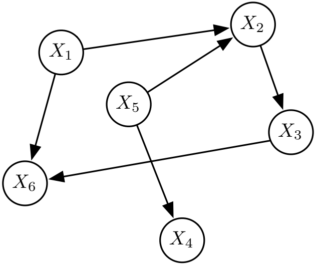
- Statistical models have long been formulated in terms of graphs in applied fields including bioinformatics, speech processing, image processing, coding theory in communications.
- Many widely known algorithms are expressed in terms of recursions operating on these graphs:
  - Kalman filtering for state-space models.
  - Forward-backward algorithm for hidden Markov models.
  - Viterbi algorithm.
  - Iterative message passing decoders for LDPC codes.
  - ...

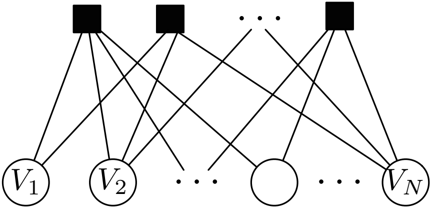
These ideas can be understood, unified and generalized within the formalism of graphical models.

GMs to represent probability distributions

Probabilistic graphical models are used to approximate such distributions by specifying a set of **conditional independence** or causality relationships, reducing the number of parameters needed to specify the distribution.

Graph theory also plays a fundamental role in assessing the **computational complexity of inference** algorithms over a GM.





GMs: learn the model/learn from the model

There are two main problems associated to graphical models:

1. Graphical Model Selection (Learn a graphical model from samples).
2. Efficient inference in Graphical Models.

Graphical Model Selection

d-dimensional random vector

$\mathbf{X} = (X_1, X_2, \ldots, X_d)$

A set of N observations
(In general $d \neq m$)

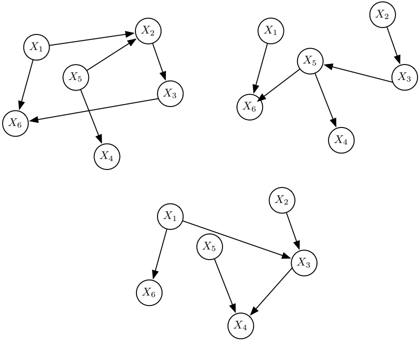
$\mathbf{Y}^1 = [Y_1^1 \; Y_2^1 \; \ldots \; Y_m^1]$

$\mathbf{Y}^2$

$\ldots$

$\mathbf{Y}^N$

d=6 $\Big\downarrow$?



- Graph structure
- Parameters of the distribution
- Sparsity
- ...

Exact Inference: the discrete case

Let $\boldsymbol{X}$ be a discrete random vector and $\boldsymbol{Y}$ a noisy observation, where $\boldsymbol{x} \in \mathcal{X}^n$, $\boldsymbol{y} \in \mathcal{Y}^m$ and the joint pmf is $p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{y})$. For a given observation $\boldsymbol{Y} = \boldsymbol{y}$, we are interested in drawing conclusions about $\boldsymbol{X}$.

Two basic kind of computations:

Given $\boldsymbol{Y} = \boldsymbol{y}$, find the most probable realization of the posterior probability distribution:

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in \mathcal{X}^n} p_{\boldsymbol{X}|\boldsymbol{y}}(\boldsymbol{x}) = \arg\max_{\boldsymbol{x} \in \mathcal{X}^n} \frac{p_{\boldsymbol{Y}|\boldsymbol{x}}(\boldsymbol{y}) p_{\boldsymbol{X}}(\boldsymbol{x})}{p_{\boldsymbol{Y}}(\boldsymbol{y})} = \arg\max_{\boldsymbol{x} \in \mathcal{X}^n} p_{\boldsymbol{Y}|\boldsymbol{x}}(\boldsymbol{y}) p_{\boldsymbol{X}}(\boldsymbol{x})$$

Complexity $\rightarrow \mathcal{O}(|\mathcal{X}|^n)$.

Exact Inference: the discrete case (II)

Given $\boldsymbol{Y} = \boldsymbol{y}$, compute the *marginal* probability distribution $p_{X_i|\boldsymbol{y}}(x_i)$:

$$\begin{aligned} p_{X_i|\boldsymbol{y}}(x_i) &= \sum_{\boldsymbol{x}_{\sim i} \in \mathcal{X}^{n-1}} p_{\boldsymbol{X}|\boldsymbol{y}}(\boldsymbol{x}) \\ &= \frac{1}{p_{\boldsymbol{Y}}(\boldsymbol{y})} \sum_{\boldsymbol{x}_{\sim i} \in \mathcal{X}^{n-1}} p_{\boldsymbol{Y}|\boldsymbol{x}}(\boldsymbol{y}) p_{\boldsymbol{X}}(\boldsymbol{x}), \end{aligned}$$

Complexity $\rightarrow \mathcal{O}(|\mathcal{X}|^n)$.

Structure helps!

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = p_{X_1}(x_1)p_{X_2|x_1}(x_2)p_{X_3|x_2}(x_3)p_{X_4|x_3}(x_4)p_{X_5|x_4}(x_5)$$

How many operations we need to compute $\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}\in\mathcal{X}^n} p_{\boldsymbol{X}}(\boldsymbol{x})$?

- **Brute force**: 5 variables $\Rightarrow |\mathcal{X}|^5$ evaluations of $p_{\boldsymbol{X}}(\boldsymbol{x})$
- **Efficient Inference**: It is easy to check that it only costs $5|\mathcal{X}|^2$ function evaluations

When the probability distribution factorizes, we can achieve huge computational gains.

In other words...

When the probability distribution is represented by a GM, we can achieve huge computational gains.

Inference and GMs

- It is critical to understand for which graphical structures efficient inference (polynomial complexity with the dimension n) is possible.
- We will use such knowledge to propose feasible and accurate **approximate inference methods** for those GMs for which **exact inference** is prohibitively complex.

Exact Inference: the continuous case

Assume now that both $\boldsymbol{X}$ and $\boldsymbol{Y}$ are real-valued random vectors with joint pdf $p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y})$ where $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \mathbb{R}^m$ and that we observe $\boldsymbol{Y} = \boldsymbol{y}$.

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}\in\mathbb{R}^n} p_{\boldsymbol{X}|\boldsymbol{y}}(\boldsymbol{x}) = \arg\max_{\boldsymbol{x}\in\mathbb{R}^n} \frac{p_{\boldsymbol{Y}|\boldsymbol{x}}(\boldsymbol{y})p_{\boldsymbol{X}}(\boldsymbol{x})}{p_{\boldsymbol{Y}}(\boldsymbol{y})} = \arg\max_{\boldsymbol{x}\in\mathbb{R}^n} p_{\boldsymbol{Y}|\boldsymbol{x}}(\boldsymbol{y})p_{\boldsymbol{X}}(\boldsymbol{x})$$

Typically $p_{\boldsymbol{X}|\boldsymbol{y}}(\boldsymbol{x})$ is not convex and multimodal

$$p_{X_i|\boldsymbol{y}}(x_i) = \int p_{\boldsymbol{X}|\boldsymbol{y}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}_{\sim\mathtt{i}} = \frac{1}{p_{\boldsymbol{Y}}(\boldsymbol{y})} \int p_{\boldsymbol{Y}|\boldsymbol{x}}(\boldsymbol{y})p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x}_{\sim\mathtt{i}}$$

This integral lacks in general of analytical solution

Special case: Gaussian distribution.

Structure helps!

- Efficient *approximate inferece* over $p_{\boldsymbol{X}|\boldsymbol{y}}(\boldsymbol{x})$ for real-valued distributions will be attained by constructing tractable approximation $q_{\boldsymbol{X}}(\boldsymbol{x})$ where inference is possible.
- Let $q_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{\theta})$ a family of distributions parameterized by a given vector $\boldsymbol{\theta}$. Assuming $p_{\boldsymbol{X}|\boldsymbol{y}}(\boldsymbol{x})$ and $q_{\boldsymbol{X}}(\boldsymbol{x}|\boldsymbol{\theta})$ have the same support, two common criterions:

$$q_{\boldsymbol{X}}(\boldsymbol{x}|\hat{\boldsymbol{\theta}}) = \max_{\boldsymbol{\theta}} KL(q_{\boldsymbol{X}}||p_{\boldsymbol{X}}) \qquad \text{Variational Inference}$$

$$q_{\boldsymbol{X}}(\boldsymbol{x}|\hat{\boldsymbol{\theta}}) = \max_{\boldsymbol{\theta}} KL(p_{\boldsymbol{X}}||q_{\boldsymbol{X}}) \qquad \text{Expectation Propagation}$$

- Feasible methods exploit the properties of the GM that represents $p_{\boldsymbol{X}|\boldsymbol{y}}(\boldsymbol{x})$!

Index

1. Course overview

2. Representing probability distributions via Factor graphs
   - Directed graphical models
   - Undirected graphical models
   - Factor graphs

3. Factor graphs in digital communications scenarios
   - Transmission of linear block codes over a memoryless channel
   - Symbol transmission and detection over ISI channels

Graphical models

- There are three main graph-based languages for describing probability distributions: **Bayessian Networks** (directed graphs), **Markov Random Fields** (undirected graphs) and **Factor graphs**.

- They differ in the set of conditional independences they can encode and in the factorization of the distribution that they induce.

- For the purpose of solving inference problems, **it is often convenient to convert both directed and undirected graphs into factor graphs**.

Bayesian networks (directed graphical models)

- They efficiently represent large joint distributions by making a set of **conditional independence** (CI) assumptions.

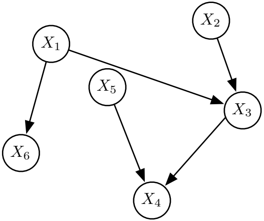
$$X \perp Y | Z \Leftrightarrow p_{X,Y|z}(x, y) = p_{X|z}(x) p_{Y|z}(y)$$

- A Bayesian network is a probability distribution of the form

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{n} p_{X_i|\boldsymbol{x}_i^\pi}(x_i)$$

  where $\boldsymbol{x}_i^\pi$ is the set of parents of X_i.

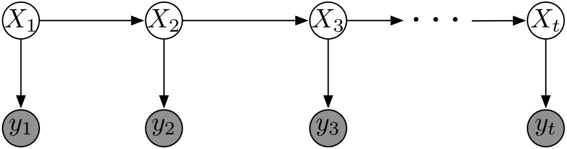
- Represented as a **directed graph**.



$$p_{\boldsymbol{X}}(\boldsymbol{x}) = p_{X_1}(x_1) p_{X_2}(x_2) p_{X_3|x_1,x_2}(x_3) p_{X_5}(x_5) p_{X_4|x_3,x_5}(x_4) p_{X_6|x_1}(x_6)$$

Hidden Markov Models

- BNs are well-suited to use known causal relationships and to express conditional dependences between variables in the graph.
- HMMs are naturally represented by BNs.



$$p_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y}) = p_{\boldsymbol{X}_1}(\boldsymbol{x}_1)p_{\boldsymbol{Y}_1|\boldsymbol{x}_1}(\boldsymbol{y}_1)p_{\boldsymbol{X}_2|\boldsymbol{x}_1}(\boldsymbol{x}_2)p_{\boldsymbol{Y}_2|\boldsymbol{x}_2}(\boldsymbol{y}_2)\dots p_{\boldsymbol{X}_t|\boldsymbol{x}_{t-1}}(\boldsymbol{x}_t)p_{\boldsymbol{Y}_t|\boldsymbol{x}_t}(\boldsymbol{y}_t)$$

Markov Random Fields (MRFs)

- Also called undirected graphical models.
- For some domains, being force to choose a direction for the edges, as required by BNs, is unnatural.
- MRFs do not require to specify orientations.
- They are more natural for domains such as spatial or relational data. Widely used in image analysis and spatial statistics.
- Less interpretable as BNs.
- **More flexible**. Factors are **arbitrary positive functions** (potentials).

For a random vector $\boldsymbol{x}$, a MRF is defined as follows

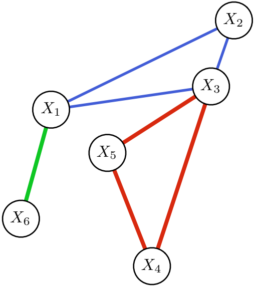
$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{Z} \prod_{c=1}^{C} \phi_c(\boldsymbol{x}_c),$$

where

1. $\boldsymbol{X}_c \; c = 1, \ldots, C$ are non-disjoint groups of variables.
2. All groups are maximal (there is no (c, d) such that $\boldsymbol{X}_c \subseteq \boldsymbol{X}_d$).
3. $\phi_c(\boldsymbol{x}_c) : \mathcal{X}^{|C|} \to \mathbb{R}^+$ (a.k.a. potential functions)
4. Z is a normalization constant:

$$Z = \int_{\mathcal{X}^n} \prod_{c=1}^{C} \phi_c(\boldsymbol{x}_c) \mathrm{d}\boldsymbol{x}, \qquad Z = \sum_{\mathcal{X}^n} \prod_{c=1}^{C} \phi_c(\boldsymbol{x}_c)$$

- Graphically, we represent a MRF by an undirected graph where all variables in $\boldsymbol{X}_c$ are connected to each other (clique).



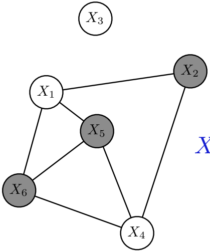
3 cliques

$$\mu(\mathbf{x}) = \frac{1}{Z}\phi_1(x_1, x_6)\phi_2(x_1, x_2, x_3)\phi_3(x_3, x_5, x_6)$$

Evaluating conditional independence in MRF

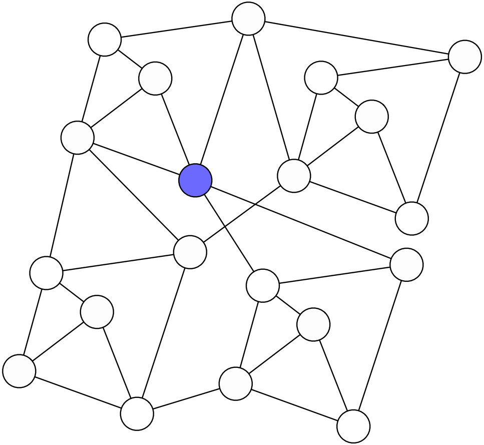
Property:
In a MRF, if a group of conditioned (observed) variables $\boldsymbol{X}_o$ is such that all paths from $\boldsymbol{X}_A$ to $\boldsymbol{X}_B$ go through $\boldsymbol{X}_o$, then

$$\boldsymbol{X}_A \perp \boldsymbol{X}_B | \boldsymbol{X}_o$$

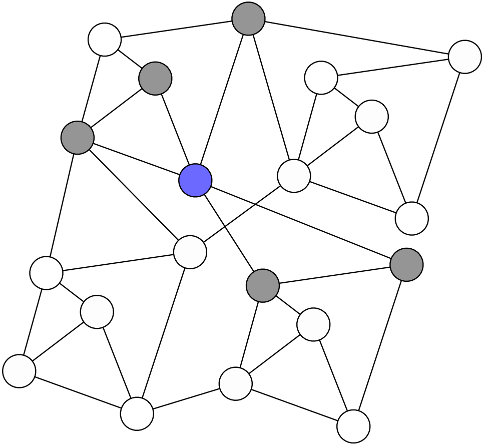


$X_4 \perp X_1, X_3 | X_2, X_5, X_6$

Markov Blanket in MRFs



Markov Blanket in MRFs



Factor Graphs

- As MRFs, they are convenient to model soft constraints between random variables that are not naturally given as a conditional probability.
- Factor graphs (FGs) make the factorization of a distribution explicit in the graph by introducing additional nodes for the factor themselves.
- They can preserve more details about the factorization.
- For the purpose of solving **inference** problems, it is often **convenient to convert both directed and undirected graphs into factor graphs.**

In many probabilistic models in communications, the joint distribution is expressed as large product of functions that overlap in their dependencies. In these scenarios, FGs are highly recommended since an equivalent MRF would produce an almost fully connected graph.

Factor Graphs (FGs)

Definition:
Given a probability distribution

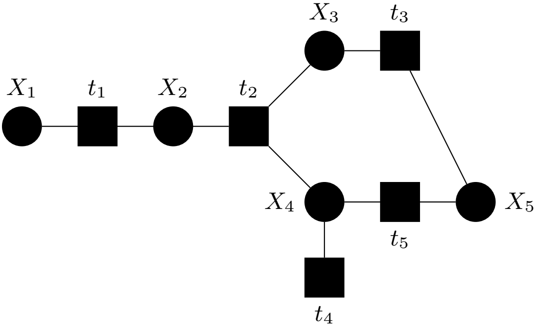
$$p_{\mathbf{X}}(\boldsymbol{x}) = \frac{1}{Z} \prod_{j \in \mathtt{J}} t_j(\boldsymbol{x}_j)$$

The FG has a node (represented by a square) for each factor t_j, $j \in \mathtt{J}$ and a variable node (represented by a circle) for each variable x_i, $i = 1, \ldots, n$. For each $x_i \in \boldsymbol{x}_j$, we place an undirected link between factor t_j and variable x_i.

FGs simply describe the factorization of functions.

They are bipartite graphs.

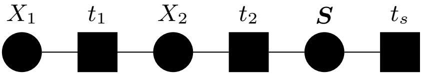
$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{Z} t_1(x_1, x_2) t_2(x_2, x_3, x_4) t_3(x_3, x_5) t_4(x_4) t_5(x_4, x_5), \qquad \boldsymbol{x} \in \mathcal{X}^5$$



By defining $\boldsymbol{S} = [\ X_3 \quad X_4 \quad X_5\ ]^T$, the same distribution factorizes as follows

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{Z} t_1(x_1, x_2) t_2(x_2, x_3, x_4) t_3(x_3, x_5) t_4(x_4) t_5(x_4, x_5)$$

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{Z} t_1(x_1, x_2) t_2(x_2, \boldsymbol{s}) t_s(\boldsymbol{s}),$$



Factors can be merged/split arbitrarily, allowing further algorithmic development for efficient inference.

Clustering of variable/factor nodes

It does not changes the global probability distribution but defines a new graph where, at the expense of more complex potential functions and losing representation of conditional independency constraints, approximate inference methods provide more accurate estimates.

Index

1. Course overview

2. Representing probability distributions via Factor graphs
   - Directed graphical models
   - Undirected graphical models
   - Factor graphs

3. Factor graphs in digital communications scenarios
   - Transmission of linear block codes over a memoryless channel
   - Symbol transmission and detection over ISI channels

Transmission of linear block codes over a memoryless channel

- A binary codeword $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is randomly selected from a linear block code $\mathcal{C}$ of length n.
- Assume a Hamming (7,4) linear block code with $n = 7$ and parity check matrix:

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

- Hence, every codeword $\boldsymbol{x} = (x_1, x_2, \ldots, x_7)$ have to fulfill the following parity-check equations:

$$x_1 \oplus x_3 \oplus x_5 \oplus x_7 = 0$$
$$x_2 \oplus x_3 \oplus x_6 \oplus x_7 = 0$$
$$x_4 \oplus x_5 \oplus x_6 \oplus x_7 = 0$$

- $\boldsymbol{x} \in \mathcal{C}$ is transmitted over a memoryless channel so that every output y_i depends just on x_i.
- For a given channel observation $\boldsymbol{y}$, the joint posterior probability of the transmitted codeword is expressed as

$$p_{\boldsymbol{X}|\boldsymbol{y}}(\boldsymbol{x}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{x}}(\boldsymbol{y})\, p_{\boldsymbol{X}}(\boldsymbol{x})}{p_{\boldsymbol{Y}}(\boldsymbol{y})} \propto p_{\boldsymbol{X}}(\boldsymbol{x}) \prod_{i=1}^{n} p_{Y_i|x_i}(y_i),$$

where

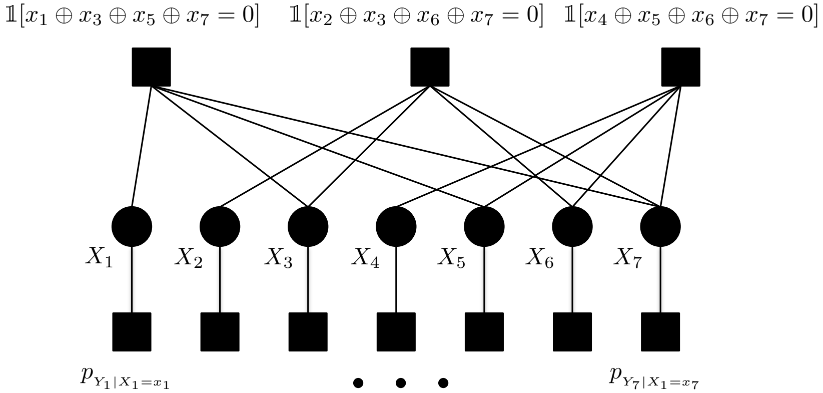
$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{\mathbb{1}[\boldsymbol{x}\boldsymbol{H}^T \overset{(\text{mod } 2)}{=} \boldsymbol{0}]}{|\mathcal{C}|}.$$

- This last term can be further factorized. Let $\boldsymbol{d}(\boldsymbol{x}) \doteq \boldsymbol{x}\boldsymbol{H}$ (mod 2) be the *syndrome* vector, of length $n-k$ (number of rows of $\boldsymbol{H}$)

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{\prod_{j=1}^{n-k} \mathbb{1}[d_j(\boldsymbol{x}) \overset{(\text{mod } 2)}{=} 0]}{|\mathcal{C}|}$$

Finally ...

$$p_{\boldsymbol{X}|\boldsymbol{y}}(\boldsymbol{x}) = \frac{1}{|\mathcal{C}|p_{\boldsymbol{Y}}(\boldsymbol{y})} \prod_{j=1}^{n-k} \mathbb{1}\left[d_j(\boldsymbol{x}) \overset{(\mathrm{mod}\ 2)}{=} 0\right] \prod_{i=1}^{n} p_{Y_i|x_i}(y_i).$$



$\mathbb{1}[x_1 \oplus x_3 \oplus x_5 \oplus x_7 = 0]$ $\mathbb{1}[x_2 \oplus x_3 \oplus x_6 \oplus x_7 = 0]$ $\mathbb{1}[x_4 \oplus x_5 \oplus x_6 \oplus x_7 = 0]$

Block-wise maximum a posteriori decoder (block-MAP decoding)

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}\in\mathcal{C}} p_{\boldsymbol{X}\,|\,\boldsymbol{y}}(\boldsymbol{x}) = \arg\max_{\boldsymbol{x}\in\mathcal{C}} \prod_{j=1}^{n-k} \mathbb{1}[d_j(\boldsymbol{x}) \overset{(\text{mod } 2)}{=} 0] \prod_{i=1}^{n} p_{Y_i\,|\,x_i}(y_i),$$

Symbol-wise maximum a posteriori decoder (symbol-MAP)

$$\begin{aligned}\hat{x}_i &= \arg\max_{x_i\in\{0,1\}} p_{X_i\,|\,\boldsymbol{y}}(x_i) = \arg\max_{x_i\in\{0,1\}} \sum_{\boldsymbol{x}_{\sim i}\in\{0,1\}^{n-1}} p_{\boldsymbol{X}\,|\,\boldsymbol{y}}(\boldsymbol{x}) \\ &= \arg\max_{x_i\in\{0,1\}} \sum_{\boldsymbol{x}_{\sim i}\in\{0,1\}^{n-1}} \prod_{j=1}^{n-k} \mathbb{1}[d_j(\boldsymbol{x}) \overset{(\text{mod } 2)}{=} 0] \prod_{i=1}^{n} p_{Y_i\,|\,x_i}(y_i)\end{aligned}$$

- **Both have $\mathcal{O}(2^n)$ complexity.**

Symbol transmission and detection over ISI channels

- Consider a sequence of $\mathtt{L}$ i.i.d. M-ary complex-valued symbols, s_k, obtained by the encoding of a sequence of information bits.
- $s_k \in \mathcal{A}$ and $|\mathcal{A}| = M$.
- ISI channel that also introduces additive white Gaussian noise (AWGN):

$$y_k = \sum_{i=0}^{\mathtt{h}} h_i s_{k-i} + w_k, \qquad w_k \sim \mathcal{CN}(\mathbf{0}, \sigma_w^2 \mathbf{I})$$

- The sequence of symbols $\boldsymbol{s}$ is preceded and terminated by a sequence of $\mathtt{h}$ deterministic known symbols: $s_k = s^*$ for $k \in \{-\mathtt{h}+1, \ldots, 0\}$ and for $k \in \{\mathtt{L}+1, \ldots, s_k + \mathtt{h}\}$.
- $\boldsymbol{y} = (y_1, y_2, \ldots, y_{\mathtt{L}+\mathtt{h}})$ is the observation vector.

$$p_{\boldsymbol{S}|\boldsymbol{y}}(\boldsymbol{s}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{s}}(\boldsymbol{y}) p_{\boldsymbol{S}}(\boldsymbol{s})}{p_{\boldsymbol{Y}}(\boldsymbol{y})}$$

$$p_{\boldsymbol{S}|\boldsymbol{y}}(\boldsymbol{s}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{s}}(\boldsymbol{y}) p_{\boldsymbol{S}}(\boldsymbol{s})}{p_{\boldsymbol{Y}}(\boldsymbol{y})}$$

$$p_{\boldsymbol{Y}|\boldsymbol{s}}(\boldsymbol{y}) = \prod_{k=1}^{\mathtt{L+h}} p_{Y_k|\boldsymbol{s}}(y_k)$$

$$p_{\boldsymbol{S}|\boldsymbol{y}}(\boldsymbol{s}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{s}}(\boldsymbol{y})p_{\boldsymbol{S}}(\boldsymbol{s})}{p_{\boldsymbol{Y}}(\boldsymbol{y})}$$

$$p_{\boldsymbol{Y}|\boldsymbol{s}}(\boldsymbol{y}) = \prod_{k=1}^{\mathtt{L}+\mathtt{h}} p_{Y_k|\boldsymbol{s}}(y_k) = \prod_{k=1}^{\mathtt{L}+\mathtt{h}} p_{Y_k|\{s_j\}_{k-\mathtt{h}}^{k}}(y_k)$$

$$p_{\boldsymbol{S}|\boldsymbol{y}}(\boldsymbol{s}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{s}}(\boldsymbol{y})p_{\boldsymbol{S}}(\boldsymbol{s})}{p_{\boldsymbol{Y}}(\boldsymbol{y})}$$

$$p_{\boldsymbol{Y}|\boldsymbol{s}}(\boldsymbol{y}) = \prod_{k=1}^{\mathtt{L}+\mathtt{h}} p_{Y_k|\boldsymbol{s}}(y_k) = \prod_{k=1}^{\mathtt{L}+\mathtt{h}} p_{Y_k|\{s_j\}_{k-\mathtt{h}}^{k}}(y_k)$$

$$p_{\boldsymbol{S}}(\boldsymbol{s}) = \prod_{u=-\mathtt{h}+1}^{\mathtt{L}+\mathtt{h}} p_{S_u}(s_u)$$

$$p_{\boldsymbol{S}|\boldsymbol{y}}(\boldsymbol{s}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{s}}(\boldsymbol{y})p_{\boldsymbol{S}}(\boldsymbol{s})}{p_{\boldsymbol{Y}}(\boldsymbol{y})}$$

$$p_{\boldsymbol{Y}|\boldsymbol{s}}(\boldsymbol{y}) = \prod_{k=1}^{\mathtt{L}+\mathtt{h}} p_{Y_k|\boldsymbol{s}}(y_k) = \prod_{k=1}^{\mathtt{L}+\mathtt{h}} p_{Y_k|\{s_j\}_{k-\mathtt{h}}^{k}}(y_k)$$

$$p_{\boldsymbol{S}}(\boldsymbol{s}) = \prod_{u=-\mathtt{h}+1}^{\mathtt{L}+\mathtt{h}} p_{S_u}(s_u)$$

$$p_{S_u}(s_u) = 1/M \text{ for } s_u \in \mathcal{A} \text{ and } u = 1, \ldots, s$$

$$p_{\boldsymbol{S}|\boldsymbol{y}}(\boldsymbol{s}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{s}}(\boldsymbol{y}) p_{\boldsymbol{S}}(\boldsymbol{s})}{p_{\boldsymbol{Y}}(\boldsymbol{y})}$$

$$p_{\boldsymbol{Y}|\boldsymbol{s}}(\boldsymbol{y}) = \prod_{k=1}^{\mathtt{L}+\mathtt{h}} p_{Y_k|\boldsymbol{s}}(y_k) = \prod_{k=1}^{\mathtt{L}+\mathtt{h}} p_{Y_k|\{s_j\}_{k-\mathtt{h}}^{k}}(y_k)$$

$$p_{\boldsymbol{S}}(\boldsymbol{s}) = \prod_{u=-\mathtt{h}+1}^{\mathtt{L}+\mathtt{h}} p_{S_u}(s_u)$$

$$p_{S_u}(s_u) = 1/M \text{ for } s_u \in \mathcal{A} \text{ and } u = 1, \ldots, \mathtt{L}$$

$$p_{S_u}(s_u) = \mathbb{1}[s_u = s^*] \text{ for } u \in \{-\mathtt{h}+1, \ldots, 0\} \text{ and } u \in \{\mathtt{L}+1, \ldots, \mathtt{h}+\mathtt{h}\}$$

$$p_{\boldsymbol{S}|\boldsymbol{y}}(\boldsymbol{s}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{s}}(\boldsymbol{y})p_{\boldsymbol{S}}(\boldsymbol{s})}{p_{\boldsymbol{Y}}(\boldsymbol{y})} \propto \prod_{k=1}^{\mathtt{L}+\mathtt{h}} p_{Y_k|\{s_j\}_{k-\mathtt{h}}^{k}}(y_k) \prod_{u=-\mathtt{h}+1}^{\mathtt{L}+\mathtt{h}} p_{S_u}(s_u)$$

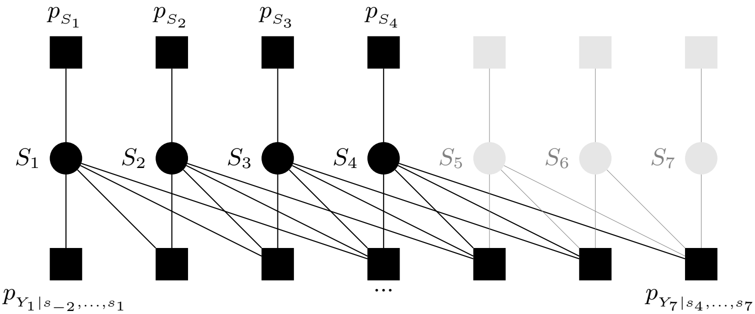


Figure: Factor graph representation of $p_{\boldsymbol{S}|\boldsymbol{y}}(\boldsymbol{s})$ for the case $\mathtt{h} = 3$ and $\mathtt{L} = 4$.

This FG has cycles. By clustering nodes we can obtain a cycle-free representation.

- *State* variable at time k, E_k. Given a realization of the $\mathtt{h}$ symbols transmitted immediately before time k, i.e., $\tilde{s}_k = \{s_{k-\mathtt{h}}, s_{k-\mathtt{h}+1}, \dots, s_{k-2}, s_{k-1}\}$, then

$$e_k = f(\tilde{s}_k) : \mathcal{A}^{\mathtt{h}} \rightarrow \{1, 2, \dots, M^{\mathtt{h}}\}$$

- E_k is a R.V. that takes $M^{\mathtt{h}}$ possible values.
- However, there is a deterministic (one-to-one) relationship between $\tilde{s}_k$ and e_k.

- The joint posterior probability of the symbols $\boldsymbol{S}$ and states $\boldsymbol{E}$ can be expressed as

$$p_{\boldsymbol{S},\boldsymbol{E}|\boldsymbol{y}}(\boldsymbol{s},\boldsymbol{e}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{e}}(\boldsymbol{y})\; p_{\boldsymbol{E}|\boldsymbol{s}}(\boldsymbol{e})\; p_{\boldsymbol{S}}(\boldsymbol{s})}{p_{\boldsymbol{Y}}(\boldsymbol{y})}$$

  where

- The joint posterior probability of the symbols $\boldsymbol{S}$ and states $\boldsymbol{E}$ can be expressed as

$$p_{\boldsymbol{S},\boldsymbol{E}|\boldsymbol{y}}(\boldsymbol{s},\boldsymbol{e}) = \frac{p_{\mathbf{Y}|\boldsymbol{e}}(\boldsymbol{y})\; p_{\boldsymbol{E}|\boldsymbol{s}}(\boldsymbol{e})\; p_{\boldsymbol{S}}(\boldsymbol{s})}{p_{\mathbf{Y}}(\boldsymbol{y})}$$

where

$$p_{\mathbf{Y}|\boldsymbol{e}}(\boldsymbol{y}) = \prod_{k=1}^{\mathtt{L+h}} p_{Y_k|e_k,e_{k+1}}(y_k)$$

- The joint posterior probability of the symbols $\boldsymbol{S}$ and states $\boldsymbol{E}$ can be expressed as

$$p_{\boldsymbol{S},\boldsymbol{E}|\boldsymbol{y}}(\boldsymbol{s},\boldsymbol{e}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{e}}(\boldsymbol{y})\; p_{\boldsymbol{E}|\boldsymbol{s}}(\boldsymbol{e})\; p_{\boldsymbol{S}}(\boldsymbol{s})}{p_{\boldsymbol{Y}}(\boldsymbol{y})}$$

where

$$p_{\boldsymbol{Y}|\boldsymbol{e}}(\boldsymbol{y}) = \prod_{k=1}^{\mathtt{L}+\mathtt{h}} p_{Y_k|e_k,e_{k+1}}(y_k)$$

$$p_{\boldsymbol{E}|\boldsymbol{s}}(\boldsymbol{e}) = \mathbb{1}[e_1 = e^*]\mathbb{1}[e_{\mathtt{L}+\mathtt{h}+1} = e^*] \prod_{k=1}^{\mathtt{L}+\mathtt{h}} \mathbb{1}[e_k \cup s_k \to e_{k+1}]$$

- The joint posterior probability of the symbols $\boldsymbol{S}$ and states $\boldsymbol{E}$ can be expressed as

$$p_{\boldsymbol{S},\boldsymbol{E}|\boldsymbol{y}}(\boldsymbol{s},\boldsymbol{e}) = \frac{p_{\boldsymbol{Y}|\boldsymbol{e}}(\boldsymbol{y})\; p_{\boldsymbol{E}|\boldsymbol{s}}(\boldsymbol{e})\; p_{\boldsymbol{S}}(\boldsymbol{s})}{p_{\boldsymbol{Y}}(\boldsymbol{y})}$$

where

$$p_{\boldsymbol{Y}|\boldsymbol{e}}(\boldsymbol{y}) = \prod_{k=1}^{\mathtt{L+h}} p_{Y_k|e_k,e_{k+1}}(y_k)$$

$$p_{\boldsymbol{E}|\boldsymbol{s}}(\boldsymbol{e}) = \mathbb{1}[e_1 = e^*]\mathbb{1}[e_{\mathtt{L+h}+1} = e^*] \prod_{k=1}^{\mathtt{L+h}} \mathbb{1}[e_k \cup s_k \rightarrow e_{k+1}]$$

$$p_{\boldsymbol{S}}(\boldsymbol{s}) = \prod_{k=1}^{\mathtt{L+h}} p_{S_k}(s_k)$$

If we finally define

$$t_k(e_k, e_{k+1}, s_k) = p_{Y_k | e_k, e_{k+1}}(y_k)\ \mathbb{1}[e_k \cup s_k \to e_{k+1}]$$

If we finally define

$$t_k(e_k, e_{k+1}, s_k) = p_{Y_k \mid e_k, e_{k+1}}(y_k)\ \mathbb{1}[e_k \cup s_k \to e_{k+1}]$$

Then...

$$p_{\boldsymbol{S},\boldsymbol{E}\mid\boldsymbol{y}}(\boldsymbol{s},\boldsymbol{e}) \propto \mathbb{1}[e_1 = e^*]\mathbb{1}[e_{\mathtt{L+h}+1} = e^*] \prod_{k=1}^{\mathtt{L+h}} p_{S_k}(s_k) t_k(e_k, e_{k+1}, s_k)$$

$$p_{\boldsymbol{S},\boldsymbol{E}|\boldsymbol{y}}(\boldsymbol{s},\boldsymbol{e}) \propto \mathbb{1}[e_1 = e^*]\mathbb{1}[e_{\mathtt{L}+\mathtt{h}+1} = e^*] \prod_{k=1}^{\mathtt{L}+\mathtt{h}} p_{S_k}(s_k) t_k(e_k, e_{k+1}, s_k)$$

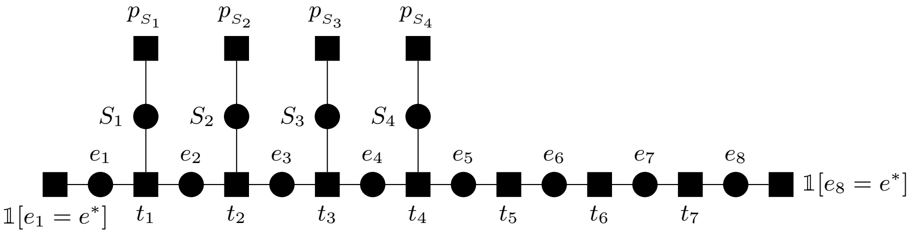


Figure: Factor graph associated to $p_{\boldsymbol{S},\boldsymbol{E}|\boldsymbol{y}}$ for the case $\mathtt{h} = 3$ and $\mathtt{L} = 4$.

Beyond visualization, the fact that the FG in the latter case is cycle-free has important practical implications to construct efficient inference methods.

Indeed, because of the FG structure the complexity of

$$\hat{s}_k = \arg\max_{s_k \in \mathcal{A}} p_{s_k|\boldsymbol{y}}(\boldsymbol{s}) = \arg\max_{s_k \in \mathcal{A}} \sum_{\boldsymbol{s}_{\sim k} \in \mathcal{A}^{\mathtt{L}-1}} \sum_{\boldsymbol{e}} p_{\boldsymbol{S},\boldsymbol{E}|\boldsymbol{y}}(\boldsymbol{s}, \boldsymbol{e})$$

$$\hat{\boldsymbol{s}} = \arg\max_{\boldsymbol{s} \in \mathcal{A}^{\mathtt{L}}} p_{\boldsymbol{S},\boldsymbol{E}|\boldsymbol{y}}(\boldsymbol{s}, \boldsymbol{e})$$

can be reduced to $\mathcal{O}(M^{\mathtt{L}})$ to $\mathcal{O}(\mathtt{L}M^{\mathtt{h}})$.