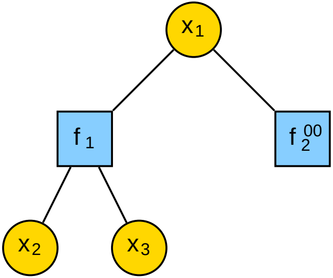
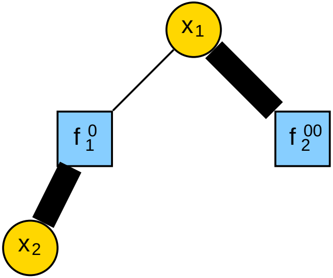
The Factor Graph and Leaf Removal



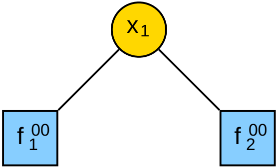
$$g_1(x_1) = \sum_{x_2}\left[\sum_{x_3} f_1(x_1;x_2;x_3)\right] f_2''(x_1)$$

The Factor Graph and Leaf Removal



$$g_1(x_1) = \left[\sum_{x_2} f_1'(x_1, x_2)\right] f_2''(x_1)$$

The Factor Graph and Leaf Removal



$$g_1(x_1) = f_1^{00}(x_1) f_2^{00}(x_1)$$

Constraint Satisfaction and Zero-One Factors

- A non-negative function $f : \mathcal{X}^n \to \mathbb{R}$ defines a distribution on $\mathcal{X}^n$:

$$P(\underline{x}) \triangleq P(X_1 = x_1, \ldots, X_n = x_n)$$
$$= \frac{1}{Z} f(\underline{x}) \triangleq \frac{1}{Z} \prod_{a=1}^{m} f_a(\underline{x}_{\partial a}),$$

  - where $\underline{x}_{\partial a}$ is the subvector of variables involved in factor a
  - and $Z \triangleq \sum_{\underline{x}} f(\underline{x})$ is called the partition function

Constraint Satisfaction and Zero-One Factors

- A non-negative function $f : \mathcal{X}^n \to \mathbb{R}$ defines a distribution on $\mathcal{X}^n$:

$$P(\underline{x}) \triangleq P(X_1 = x_1, \ldots, X_n = x_n)$$
$$= \frac{1}{Z} f(\underline{x}) \triangleq \frac{1}{Z} \prod_{a=1}^{m} f_a(\underline{x}_{\partial a}),$$

  - where $\underline{x}_{\partial a}$ is the subvector of variables involved in factor a
  - and $Z \triangleq \sum_{\underline{x}} f(\underline{x})$ is called the partition function

- For Constraint Satisfaction Problems (CSPs)
  - All factors $f_a(\underline{x}_{\partial a})$ take values in $\{0, 1\}$
  - The set of valid configurations is $\{\underline{x} \in \mathcal{X}^n \mid f(\underline{x}) = 1\}$
  - Thus, Z equals the number of valid configurations
  - $P(\underline{x})$ is uniform over the set of valid configurations

Outline

Introduction

Factor Graphs

Message Passing

Applications of Factor Graphs

Applications of EXIT Curves

Spatially-Coupled Factor Graphs

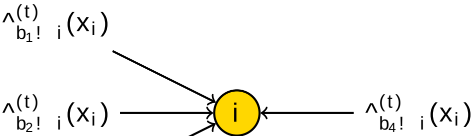
Universality for Multiuser Scenarios

Marginalization via Belief Propagation

- Factor Graph $G = (V \cup F, E)$
  - Variable nodes V, Factor nodes F
  - Edges: $(i, a) \in E \subseteq V \times F$
  - $\partial F(i)/\partial V(a) =$ set of neighbors for node i/a
  - Messages: $\nu_{i \to a}^{(t)}(x_i)$ and $\hat{\nu}_{a \to i}^{(t)}(x_i)$

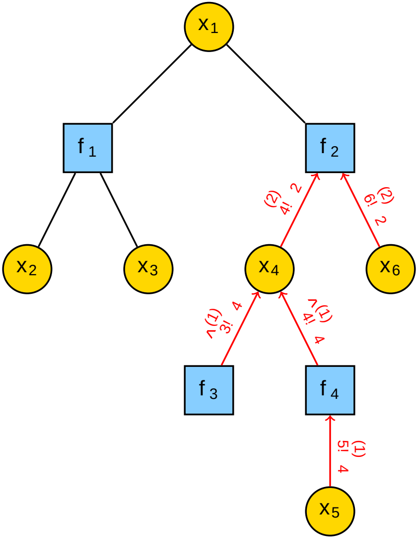
Marginalization via Belief Propagation

- Factor Graph $G = (V \cup F, E)$
  - Variable nodes V, Factor nodes F
  - Edges: $(i, a) \in E \subseteq V \times F$
  - $F(i)/V(a) =$ set of neighbors for node i/a
  - Messages: $\mu^{(t)}_{i \to a}(x_i)$ and $\hat{\mu}^{(t)}_{a \to i}(x_i)$
- variable-i marginal



$$\mu_i^{(t+1)}(x_i) = \prod_{b \in F(i)} \hat{\mu}^{(t)}_{b \to i}(x_i)$$

Marginalization via Belief Propagation: Example



iteration 1: factor to variable

$$\wedge^{(1)}_{4!\quad 4}(x_4) = \sum_{x_5} f_4(x_4; x_5)\ {}^{(1)}_{5!\quad 4}(x_i)$$

$$= \sum_{x_5} f_4(x_4; x_5)$$

$$\wedge^{(1)}_{3!\quad 4}(x_4) = f_3(x_4)$$

iteration 2: variable to factor

$$ {}^{(2)}_{4!\quad 2}(x_4) = \wedge\ {}^{(1)}_{4!\quad 4}(x_4)\wedge^{(1)}_{3!\quad 4}(x_4)$$

$$= f_3(x_4) \sum_{x_5} f_4(x_4; x_5)$$

$$ {}^{(2)}_{6!\quad 2}(x_6) = 1$$

Solving Sudoku with a Factor Graph

- Consider any constraint satisfaction problem with observed 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 with a Factor Graph

- Consider any constraint satisfaction problem with observed 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

- Low-complexity peeling solution
  - Set elements of $\underline{x}$ one at a time
  - Each step looks for $i \in [n]$ and $x' \in \mathcal{X}$ such that:
    - For currently set variables, $f_i(\underline{x}) = 0$ for all 0f2 ,16.113 678 Tf 8.717 2

Boolean Satisfiability: K-SAT

- One instance of 3-SAT is given, for example, by

$$f(\underline{x}) = (\bar{x}_1 \vee \bar{}$$

Boolean Satisfiability: K-SAT

- One instance of 3-SAT is given, for example, by

$$f(\underline{x}) = (\bar{x}_1 \vee \bar{x}_3 \vee x_7) \wedge (x_1 \vee \bar{x}_2 \vee x_5) \wedge (x_2 \vee \bar{x}_4 \vee x_6).$$

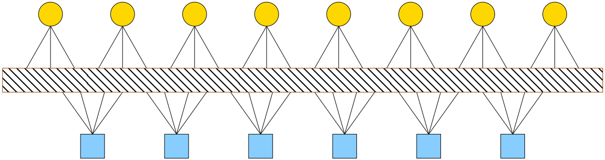
  - In the FG, clause $a \in [m]$ is enforced by the function f_a

- Marginalization allows uniform sampling from valid set
  - For $i = 1, 2, \ldots, n$, fix x_j for $j < i$ and compute marginal

$$g_i(x_i) = \frac{1}{Z_i} \sum_{x_{i+1}, \ldots, x_n} f(\underline{x}) = P(X_i = x_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1})$$

  - Then, sample $x_i \sim g_i(\cdot)$ and repeat

Low-Density Parity-Check (LDPC) Codes

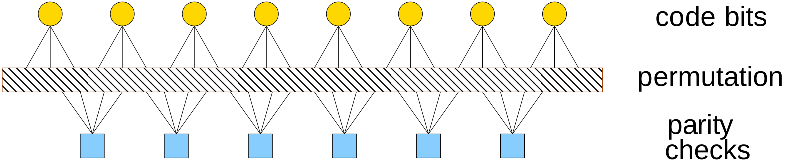


code bits

permutation

parity checks

Low-Density Parity-Check (LDPC) Codes

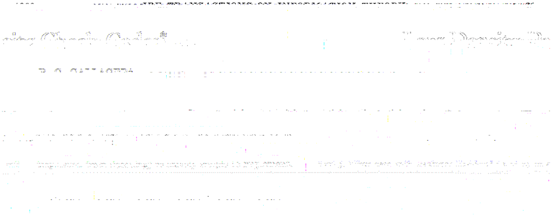


- Linear codes defined by $\underline{x} H^T = \underline{0}$ for all c.w. $\underline{x} \in C \subseteq \{0,1\}^n$
    - H is an $m \times n$ sparse parity-check matrix for the code
    - Code bits and parity checks associated with cols/rows of H
- Factor graph: H is the biadjacency matrix for variable/factor 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 $\underline{x}_{\partial a}$ be the subvector of variables in the a-th check and

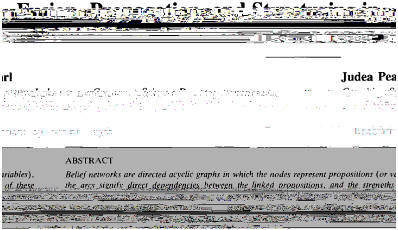
$$f(x_1, \ldots, x_n) = \left(\prod_{a=1}^{m} f_{\text{even}}(\underline{x}_{\partial a}) \right) \left(\prod_{i=1}^{n} P_{Y|X}(y_i|x_i) \right)$$

A Little History

Robert Gallager introduced LDPC codes in 1962 paper



Judea Pearl defined general belief-propagation in 1986 paper



rl

Judea Pea

ABSTRACT

riables),
of these

Belief networks are directed acyclic graphs in which the nodes represent propositions (or v
the arcs signify direct dependencies between the linked propositions, and the strengths

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

Simple Message-Passing Decoding for the BEC

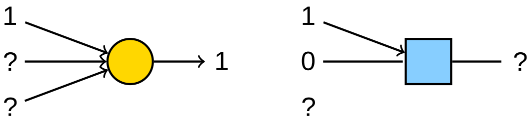
- Constraint nodes define the valid patterns
  - Circles represent a single value shared by factors
  - Squares assert attached variables sum to 0 mod 2
- Iterative decoding on the binary erasure channel (BEC)
  - Messages passed in phases: bit-to-check and check-to-bit
  - Each output message depends 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
  -

Simple Message-Passing Decoding for the BEC

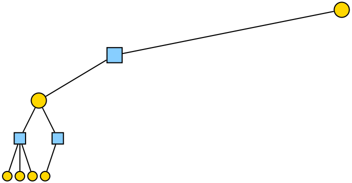
- Constraint nodes define the valid patterns
  - Circles represent a single value shared by factors
  - Squares assert attached variables sum to 0 mod 2
- Iterative decoding on the binary erasure channel (BEC)
  - Messages passed in phases: bit-to-check and check-to-bit
  - Each output message depends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

Ch

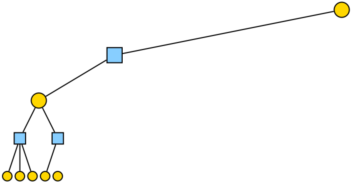
- Message passing rules for the BEC
  - Bits pass an erasure only if all other inputs are erased
  - Checks pass the correct value only if all other inputs are correct



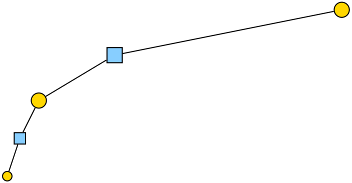
Computation Graph and Density Evolution



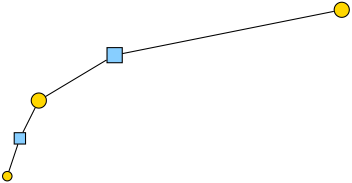
Computation Graph and Density Evolution



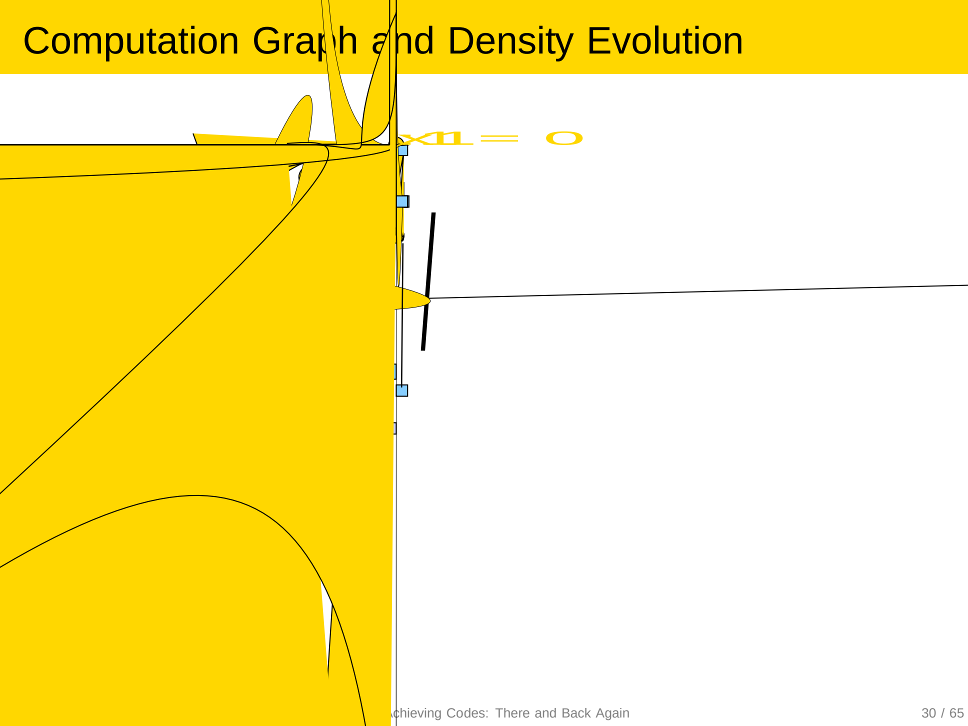
Computation Graph and Density Evolution



Computation Graph and Density Evolution



Computation Graph and Density Evolution



EXtrinsic Information Transfer (EXIT) Curves

- Introduced by ten Brink in 1999 to understand iterative decoding
  - For the BEC, the MAP EXIT curve is

$$h^{\mathrm{MAP}}(\epsilon) \triangleq \frac{1}{n}\sum_{i=1}^{n} H(X_i \mid \underline{Y}_{\sim i}(\epsilon))$$

EXtrinsic Information Transfer (EXIT) Curves

- Introduced by ten Brink in 1999 to understand iterative decoding
    - For the BEC, the MAP EXIT curve is

$$h^{\mathrm{MAP}}(\epsilon) \triangleq \frac{1}{n}\sum_{i=1}^{n} H(X_i \mid \underline{Y}_{\sim i}(\epsilon))$$

    - EXIT Area Theorem [ABK04]

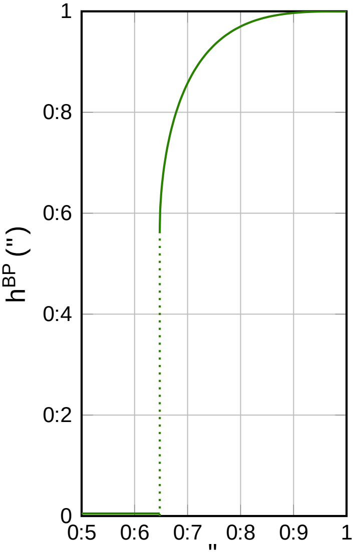
$$\frac{1}{n} H(\underline{X} \mid \underline{Y}(\epsilon)) = \int_0^{\epsilon} h^{\mathrm{MAP}}(\delta)\, d\delta$$

    - BP EXIT curve

$$h^{\mathrm{BP}}(\epsilon) \triangleq \frac{1}{n}\sum_{i=1}^{n} H\left(X_i \mid \phi_i^{\mathrm{BP}}(\underline{Y}_{\sim i}(\epsilon))\right)$$

        - where $\phi_i^{\mathrm{BP}}(Z)$ is the BP estimate of X_i given Z
        - Data processing inequality: $h^{\mathrm{BP}}(\epsilon) \geq h^{\mathrm{MAP}}(\epsilon)$

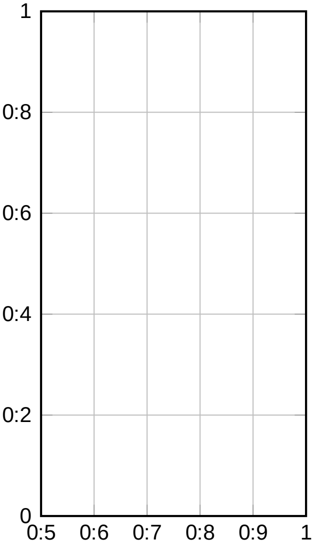
EXtrinsic Information Transfer (EXIT) Curves



- (3,4)-regular LDPC code
  - Codeword $(X_1, \ldots, X_n)$
  - Received $(Y_1, \ldots, Y_n)$

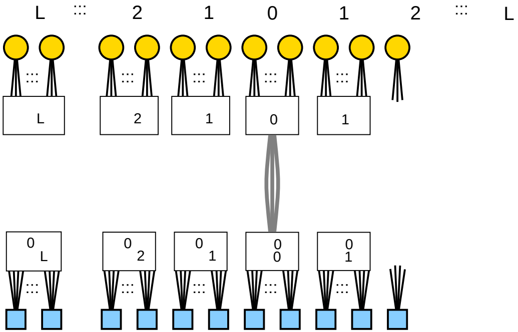
- BP EXIT curve via DE
  - This code: $h^{BP}(\epsilon) = (\;x_\infty(\epsilon))^3$
  - 0 below BP threshold 0.647

EXtrinsic Information Transfer (EXIT) Curves



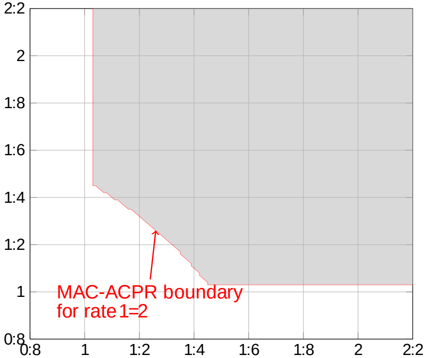
What is Spatial Coupling?

Spatially-Coupled LDPC Codes: (l; r; L; w) Ensemble



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

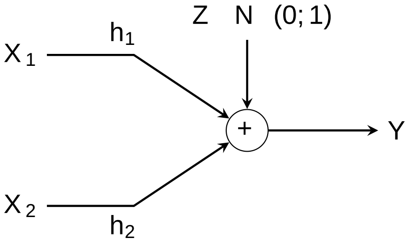


1

Universality over Unknown Parameters

- The Achievable Channel Parameter Region (ACPR)
    - For a sequence of coding schemesg arameters

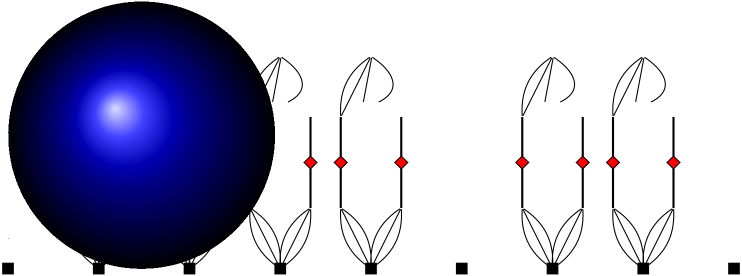
2-User Binary-Input Gaussian Multiple Access Channel



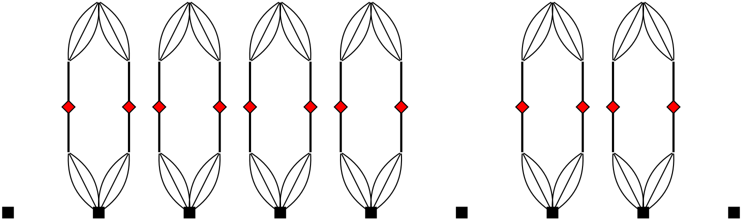
- Fixed noise variance
- Real channel gains h_1 and h_2 not known at transmitter
- Each code has rate R

- MAC-ACPR denotes the information-theoretic optimal region

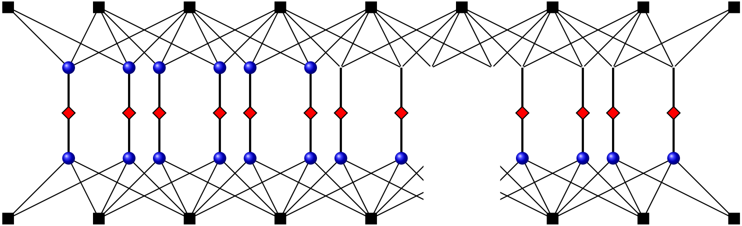
Spatially-Coupled Factor Graph for Joint Decoder



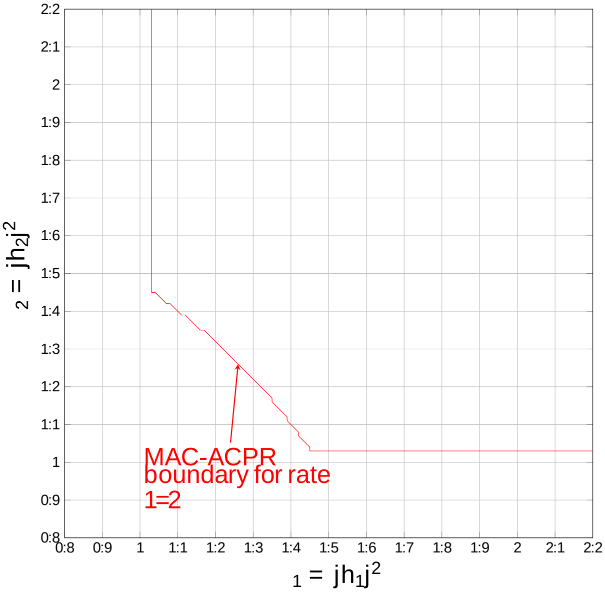
Spatially-Coupled Factor Graph for Joint Decoder



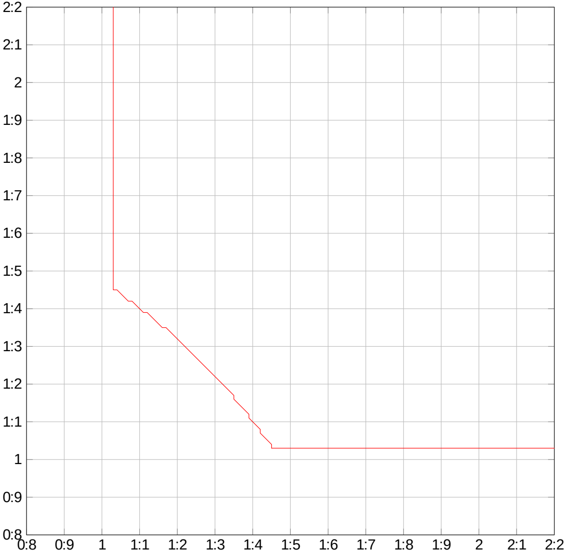
Spatially-Coupled Factor Graph for Joint Decoder



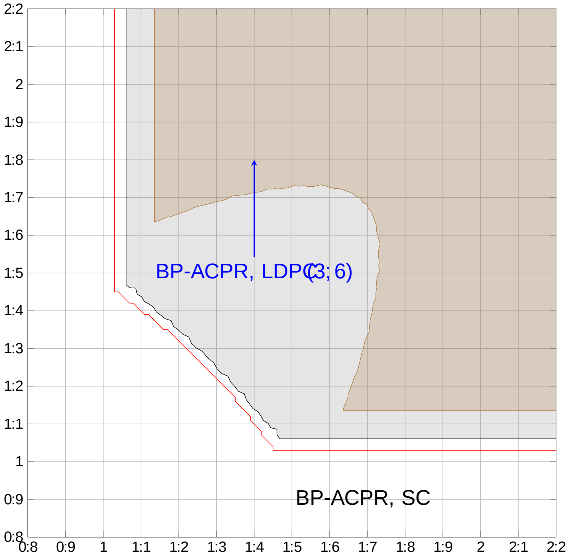
DE Performance of the Joint Decoder



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

Introduction

Factor Graphs

Message Passing

Applications of Factor Graphs

Applications of EXIT Curves

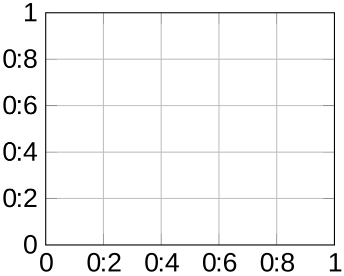
Spatially-Coupled Factor Graphs

Universality for Multiuser Scenarios

Abstract Formulation of Threshold Saturation

Single Monotone Recursion

- Smooth increasing $g : [0,1] \to [0,1]$



Single Monotone Recursion

- Smooth increasing $f : [0,1] \to [0,1]$
- Discrete-time recursion

Single Monotone Recursion

- Smooth increasing $f:[0,1]\to[0,1]$
- Discrete-time recursion

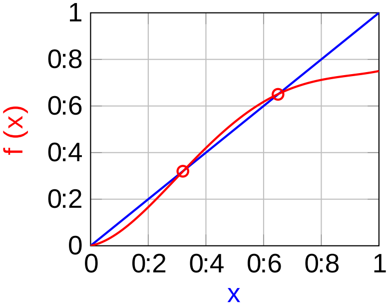
$$x^{(\ell+1)} = f(x^{(\ell)})$$

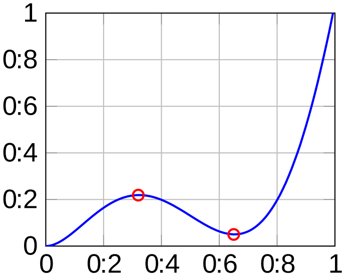
- "Potential energy" $U_s(x)$

$$U_s(x) = \int_0^x z - f(z)\,dz = \frac{x^2}{2} - F(x)$$

- Continuous (small step) dynamics

$$\frac{d}{dt}x(t) = f\big(x(t)\big) - x(t) = -\nabla U_s\big(x(t)\big)$$





Single Monotone Recursion

- Smooth increasing $f : [0,1] \to [0,1]$
- Discrete-time recursion

$$x^{(\ell+1)} = f(x^{(\ell)})$$

- "Potential energy" $U_s(x)$

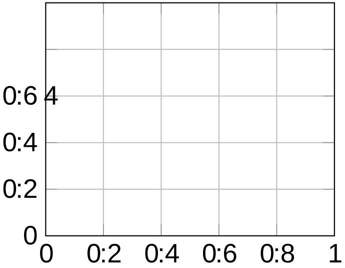
$$U_s(x) = \int_0^x \big(z - f(z)\big)\, dz = \frac{x^2}{2} - F(x)$$

- Continuous (small step) dynamics

$$\frac{d}{dt}x(t) = f\big(x(t)\big) - x(t) = -\nabla U_s\big(x(t)\big)$$

- Lyapunov stability

$$\frac{d}{dt}U_s\big(x(t)\big) = -\big(x(t) - f(x(t))\big)^2$$



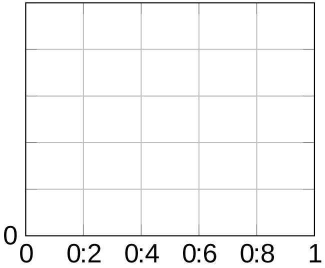
Single Monotone Recursion

- Smooth increasing $f : [0,1] \to [0,1]$
- Discrete-time recursion

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

- “Potential energy” $U_s(x)$

$$U_s(x) = \int_0^x z \quad f(z)\; dz = \frac{x^2}{2} \quad F(x)$$



- Continuous (small step) dynamics

$$\frac{d}{dt}x(t) = f \;\; x(t) \quad x(t) = \;\; \nabla \;\; U_s \; x(t)$$

- Lyapunov stability

$$\frac{d}{dt}U_s \; x(t) \;\; = \qquad x(t) \quad f(x(t)) \;\; ^{2}$$

Both $\downarrow 0$ iff no fixed points in $(0,1]$

Coupled Monotone Recursion (1)

- Coupled recursion $\underline{x}^{(\ell+1)} = T\underline{x}^{(\ell)}$ with $\underline{x}^{(\ell)} = \left(x_0^{(\ell)}, x_1^{(\ell)}, \ldots\right)$ and

$$T\underline{x} \triangleq A^{\top} \underline{f}\left(A\underline{x}\right),$$

where $[\underline{f}(\underline{x})]_i = f(x_i)$ and A averages w adjacent values

$$A = \frac{1}{w}\begin{bmatrix} 1 & 1 & & 1 & 0 & \\ 0 & 1 & 1 & \ddots & 1 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

  - i.e., avg right w positions, apply f, then avg left w positions

- Coupled potential: U_c-1.245 Td d67 115ei.

Coupled Monotone Recursion (2)

- Properties of T (note: $\underline{x} \quad \underline{y}$, $\quad x_i \quad y_i$ for all i)
  - T is monotone: $\underline{x} \quad \underline{y}$ implies $T\underline{x} \quad T\underline{y}$
  - T preserves spatial order: $x_{i+1} \quad x_i$ implies $[T\underline{x}]_{i+1} \quad [T\underline{x}]_i$