Density Evolution and LDPC Convolutional Codes

Lentmaier *et al.* (2009) – Approaching Capacity with Asymptotically Regular LDPC Codes

Urbanke (2010) – Spatially Coupled Codes – A New Paradigm for Code Design (Talk at KTH)

Christian Häger

Department of Signals and Systems
Communication Systems Group
Chalmers University of Technology
Gothenburg, Sweden

ECC Seminar – May 29, 2012

CHALMERS

Outline

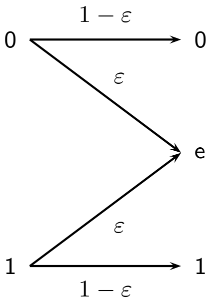
Recap: Density Evolution

Recap: Density Evolution

- Density evolution provides a threshold that divides the channel parameter range into an admissable and a non-admissable region.

Recap: Density Evolution

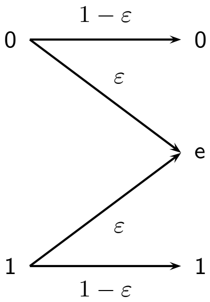
- Density evolution provides a threshold that divides the channel parameter range into an admissable and a non-admissable region.
- Consider the binary erasure channel:



$$C = 1 - \varepsilon$$

Recap: Density Evolution

- Density evolution provides a threshold that divides the channel parameter range into an admissable and a non-admissable region.
- Consider the binary erasure channel:



$C = 1 - \varepsilon$

- Here, density evolution is particularly simple: Track the average probability $p^{(l)}$ of an erasure after l iterations.
  - $(l = 0)$: $0\ 1\ 0\ e\ 1\ 0\ 1\ e\ e\ 1\ 0\ \ldots$ $\quad p^{(0)} = \varepsilon$
  - $(l = 1)$: $0\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ e\ 1\ 0\ \ldots$ $\quad p^{(1)} = ?$
  - $\ldots$

Recall the Sum-Product Algorithm

Recall the Sum-Product Algorithm

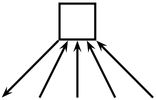
- Channel:

$$L_j = \begin{cases} +\infty, & y_j = 0 \\ -\infty, & y_j = 1 \\ 0, & y_j = e \end{cases}$$

Recall the Sum-Product Algorithm

- Channel:

$$L_j = \begin{cases} +\infty, & y_j = 0 \\ -\infty, & y_j = 1 \\ 0, & y_j = e \end{cases}$$



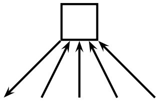
- Check node (CN) update:

$$L_{i\to j} = 2\tanh^{-1}\left(\prod_{j'\in N(i)-\{j\}} \tanh\left(\frac{1}{2}L_{j'\to i}\right)\right)$$

Recall the Sum-Product Algorithm

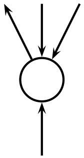
- Channel:

$$L_j = \begin{cases} +\infty, & y_j = 0 \\ -\infty, & y_j = 1 \\ 0, & y_j = e \end{cases}$$



- Check node (CN) update:

$$L_{i\to j} = 2\tanh^{-1}\left(\prod_{j'\in N(i)-\{j\}} \tanh\left(\frac{1}{2}L_{j'\to i}\right)\right)$$

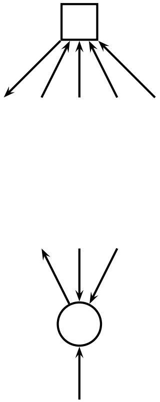


- Variable node (VN) update:

$$L_{j\to i} = L_j + \sum_{i'\in N(j)-\{i\}} L_{i'\to j}$$

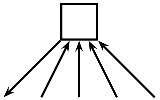
Recall the Sum-Product Algorithm

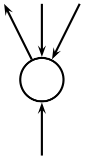
- $p^{(l)}$: probability that an outgoing VN message is an erasure, with $p^{(0)} = \varepsilon$



Recall the Sum-Product Algorithm

- $p^{(l)}$: probability that an outgoing VN message is an erasure, with $p^{(0)} = \varepsilon$
- $q^{(l)}$: probability that an outgoing CN message is an erasure

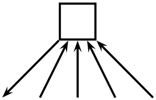


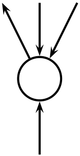


Recall the Sum-Product Algorithm

- $p^{(l)}$: probability that an outgoing VN message is an erasure, with $p^{(0)} = \varepsilon$
- $q^{(l)}$: probability that an outgoing CN message is an erasure
- CN update (degree k):

$$q^{(l)} = 1 - (1 - p^{(l-1)})^{k-1}$$

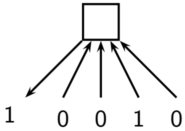


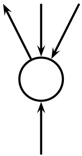


Recall the Sum-Product Algorithm

- $p^{(l)}$: probability that an outgoing VN message is an erasure, with $p^{(0)} = \varepsilon$
- $q^{(l)}$: probability that an outgoing CN message is an erasure
- CN update (degree k):

$$q^{(l)} = 1 - (1 - p^{(l-1)})^{k-1}$$

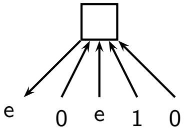


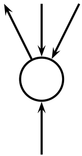


Recall the Sum-Product Algorithm

- $p^{(l)}$: probability that an outgoing VN message is an erasure, with $p^{(0)} = \varepsilon$
- $q^{(l)}$: probability that an outgoing CN message is an erasure
- CN update (degree k):

$$q^{(l)} = 1 - (1 - p^{(l-1)})^{k-1}$$





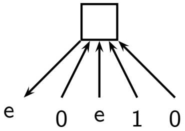
Recall the Sum-Product Algorithm

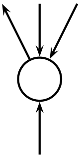
- $p^{(l)}$: probability that an outgoing VN message is an erasure, with $p^{(0)} = \varepsilon$
- $q^{(l)}$: probability that an outgoing CN message is an erasure
- CN update (degree k):

$$q^{(l)} = 1 - (1 - p^{(l-1)})^{k-1}$$

- VN update (degree k):

$$p^{(l)} = \varepsilon (q^{(l)})^{k-1}$$





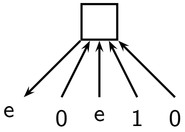
Recall the Sum-Product Algorithm

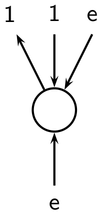
- $p^{(l)}$: probability that an outgoing VN message is an erasure, with $p^{(0)} = \varepsilon$
- $q^{(l)}$: probability that an outgoing CN message is an erasure
- CN update (degree k):

$$q^{(l)} = 1 - (1 - p^{(l-1)})^{k-1}$$

- VN update (degree k):

$$p^{(l)} = \varepsilon (q^{(l)})^{k-1}$$





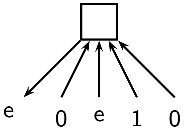
Recall the Sum-Product Algorithm

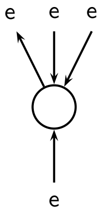
- $p^{(l)}$: probability that an outgoing VN message is an erasure, with $p^{(0)} = \varepsilon$
- $q^{(l)}$: probability that an outgoing CN message is an erasure
- CN update (degree k):

$$q^{(l)} = 1 - (1 - p^{(l-1)})^{k-1}$$

- VN update (degree k):

$$p^{(l)} = \varepsilon (q^{(l)})^{k-1}$$

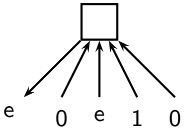




Recall the Sum-Product Algorithm

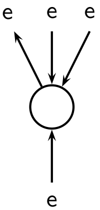
- $p^{(l)}$: probability that an outgoing VN message is an erasure, with $p^{(0)} = \varepsilon$
- $q^{(l)}$: probability that an outgoing CN message is an erasure
- CN update (degree k):

$$q^{(l)} = 1 - (1 - p^{(l-1)})^{k-1}$$



- VN update (degree k):

$$p^{(l)} = \varepsilon (q^{(l)})^{k-1}$$



- Averaged over all VNs and CNs ($\lambda(x)$ is the VN degree distribution, $\rho(x)$ is the CN degree distribution):

$$p^{(l)} = \varepsilon \lambda(1 - \rho(1 - p^{(l-1)}))$$

Threshold Example (Capacity $C = 0.5$)

Example

- Regular $(3, 6)$ LDPC block code ensemble, rate $R = 1/2$:

$$\rho(x) = x^5 \qquad \lambda(x) = x^2$$

- Threshold $\varepsilon^* = 0.429$

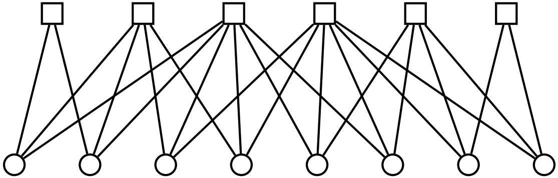
Example

- Regular $(5, 10)$ LDPC block code ensemble, rate $R = 1/2$:

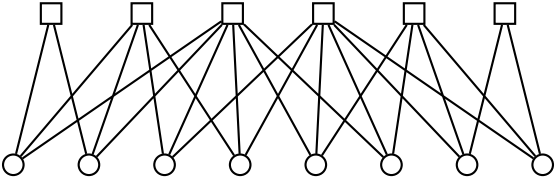
$$\rho(x) = x^9 \qquad \lambda(x) = x^4$$

- Threshold $\varepsilon^* = 0.341$

Density Evolution for Protograph Ensembles

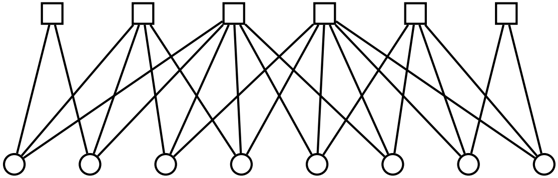


Density Evolution for Protograph Ensembles



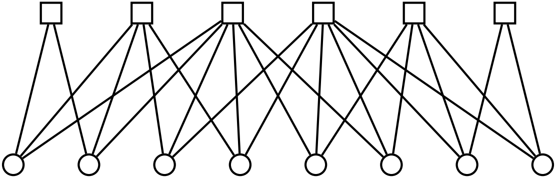
- Protograph = prototype graph

Density Evolution for Protograph Ensembles



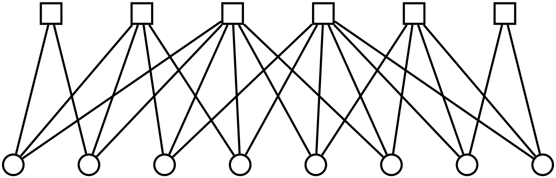
- Protograph = prototype graph
- Tanner graph of the code results from a copy-and-permute procedure that preserves the degree distribution of all nodes

Density Evolution for Protograph Ensembles



- Protograph = prototype graph
- Tanner graph of the code results from a copy-and-permute procedure that preserves the degree distribution of all nodes
- Imposes structure on the resulting LDPC code ensemble

Density Evolution for Protograph Ensembles

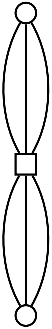


- Protograph = prototype graph
- Tanner graph of the code results from a copy-and-permute procedure that preserves the degree distribution of all nodes
- Imposes structure on the resulting LDPC code ensemble
- Now the erasure probabilities are functions of the edges:

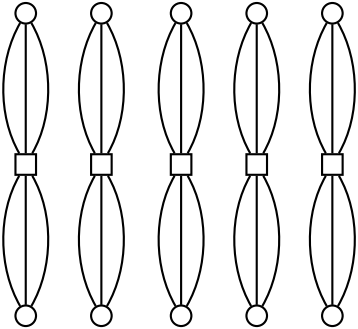
$$q^{(l)}(e_{i\to j}) = 1 - \prod_{j'\in N(i)-\{j\}} (1 - p^{(l-1)}(e_{j'\to i}))$$

$$p^{(l)}(e_{j\to i}) = \varepsilon \prod_{i'\in N(i)-\{j\}} q^{(l)}(e_{i'\to j})$$

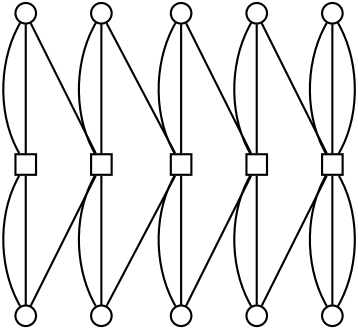
Spatial Coupling via "Edge Spreading"



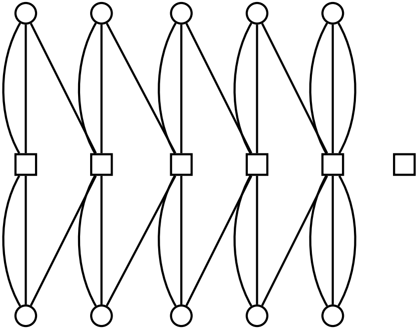
Spatial Coupling via “Edge Spreading”



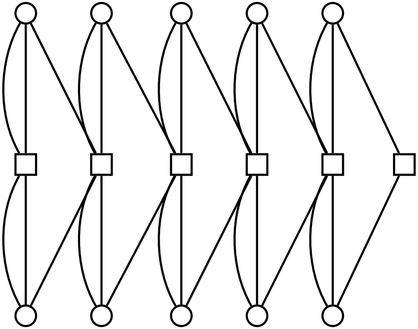
Spatial Coupling via “Edge Spreading”



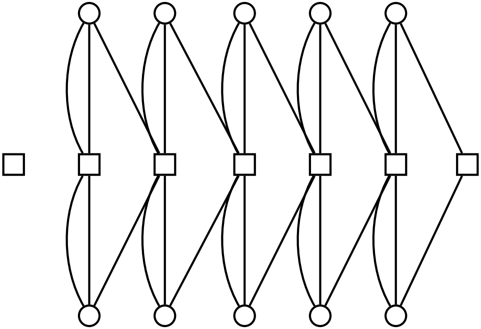
Spatial Coupling via “Edge Spreading”



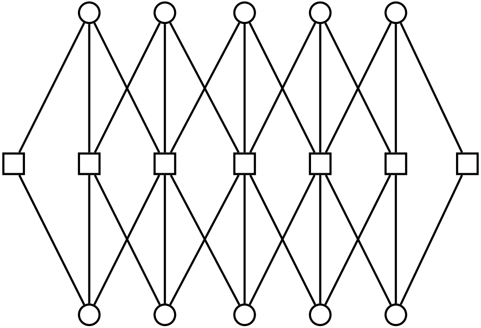
Spatial Coupling via "Edge Spreading"



Spatial Coupling via "Edge Spreading"



Spatial Coupling via "Edge Spreading"



Properties

Properties

- Parity check matrix has a convolutional-like structure:

$$B = \begin{pmatrix} 1&1&0&0&0&0&0&0&0&0&0&0&0&0\\ 1&1&1&1&0&0&0&0&0&0&0&0&0&0\\ 1&1&1&1&1&1&0&0&0&0&0&0&0&0\\ 0&0&1&1&1&1&1&1&0&0&0&0&0&0\\ 0&0&0&0&1&1&1&1&1&1&0&0&0&0\\ 0&0&0&0&0&0&1&1&1&1&1&1&0&0\\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1\\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1\\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1 \end{pmatrix}$$

Properties

- Parity check matrix has a convolutional-like structure:

$$B = \begin{pmatrix}
1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&1&0&0&0&0&0&0&0&0&0&0\\
1&1&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&1&1&1&1&1&1&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&0&0&0&0\\
0&0&0&0&0&0&1&1&1&1&1&1&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&1&1
\end{pmatrix}$$

- Rate loss due to termination (additional check nodes)

Properties

- Parity check matrix has a convolutional-like structure:

$$B = \begin{pmatrix} 1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&1&1&1&1&1&1&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&1&1&1&0&0&0&0 \\ 0&0&0&0&0&0&1&1&1&1&1&1&0&0 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1 \end{pmatrix}$$

- Rate loss due to termination (additional check nodes)
- Slight irregularities of the CN degrees at the beginning and end of the protograph

Properties

- Parity check matrix has a convolutional-like structure:

$$B = \begin{pmatrix} 1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&0&0&0&0&0&0&0&0&0&0 \\ 1&1&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&1&1&1&1&1&1&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&1&1&1&0&0&0&0 \\ 0&0&0&0&0&0&1&1&1&1&1&1&0&0 \\ 0&0&0&0&0&0&0&0&1&1&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&1&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&1 \end{pmatrix}$$

- Rate loss due to termination (additional check nodes)
- Slight irregularities of the CN degrees at the beginning and end of the protograph
- However, as the number of protograph copies L grows large, the code becomes asymptotically regular and the rate loss becomes negligable

Thresholds for Protograph-based LDPC Convolutional Codes

Thresholds for Protograph-based LDPC Convolutional Codes

- Based on (3,6) LDPC Block protograph ($\varepsilon^* = 0.429$):

L	R_L	ε_L^*	$\varepsilon_{Sh}(R_L)$
5	0.300	0.587	0.700
10	0.400	0.504	0.600
20	0.450	0.488	0.550
40	0.475	0.488	0.525
∞	0.500	0.488	0.500

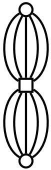
Thresholds for Protograph-based LDPC Convolutional Codes

- Based on (3,6) LDPC Block protograph ($\varepsilon^* = 0.429$):

L	R_L	ε_L^*	$\varepsilon_{Sh}(R_L)$
5	0.300	0.587	0.700
10	0.400	0.504	0.600
20	0.450	0.488	0.550
40	0.475	0.488	0.525
∞	0.500	0.488	0.500

- Based on (5,10) LDPC Block protograph ($\varepsilon^* = 0.341$):

L	R_L	ε_L^*	$\varepsilon_{Sh}(R_L)$
5	0.100	0.625	0.900
10	0.300	0.512	0.700
20	0.400	0.499	0.600
40	0.450	0.499	0.550
∞	0.500	0.499	0.500



Erasure Probability Evolution at $\varepsilon = 0.483$, $L = 20$, (3,6) LDPC code

Conclusion

Conclusion

- Asymptotically regular protograph ensembles are capable of approaching capacity

Conclusion

- Asymptotically regular protograph ensembles are capable of approaching capacity
- Spatial coupling and termination leads to slight irregularities of the CN degrees

Conclusion

- Asymptotically regular protograph ensembles are capable of approaching capacity
- Spatial coupling and termination leads to slight irregularities of the CN degrees
- Same effect is also present for other channels, e.g., AWGN

Thank you!