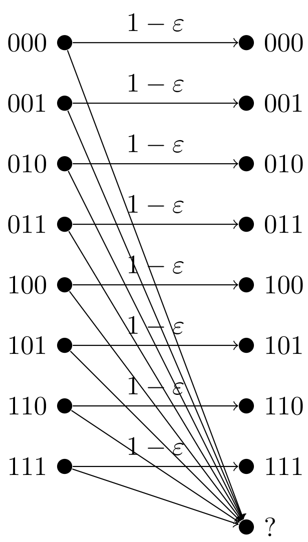
Chapter 8

Low Density Codes and Message Passing Decoding

8.1 Coding for Erasure Channels

Erasure channel

- Model packet loss in networks (e.g. Internet, wireless networks)
- Capacity: $1 - \varepsilon$ symbol per use
- Solutions:
  - Retransmission
  - Forward error correction



Retransmission

- Example: TCP, 802.11 MAC, cellular networks
- Achieve capacity
- Require feedbacks
- Not good for many scenarios
  1. wireless transmissions
  2. deep-space (satellite), underwater communications
  3. multicast transmissions

Forward error correction

- Capacity achieving without feedbacks
- Reed-Solomon code $(n, k, n-k+1)$
  - Encoding and decoding complexity: $O((n-k)\log n)$ per symbol.
- Can we have better solutions?
  - $O(1)$ complexity (per symbol).
  - Adaptive for different erasure rates/patterns.

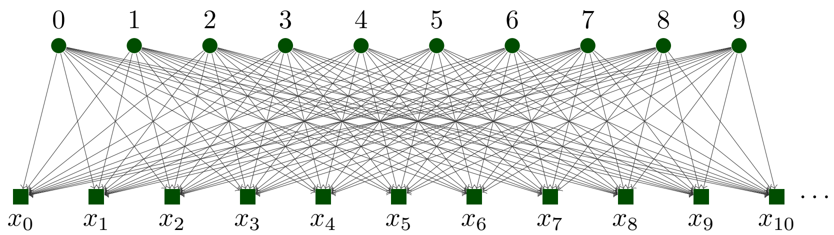
What are fountain codes?

- Transmit a file of k packets: $\mathbf{B} = [b_1, b_2, \ldots, b_k]$ where $b_i \in \mathbb{F}_q^T$
- Encoder generates potentially an infinite number of coded packets
- The file can be recovered from any set of n coded packets, where n is slightly larger than k.
- Also known as *rateless codes*

8.2 Rateless Random Linear Codes

Rateless random linear codes

- Encoding: $x_j = \sum_{i=1}^{k} \alpha_{j,i} b_i$.
- Decode from any k coded packets with linearly independent coding vectors.
- Work for any erasure patterns – universal.



Error Probability as a block code of length n

- When n packets are transmitted, the number of received packets N is $B(n, 1-\epsilon)$.
- The received packets $\mathbf{Y} = [y_1, y_2, \ldots y_N]$ is given by $\mathbf{Y} = \mathbf{BA}$.
- The decoding is correct iff $\mathrm{rank}(\mathbf{A}) = k$.
- Error probability

$$
\begin{aligned}
P_e &= 1 - \Pr\{\mathrm{rank}(\mathbf{A}) = k\} \\
&= 1 - \sum_{j=k}^{n} \binom{n}{j} \epsilon^{n-j}(1-\epsilon)^j \Pr\{\mathrm{rank}(\mathbf{A}) = k | N = j\} \\
&= 1 - \sum_{j=k}^{n} \binom{n}{j} \epsilon^{n-j}(1-\epsilon)^j \zeta_k^j,
\end{aligned}
$$

  where ζ_k^j is the probability that a $k \times j$ totally random matrix has rank k.

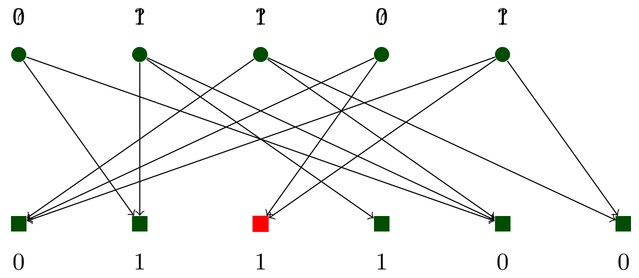
  Note that $\zeta_k^j = (1-q^{-j})(1-q^{-j+1})\cdots(1-q^{-j+k-1})$ when $0 < k \le j$.

Coding overhead (as a rateless code)

- Coding overhead is the number of packets received minus k when decoding.
- Expected coding overhead:

$$
\mathrm{CO} = \sum_{i=1}^{\infty} i \Pr\{E_i \cap E'_{i-1}\} = \sum_{i=1}^{\infty} i \zeta_{k-1}^{k,k+i-1}(1-q^{-1}),
$$

  where
  - E_i: the first $k+i$ received packets have rank k.



– $\zeta_r^{m,n}$: the probability that an $m \times n$ tatally random matrix has rank r.

Note that

$$\zeta_r^{m,n} = \frac{\zeta_r^m \zeta_r^n}{\zeta_r^r q^{(m-r)(n-r)}}.$$

8.3 LT codes [Luby 98]

LT codes

- Sparse encoding
  1. pick a degree d by sampling a degree distribution $\Psi = (\Psi_1, \Psi_2, \ldots, \Psi_K)$.
  2. uniformly at random pick d input packets.
  3. generate a coded packet by linearly combinate of the d input packets.
  4. repeat 1 - 3.
- Belief propogation decoding
  1. find a coded packet with degree one, which recovers the corresponding input packet.
  2. substitute the recovered input packet into the other coded packets that it involves.
  3. repeat 1 - 2 until there is no coded packets with degree one.
- Encoding/decoding complexity: $O(\log K)$ per packet, determined by the average degree $\mathbb{E}[\Psi]$.

Tanner graph of LT codes

A bound on degree distribution

Theorem 8.1 For an LT code with k input packets and n coded packets, if there exists a decoding algorithm with $P_e \leq k^{-c}$, then $\mathbb{E}[\Psi] \geq c' \frac{k}{n} \ln k$.

- So when n is close to k, $\mathbb{E}[\Psi] \geq c' \ln k$.
- Luby showed that there exists a degree distribution such that
  1. $\mathbb{E}[\Psi] = O(\log(k))$,
  2. the BP decoding succeeds with vanishing error probability for n coded packets with $\frac{n-k}{k} \to 0$.

Degree distribution of LT codes

- Ideal Soliton distribution:

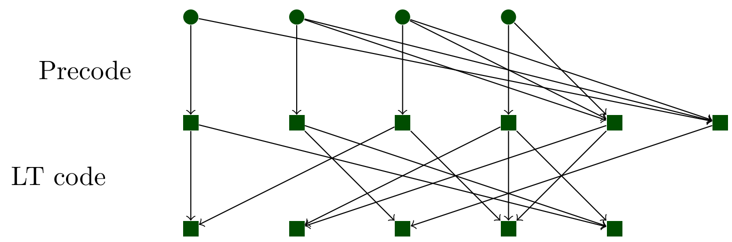
$$\rho_1 = 1/k$$
$$\rho_i = \frac{1}{i(i-1)}, \forall i = 2, 3, \ldots, k.$$

- Robust Soliton distribution:

8.4 Raptor codes

Raptor codes [Shokrollahi 2000]

- The trick: precode
- Encoding/decoding complexity $O(1)$ per packet



Degree distribution of Raptor codes

- BP decoding recovers at least $1 - \eta$ fraction of the (intermediate) input packets.
- The maximum degree $D \leq 1/\eta$. So $\mathbb{E}[\Psi] = O(1)$.
- The gap $\frac{n-k}{k}$ can be any positve value but is not vanishing for a fixed degree distribution when $k \to \infty$.

Performance analysis

- Asymptotic analysis: performance when $k \to \infty$.
  - Tree analysis [LMS98]
  - Differential equation approach (see [Wor99])
- Finite-length analysis: performance when k is relative small.
  - Iterative formula for the distribution of the decoder status [KLS04]

[LMS98] M. Luby, M. Mitzenmacher, and M. A. Shokrollahi, "Analysis of Random Processes via And-Or Tree Evaluation", in Proc. *SODA*, 1998, pp. 364–373.
[Wor99] N. C. Wormald, "The differential equation method for random graph processes and greedy algorithms," Karonsky and Proemel, eds., Lectures on Approximation and Randomized Algorithms PWN, Warsaw, pp. 73–155, 1999.
[KLS04] R. Karp, M. Luby, and A. Shokrollahi, "Finite length analysis of LT- codes," in Proc. IEEE ISIT'04, 2004.

8.5 Tree based analysis

Random Bipartite Graph

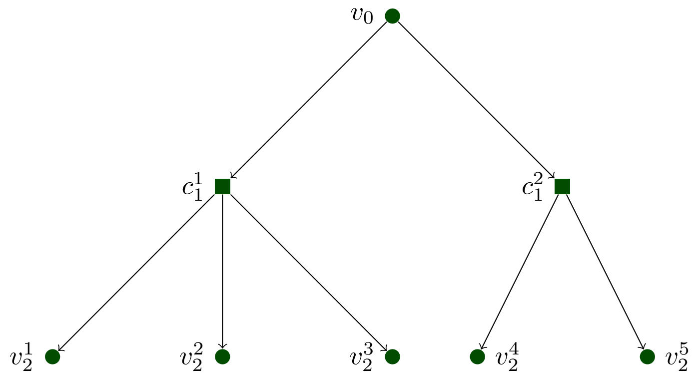
- Consider a sequence of random bipartite graph with n check nodes and $K = nR$ variable nodes.
  - A check node has degree d with probability Ψ_d.
  - A check node with degree d connects to d variable nodes chosen uniformly at random.
- BP decoding of the bipartite graph has multiple iterations. In each iteration,
  - All variable nodes adjacent to a check node of degree one are recovered.
  - All edges adjacent to recovered variable nodes are removed.

Outline of Tree based Analysis

- Analyze the decoding process on a random And-OR tree.
- Show that the neighbor-hood of a variable node in the random bipartite graph is similar to a random tree in probability.
- Determine a proper degree distribution.

Random Tree

- Random tree T_l has $l + 1$ levels.
- The root is a variable node at level 0 and the leaf nodes are at level l.
- Each variable node has $i - 1$ children (check nodes) with probability α_i.
- Each check node has $i - 1$ children (variable nodes) with probability β_i.



And-OR Tree Analysis

- Let x_l be the probability that the root of T_{2l-1} can be recovered by the BP decoding of LT codes.
- Let y_l be the probability that a children of the root of T_{2l-1} has all its children recovered.
- We have

$$x_l = 1 - \sum_{i=1}^{\infty} \alpha_i (1-y_l)^{i-1} \qquad\qquad y_l = \sum_{i=1}^{\infty} \beta_i x_{l-1}^{i-1}$$

- We have $x_l = f(x_{l-1})$, where

$$f(x) = 1 - \sum_{i=1}^{\infty} \alpha_i \left(1 - \sum_{i=1}^{\infty} \beta_i x^{i-1}\right)^{i-1}$$

- Note that $f(0) \geq 0$, $f(1) = 1$ and $f(x)$ is non-decreasing.
- x_l tends to the first $x \in [0,1)$ such that $f(x) = x$.

Variable node degree distribution

- The variable node degree distribution converges to a Poisson distribution

$$\Lambda_k = \frac{\lambda^k e^{-\lambda}}{k!},$$

where $\lambda = \frac{\mathbb{E}\Psi}{R}$.

Computation graph

- Computation graph G_l: Choose a variable node v and G_l is the subgraph induced by v and all neighbors of v within distance l.
- Random tree T_l with

$$\alpha_k = \frac{\lambda^{k-1} e^{-\lambda}}{(k-1)!}, \quad k = 1, \ldots,$$
$$\beta_k = \frac{k\Psi_k}{\sum_i i\Psi_i}, \quad k = 1, \ldots, D,$$

where $k_{\max}$ is a fixed integer and α_k is the truncated Poisson distribution.
- Convergence to Tree: for a fixed tree T of level at most $l+1$ with the maximum variable node degree $k_{\max}$,

$$\Pr\{G_l = \mathrm{T}\} \to \Pr\{T_l = \mathrm{T}\}.$$

We prove the above convergence by induction. Both T_l and G_l has only one variable node, the convergence holds for $l = 0$. Assume that the convergence holds for certain $l \geq 0$. For a tree T of at most $l+2$ levels, let T$'$ be its subgraph of the first $l+1$ levels. We have $\Pr\{G_{l+1} = \mathrm{T}\} =$

$\Pr\{G_l = \mathrm{T}'\}\Pr\{G_{l+1} = \mathrm{T}|G_l = \mathrm{T}'\}$, where the first term converges by the induction hypothesis, and we only need to study the second term.

Suppose that T' has N leaf nodes. Let $G_l^0 = G_l$. For $i = 1, \ldots, N$, let G_l^i be the subgraph of G_{l+1} formed by the first $l+1$ levels and the children of the first i nodes at level l. Under the condition that $G_l = \mathrm{T}'$, $G_{l+1} = G_l^N$. We can write

$$\Pr\{G_{l+1} = \mathrm{T}|G_l = \mathrm{T}'\} = \prod_{i=1}^{N} \Pr\{G_l^i = \mathrm{T}^i|G_l^{i-1} = \mathrm{T}^{i-1}\},$$

where $\mathrm{T}^0 = \mathrm{T}$ and for $i = 1, \ldots, N$, T^i is the subgraph of T that includes the first $l+1$ levels and the children of the first i nodes at level l.

To characterize $\Pr\{G_l^i = \mathrm{T}^i|G_l^{i-1} = \mathrm{T}^{i-1}\}$, suppose that the ith node u at level l of T has $s-1$ children. Consider two cases:

- When l is even, u is a variable node. Similar to the study of the variable node degree distribution, we have $\Pr\{G_l^i = \mathrm{T}^i|G_l^{i-1} = \mathrm{T}^{i-1}\} \to \alpha_s$.
- When l is odd, u is a check node. Let v be the parent of u in T. We have

$$\begin{aligned}\Pr\{G_l^i = \mathrm{T}^i|G_l^{i-1} = \mathrm{T}^{i-1}\} &= \frac{\Pr\{G_l^i = \mathrm{T}^i\}}{\Pr\{G_l^{i-1} = \mathrm{T}^{i-1}\}} \\ &= \frac{\Pr\{u \text{ has degree } s \text{ and connects to } v|E\}}{\Pr\{u \text{ connects to } v|E\}} \\ &\to \frac{s\Psi_s}{\mathbb{E}\Psi} = \beta_s\end{aligned}$$

  where E is the event that $G_l^{i-1} \setminus \{u\}$ is given.

Note that the above analysis depends on that the maximum variable node degree is bounded by a constant $k_{\max}$. The probability that T_{2l-1} has the maximum variable node degree $k_{\max}$ is at least $(\sum_{i=1}^{k_{\max}} \Lambda_i)^{k_{\max}^l D^l}$, which tends to 1 as $k_{\max}$ tends to infinity. For any $\epsilon > 0$, we can find a sufficiently large $k_{\max}$ to show that a variable node can be recovered by l iterations of the BP decoding with probability at least $x_l - \epsilon/2$.

To complete the proof, we can use a martingale argument to show that the total number of variable node recovered by the BP decoding is at least $(x_l - \epsilon)K$ with high probability.

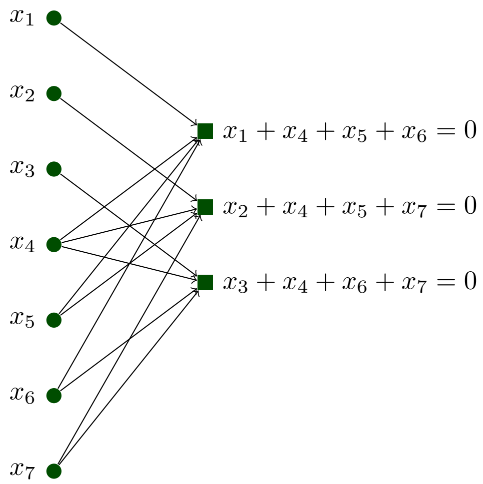
Sufficient condition

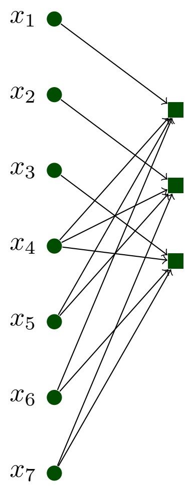
- Substituting α_i and β_i, we get
$$f(x) = 1 - \exp\left(-\frac{\Psi'(x)}{R}\right).$$
- To guarantee the success of decoding with high probability, we can require
$$x < 1 - \exp\left(-\frac{\Psi'(x)}{R}\right), \quad \text{for } x \in [0, 1-\eta],$$
  which implies
$$\Psi'(x) + R\ln(1-x) > 0, \quad \text{for } x \in [0, 1-\eta].$$
- Let $D = \lfloor 1/\eta \rfloor - 1$. For any $R < 1$, let
$$\Psi(x) = R\left((1/R-1)x + \sum_{i=2}^{D-1} \frac{x^i}{(i-1)i} + \frac{x^D}{D-1}\right).$$

8.6 LDPC Codes

A brief history

- Invented by Robert G. Gallager in his Ph.D. thesis (1960)
- Reinvented several times for the next 30 years
- Factor graph, message passing algorithm
- Very hot research topic since around 2000





LDPC code from sparse bipartite graph

LDPC code from sparse parity check matrix

- A bipartite graph with n variable (left) nodes and r (right) check nodes.
- Let H be a binary adjacency matrix of the graph.
- The LDPC code is the set of vectors $\mathbf{c} = (c_1, \ldots, c_n)$ such that $H\mathbf{c}^\top = 0$.
- Design rate of the LDPC code is $1 - r/n$.
- How to define sparsity?

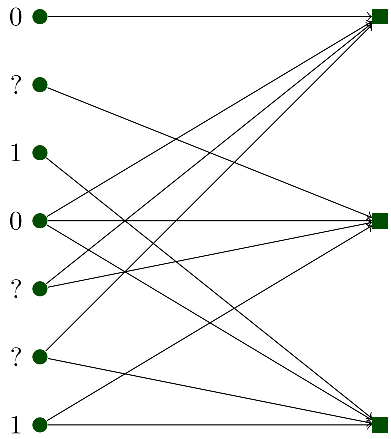
LDPC Ensemble

- (d_v, d_c)-regular LDPC code: every variable node has degree d_v and every check node has degree d_c.
- Let Λ_i (P_i) be the number of variable (check) nodes of degree i.
- LDPC ensemble: a set of sparse graphs with constraint (Λ, P) associated with a distribution.
- Let λ_i (ρ_i) be the fraction of edges that connect to variable (check) nodes of degree i.
- Let $\rho(x) = \sum_i \rho_i x^{i-1}$.

As the numbers of edges counted from the variable nodes and the check nodes should be the same, we have

$$\sum_i i\Lambda_i = \sum_i iP_i.$$

The block length of the ensemble is $n = \sum_i \Lambda_i$, and the design rate of the ensemble is $1 - \frac{\sum_i P_i}{n}$.



Moreover,

$$\lambda_i = \frac{i\Lambda_i}{\sum_i i\Lambda_i}$$

and

$$\rho_i = \frac{iP_i}{\sum_i iP_i}.$$

General message passing decoding rule

- In each round of the algorithm,
    1. messages are passed from variable nodes to check nodes, and
    2. from check nodes back to variable nodes.
- The messages from variable nodes to check nodes are computed based on
    1. the observed value of the variable node, and
    2. some of the messages passed from the neighboring check nodes.
- The message sent from a variable node v to a check node c does not take into account the message previously send from c to v.

8.7 Binary Erasure Channel

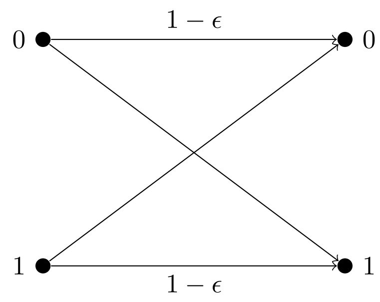
Message passing decoding

1. In the first round, a variable node just sends the value received from the channel.
2. For other other rounds, a variable node sends "erasure" to check node c if its received message is erasure and all the incoming messages other than from c are erasures. Otherwise, it sends its received message or the non-erasure incoming message.
3. A check node sends "erasure" to node v if any of the incoming message from nodes other than v is an erasure. Otherwise, it sends the mod-2 sum of the incoming messages other than from v.

Compute Graph and Tree

- Computation graph G_l:
    - uniformly at random pick an edge e and denote the variable node v of this edge as the root
    - G_l is the subgraph of the Tanner graph deleting e that contains the nodes with distance at most l to v
    - G_0 includes v only.
- Random tree T_l:
    - T_l has $l+1$ levels. The root, at level 0, is a variable node.
    - A variable node has $i-1$ children (check nodes) with probability λ_i.
    - A check node has $i-1$ children (variable nodes) with probability ρ_i.
- The decoding performance of G_l converges to that of T_l when $n \to \infty$.

l	r	rate	$\epsilon^{\text{Shannon}}$	ϵ^{BP}
3	6	1/2	0.5	0.4294
4	8	1/2	0.5	0.3834
3	5	2/5	0.6	0.5176
4	6	1/3	0.667	0.5061
3	4	1/4	0.75	0.6474



Density evolution

- Suppose the variable nodes are transmitted through BEC(ϵ).
- Let x_l be the probability that the root of $\mathcal{T}_{2l}$ is "erasure" after BP decoding.
- Let $f_\epsilon(x) = \epsilon\lambda(1-\rho(1-x))$.
- $x_0 = \epsilon$ and for $l > 1$,
$$x_l = f_\epsilon(x_{l-1}) = \epsilon\lambda(1-\rho(1-x_{l-1})).$$
- To have the erasure probability converging to zero, we require
$$x > \epsilon\lambda(1-\rho(1-x)), \quad x \in (0,1).$$
or
$$\lambda^{-1}(x/\epsilon) > 1-\rho(1-x), \quad x \in (0,1).$$

Threshold

Define
$$\epsilon^{\text{BP}}(\lambda,\rho) = \sup\{\epsilon : x_l(\epsilon) \stackrel{l\to\infty}{\longrightarrow} 0\}.$$

Fixed point characterization of threshold

Theorem 8.2 1. $\epsilon^{\text{BP}}(\lambda,\rho) = \sup\{\epsilon : x = f_\epsilon(x) \text{ has no solution in } (0,1]\}$.
2. $\epsilon^{\text{BP}}(\lambda,\rho) = \inf\{\epsilon : x = f_\epsilon(x) \text{ has a solution in } (0,1]\}$.

8.8 Binary Memoryless Symmeric Channel

Binary Memoryless Symmetric Channel

Gallager algorithm A

- In round 0, the variable nodes send their received values to all their neighboring check nodes.
- In the following rounds, a variable node sends to the neighboring check node c:
  - send value b if all the incoming messages from check nodes other than c are the same value b,
  - its received value otherwise.
- A check node sends to the neighboring variable node v the addition of the incoming messages from incident variable nodes other than v.

Asymptotic analysis

- All-zero codeword is transmitted
- Converge to tree.

Asymptotic analysis

- Let x_l be the error probability of the root of T_{2l}.
- For $l > 1$

$$\begin{aligned} x_l &= \epsilon\left(1-\lambda\left(1-z_{l-1}\right)\right)+(1-\epsilon)\lambda\left(z_{l-1}\right) \\ z_l &= \frac{1-\rho(1-2x_l)}{2}, \end{aligned}$$

where $x_0 = \epsilon$.

[LMSS01] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs," IEEE Trans. Inform. Theory, vol. 47, pp. 585–598, 2001.

Example: $(3,6)$**-LDPC code**

- Expect that x_l is monotonically decreasing
- A sufficient condition:

$$\epsilon\left(1-\lambda\left(\frac{1+\rho(1-2x)}{2}\right)\right)+(1-\epsilon)\lambda\left(\frac{1-\rho(1-2x)}{2}\right) > x$$

for $x \in (0, \epsilon]$.
- Rate of the code is $1/2$
- $\lambda(x) = x^2$ and $\rho(x) = x^5$
- Maximum ϵ is around 0.039

8.9 Bit-wise MAP Decoding

Bit-wise MAP Decoding

- A binary code word $(x_1, \ldots, x_n)$ is transmitted through a memoryless channel $p(y|x)$.
- The rule for bit-wise Maximum a posteriori decoder:

$$\begin{aligned} \hat{x}_i^{\mathrm{MAP}} &= \operatorname*{argmax}_{x_i} p_{X_i|Y^n}(x_i|y^n) \\ &= \operatorname*{argmax}_{x_i} \sum_{\sim x_i} p_{X^n|Y^n}(x^n|y^n) \\ &= \operatorname*{argmax}_{x_i} \sum_{\sim x_i} p_{Y^n|X^n}(y^n|x^n)p(x^n) \\ &= \operatorname*{argmax}_{x_i} \sum_{\sim x_i} \prod_j p(y_j|x_j)\mathbf{1}_{x^n \in C} \end{aligned}$$

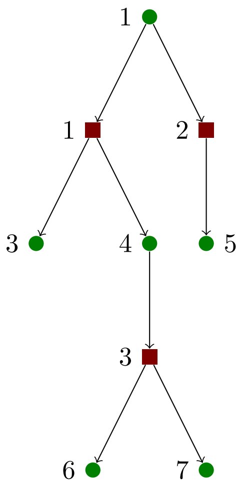
Note that for LDPC codes, the indicator function $\mathbf{1}_{x^n \in C}$ can be written as

$$\mathbf{1}_{x^n \in C} = \prod_{i=1}^{r} \mathbf{1}_{\langle h_i, x^n \rangle = 0},$$

where h_i is the ith row of H.

Let $f(x_1, \ldots, x_n) = \prod_j p(y_j|x_j)\mathbf{1}_{x^n \in C}$, which is the production of $n+r$ functions of $x_1, \ldots, x_n$. The bit-wise MAP decoding is just the marginalization of this function for each variables x_i. The message passing algorithm enables simple and parallel computation of marginalizations, when the factor graph of f is a tree topology.

- Fix a tree T.



- Each variable node i at round t has a message $(\mu_i^t(0), \mu_i^t(1))$ such that $\frac{\mu_i^t(0)}{\mu_i^t(1)} = \frac{p_{X_i|\tilde{Y}_i^t}(0|y)}{p_{X_i|\tilde{Y}_i^t}(1|y)}$, where $\tilde{Y}_i^t$ is the vector of received symbols corresponding to the variable nodes in the subtree rooted in variable node i with depth $2t+1$.
- $\mu_i^0(a) = p(y_i|a)$ for $i = 1, 3, 4, 5, 6, 7$.
- Each check node i at round t has a message $(\tilde{\mu}_i^t(0), \tilde{\mu}_i^t(1))$.

Since

$$\begin{aligned}
\mu_4^1(a) &\propto p_{X_4|\tilde{Y}_4^1}(a|y_4y_6y_7) \\
&= \sum_{x_6,x_7} p_{\tilde{X}_4^1|\tilde{Y}_4^1}(ax_6x_7|y_4y_6y_7) \\
&\propto \sum_{x_6,x_7} p_{\tilde{Y}_4^1|\tilde{X}_4^1}(y_4y_6y_7|ax_6x_7)p_{\tilde{X}_4^1}(ax_6x_7) \\
&\propto \mu_4^0(a)\sum_{x_6,x_7} \mu_6^0(x_6)\mu_7^0(x_7)1_{x_6+x_7=a}
\end{aligned}$$

We can write

$$\begin{aligned}
\tilde{\mu}_3^0(a) &= \sum_{x_6,x_7:x_6+x_7=a} \mu_6^0(x_6)\mu_7^0(x_7) \\
\mu_4^1(a) &= \mu_4^0(a)\tilde{\mu}_3^0(a)
\end{aligned}$$

Further

- $\tilde{\mu}_1^1(a) = \sum_{x_3,x_4:x_3+x_4=a} \mu_3^1(x_3)\mu_4^1(x_4)$
- $\tilde{\mu}_2^1(a) = \sum_{x:x=a} \mu_5^1(x)$
- $\mu_1^2(a) = \mu_1^0(a)\tilde{\mu}_2^1(a)\tilde{\mu}_1^1(a)$.

We can check that $\mu_1^2(a) \propto p_{X_1|\tilde{Y}_1^2}(a|y_3^7)$.

Bit-wise MAP message passing rule

- For each variable v with a set of children (check nodes) C, $\mu_v^t(a) = \mu_v^0(a)\prod_{c\in C}\tilde{\mu}_c^{t-1}(a)$.
- For each check node c with a set of children (variable nodes) V, $\tilde{\mu}_c^t(a) = \sum_{x_v:v\in V:\sum_{v\in V}x_v=a}\prod_{v\in V}\mu_v^t(x_v)$.
- Since we only care about the ratio of $\mu_v(0)$ and $\mu_v(1)$, the message passing rule can be simplified:
  - Let $L_v^t = \ln\frac{\mu_v^t(0)}{\mu_v^t(1)}$ and $\tilde{L}_c^t = \ln\frac{\tilde{\mu}_c^t(0)}{\tilde{\mu}_c^t(1)}$. Define $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$. Then,
  - $\tilde{L}_c^t = 2\tanh^{-1}(\prod_{v\in V}\tanh(L_v^t/2))$
  - $L_v^t = L_v^0 + \sum_{c\in C}\tilde{L}_c^{t-1}$.

References

[Sho03] A. Shokrolianhi, “LDPC Codes: An Introduction”.

[RU09] T. Richardson and R. Urbanke, *Modern Coding Theory*, Cambridge, 2009.