Graphical Models and Inference: Breakthroughs and Insight from Spatial Coupling

Henry D. Pfister

Based on joint work with Yung-Yih Jian, Santhosh Kumar, Krishna R. Narayanan, Phong S. Nguyen, and Arvind Yedla

Duke University

March 17th, 2014

Outline

Outline

Graphical Models and LDPC Codes

Spatially-Coupled Graphical Models

Universality for Multiuser Scenarios

General Formulation of Threshold Saturation

Wyner-Ziv and Gelfand-Pinsker

Conclusions

Graphical Models

- A graphical model provides a graphical representation of the local dependence structure for a set of random variables
  - Examples: factor graphs, bayesian networks, etc...

Graphical Models

- A graphical model provides a graphical representation of the local dependence structure for a set of random variables
  - Examples: factor graphs, bayesian networks, etc...
- Consider random variables $(X_1, X_2, \ldots, X_4) \in \mathcal{X}^4$ and Y where:

$$\begin{aligned} P(x_1, x_2, x_3, x_4) &\triangleq \mathbb{P}(X_1 = x_1, X_2 = x_2, \ldots, X_4 = x_4 | Y = y) \\ &\propto f(x_1, x_2, x_3, x_4) \\ &\triangleq f_1(x_1, x_2) f_2(x_2, x_3) f_3(x_3, x_4) \end{aligned}$$

Graphical Models

- A graphical model provides a graphical representation of the local dependence structure for a set of random variables
  - Examples: factor graphs, bayesian networks, etc...
- Consider random variables $(X_1, X_2, \ldots, X_4) \in \mathcal{X}^4$ and Y where:

$$\begin{aligned} P(x_1, x_2, x_3, x_4) &\triangleq \mathbb{P}(X_1 = x_1, X_2 = x_2, \ldots, X_4 = x_4 | Y = y) \\ &\propto f(x_1, x_2, x_3, x_4) \\ &\triangleq f_1(x_1, x_2) f_2(x_2, x_3) f_3(x_3, x_4) \end{aligned}$$

- Given $Y = y$, this describes a Markov chain whose factor graph is

Inference via Marginalization

- Marginalizing out all variables except X_1 gives

$$\mathbb{P}(X_1 = x_1 | Y = y) \propto g_1(x_1) \triangleq \sum_{(x_2,\ldots,x_4)\in\mathcal{X}^3} f(x_1, x_2, x_3, x_4)$$

- Thus, the maximum a posteriori decision for X_1 given $Y = y$ is

$$\hat{x}_1 = \arg\max_{x_1\in\mathcal{X}} \sum_{(x_2,\ldots,x_4)\in\mathcal{X}^3} f(x_1, x_2, x_3, x_4)$$

  - In general, this requires roughly $|\mathcal{X}|^4$ operations

Inference via Marginalization

- Marginalizing out all variables except X_1 gives

$$\mathbb{P}(X_1 = x_1 | Y = y) \propto g_1(x_1) \triangleq \sum_{(x_2,\ldots,x_4)\in\mathcal{X}^3} f(x_1, x_2, x_3, x_4)$$

- Thus, the maximum a posteriori decision for X_1 given $Y = y$ is

$$\hat{x}_1 = \arg\max_{x_1\in\mathcal{X}} \sum_{(x_2,\ldots,x_4)\in\mathcal{X}^3} f(x_1, x_2, x_3, x_4)$$

  - In general, this requires roughly $|\mathcal{X}|^4$ operations

- Marginalization is efficient for tree-structured factor graphs
  - For this Markov chain, roughly $5\,|\mathcal{X}|^2$ operations required

$$g_1(x_1) = \sum_{x_2\in\mathcal{X}} f_1(x_1, x_2) \sum_{x_3\in\mathcal{X}} f_2(x_2, x_3) \sum_{x_4\in\mathcal{X}} f_3(x_3, x_4)$$

Sudoku: A Well-Known Example

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

rows are permutations of $\{1, 2, \ldots, 9\}$

Sudoku: A Well-Known Example

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

rows are permutations of $\{1, 2, \ldots, 9\}$
columns are permutations of $\{1, 2, \ldots, 9\}$

Sudoku: A Well-Known Example

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

rows are permutations of $\{1, 2, \ldots, 9\}$
columns are permutations of $\{1, 2, \ldots, 9\}$
subblocks are permutations of $\{1, 2, \ldots, 9\}$

Sudoku: A Well-Known Example

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

rows are permutations of $\{1, 2, \ldots, 9\}$
columns are permutations of $\{1, 2, \ldots, 9\}$
subblocks are permutations of $\{1, 2, \ldots, 9\}$

x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}
x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}	x_{29}
x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}	x_{37}	x_{38}	x_{39}
x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}	x_{49}
x_{51}	x_{52}	x_{53}	x_{54}	x_{55}	x_{56}	x_{57}	x_{58}	x_{59}
x_{61}	x_{62}	x_{63}	x_{64}	x_{65}	x_{66}	x_{67}	x_{68}	x_{69}
x_{71}	x_{72}	x_{73}	x_{74}	x_{75}	x_{76}	x_{77}	x_{78}	x_{79}
x_{81}	x_{82}	x_{83}	x_{84}	x_{85}	x_{86}	x_{87}	x_{88}	x_{89}
x_{91}	x_{92}	x_{93}	x_{94}	x_{95}	x_{96}	x_{97}	x_{98}	x_{99}

implied factor graph has
81 variable and 27 factor nodes

$$f(\underline{x}) = \left(\prod_{i=1}^{9} f_\sigma(x_{i*}) \right) \left(\prod_{j=1}^{9} f_\sigma(x_{*j}) \right) \left(\prod_{k=1}^{9} f_\sigma(x_{B(k)}) \right) \prod_{(i,j)\in O} \mathbb{I}(x_{ij} = y_{ij})$$

Solving Sudoku via Marginalization

- Consider any constraint satisfaction problem with erased entries
  - One can write $f(\underline{x})$ as the product of indicator functions
  - Some factors force $\underline{x}$ to be valid (i.e., satisfy constraints)
  - Other factors force $\underline{x}$ to be compatible with observed values
  - Summing over $\underline{x}$ counts the $\#$ of valid compatible sequences

Solving Sudoku via Marginalization

- Consider any constraint satisfaction problem with erased entries
  - One can write $f(\underline{x})$ as the product of indicator functions
  - Some factors force $\underline{x}$ to be valid (i.e., satisfy constraints)
  - Other factors force $\underline{x}$ to be compatible with observed values
  - Summing over $\underline{x}$ counts the $\#$ of valid compatible sequences
- Marginalization allows uniform sampling from valid compatible set
  - Sample $x_1' \sim g_1(\cdot)$, fix $x_1 = x_1'$, sample $x_2' \sim g_2(\cdot|x_1)$, etc...
    - For Sudoku, this always works because only one solution!
  - Low complexity if factor graph forms a tree
  - If not a tree, low-complexity approximation sometimes possible
  - But, in general, marginalization is $\#$P-complete

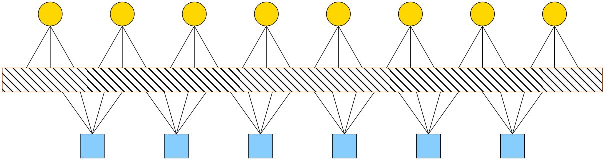
Solving Sudoku via Marginalization

- Consider any constraint satisfaction problem with erased entries
  - One can write $f(\underline{x})$ as the product of indicator functions
  - Some factors force $\underline{x}$ to be valid (i.e., satisfy constraints)
  - Other factors force $\underline{x}$ to be compatible with observed values
  - Summing over $\underline{x}$ counts the # of valid compatible sequences
- Marginalization allows uniform sampling from valid compatible set
  - Sample $x_1' \sim g_1(\cdot)$, fix $x_1 = x_1'$, sample $x_2' \sim g_2(\cdot|x_1)$, etc...
    - For Sudoku, this always works because only one solution!
  - Low complexity if factor graph forms a tree
  - If not a tree, low-complexity approximation sometimes possible
  - But, in general, marginalization is #P-complete
- Enough fun and games, how about some engineering problems!

Point-to-Point Communication

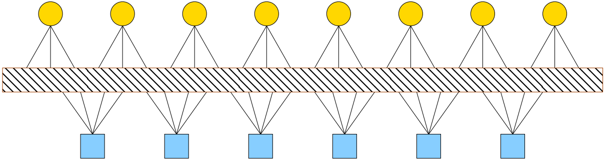
- Channel Model
  - Transition probability: $P_{Y|X}(y|x)$ for $x \in \mathcal{X}$ and $y \in \mathcal{Y}$
  - Transmit a length-n codeword $\underline{x} \in \mathcal{C} \subset \mathcal{X}^n$
- Shannon Capacity
  - Code rate: $R \triangleq \frac{1}{n} \log_2 |\mathcal{C}|$ (bits per channel use)
  - As $n \to \infty$, reliable transmission iff $R < C \triangleq \max_{p(x)} I(X;Y)$
- Example: the binary erasure channel BEC(ε)
  - Bits sent perfectly (with prob. $1-\varepsilon$) or erased (with prob. ε)
  - Capacity: $C = 1 - \varepsilon =$ fraction unerased bits
  - Roughly one info bit transmitted for each unerased reception

Low-Density Parity-Check (LDPC) Codes



- Linear codes defined by $\underline{x}H^T = \underline{0}$ for all c.w. $\underline{x} \in \mathcal{C} \subset \{0,1\}^n$
  - H is an $r \times n$ sparse parity-check matrix for the code
  - Code bits and parity checks associated with cols/rows of H

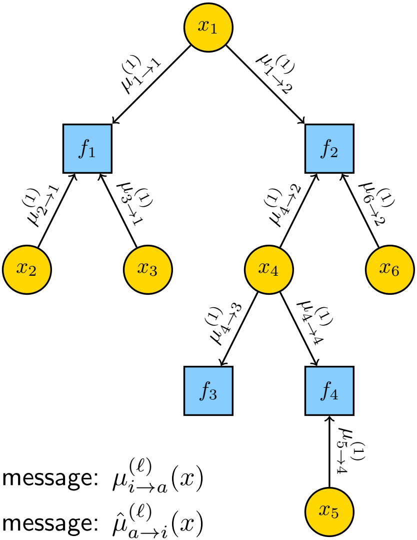
Low-Density Parity-Check (LDPC) Codes



- Linear codes defined by $\underline{x}H^T = \underline{0}$ for all c.w. $\underline{x} \in \mathcal{C} \subset \{0,1\}^n$
  - H is an $r \times n$ sparse parity-check matrix for the code
  - Code bits and parity checks associated with cols/rows of H
- Factor graph: H is the biadjacency matrix for variable/factor nodes
  - Ensemble defined by configuration model for random graphs
  - Checks define factors: $f_{\text{even}}(x_1^d) = \mathbb{I}(x_1 \oplus \cdots \oplus x_d = 0)$
  - Let $x_{F(a)}$ be the x-subvector for the a-th check and

$$f(x_1, \ldots, x_n) = \underbrace{\left(\prod_{a=1}^{r} f_{\text{even}}(x_{F(a)}) \right)}_{\mathbf{1}_{\mathcal{C}}(x_1^n)} \left(\prod_{i=1}^{n} P_{Y|X}(y_i|x_i) \right)$$

Marginalization via Belief Propagation

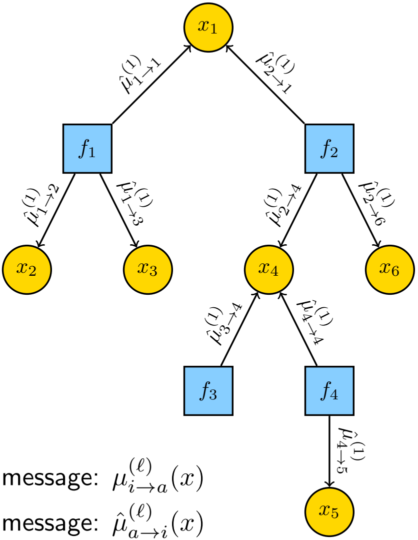


variable-to-factor message: $\mu^{(\ell)}_{i\to a}(x)$

factor-to-variable message: $\hat{\mu}^{(\ell)}_{a\to i}(x)$

$$f(x_1,x_2,x_3,x_4,x_5,x_6) = f_1(x_1,x_2,x_3)f_2(x_1,x_4,x_6)f_3(x_4)f_4(x_4,x_5)$$

Marginalization via Belief Propagation

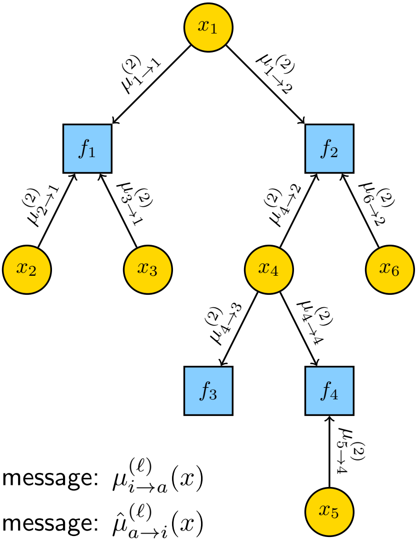


variable-to-factor message: $\mu^{(\ell)}_{i\to a}(x)$

factor-to-variable message: $\hat{\mu}^{(\ell)}_{a\to i}(x)$

$$f(x_1,x_2,x_3,x_4,x_5,x_6) = f_1(x_1,x_2,x_3)f_2(x_1,x_4,x_6)f_3(x_4)f_4(x_4,x_5)$$

Marginalization via Belief Propagation

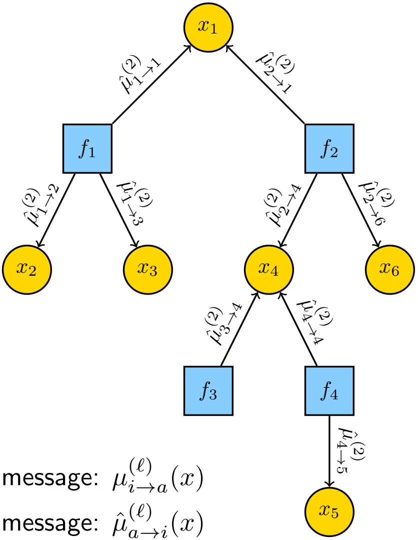


variable-to-factor message: $\mu^{(\ell)}_{i\to a}(x)$

factor-to-variable message: $\hat{\mu}^{(\ell)}_{a\to i}(x)$

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)$$

Marginalization via Belief Propagation



variable-to-factor message: $\mu^{(\ell)}_{i\to a}(x)$

factor-to-variable message: $\hat{\mu}^{(\ell)}_{a\to i}(x)$

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5)$$

A Little History

Robert Gallager

introduced LDPC codes in 1962 paper

1962 *IRE TRANSACTIONS ON INFORMATION THEORY* 21

Low-Density Parity-Check Codes*

R. G. GALLAGER†

Summary—A low-density parity-check code is a code specified by a parity-check matrix with the following properties: each column contains a small fixed number $j \geq 3$ of 1's and each row contains a small fixed number $k > j$ of 1's. The typical minimum distance of these codes increases linearly with block length for a fixed rate and fixed j. When used with maximum likelihood decoding on a sufficiently quiet binary-input symmetric channel, the typical probability of decoding error decreases exponentially with block length for a fixed rate and fixed j.

A simple but nonoptimum decoding scheme operating directly from the channel a posteriori probabilities is described. Both the

equations. We call the set of digits contained in a parity-check equation a parity-check set. For example, the first parity-check set in Fig. 1 is the set of digits (1, 2, 3, 5).

The use of parity-check codes makes coding (as distinguished from decoding) relatively simple to implement. Also, as Elias [3] has shown, if a typical parity-check code of long block length is used on a binary symmetric channel, and if the code rate is between *critical rate* and channel capacity, then the probability of decoding error

Judea Pearl

defined general belief-propagation in 1986 paper

Fusion, Propagation, and Structuring in Belief Networks*

Judea Pearl
Cognitive Systems Laboratory, Computer Science Department, University of California, Los Angeles, CA 90024, U.S.A.

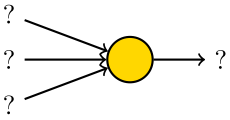
Recommended by Patrick Hayes

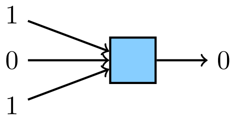
ABSTRACT

Belief networks are directed acyclic graphs in which the nodes represent propositions (or variables), the arcs signify direct dependencies between the linked propositions, and the strengths of these dependencies are quantified by conditional probabilities. A network of this sort can be used to represent the generic knowledge of a domain expert, and it turns into a computational architecture if the links are used not merely for storing factual knowledge but also for directing and activating the data flow in the computations which manipulate this knowledge.

Simple Message-Passing Decoding for the BEC

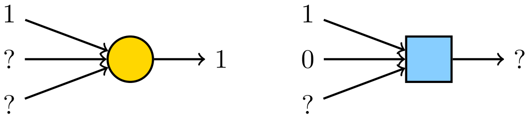
- Constraint nodes define the valid patterns
  - Circles represent a single value shared by factors
  - Squares assert attached variables sum to 0 mod 2
- Iterative decoding on the binary erasure channel (BEC)
  - Messages passed in phases: bit-to-check and check-to-bit
  - Each output message depends on other input messages
  - Each message is either the correct value or an erasure
- Message passing rules for the BEC
  - Bits pass an erasure only if all other inputs are erased
  - Checks pass the correct value only if all other inputs are correct



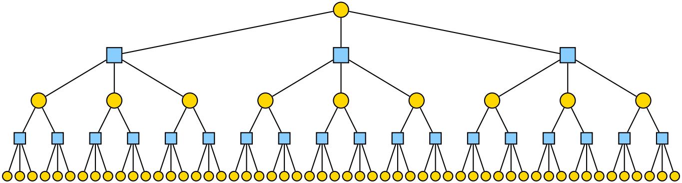


Simple Message-Passing Decoding for the BEC

- Constraint nodes define the valid patterns
  - Circles represent a single value shared by factors
  - Squares assert attached variables sum to 0 mod 2
- Iterative decoding on the binary erasure channel (BEC)
  - Messages passed in phases: bit-to-check and check-to-bit
  - Each output message depends on other input messages
  - Each message is either the correct value or an erasure
- Message passing rules for the BEC
  - Bits pass an erasure only if all other inputs are erased
  - Checks pass the correct value only if all other inputs are correct



Computation Graph and Density Evolution



$\tilde{x}_3 = \varepsilon y_2^3$

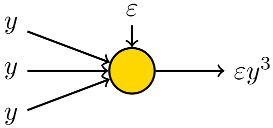
$y_2 = 1-(1-x_2)^3$

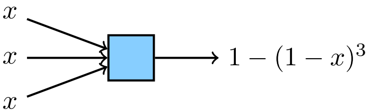
$x_2 = \varepsilon y_1^2$

$y_1 = 1-(1-x_1)^3$

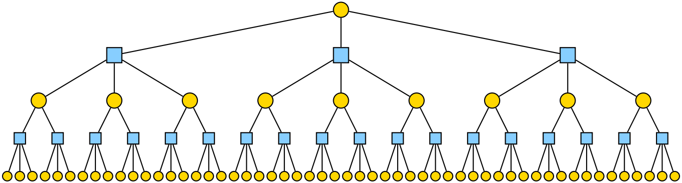
$x_1 = \varepsilon$

- Computation graph for a (3,4)-regular LDPC code
  - Illustrates decoding from the perspective of a single bit-node
  - For long random LDPC codes, the graph is typically a tree
  - Allows density evolution to track message erasure probability
  - If x/y are erasure prob. of bit/check output messages, then





Computation Graph and Density Evolution



$\tilde{x}_3 = \varepsilon y_2^3$

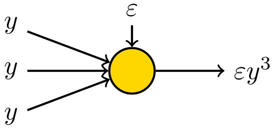
$y_2 = 1-(1-x_2)^3$

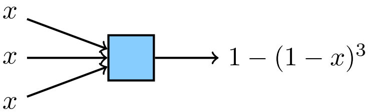
$x_2 = \varepsilon y_1^2$

$y_1 = 1-(1-x_1)^3$

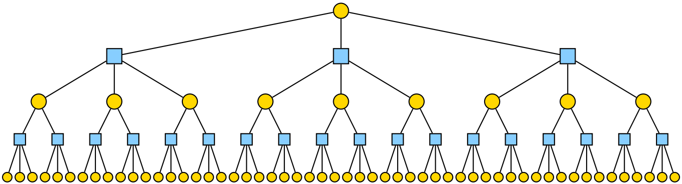
$x_1 = 0.600$

- Computation graph for a (3,4)-regular LDPC code
  - Illustrates decoding from the perspective of a single bit-node
  - For long random LDPC codes, the graph is typically a tree
  - Allows density evolution to track message erasure probability
  - If x/y are erasure prob. of bit/check output messages, then





Computation Graph and Density Evolution



$$\tilde{x}_3 = \varepsilon y_2^3$$

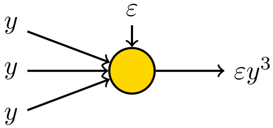
$$y_2 = 1-(1-x_2)^3$$

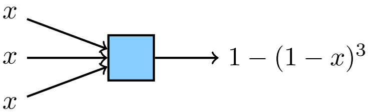
$$x_2 = \varepsilon y_1^2$$

$$y_1 = 0.936$$

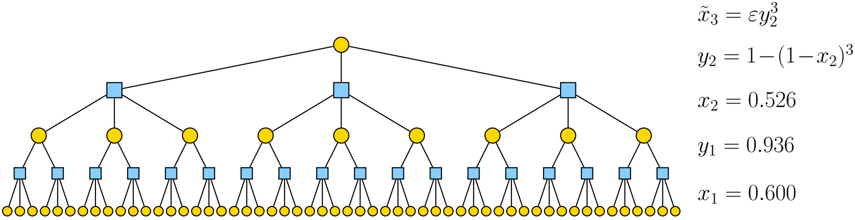
$$x_1 = 0.600$$

- Computation graph for a (3,4)-regular LDPC code
  - Illustrates decoding from the perspective of a single bit-node
  - For long random LDPC codes, the graph is typically a tree
  - Allows density evolution to track message erasure probability
  - If x/y are erasure prob. of bit/check output messages, then

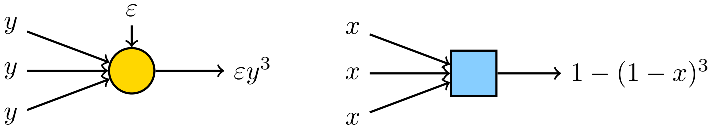




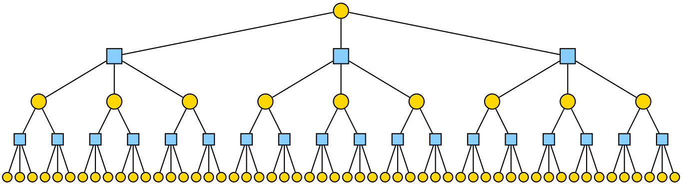
Computation Graph and Density Evolution



- Computation graph for a (3,4)-regular LDPC code
  - Illustrates decoding from the perspective of a single bit-node
  - For long random LDPC codes, the graph is typically a tree
  - Allows density evolution to track message erasure probability
  - If x/y are erasure prob. of bit/check output messages, then



Computation Graph and Density Evolution



$\tilde{x}_3 = \varepsilon y_2^3$

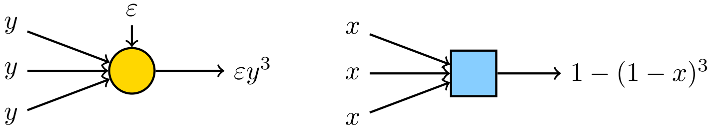
$y_2 = 0.894$

$x_2 = 0.526$

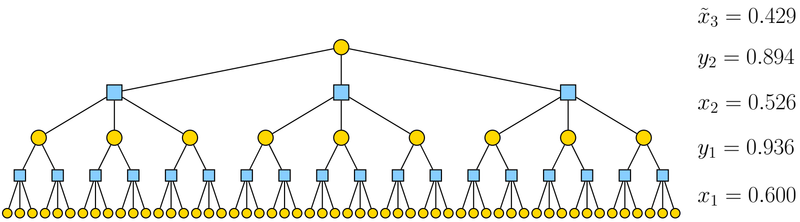
$y_1 = 0.936$

$x_1 = 0.600$

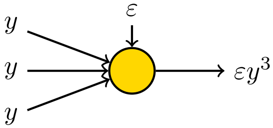
- Computation graph for a (3,4)-regular LDPC code
  - Illustrates decoding from the perspective of a single bit-node
  - For long random LDPC codes, the graph is typically a tree
  - Allows density evolution to track message erasure probability
  - If x/y are erasure prob. of bit/check output messages, then

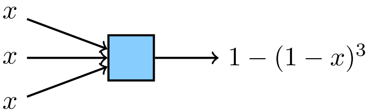


Computation Graph and Density Evolution

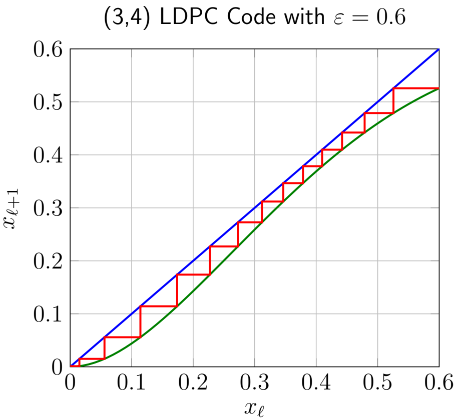


- Computation graph for a (3,4)-regular LDPC code
  - Illustrates decoding from the perspective of a single bit-node
  - For long random LDPC codes, the graph is typically a tree
  - Allows density evolution to track message erasure probability
  - If x/y are erasure prob. of bit/check output messages, then





Density Evolution (DE) for LDPC Codes



Density evolution for a $(3,4)$-regular LDPC code:

$$x_{\ell+1} = \varepsilon \left(1 - (1 - x_\ell)^3\right)^2$$

Decoding Thresholds:

$$\varepsilon^{\mathrm{BP}} \approx 0.647$$
$$\varepsilon^{\mathrm{MAP}} \approx 0.746$$
$$\varepsilon^{\mathrm{Sh}} = 0.750$$

- Binary erasure channel (BEC) with erasure prob. ε
- DE tracks bit-to-check msg erasure rate x_ℓ after ℓ iterations
- Defines noise threshold $\varepsilon^{\mathrm{BP}}$ for the large system limit
  - Easily computed numerically for given code ensemble

Quick Review

- Factor Graphs
  - Represent a combination of constraints and observations
  - Marginalization enables inference
  - Efficient approximate marginalization via belief propagation

Quick Review

- Factor Graphs
  - Represent a combination of constraints and observations
  - Marginalization enables inference
  - Efficient approximate marginalization via belief propagation
- Low-Density Parity-Check Codes
  - Low-complexity decoding via belief-propagation (BP)
  - Density evolution computes noise threshold for BP decoding

Quick Review

- Factor Graphs
  - Represent a combination of constraints and observations
  - Marginalization enables inference
  - Efficient approximate marginalization via belief propagation
- Low-Density Parity-Check Codes
  - Low-complexity decoding via belief-propagation (BP)
  - Density evolution computes noise threshold for BP decoding
- Historical Notes
  - DE for LDPC on BEC introduced by LMSSS in 1997
  - Extended to general channels by RU in 2001

Outline

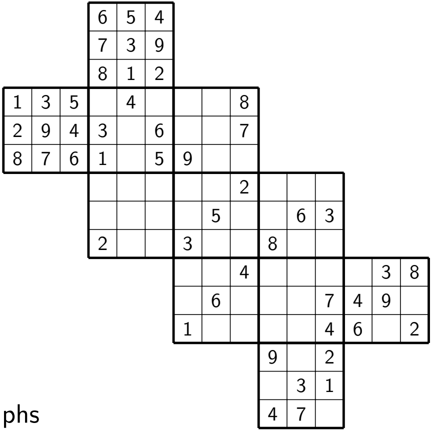
What is Spatial Coupling?

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

			6	5	4									
			7	3	9									
			8	1	2									
1	3	5		4				8						
2	9	4	3		6			7						
8	7	6	1		5	9								
								2						
							5			6	3			
			2			3			8					
								4					3	8
							6				7	4	9	
						1					4	6		2
									9		2			
										3	1			
									4	7				

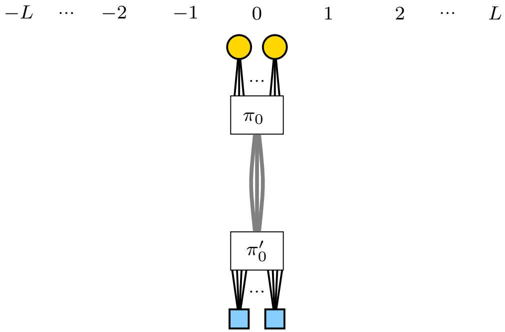
What is Spatial Coupling?

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

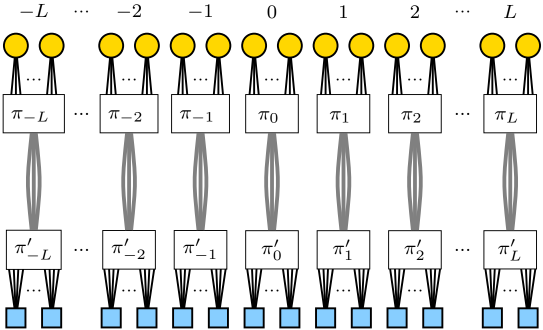


- Spatially-Coupled Factor Graphs
  - Variable nodes have a natural global orientation
  - Boundaries help variables to be recovered in an ordered fashion

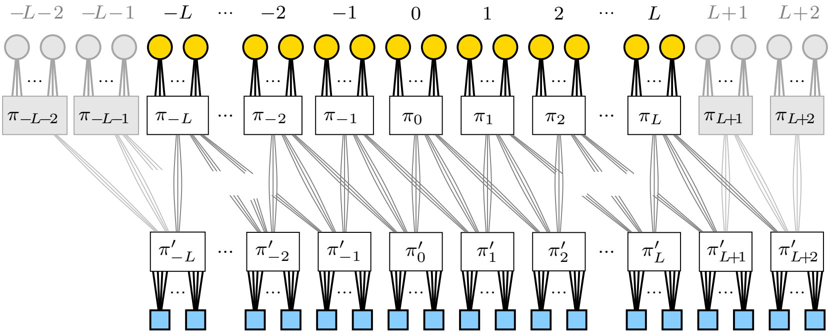
Spatially-Coupled LDPC Codes



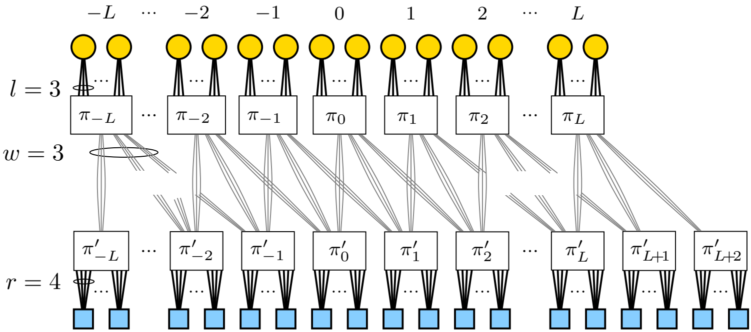
Spatially-Coupled LDPC Codes



Spatially-Coupled LDPC Codes



Spatially-Coupled LDPC Codes



- Historical Notes
  - LDPC convolutional codes introduced by FZ in 1999
  - Shown to have near optimal noise thresholds by LSZC in 2005
  - (l, r, L, w) ensemble proven to achieve capacity by KRU in 2011

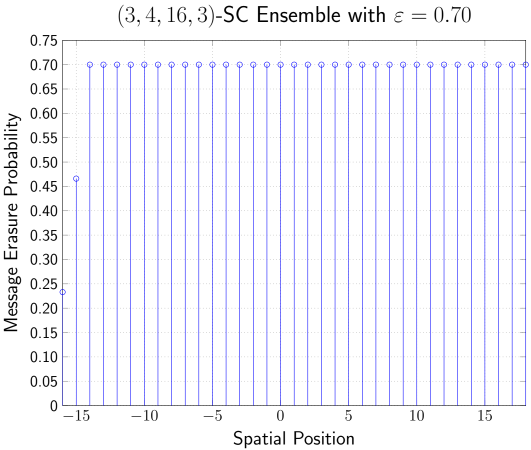
The Spatial Coupling KRU

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 57, NO. 2, FEBRUARY 2011 803

Threshold Saturation via Spatial Coupling: Why Convolutional LDPC Ensembles Perform So Well over the BEC

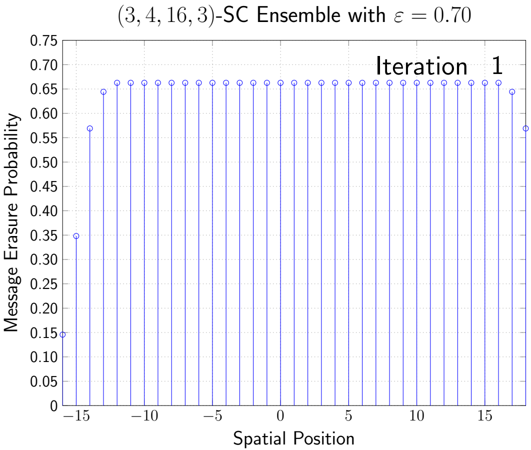
Shrinivas Kudekar, *Member, IEEE*, Thomas J. Richardson, *Fellow, IEEE*, and Rüdiger L. Urbanke

Density Evolution for the (l, r, L, w)-SC LDPC Ensemble



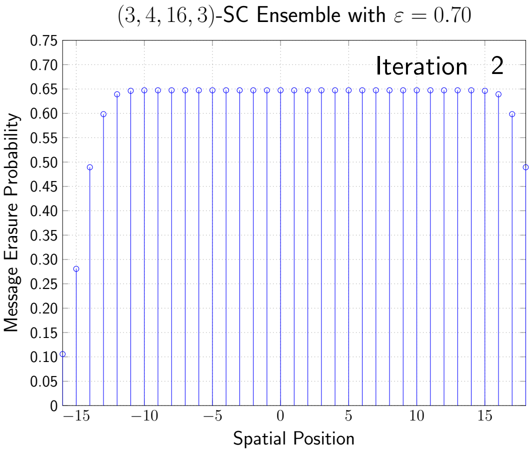
$$z_i^{(\ell+1)} = \varepsilon \left(1 - \frac{1}{w}\sum_{j=0}^{w-1}\left(1 - \frac{1}{w}\sum_{k=0}^{w-1} z_{i+j-k}^{(\ell)}\right)^{r-1}\right)^{l-1}$$

Density Evolution for the (l, r, L, w)-SC LDPC Ensemble



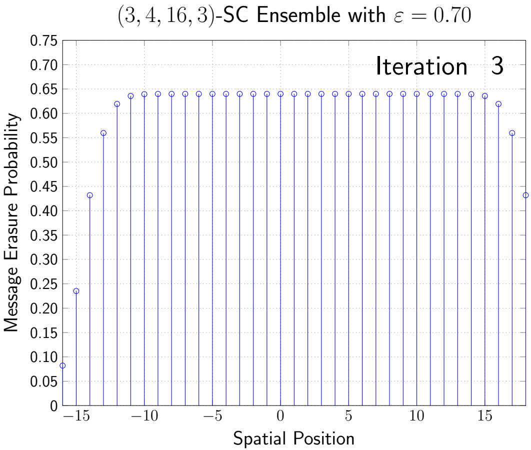
$$z_i^{(\ell+1)} = \varepsilon \left(1 - \frac{1}{w} \sum_{j=0}^{w-1} \left(1 - \frac{1}{w} \sum_{k=0}^{w-1} z_{i+j-k}^{(\ell)} \right)^{r-1} \right)^{l-1}$$

Density Evolution for the (l, r, L, w)-SC LDPC Ensemble



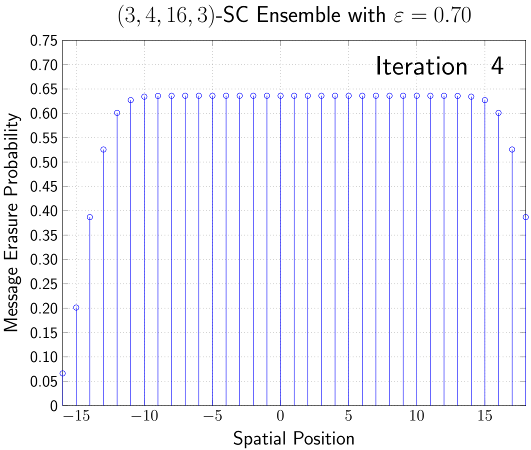
$$z_i^{(\ell+1)} = \varepsilon \left(1 - \frac{1}{w} \sum_{j=0}^{w-1} \left(1 - \frac{1}{w} \sum_{k=0}^{w-1} z_{i+j-k}^{(\ell)}\right)^{r-1}\right)^{l-1}$$

Density Evolution for the (l, r, L, w)-SC LDPC Ensemble



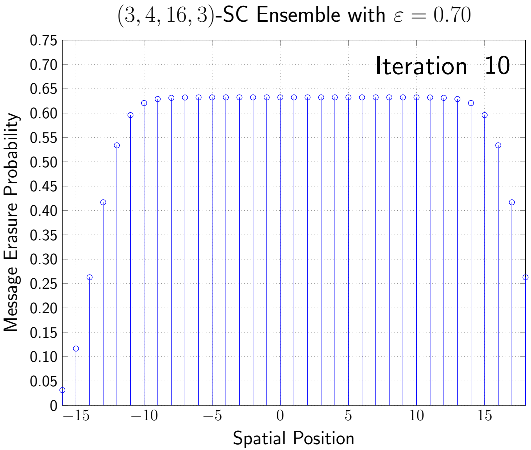
$$z_i^{(\ell+1)} = \varepsilon \left(1 - \frac{1}{w} \sum_{j=0}^{w-1} \left(1 - \frac{1}{w} \sum_{k=0}^{w-1} z_{i+j-k}^{(\ell)} \right)^{r-1} \right)^{l-1}$$

Density Evolution for the (l, r, L, w)-SC LDPC Ensemble



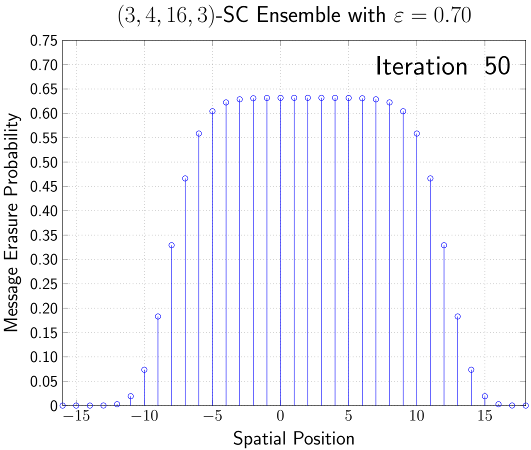
$$z_i^{(\ell+1)} = \varepsilon \left(1 - \frac{1}{w} \sum_{j=0}^{w-1} \left(1 - \frac{1}{w} \sum_{k=0}^{w-1} z_{i+j-k}^{(\ell)}\right)^{r-1}\right)^{l-1}$$

Density Evolution for the (l, r, L, w)-SC LDPC Ensemble



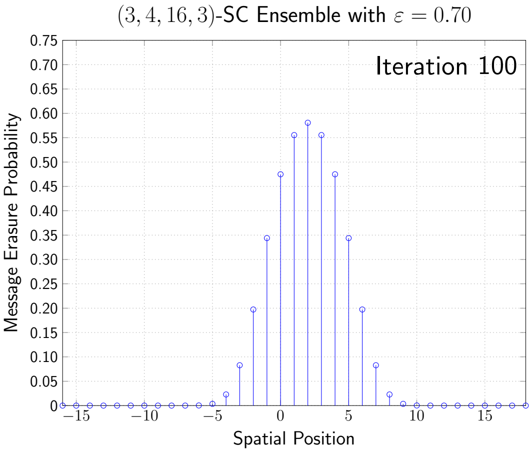
$$z_i^{(\ell+1)} = \varepsilon \left(1 - \frac{1}{w}\sum_{j=0}^{w-1}\left(1 - \frac{1}{w}\sum_{k=0}^{w-1} z_{i+j-k}^{(\ell)}\right)^{r-1}\right)^{l-1}$$

Density Evolution for the (l, r, L, w)-SC LDPC Ensemble



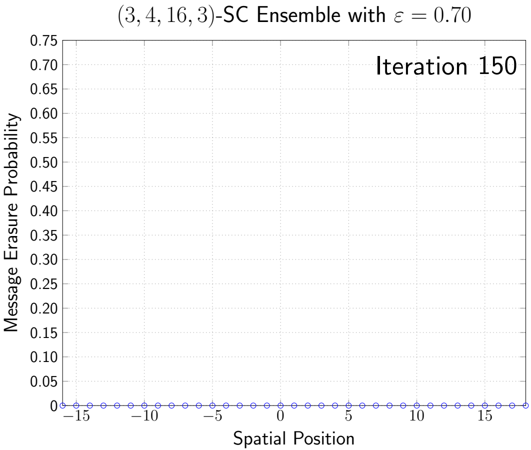
$$z_i^{(\ell+1)} = \varepsilon \left(1 - \frac{1}{w} \sum_{j=0}^{w-1} \left(1 - \frac{1}{w} \sum_{k=0}^{w-1} z_{i+j-k}^{(\ell)} \right)^{r-1} \right)^{l-1}$$

Density Evolution for the (l, r, L, w)-SC LDPC Ensemble



$$z_i^{(\ell+1)} = \varepsilon \left(1 - \frac{1}{w} \sum_{j=0}^{w-1} \left(1 - \frac{1}{w} \sum_{k=0}^{w-1} z_{i+j-k}^{(\ell)} \right)^{r-1} \right)^{l-1}$$

Density Evolution for the (l, r, L, w)-SC LDPC Ensemble



$$z_i^{(\ell+1)} = \varepsilon \left(1 - \frac{1}{w} \sum_{j=0}^{w-1} \left(1 - \frac{1}{w} \sum_{k=0}^{w-1} z_{i+j-k}^{(\ell)} \right)^{r-1} \right)^{l-1}$$

Threshold Saturation via Spatial Coupling

- **General Phenomenon** (observed by Kudekar, Richardson, Urbanke)
  - BP threshold of the spatially-coupled system converges to the MAP threshold of the uncoupled system
  - Can be proven rigorously in many cases!

Threshold Saturation via Spatial Coupling

- **General Phenomenon** (observed by Kudekar, Richardson, Urbanke)
  - BP threshold of the spatially-coupled system converges to the MAP threshold of the uncoupled system
  - Can be proven rigorously in many cases!
- Connection to statistical physics
  - Factor graph defines system of coupled particles
  - Valid sequences are ordered crystalline structures

Threshold Saturation via Spatial Coupling

- **General Phenomenon** (observed by Kudekar, Richardson, Urbanke)
  - BP threshold of the spatially-coupled system converges to the MAP threshold of the uncoupled system
  - Can be proven rigorously in many cases!
- Connection to statistical physics
  - Factor graph defines system of coupled particles
  - Valid sequences are ordered crystalline structures
- Between BP and MAP threshold, system acts as supercooled liquid
  - Correct answer (crystalline state) has minimum energy.
  - Spontaneous crystallization (i.e., decoding) does not occur

http://www.youtube.com/watch?v=Xe8vJrIvDQM

Why is Spatial Coupling Important?

- Breakthroughs: first practical constructions of
  - universal codes for binary-input memoryless channels [KRU12]
  - information-theoretically optimal compressive sensing [DJM11]
  - universal codes for Slepian-Wolf and MAC problems [YJNP11]
  - codes $\rightarrow$ capacity with iterative hard-decision decoding [JNP12]
  - codes $\rightarrow$ rate-distortion limit with iterative decoding [AMUV12]

Why is Spatial Coupling Important?

- Breakthroughs: first practical constructions of
  - universal codes for binary-input memoryless channels [KRU12]
  - information-theoretically optimal compressive sensing [DJM11]
  - universal codes for Slepian-Wolf and MAC problems [YJNP11]
  - codes $\rightarrow$ capacity with iterative hard-decision decoding [JNP12]
  - codes $\rightarrow$ rate-distortion limit with iterative decoding [AMUV12]
- It allows rigorous proof in many cases
  - Original proofs [KRU11/12] quite specific to LDPC codes
  - Our proof is for increasing scalar/vector recursions [YJNP12/13]

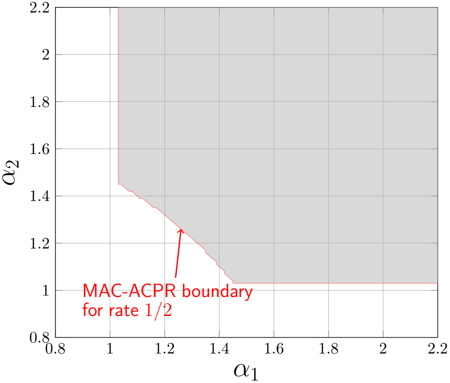
Why is Spatial Coupling Important?

- Breakthroughs: first practical constructions of
  - universal codes for binary-input memoryless channels [KRU12]
  - information-theoretically optimal compressive sensing [DJM11]
  - universal codes for Slepian-Wolf and MAC problems [YJNP11]
  - codes $\rightarrow$ capacity with iterative hard-decision decoding [JNP12]
  - codes $\rightarrow$ rate-distortion limit with iterative decoding [AMUV12]
- It allows rigorous proof in many cases
  - Original proofs [KRU11/12] quite specific to LDPC codes
  - Our proof is for increasing scalar/vector recursions [YJNP12/13]
- Spatial coupling as a proof technique [GMU13]
  - For a large random factor graph, construct a coupled version
  - Use DE to analyze BP decoding of coupled system
  - Compare uncoupled MAP with coupled BP via interpolation

Outline

Universality over Unknown Parameters

- The Achievable Channel Parameter Region (ACPR)
  - For a sequence of coding schemes involving one or more parameters, the parameter region where decoding succeeds in the limit
  - In contrast, a capacity region is a rate region for fixed channels



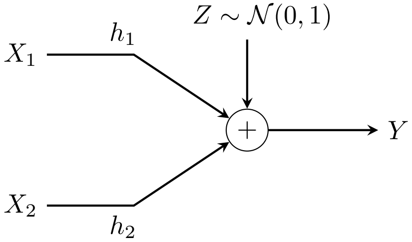
Universality over Unknown Parameters

- The Achievable Channel Parameter Region (ACPR)
  - For a sequence of coding schemes involving one or more parameters, the parameter region where decoding succeeds in the limit
  - In contrast, a capacity region is a rate region for fixed channels
- Properties
  - For fixed encoders, the ACPR depends on the decoders
  - For example, one has BP-ACPR $\subseteq$ MAP-ACPR
  - Often, $\exists$ unique maximal ACPR given by information theory

Universality over Unknown Parameters

- The Achievable Channel Parameter Region (ACPR)
  - For a sequence of coding schemes involving one or more parameters, the parameter region where decoding succeeds in the limit
  - In contrast, a capacity region is a rate region for fixed channels
- Properties
  - For fixed encoders, the ACPR depends on the decoders
  - For example, one has BP-ACPR $\subseteq$ MAP-ACPR
  - Often, $\exists$ unique maximal ACPR given by information theory
- Universality
  - A sequence of encoding/decoding schemes is called universal if:
    its ACPR equals the optimal ACPR
  - Channel parameters are assumed unknown at the transmitter
  - At the receiver, the channel parameters are easily estimated

2-User Binary-Input Gaussian Multiple Access Channel



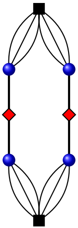
- Fixed noise variance
- Real channel gains h_1 and h_2 not known at transmitter
- Each code has rate R

- MAC-ACPR denotes the information-theoretic optimal region

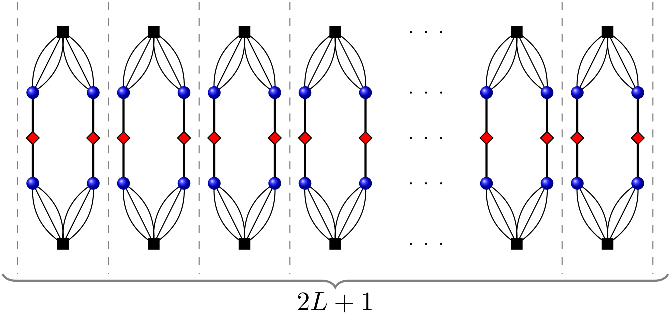
A Little History: SC for Multiple-Access (MAC) Channels

- KK consider a binary-adder erasure channel (ISIT 2011)
  - SC exhibits threshold saturation for the joint decoder
- YNPN consider the Gaussian MAC (ISIT/Allerton 2011)
  - SC exhibits threshold saturation for the joint decoder
  - For channel gains h_1, h_2 unknown at transmitter, SC provides universality
- Others consider CDMA systems without coding
  - TTK show SC improves BP demod of standard CDMA
  - ST prove saturation for a SC protograph-style CDMA

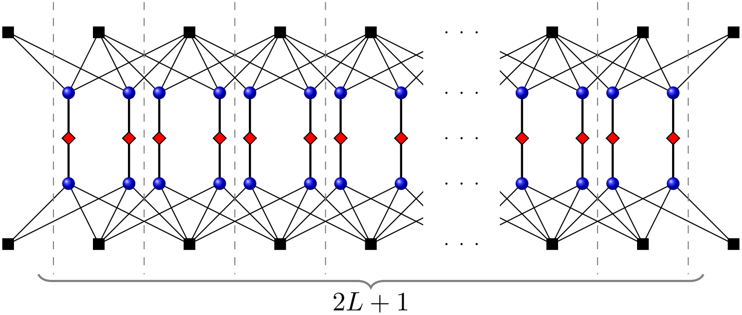
Spatially-Coupled Factor Graph for Joint Decoder



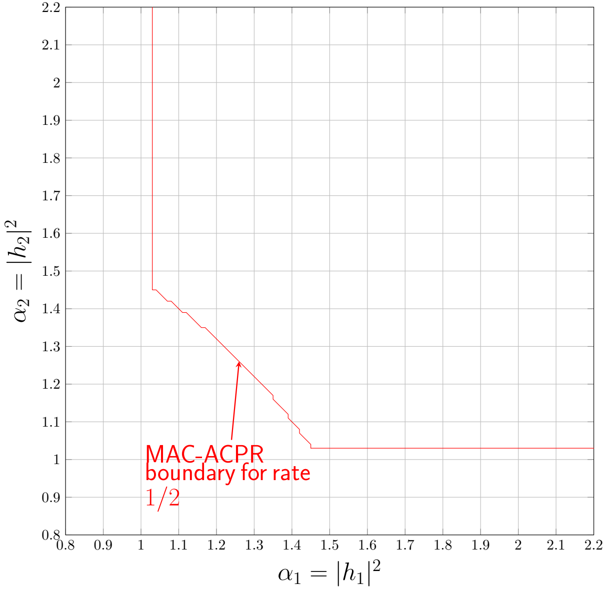
Spatially-Coupled Factor Graph for Joint Decoder



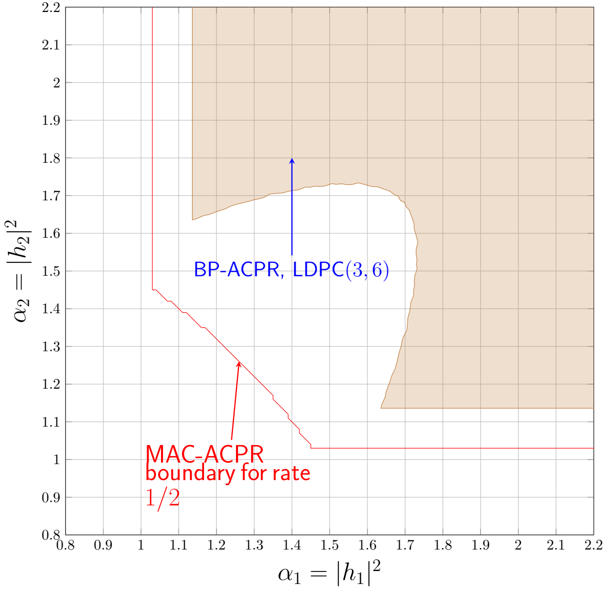
Spatially-Coupled Factor Graph for Joint Decoder



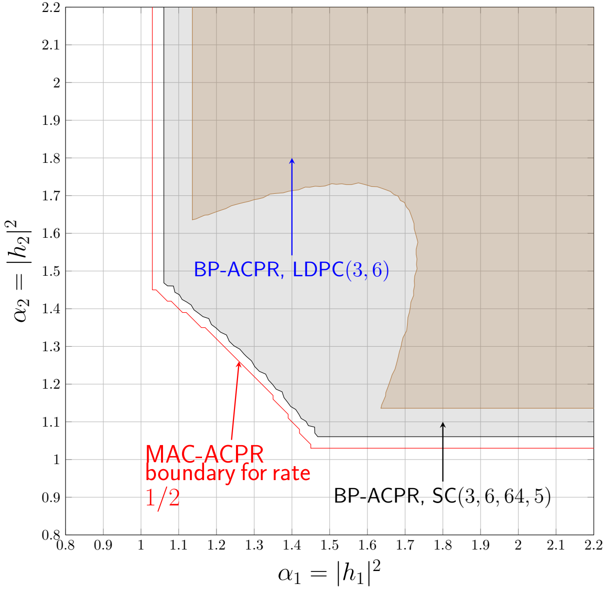
DE Performance of the Joint Decoder



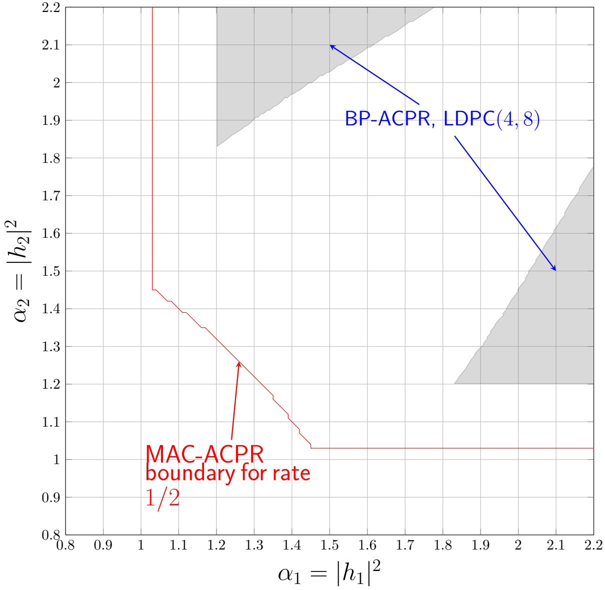
DE Performance of the Joint Decoder



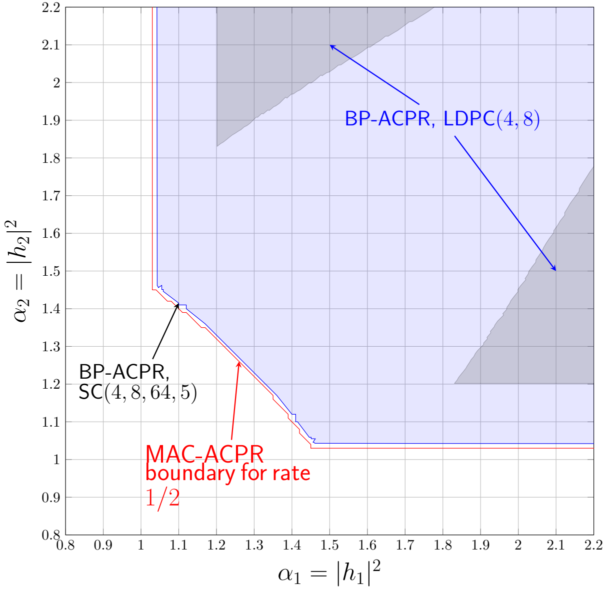
DE Performance of the Joint Decoder



DE Performance of the Joint Decoder



DE Performance of the Joint Decoder



Outline

Proving Threshold Saturation: General Approach

Let $f\colon \mathcal{X} \to \mathcal{X}$ and $g\colon \mathcal{X} \to \mathcal{X}$ be strictly increasing C^2 functions on $\mathcal{X} = [0,1]$. The scalar recursion (from $x^{(0)} = 1$)

$$y^{(\ell+1)} = g\left(x^{(\ell)}\right)$$
$$x^{(\ell+1)} = f\left(y^{(\ell+1)}\right)$$

Proving Threshold Saturation: General Approach

Let $f\colon \mathcal{X} \to \mathcal{X}$ and $g\colon \mathcal{X} \to \mathcal{X}$ be strictly increasing C^2 functions on $\mathcal{X} = [0,1]$. The scalar recursion (from $x^{(0)} = 1$)

$$y^{(\ell+1)} = g\left(x^{(\ell)}\right) = 1 - (1-x)^3$$
$$x^{(\ell+1)} = f\left(y^{(\ell+1)}\right) = \varepsilon x^2$$

Ex. (3,4) LDPC

Proving Threshold Saturation: General Approach

Let $f\colon \mathcal{X}\to\mathcal{X}$ and $g\colon \mathcal{X}\to\mathcal{X}$ be strictly increasing C^2 functions on $\mathcal{X}=[0,1]$. The scalar recursion (from $x^{(0)}=1$)

$$y^{(\ell+1)} = g\left(x^{(\ell)}\right) = 1-(1-x)^3$$
$$x^{(\ell+1)} = f\left(y^{(\ell+1)}\right) = \varepsilon x^2$$

Ex. (3,4) LDPC

characterizes fixed point of the coupled recursion ($x_i^{(0)}=1, i\in[N+w-1]$)

$$y_i^{(\ell+1)} = g\left(x_i^{(\ell)}\right)$$
$$x_i^{(\ell+1)} = \sum_{j=1}^{N+w-1} A_{j,i}\, f\left(\sum_{k=1}^{N} A_{j,k}\, y_k^{(\ell+1)}\right)$$
$$[A_{j,k}] = \boldsymbol{A} = \frac{1}{w}\begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & \ddots & 1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 1 & 1 & \cdots & 1 \end{bmatrix}$$

Proving Threshold Saturation: General Approach

Let $f\colon \mathcal{X} \to \mathcal{X}$ and $g\colon \mathcal{X} \to \mathcal{X}$ be strictly increasing C^2 functions on $\mathcal{X} = [0,1]$. The scalar recursion (from $x^{(0)} = 1$)

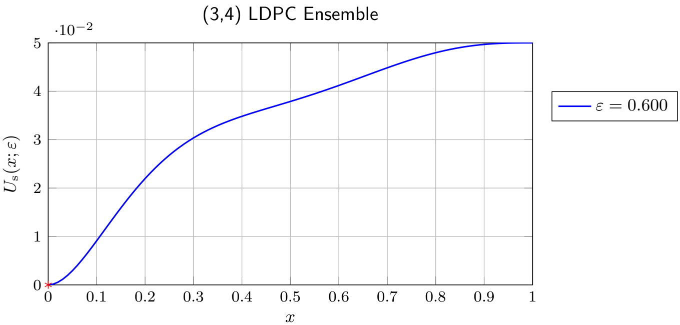
$$\begin{aligned} y^{(\ell+1)} &= g\left(x^{(\ell)}\right) = 1 - (1-x)^3 \\ x^{(\ell+1)} &= f\left(y^{(\ell+1)}\right) = \varepsilon x^2 \end{aligned}$$

Ex. (3,4) LDPC

characterizes fixed point of the coupled recursion ($\boldsymbol{x}^{(0)} = \mathbf{1}$)

$$\begin{aligned} \boldsymbol{y}^{(\ell+1)} &= \boldsymbol{g}\left(\boldsymbol{x}^{(\ell)}\right) \\ \boldsymbol{x}^{(\ell+1)} &= \boldsymbol{A}^\top \boldsymbol{f}\left(\boldsymbol{A}\, \boldsymbol{y}^{(\ell+1)}\right) \\ \boldsymbol{A} &= \frac{1}{w} \begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & \ddots & 1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 1 & 1 & \cdots & 1 \end{bmatrix} \end{aligned}$$

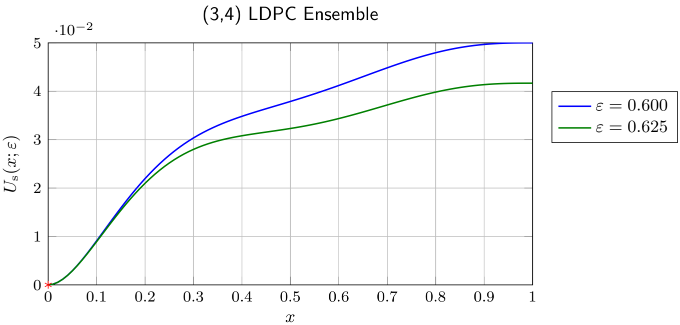
The Potential Function of the Scalar Recursion



Let the potential function $U_{\mathrm{s}} \colon \mathcal{X} \to \mathbb{R}$ of the scalar recursion be

$$U_{\mathrm{s}}(x) \triangleq \int_0^x \big(z - f\left(g(z)\right)\big) g'(z) \mathrm{d}z.$$

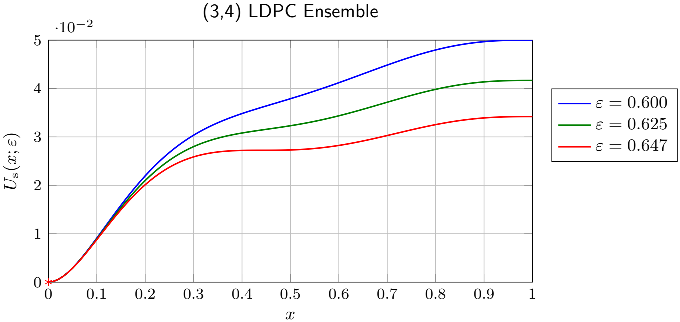
The Potential Function of the Scalar Recursion



Let the potential function $U_{\mathrm{s}} \colon \mathcal{X} \to \mathbb{R}$ of the scalar recursion be

$$U_{\mathrm{s}}(x) \triangleq \int_0^x \big(z - f\left(g(z)\right)\big) g'(z) \mathrm{d}z.$$

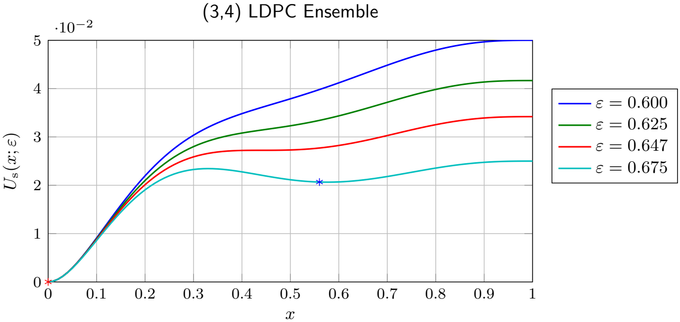
The Potential Function of the Scalar Recursion



Let the potential function $U_\mathrm{s} \colon \mathcal{X} \to \mathbb{R}$ of the scalar recursion be

$$U_\mathrm{s}(x) \triangleq \int_0^x \big(z - f\left(g(z)\right)\big) g'(z) \mathrm{d}z.$$

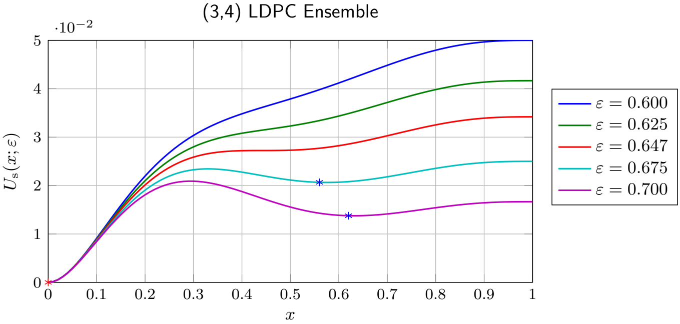
The Potential Function of the Scalar Recursion



Let the potential function $U_\mathrm{s} \colon \mathcal{X} \to \mathbb{R}$ of the scalar recursion be

$$U_\mathrm{s}(x) \triangleq \int_0^x \big(z - f\left(g(z)\right)\big) g'(z) \mathrm{d}z.$$

The Potential Function of the Scalar Recursion



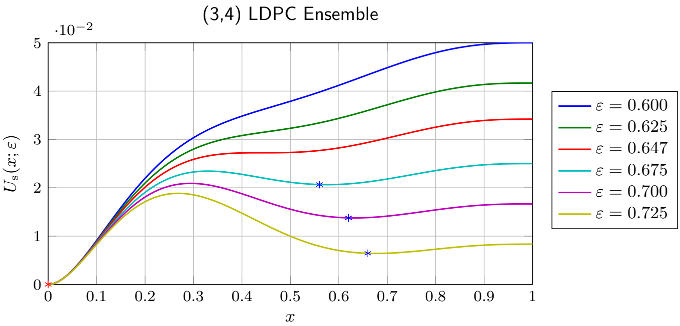
Let the potential function $U_s : \mathcal{X} \to \mathbb{R}$ of the scalar recursion be

$$U_s(x) \triangleq \int_0^x \big(z - f\left(g(z)\right)\big) g'(z) \mathrm{d}z.$$

Theorem (YJNP13) (arXiv:1309.7910)

$$\lim_{w\to\infty} \lim_{M\to\infty} \max_{i\in\{1,\ldots,M\}} x_i^{(\infty)} \leq \max\left(\arg\min_{x\in\mathcal{X}} U_s(x)\right)$$

The Potential Function of the Scalar Recursion



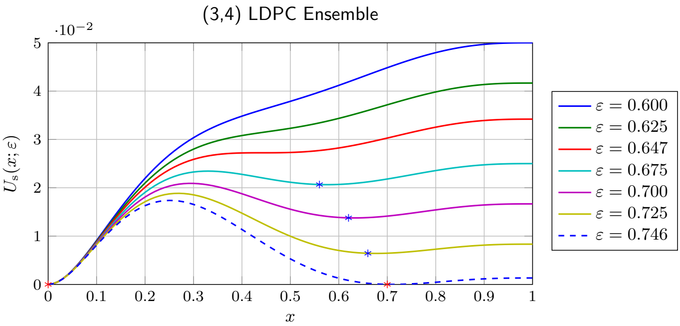
Let the potential function $U_s \colon \mathcal{X} \to \mathbb{R}$ of the scalar recursion be

$$U_s(x) \triangleq \int_0^x \big(z - f\left(g(z)\right)\big) g'(z) \mathrm{d}z.$$

Theorem (YJNP13) (arXiv:1309.7910)

$$\lim_{w\to\infty} \lim_{M\to\infty} \max_{i\in\{1,\ldots,M\}} x_i^{(\infty)} \leq \max\left(\arg\min_{x\in\mathcal{X}} U_s(x)\right)$$

The Potential Function of the Scalar Recursion



Let the potential function $U_{\mathrm{s}} \colon \mathcal{X} \to \mathbb{R}$ of the scalar recursion be

$$U_{\mathrm{s}}(x) \triangleq \int_0^x \big(z - f\left(g(z)\right)\big) g'(z) \mathrm{d}z.$$

Theorem (YJNP13) (arXiv:1309.7910)

$$\lim_{w\to\infty} \lim_{M\to\infty} \max_{i\in\{1,\ldots,M\}} x_i^{(\infty)} \le \max\left(\arg\min_{x\in\mathcal{X}} U_{\mathrm{s}}(x) \right)$$

Compressive Sensing (CS)

- Basic Idea
  - For a signal vector in $u \in \mathbb{R}^n$
  - Let $\Phi \in \mathbb{R}^{m \times n}$ be an $m \times n$ measurement matrix
  - Reconstruct u from the sample vector $v = \Phi u$ of length $m < n$

Compressive Sensing (CS)

- Basic Idea
  - For a signal vector in $u \in \mathbb{R}^n$
  - Let $\Phi \in \mathbb{R}^{m \times n}$ be an $m \times n$ measurement matrix
  - Reconstruct u from the sample vector $v = \Phi u$ of length $m < n$
- Details are skipped as people here are quite familiar with CS!
  - Brady et al. applied CS to spectral imaging and holography
  - Calderbank et al. designed fast deterministic measurement matrices
  - Carin et al. built Bayesian models for wavelet-sparse signals
  - Gehm et al. applied to CS tracking problems
  - Reeves et al. studied the limits of sparse support recovery
  - Sapiro et al. considered CS of Gaussian mixture models

Compressive Sensing (CS)

- Basic Idea
  - For a signal vector in $u \in \mathbb{R}^n$
  - Let $\Phi \in \mathbb{R}^{m \times n}$ be an $m \times n$ measurement matrix
  - Reconstruct u from the sample vector $v = \Phi u$ of length $m < n$
- Details are skipped as people here are quite familiar with CS!
  - Brady et al. applied CS to spectral imaging and holography
  - Calderbank et al. designed fast deterministic measurement matrices
  - Carin et al. built Bayesian models for wavelet-sparse signals
  - Gehm et al. applied to CS tracking problems
  - Reeves et al. studied the limits of sparse support recovery
  - Sapiro et al. considered CS of Gaussian mixture models
- My interest in CS is related to coding theory and factor graphs
  - Introduced (with Kudekar) first application of spatial-coupling to CS
  - The suboptimal decoders we analyzed showed moderate gains
  - Under GABP decoding, spatial coupling is nearly optimal

Spatially-Coupled (SC) Compressed Sensing

- Consider the compressive sensing reconstruction of a length-n signal
  - whose entries are i.i.d. copies of a r.v. X with $\mathbb{E}[X^2] < \infty$
  - from δn linear measurements with i.i.d. noise $Z \sim \mathcal{N}(0, \sigma^2)$
  - Assume SC measurements with chain length N and width w

- The MSE x^* for SC measurements with BP reconstruction [DJM11][KMSSZ11] satisfies (asymptotically for $M \gg w \to \infty$)

$$x^* \leq \max \left\{ \operatorname*{argmin}_{x \in \mathcal{X}} \left(-\frac{x}{\sigma^2 + \frac{1}{\delta} x} + \delta \ln \left(1 + \frac{x}{\delta \sigma^2} \right) \right.\right. \\ \left.\left. -2I \left(X; \sqrt{\frac{1}{\sigma^2}} X + Z \right) + 2I \left(X; \sqrt{\frac{1}{\sigma^2 + x/\delta}} X + Z \right) \right) \right\}$$

- RHS equals the replica method prediction for the optimal MSE

History of Threshold Saturation Proofs

- the BEC by KRU in 2010
  - Established many properties and tools used by later approaches
- the Curie-Weiss model in physics by HMU in 2010
- CDMA using a GA by TTK in 2011
- CDMA with outer code via GA by Truhachev in 2011
- compressive sensing using a GA by DJM in 2011
- regular codes on BMS channels by KRU in 2012
- increasing scalar and vector recursions by YJNP in 2012
- irregular LDPC codes on BMS channels by KYMP in 2012
- non-decreasing scalar recursions by KRU in 2012

Outline

Graphical Models and LDPC Codes

Spatially-Coupled Graphical Models

Universality for Multiuser Scenarios

General Formulation of Threshold Saturation

Wyner-Ziv and Gelfand-Pinsker

Conclusions

Rate-Distortion, Wyner-Ziv, and Gelfand-Pinsker

- Rate Distortion (RD) Problem
  - What is the minimum data rate to transmit a source with average distortion less than D?

Rate-Distortion, Wyner-Ziv, and Gelfand-Pinsker

- Rate Distortion (RD) Problem
  - What is the minimum data rate to transmit a source with average distortion less than D?
- Wyner-Ziv (WZ) Problem
  - WZ extends RD to the case of side-information at the decoder

Rate-Distortion, Wyner-Ziv, and Gelfand-Pinsker

- Rate Distortion (RD) Problem
  - What is the minimum data rate to transmit a source with average distortion less than D?
- Wyner-Ziv (WZ) Problem
  - WZ extends RD to the case of side-information at the decoder
- Gelfand-Pinsker (GP) Problem
  - Channel coding with non-causal side-information at transmitter

Rate-Distortion, Wyner-Ziv, and Gelfand-Pinsker

- Rate Distortion (RD) Problem
  - What is the minimum data rate to transmit a source with average distortion less than D?
- Wyner-Ziv (WZ) Problem
  - WZ extends RD to the case of side-information at the decoder
- Gelfand-Pinsker (GP) Problem
  - Channel coding with non-causal side-information at transmitter
- WZ and GP problems arise naturally in network information theory

Belief-Propagation Guided Decimation (BPGD)

- RD-type problems are challenging for graph codes with BP decoding
  - They require quantization of an arbitrary sequence to a codebook
  - BP converges only if received sequence is "close" to a codeword
  - But, vanishing fraction of total space is "close" to codewords

Belief-Propagation Guided Decimation (BPGD)

- RD-type problems are challenging for graph codes with BP decoding
  - They require quantization of an arbitrary sequence to a codebook
  - BP converges only if received sequence is "close" to a codeword
  - But, vanishing fraction of total space is "close" to codewords
- When the received vector is not "close" to a codeword
  - BP decoder typically converges to a non-informative fixed point
  - There are exponentially many codewords with low distortion
  - But, the decoder just cannot pick one
  - The bias of a bit is defined to be $|LLR| = \left|\ln \frac{P(X=0)}{P(X=1)}\right|$

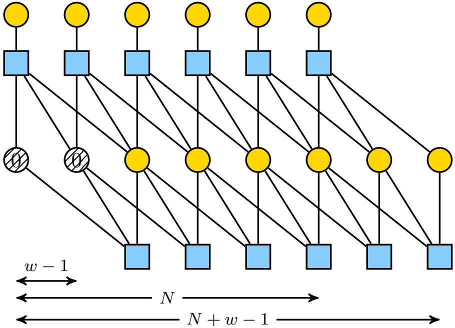
Belief-Propagation Guided Decimation (BPGD)

- RD-type problems are challenging for graph codes with BP decoding
  - They require quantization of an arbitrary sequence to a codebook
  - BP converges only if received sequence is "close" to a codeword
  - But, vanishing fraction of total space is "close" to codewords

- When the received vector is not "close" to a codeword
  - BP decoder typically converges to a non-informative fixed point
  - There are exponentially many codewords with low distortion
  - But, the decoder just cannot pick one
  - The bias of a bit is defined to be $|LLR| = \left|\ln \frac{P(X=0)}{P(X=1)}\right|$

- To force convergence, bits are sequentially "decimated":
  1. The BP decoder is run for a fixed number of iterations
  2. A bit with large bias is sampled and "decimated"

Once Again, Spatial-Coupling Comes to the Rescue



- Rate Distortion
  - SC low-density generator matrix (LDGM) codes can approach the RD limit with BPGD [AMUV12]
- Wyner-Ziv and Gelfand-Pinsker
  - SC compound LDGM/LDPC codes can approach the WZ/GP limits with BPGD [KVNP14]

Outline

Summary

- Spatial coupling
  - Powerful technique for designing and understanding factor graphs
  - Related to the statistical physics of supercooled liquids
  - General proof of threshold saturation for scalar recursions
  - For many multiuser problems, it provides universality
  - For RD/WZ/GP problems, it gives the only LDPC-based solutions

Thanks for your attention