Message Passing for Decoding and Inference (§21/25/26)

① Decoding Problem:

$$S \longrightarrow x^N \longrightarrow y^N \overset{???}{\dashrightarrow} \hat{S}$$

Given channel Q, encoder, and output y^N, how to decode?

* Clever algebra, like for Reed-Solomon: Not always possible!

* Maximum likelihood codeword decoder: Assume $P(S)$ uniform.
  Given y^N, find S that maximizes $P(S|y^N)$.
  Eqv: Maximize $P(x^N|y^N)$ over $C = \{x^N \text{ codeword}\}$

We assume no two S have same codeword x^N !!!

$$P(x^N|y^N) \overset{\text{Bayes}}{=} \frac{P(y^N|x^N)\,P(x^N)}{P(y^N)} = \cancel{\frac{1}{P(y^N)\cdot \#C}}\, P(y^N|x^N)\cdot \mathbb{1}[x^N \in C]$$

Can ignore if y^N known

remember this notation? $= \begin{cases} 1 & \text{if } x^N \in C \\ 0 & \text{if } x^N \notin C \end{cases}$

Thus: Given y^N, find x^N that maximizes

$$\boxed{G(x^N) = Q(y_1|x_1)\cdots Q(y_N|x_N)\,\mathbb{1}[x^N \in C]}$$

$\delta_{ab} = \begin{cases} 1 & \text{if } a=b \\ 0 & \text{if } a\neq b \end{cases}$ notation ok?

e.g. repetition code R_3: $C = \{000, 111\}$

$$G(x_1,x_2,x_3) = Q(y_1|x_1)\,Q(y_2|x_2)\,Q(y_3|x_3)\,\delta_{x_1,x_2}\,\delta_{x_2,x_3}$$

* Bitwise decoder: Maximize $P(x_i|y^N)$ for each $i=1,\dots,N$. Equivalently:
  Given y^N, find x_i that maximizes

$$G_i(x_i) := \sum_{\{x_j\}_{j\neq i}} G(x^N)$$

How to avoid having to compute all numbers $G(x^N)$???

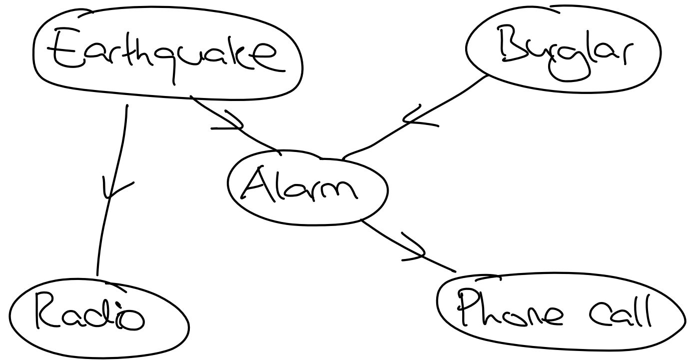
exponential in N

② Inference in Bayesian networks:

Assume we have a model

$$P(e,b,r,a,p)$$
$$= \underline{P(e)}\,\underline{P(b)}\,\underline{P(r|e)}\,\underline{P(a|eb)}\,\underline{P(p|a)}$$

all functions are known (= model)



Inference Problems:

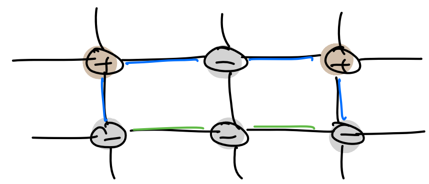
$$? = \Pr(B=1 | A=1) = \frac{\Pr(B=1, A=1)}{\sum_b \Pr(B=b, A=1)}$$

↳ enough to compute $\underbrace{\Pr(B=b, A=1)}_{\text{marginal of } B} = \sum_{e,r,p} \underbrace{P(e,b,r,1,p)}_{\text{product of local factors}}$

How to avoid having to first compute $P(e,b,r,1,p)$ for all e,b,r,p?

③ Statistical Physics: Ising model on lattice

* one particle per site with states $x_i \in \{\pm 1\}$
* total energy $E[\{x_i\}] = \sum_{i\sim j} J(1-\delta_{x_i,x_j})$



energy cost J if NOT same

ferromagnet if $J>0$

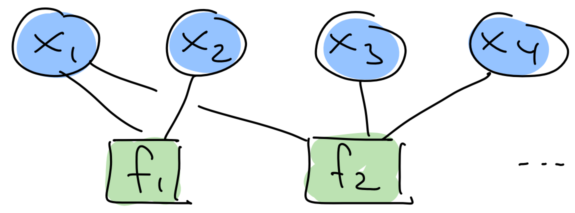
Partition function at temperature T:

$$Z = \sum_{\{x_i\}} e^{-E[\{x_i\}]/T} = \sum_{\{x_i\}} \underbrace{\prod_{i\sim j} e^{-J/T\cdot(1-\delta_{x_i,x_j})}}_{\text{product of local factors}}$$

General Setup:

$$G(x^N) = \prod_m \overset{\text{factor}}{f_m}\big(\{x_i\}_{\underbrace{i\in I(m)}_{\text{Subset of variables}}}\big)$$

where $x_i \in \mathcal{A}_i$



$$G(x^N) = f_1(x_1,x_2)\, f_2(x_1,x_3,x_4) \cdots$$

Factor graph:

* vertices: x_i for each variable, $\boxed{f_m}$ for each factor
* edge: x_i — $\boxed{f_m}$ if f_m depends on x_i

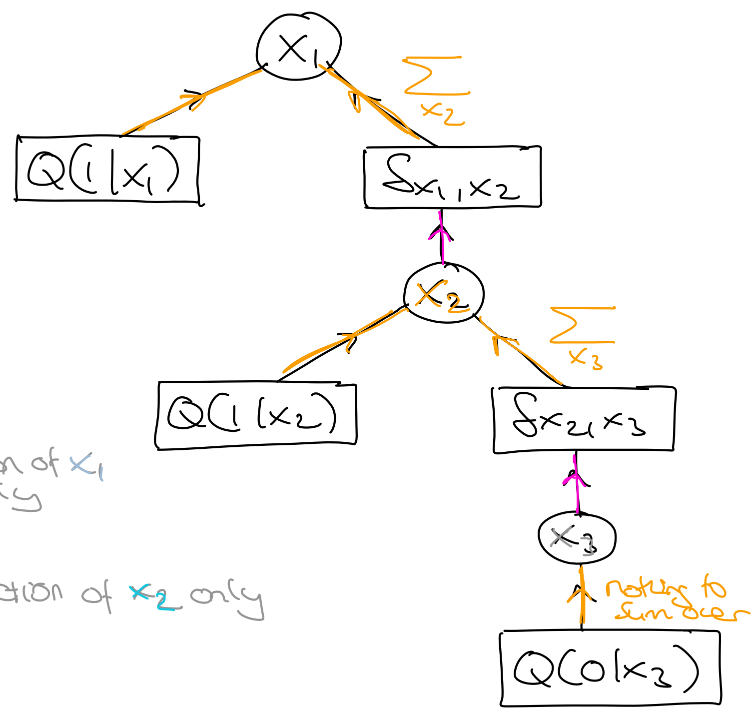
Problem: Compute marginals → ① Bitwise decoding, ② Bayesian inference, ...

$$\boxed{G_i(x_i) := \sum_{\{x_j\}_{j\neq i}} G(x^N) = \sum_{\substack{x_1,\dots,x_{i-1},\\ x_{i+1},\dots,x_N}} G(x_1,\dots,x_N)}$$

e.g. for repetition code: If we receive $y^3 = 110$:

$$G(x_1,x_2,x_3) = \delta_{x_1,x_2}\,\delta_{x_2,x_3}\, Q(1|x_1)\, Q(1|x_2)\, Q(0|x_3)$$

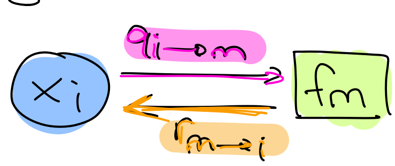
* factor graph is tree:
* if rooted at x_1 gives natural "algo" for computing marginal:



$$G_1(x_1) = Q(1|x_1)$$
$$\left.\sum_{x_2} \delta_{x_1,x_2}\, Q(1|x_2)\right\} \text{function of } x_1 \text{ only}$$
$$\left.\sum_{x_3} \delta_{x_2,x_3}\, Q(0|x_3)\right\} \text{function of } x_2 \text{ only}$$

⚡ If interested in G_2 or G_3, need to change root and run again

The following "message passing" algorithm computes all $G_i(x_i)$ at the same time:



NB: Messages are functions/tuples ! (one real number for each $x_i \in \mathcal{A}_i$)

Sum-product algorithm ("belief propagation"):

Input: Factor graph & factors $\{f_i\}$ & integer T

① For all edges $(x_i) - \boxed{f_m}$ and all $x_i \in \mathcal{A}_i$:

$$q_{i\to m}(x_i) \leftarrow 1$$

② For T steps:

For all edges $(x_i) - \boxed{f_m}$ and all $x_i \in \mathcal{A}_i$:

$$r_{m\to i}(x_i) \leftarrow \sum_{\{x_j\}_{j\in I(m), j\neq i}} f_m(\{x_j\}_{j\in I(m)}) \cdot \prod_{j\in I(m), j\neq i} q_{j\to m}(x_j)$$

$I(m)$ = variables that appear in f_m

For all edges $(x_i) - \boxed{f_m}$ and all $x_i \in \mathcal{A}_i$:

$$q_{i\to m}(x_i) \leftarrow \prod_{n\in M(i), n\neq m} r_{n\to i}(x_i)$$

$M(i)$ = factors that depend on x_i

③ For all vertices (x_i) and all $x_i \in \mathcal{A}_i$:

$$G_i(x_i) \leftarrow \prod_{m\in M(j)} r_{m\to j}(x_i)$$

* Sum-product algo works provably for trees
* computes all G_i at the same time if $T \geq$ diameter of graph.
* in practice also used for general graphs

  but only a heuristic: problem is NP-hard

Variations:

* Partition function: $Z = \sum_{x^N} G(x^N) = ?$ → ③ physics $\left[Z = \sum_{x_i} G_i(x_i)\right]$
* Maximum: $\max_{x^N} G(x^N) = ?$ → ① ML decoding

[Replace $\sum$ by max ⇝ "max-product" algo $\overset{-\log}{⇝}$ "min-sum" algo]

Outlook

What did we NOT cover?

* Channels with memory
* Multi-user information theory
* More connections to inference, machine learning, etc.

Bachelor project?

new specialization DMQI

* Quantum information theory → MasterMath course with Maris Ozols
  & Quantum Computing → Courses by Maris (BSc) + Ronald de Wolf (MSc)
* Connections to Cryptography → Course by Chris Schaffner

How to prepare for the exam?

* Learning objectives @ homepage
* Lecture notes, homework, practice problems, last year's exam
* Don't forget to prepare your cheat sheet
* Structure: mix of problems of type ① + ② from HW

THANKS + SEE YOU AGAIN SOON!