Statistical Physics of Algorithms
(subjective mini-course)

Michael Chertkov

Center for Nonlinear Studies & Theory Division, LANL

March 3-5, 2008 KITP-China

Books, Reviews, Papers

No perfect book on the subject, yet

Good books on related subjects

- David J. C. MacKay, *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, 2003
- Marc Mezard & Anrea Montanari, *Information, Physics and Computation*, in progress see Mezard's webpage
- Tom Richardson, Rüdiger Urbanke, *Modern Coding Theory* Cambridge University Press, 2005
- Alexander K. Hartmann, Heiko Rieger, *Optimization Algorithms in Physics*, Wiley-VCH, 2002

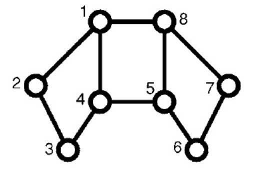
Many recent research papers, and few reviews scattered over Physics, Computer Science and Information Theory journals

Boolean Graphical Models = The Language

Forney style - variables on the edges

$$\mathcal{P}(\vec{\sigma}) = Z^{-1}\prod_a f_a(\vec{\sigma}_a)$$

$$\underbrace{Z = \sum_{\boldsymbol{\sigma}}\prod_a f_a(\vec{\sigma}_a)}_{\text{partition function}}$$



$$f_a \geq 0$$

$$\sigma_{ab} = \sigma_{ba} = \pm 1$$

$$\vec{\sigma}_1 = (\sigma_{12}, \sigma_{14}, \sigma_{18})$$

$$\vec{\sigma}_2 = (\sigma_{12}, \sigma_{23})$$

Objects of Interest

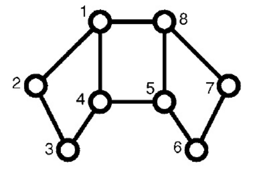
- Most Probable Configuration = Maximum Likelihood = Ground State: $\arg\max \mathcal{P}(\vec{\sigma})$
- Marginal Probability: e.g. $\mathcal{P}(\sigma_{ab}) \equiv \sum_{\vec{\sigma}\setminus\sigma_{ab}} \mathcal{P}(\vec{\sigma})$
- Partition Function: Z

Boolean Graphical Models = The Language

Forney style - variables on the edges

$$\mathcal{P}(\vec{\sigma}) = Z^{-1} \prod_a f_a(\vec{\sigma}_a)$$

$$\underbrace{Z = \sum_{\boldsymbol{\sigma}} \prod_a f_a(\vec{\sigma}_a)}_{\text{partition function}}$$



$$f_a \geq 0$$

$$\sigma_{ab} = \sigma_{ba} = \pm 1$$

$$\vec{\sigma}_1 = (\sigma_{12}, \sigma_{14}, \sigma_{18})$$

$$\vec{\sigma}_2 = (\sigma_{12}, \sigma_{23})$$

Objects of Interest

- Most Probable Configuration = Maximum Likelihood = Ground State: $\arg\max \mathcal{P}(\vec{\sigma})$
- Marginal Probability: e.g. $\mathcal{P}(\sigma_{ab}) \equiv \sum_{\vec{\sigma} \setminus \sigma_{ab}} \mathcal{P}(\vec{\sigma})$
- Partition Function: Z

Example (1): Statistical Physics

Ising model $\sigma_i = \pm 1$

$$\mathcal{P}(\vec{\sigma}) = Z^{-1} \exp\left(\sum_{(i,j)} J_{ij}\sigma_i\sigma_j\right)$$

J_{ij} defines the graph (lattice)

Graphical Representation

Variables are usually associated with vertexes ... but transformation to the Forney graph (variables on the edges) is straightforward

- Ferromagnetic ($J_{ij} < 0$), Anti-ferromagnetic ($J_{ij} > 0$) and Frustrated/Glassy
- Magnetization (order parameter) and Ground State
- Thermodynamic Limit, $N \to \infty$
- Phase Transitions

Example (2): Information Theory, Machine Learning, etc

Probabilistic Reconstruction (Statistical Inference)

$\vec{\sigma}_{\rm orig}$	$\Rightarrow$	$\vec{x}$	$\Rightarrow$	$\vec{\sigma}$
original *data* $\vec{\sigma}_{\rm orig}\in\mathcal{C}$ codeword	noisy channel $\mathcal{P}(\vec{x}\vert\vec{\sigma})$	corrupted data: log-likelihood magnetic field	statistical inference	possible preimage $\vec{\sigma}\in\mathcal{C}$

Maximum Likelihood [ground state]

Marginalization

$$\mathrm{ML}(\vec{x}) = \arg\max_{\vec{\sigma}} \mathcal{P}(\vec{x}|\vec{\sigma}) \qquad \sigma_i^*(\vec{x}) = \arg\max_{\sigma_i} \sum_{\vec{\sigma}\setminus\sigma_i} \mathcal{P}(\vec{x}|\vec{\sigma})$$

forward error correction - to be discussed later

Example (2): Information Theory, Machine Learning, etc

Probabilistic Reconstruction (Statistical Inference)

$\vec{\sigma}_{\text{orig}}$	$\Rightarrow$	$\vec{x}$	$\Rightarrow$	$\vec{\sigma}$
original *data* $\vec{\sigma}_{\text{orig}} \in \mathcal{C}$ codeword	noisy channel $\mathcal{P}(\vec{x}\|\vec{\sigma})$	corrupted data: log-likelihood magnetic field	statistical inference	possible preimage $\vec{\sigma} \in \mathcal{C}$

Maximum Likelihood [ground state]

Marginalization

$$\text{ML}(\vec{x}) = \arg\max_{\vec{\sigma}} \mathcal{P}(\vec{x}|\vec{\sigma}) \qquad \sigma_i^*(\vec{x}) = \arg\max_{\sigma_i} \sum_{\vec{\sigma}\setminus\sigma_i} \mathcal{P}(\vec{x}|\vec{\sigma})$$

forward error correction - to be discussed later

Example (2): Information Theory, Machine Learning, etc

Probabilistic Reconstruction (Statistical Inference)

$\vec{\sigma}_{\text{orig}}$	$\Rightarrow$	$\vec{x}$	$\Rightarrow$	$\vec{\sigma}$
original *data* $\vec{\sigma}_{\text{orig}} \in \mathcal{C}$ codeword	noisy channel $\mathcal{P}(\vec{x}\vert\vec{\sigma})$	corrupted data: log-likelihood magnetic field	statistical inference	possible preimage $\vec{\sigma} \in \mathcal{C}$

Maximum Likelihood [ground state]

Marginalization

$$\text{ML}(\vec{x}) = \arg\max_{\vec{\sigma}} \mathcal{P}(\vec{x}|\vec{\sigma}) \qquad \sigma_i^*(\vec{x}) = \arg\max_{\sigma_i} \sum_{\vec{\sigma}\setminus\sigma_i} \mathcal{P}(\vec{x}|\vec{\sigma})$$

forward error correction - to be discussed later

Example (2): Information Theory, Machine Learning, etc

Probabilistic Reconstruction (Statistical Inference)

$\vec{\sigma}_{\text{orig}}$	$\Rightarrow$	$\vec{x}$	$\Rightarrow$	$\vec{\sigma}$
original *data* $\vec{\sigma}_{\text{orig}} \in \mathcal{C}$ codeword	noisy channel $\mathcal{P}(\vec{x}\vert\vec{\sigma})$	corrupted data: log-likelihood magnetic field	statistical inference	possible preimage $\vec{\sigma} \in \mathcal{C}$

Maximum Likelihood [ground state]

Marginalization

$$\text{ML}(\vec{x}) = \arg\max_{\vec{\sigma}} \mathcal{P}(\vec{x}|\vec{\sigma})$$

$$\sigma_i^*(\vec{x}) = \arg\max_{\sigma_i} \sum_{\vec{\sigma}\setminus\sigma_i} \mathcal{P}(\vec{x}|\vec{\sigma})$$

forward error correction - to be discussed later

Example (3): Combinatorial Optimization, K-SAT

$$F(\vec{x}) = \begin{array}{l} (x_1 \vee x_2 \vee \bar{x}_3)\wedge \\ (x_5 \vee \bar{x}_1 \vee \bar{x}_4)\wedge \\ (x_2 \vee x_7 \vee x_3)\wedge \\ (\bar{x}_7 \vee x_5 \vee \bar{x}_5)\wedge \\ \cdots \end{array}$$

$1, 2, \cdots, N$ – *variables*

$F(\vec{x})$ is a conjunction of M clauses

$x_i = 0(\text{bad}), 1(\text{good})$

$\bar{x}_i$ is negation of x_i

$\vee = \text{OR} \quad \wedge = \text{AND}$

$\vec{x}$ is a "valid assignment" if $F(\vec{x}) = 1$

Probabilistic interpretation

$$P(\vec{x}) = Z^{-1} F(\vec{x}), \quad Z \equiv \sum_{\vec{x}} F(\vec{x})$$

- Finding a Valid Assignment, Counting Number of Assignments
- Graphical Representation, Sparseness
- Random, non-Random formulas
- SAT/UNSAT transition wrt $\alpha = M/N$, $M, N \to \infty$

Complexity & Algorithms

- How many operations are required to evaluate a graphical model of size N?
- What is the exact algorithm with the least number of operations?
- If one is ready to trade optimality for efficiency, what is the best (or just good) approximate algorithm he/she can find for a given (small) number of operations?
- Given an approximate algorithm, how to decide if the algorithm is good or bad? What is the measure of success?
- How one can systematically improve an approximate algorithm?

- Linear (or Algebraic) in N is EASY, Exponential is DIFFICULT

Complexity & Algorithms

- How many operations are required to evaluate a graphical model of size N?
- What is the exact algorithm with the least number of operations?
- If one is ready to trade optimality for efficiency, what is the best (or just good) approximate algorithm he/she can find for a given (small) number of operations?
- Given an approximate algorithm, how to decide if the algorithm is good or bad? What is the measure of success?
- How one can systematically improve an approximate algorithm?

- Linear (or Algebraic) in N is EASY, Exponential is DIFFICULT

Complexity & Algorithms

- How many operations are required to evaluate a graphical model of size N?
- What is the exact algorithm with the least number of operations?
- If one is ready to trade optimality for efficiency, what is the best (or just good) approximate algorithm he/she can find for a given (small) number of operations?
- Given an approximate algorithm, how to decide if the algorithm is good or bad? What is the measure of success?
- How one can systematically improve an approximate algorithm?

- Linear (or Algebraic) in N is EASY, Exponential is DIFFICULT

Complexity & Algorithms

- How many operations are required to evaluate a graphical model of size N?
- What is the exact algorithm with the least number of operations?
- If one is ready to trade optimality for efficiency, what is the best (or just good) approximate algorithm he/she can find for a given (small) number of operations?
- Given an approximate algorithm, how to decide if the algorithm is good or bad? What is the measure of success?
- How one can systematically improve an approximate algorithm?

- Linear (or Algebraic) in N is EASY, Exponential is DIFFICULT

Complexity & Algorithms

- How many operations are required to evaluate a graphical model of size N?
- What is the exact algorithm with the least number of operations?
- If one is ready to trade optimality for efficiency, what is the best (or just good) approximate algorithm he/she can find for a given (small) number of operations?
- Given an approximate algorithm, how to decide if the algorithm is good or bad? What is the measure of success?
- How one can systematically improve an approximate algorithm?

- Linear (or Algebraic) in N is EASY, Exponential is DIFFICULT

Complexity & Algorithms

- How many operations are required to evaluate a graphical model of size N?
- What is the exact algorithm with the least number of operations?
- If one is ready to trade optimality for efficiency, what is the best (or just good) approximate algorithm he/she can find for a given (small) number of operations?
- Given an approximate algorithm, how to decide if the algorithm is good or bad? What is the measure of success?
- How one can systematically improve an approximate algorithm?

- Linear (or Algebraic) in N is EASY, Exponential is DIFFICULT

Easy & Difficult Boolean Problems

EASY

- Any graphical problems on a tree (Bethe-Pieirls, dynamical programming, belief propagation, and other names)
- Ground State of a Rand. Field Ferrom. Ising model on any graph
- Partition function of a planar Ising model
- Finding if 2-SAT is satisfiable
- Decoding over Binary Erasure Channel = XOR-SAT
- Some network flow problems (max-flow, min-cut, shortest path, etc)
- Minimal Perfect Matching Problem
- Some special cases of Integer Programming (TUM)

Typical graphical problem, with loops and factor functions of a general position, is DIFFICULT

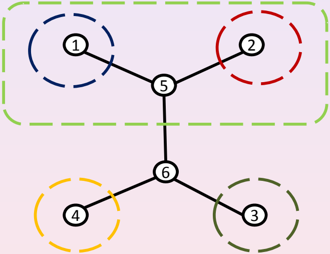
Easy & Difficult Boolean Problems

EASY

- Any graphical problems on a tree (Bethe-Pieirls, dynamical programming, belief propagation, and other names)
- Ground State of a Rand. Field Ferrom. Ising model on any graph
- Partition function of a planar Ising model
- Finding if 2-SAT is satisfiable
- Decoding over Binary Erasure Channel = XOR-SAT
- Some network flow problems (max-flow, min-cut, shortest path, etc)
- Minimal Perfect Matching Problem
- Some special cases of Integer Programming (TUM)

Typical graphical problem, with loops and factor functions of a general position, is DIFFICULT

BP is Exact on a Tree — Bethe '35, Pieirls '36



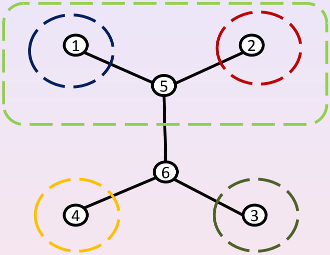
$$Z_{15}(\sigma_{15}) = f_1(\sigma_{15}), \quad Z_{25}(\sigma_{25}) = f_2(\sigma_{25}),$$

$$Z_{36}(\sigma_{36}) = f_3(\sigma_{36}), \quad Z_{46}(\sigma_{46}) = f_4(\sigma_{46})$$

$$Z_{56}(\sigma_{56}) = \sum_{\vec{\sigma}_5 \setminus \sigma_{56}} f_5(\vec{\sigma}_5) Z_{15}(\sigma_{15}) Z_{25}(\sigma_{25})$$

$$Z = \sum_{\vec{\sigma}_6} f_6(\vec{\sigma}_6) Z_{36}(\sigma_{36}) Z_{46}(\sigma_{46}) Z_{56}(\sigma_{56})$$

BP is Exact on a Tree — Bethe '35, Pieirls '36



$$Z_{15}(\sigma_{15}) = f_1(\sigma_{15}), \quad Z_{25}(\sigma_{25}) = f_2(\sigma_{25}),$$
$$Z_{36}(\sigma_{36}) = f_3(\sigma_{36}), \quad Z_{46}(\sigma_{46}) = f_4(\sigma_{46})$$
$$Z_{56}(\sigma_{56}) = \sum_{\vec{\sigma}_5 \setminus \sigma_{56}} f_5(\vec{\sigma}_5) Z_{15}(\sigma_{15}) Z_{25}(\sigma_{25})$$
$$Z = \sum_{\vec{\sigma}_6} f_6(\vec{\sigma}_6) Z_{36}(\sigma_{36}) Z_{46}(\sigma_{46}) Z_{56}(\sigma_{56})$$

$$Z_{ba}(\sigma_{ab}) = \sum_{\vec{\sigma}_a \setminus \sigma_{ab}} f_a(\vec{\sigma}_a) Z_{ac}(\sigma_{ac}) Z_{ad}(\sigma_{ad}) \Rightarrow Z_{ab}(\sigma_{ab}) = A_{ab} \exp(\eta_{ab}\sigma_{ab})$$

Belief Propagation Equations

$$\sum_{\vec{\sigma}_a} f_a(\vec{\sigma}_a) \exp\Big(\sum_{c \in a} \eta_{ac}\sigma_{ac}\Big) \left(\sigma_{ab} - \tanh\left(\eta_{ab} + \eta_{ba}\right)\right) = 0$$

Variational Method in Statistical Mechanics

$$P(\vec{\sigma}) = \frac{\prod_a f_a(\vec{\sigma}_a)}{Z}, \quad Z \equiv \sum_{\vec{\sigma}} \prod_a f_a(\vec{\sigma}_a)$$

Exact Variational Principe

J.W. Gibbs 1903 (or earlier)

also known as Kullback-Leibler (1951) in CS and IT

$$F\{b(\vec{\sigma})\} = -\sum_{\vec{\sigma}} b(\vec{\sigma}) \sum_a \ln f_a(\vec{\sigma}_a) + \sum_{\vec{\sigma}} b(\vec{\sigma}) \ln b(\vec{\sigma})$$

$$\left.\frac{\delta F}{\delta b(\vec{\sigma})}\right|_{b(\vec{\sigma})=p(\vec{\sigma})} = 0 \quad \text{under} \quad \sum_{\vec{\sigma}} b(\vec{\sigma}) = 1$$

Variational Ansatz

- Mean-Field: $p(\vec{\sigma}) \approx b(\vec{\sigma}) = \prod_{(a,b)} b_{ab}(\sigma_{ab})$
- Belief Propagation:

$$p(\vec{\sigma}) \approx b(\vec{\sigma}) = \frac{\prod_a b_a(\vec{\sigma}_a)}{\prod_{(a,b)} b_{ab}(\sigma_{ab})} \quad \text{(exact on a tree)}$$

$$\forall\ a;\ c \in a: \quad \sum_{\vec{\sigma}_a} b_a(\vec{\sigma}_a) = 1, \quad b_{ac}(\sigma_{ac}) = \sum_{\vec{\sigma}_a \setminus \sigma_{ac}} b_a(\vec{\sigma}_a)$$

Variational Method in Statistical Mechanics

$$P(\vec{\sigma}) = \frac{\prod_a f_a(\vec{\sigma}_a)}{Z}, \quad Z \equiv \sum_{\vec{\sigma}} \prod_a f_a(\vec{\sigma}_a)$$

Exact Variational Principle — J.W. Gibbs 1903 (or earlier)

also known as Kullback-Leibler (1951) in CS and IT

$$F\{b(\vec{\sigma})\} = -\sum_{\vec{\sigma}} b(\vec{\sigma}) \sum_a \ln f_a(\vec{\sigma}_a) + \sum_{\vec{\sigma}} b(\vec{\sigma}) \ln b(\vec{\sigma})$$

$$\left.\frac{\delta F}{\delta b(\vec{\sigma})}\right|_{b(\vec{\sigma})=p(\vec{\sigma})} = 0 \quad \text{under} \quad \sum_{\vec{\sigma}} b(\vec{\sigma}) = 1$$

Variational Ansatz

- Mean-Field: $p(\vec{\sigma}) \approx b(\vec{\sigma}) = \prod_{(a,b)} b_{ab}(\sigma_{ab})$
- Belief Propagation:

$$p(\vec{\sigma}) \approx b(\vec{\sigma}) = \frac{\prod_a b_a(\vec{\sigma}_a)}{\prod_{(a,b)} b_{ab}(\sigma_{ab})} \quad \text{(exact on a tree)}$$

$$\forall\ a;\ c \in a: \quad \sum_{\vec{\sigma}_a} b_a(\vec{\sigma}_a) = 1, \quad b_{ac}(\sigma_{ac}) = \sum_{\vec{\sigma}_a \setminus \sigma_{ac}} b_a(\vec{\sigma}_a)$$

Bethe Free Energy: variational approach

(Yedidia,Freeman,Weiss '01 - inspired by Bethe '35, Peierls '36)

$$F = \underbrace{-\sum_a \sum_{\vec{\sigma}_a} b_a(\vec{\sigma}_a) \ln f_a(\vec{\sigma}_a)}_{\text{self-energy}} + \underbrace{\sum_a \sum_{\vec{\sigma}_a} b_a(\vec{\sigma}_a) \ln b_a(\vec{\sigma}_a) - \sum_{(a,c)} b_{ac}(\sigma_{ac}) \ln b_{ac}(\sigma_{ac})}_{\text{configurational entropy}}$$

$\forall\ a;\ c \in a:\quad \sum_{\vec{\sigma}_a} b_a(\vec{\sigma}_a) = 1,\quad b_{ac}(\sigma_{ac}) = \sum_{\vec{\sigma}_a \setminus \sigma_{ac}} b_a(\vec{\sigma}_a)$

$\Rightarrow$ Belief-Propagation Equations: $\left.\frac{\delta F}{\delta b}\right|_{\text{constr.}} = 0$

Belief-Propagation as an approximation: iterative $\Rightarrow$ Gallager '61; MacKay '98

- Exact on a tree
- Trading optimality for reduction in complexity: $\sim 2^L \to \sim L$
- (BP = solving equations on the graph) $\neq$ (Message Passing = iterative BP)
- Convergence of MP to minimum of Bethe Free energy can be enforced
- $Z_{BP} \gtrless Z_{exact}$: BP ansatz in exact Gibbs Functional is not a truly variational substitution ($\sum_{\vec{\sigma}} b(\vec{\sigma}) = 1$ is not guaranteed)

Bethe Free Energy: variational approach
(Yedidia,Freeman,Weiss '01 - inspired by Bethe '35, Peierls '36)

$$F = \underbrace{-\sum_a \sum_{\vec\sigma_a} b_a(\vec\sigma_a)\ln f_a(\vec\sigma_a)}_{\text{self-energy}} + \underbrace{\sum_a \sum_{\vec\sigma_a} b_a(\vec\sigma_a)\ln b_a(\vec\sigma_a) - \sum_{(a,c)} b_{ac}(\sigma_{ac})\ln b_{ac}(\sigma_{ac})}_{\text{configurational entropy}}$$

$\forall\ a;\ c \in a:\quad \sum_{\vec\sigma_a} b_a(\vec\sigma_a) = 1,\quad b_{ac}(\sigma_{ac}) = \sum_{\vec\sigma_a \setminus \sigma_{ac}} b_a(\vec\sigma_a)$

$\Rightarrow$ Belief-Propagation Equations: $\left.\frac{\delta F}{\delta b}\right|_{\text{constr.}} = 0$

Belief-Propagation as an approximation: iterative $\Rightarrow$ Gallager '61; MacKay '98

- Exact on a tree
- Trading optimality for reduction in complexity: $\sim 2^L \to \sim L$
- (BP = solving equations on the graph) $\neq$ (Message Passing = iterative BP)
- Convergence of MP to minimum of Bethe Free energy can be enforced
- $Z_{BP} \gtrless Z_{\text{exact}}$: BP ansatz in exact Gibbs Functional is not a truly variational substitution ($\sum_{\vec\sigma} b(\vec\sigma) = 1$ is not guaranteed)

Linear Programming version of Belief Propagation

In the limit of large SNR, $\ln f_a \to \pm\infty$: BP$\to$LP

Minimize $F \approx E = -\sum_a \sum_{\vec{\sigma}_a} b_a(\vec{\sigma}_a) \ln f_a(\vec{\sigma}_a) =$ self energy

under set of linear constraints

LP decoding of LDPC codes Feldman, Wainwright, Karger '03

- ML can be restated as an LP over a codeword polytope
- LP decoding is a "local codewords" relaxation of LP-ML
- Codeword convergence certificate
- Discrete and Nice for Analysis
- Large polytope $\{b_\alpha, b_i\} \Rightarrow$ Small polytope $\{b_i\}$

Linear Programming version of Belief Propagation

In the limit of large SNR, $\ln f_a \to \pm\infty$: BP→LP

Minimize $F \approx E = -\sum_a \sum_{\vec{\sigma}_a} b_a(\vec{\sigma}_a) \ln f_a(\vec{\sigma}_a) =$ self energy

under set of linear constraints

LP decoding of LDPC codes — Feldman, Wainwright, Karger '03

- ML can be restated as an LP over a codeword polytope
- LP decoding is a "local codewords" relaxation of LP-ML
- Codeword convergence certificate
- Discrete and Nice for Analysis
- Large polytope $\{b_\alpha, b_i\} \Rightarrow$ Small polytope $\{b_i\}$

BP does not account for Loops

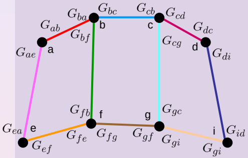
Questions:

- Is BP just a heuristic in a loopy case?
- Why does it (often) work so well?
- Does exact inference allow an expression in terms of BP?
- Can one correct BP systematically?

Gauge Transformations

Chertkov, Chernyak '06

Local Gauge, G, Transformations



$$Z=\sum_{\vec\sigma}\prod_a f_a(\vec\sigma_a),\ \ \vec\sigma_a=(\sigma_{ab},\sigma_{ac},\cdots)$$

$$\sigma_{ab}=\sigma_{ba}=\pm 1$$

$$f_a(\vec\sigma_a=(\sigma_{ab},\cdots))\to \sum_{\sigma'_{ab}} G_{ab}(\sigma_{ab},\sigma'_{ab})\, f_a(\sigma'_{ab},\cdots)$$

$$\sum_{\sigma_{ab}} G_{ab}(\sigma_{ab},\sigma')G_{ba}(\sigma_{ab},\sigma'')=\delta(\sigma',\sigma'')$$

The partition function is invariant under any G-gauge!

$$Z=\sum_{\vec\sigma}\prod_a f_a(\vec\sigma_a)=\sum_{\vec\sigma}\prod_a\left(\sum_{\vec\sigma'_a} f_a(\vec\sigma'_a)\prod_{b\in a} G_{ab}(\sigma_{ab},\sigma'_{ab})\right)$$

Belief Propagation as a Gauge Fixing

Chertkov, Chernyak '06

$$Z=\sum_{\vec\sigma}\prod_a f_a(\vec\sigma_a)=\sum_{\boldsymbol{\sigma}}\prod_a\left(\sum_{\vec\sigma'_a} f_a(\vec\sigma'_a)\prod_{b\in a} G_{ab}(\sigma_{ab},\sigma'_{ab})\right)$$

$$Z=\underbrace{Z_0(G)}_{\substack{\text{ground state}\\ \vec\sigma=+\vec 1}}+\underbrace{\sum_{\text{all possible colorings of the graph}} Z_c(G)}_{\vec\sigma\neq+\vec 1,\quad \text{excited states}}$$

Belief Propagation Gauge $\quad \forall a\ \&\ \forall b\in a:$

$$\sum_{\vec\sigma'_a} f_a(\vec\sigma')G^{(bp)}_{ab}(\sigma_{ab}=-1,\sigma'_{ab})\prod_{c\in a}^{c\neq b} G^{(bp)}_{ac}(+1,\sigma'_{ac})=0$$

No loose BLUE=colored edges at any vertex of the graph!

Belief Propagation as a Gauge Fixing (II)

$\underline{\forall a \ \& \ \forall b \in a:}$

$$\left\{\begin{array}{l} \sum_{\vec{\sigma}'_a} f_a(\vec{\sigma}') G^{(bp)}_{ab}(-1, \sigma'_{ab}) \prod_{c\in a}^{c\neq b} G^{(bp)}_{ac}(+1, \sigma'_{ac}) = 0 \\ \sum_{\sigma_{ab}} G_{ab}(\sigma_{ab}, \sigma') G_{ba}(\sigma_{ab}, \sigma'') = \delta(\sigma', \sigma'') \end{array}\right. \Rightarrow \left\{\begin{array}{l} G^{(bp)}_{ba}(+1, \sigma'_{ab}) = \rho_a^{-1} \overbrace{\sum_{\vec{\sigma}'_a \setminus \sigma'_{ab}} f_a(\vec{\sigma}') \prod_{c\in a}^{c\neq b} G^{(bp)}_{ac}(+1, \sigma'_{ac})}^{sum-product} \\ \rho_a = \sum_{\vec{\sigma}'_a} f_a(\vec{\sigma}') \prod_{c\in a} G^{(bp)}_{ac}(+1, \sigma'_{ac}) \end{array}\right.$$

Belief Propagation in terms of Messages

$$G^{(bp)}_{ab}(+1, \sigma) = \frac{\exp(\sigma\eta_{ab})}{2\sqrt{\cosh(\eta_{ab} + \eta_{ba})}}, \quad G^{(bp)}_{ab}(-1, \sigma) = \sigma \frac{\exp(-\sigma\eta_{ba})}{2\sqrt{\cosh(\eta_{ab} + \eta_{ba})}} \Longrightarrow$$

$$\sum_{\vec{\sigma}_a \setminus \sigma_{ab}} f_a(\vec{\sigma}_a) \exp\left(\sum_{c\in a} \sigma_{ac}\eta_{ac}\right) (\sigma_{ab} - \tanh(\eta_{ab} + \eta_{ba})) = 0$$

$$b_a(\vec{\sigma}_a) = \frac{f_a(\vec{\sigma}_a) \exp\left(\sum_{b\in a} \sigma_{ab}\eta_{ab}\right)}{\sum_{\vec{\sigma}_a} f_a(\vec{\sigma}_a) \exp\left(\sum_{b\in a} \sigma_{ab}\eta_{ab}\right)}, \quad b_{ab}(\sigma) = \frac{\exp(\sigma(\eta_{ab}+\eta_{ba}))}{\sum_\sigma \exp(\sigma(\eta_{ab}+\eta_{ba}))}$$

Variational Principe and Gauge Fixing

$$Z = \underbrace{Z_0(G)}_{\vec{\sigma}=+\vec{1}} + \sum_{\vec{\sigma}\neq+\vec{1}} Z_c(G), \quad Z_0(G) \Rightarrow \underbrace{Z_0(\epsilon), \quad \epsilon_{ab}(\sigma_{ab}) = G_{ab}(+1,\sigma_{ab})}_{\text{depends only on the ground state gauges}}$$

Variational formulation of Belief Propagation

$$\left.\frac{\partial Z_0(\epsilon)}{\partial \epsilon_{ab}(\sigma_{ab})}\right|^{(bp)} = 0 \quad \Leftrightarrow \quad \text{Belief Propagation Equations}$$

$\mathcal{F}_0(\epsilon) = -\ln Z_0(\epsilon)$ is directly related to the Bethe Free Energy of *Yedidia, Freeman, Weiss '01* ▸ Bethe Free Energy

General Remarks on Gauge Fixing

- Related to the Re-parametrization Framework of Wainwright, Jaakkola and Willsky '03
- Generalizable to q-ary alphabet Chernyak, Chertkov '07
- ... suggests Loop Series for the Partition Function $\Rightarrow$

Variational Principe and Gauge Fixing

$$Z = \underbrace{Z_0(G)}_{\vec{\sigma}=+\vec{1}} + \sum_{\vec{\sigma}\neq+\vec{1}} Z_c(G), \quad Z_0(G) \Rightarrow \underbrace{Z_0(\epsilon), \quad \epsilon_{ab}(\sigma_{ab}) = G_{ab}(+1,\sigma_{ab})}_{\text{depends only on the ground state gauges}}$$

Variational formulation of Belief Propagation

$$\left.\frac{\partial Z_0(\epsilon)}{\partial \epsilon_{ab}(\sigma_{ab})}\right|^{(bp)} = 0 \quad \Leftrightarrow \quad \text{Belief Propagation Equations}$$

$\mathcal{F}_0(\epsilon) = -\ln Z_0(\epsilon)$ is directly related to the Bethe Free Energy of *Yedidia, Freeman, Weiss '01* ▸ Bethe Free Energy

General Remarks on Gauge Fixing

- Related to the Re-parametrization Framework of Wainwright, Jaakkola and Willsky '03
- Generalizable to q-ary alphabet Chernyak, Chertkov '07
- ... suggests Loop Series for the Partition Function ⇒

Variational Principle and Gauge Fixing

$$Z = \underbrace{Z_0(G)}_{\vec{\sigma}=+\vec{1}} + \sum_{\vec{\sigma}\neq+\vec{1}} Z_c(G), \quad Z_0(G) \Rightarrow \underbrace{Z_0(\epsilon), \quad \epsilon_{ab}(\sigma_{ab}) = G_{ab}(+1,\sigma_{ab})}_{\text{depends only on the ground state gauges}}$$

Variational formulation of Belief Propagation

$$\left.\frac{\partial Z_0(\epsilon)}{\partial \epsilon_{ab}(\sigma_{ab})}\right|^{(bp)} = 0 \quad \Leftrightarrow \quad \text{Belief Propagation Equations}$$

$\mathcal{F}_0(\epsilon) = -\ln Z_0(\epsilon)$ is directly related to the Bethe Free Energy of *Yedidia, Freeman, Weiss '01* ▸ Bethe Free Energy

General Remarks on Gauge Fixing

- Related to the Re-parametrization Framework of Wainwright, Jaakkola and Willsky '03
- Generalizable to q-ary alphabet Chernyak, Chertkov '07
- ... suggests Loop Series for the Partition Function $\Rightarrow$

Loop Series: Chertkov,Chernyak '06

Exact (!!) expression in terms of BP

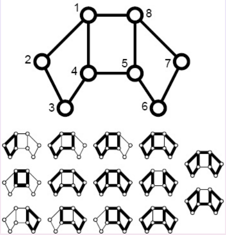
$$Z=\sum_{\vec{\sigma}_\sigma}\prod_a f_a(\vec{\sigma}_a)=Z_0\left(1+\sum_C r(C)\right)$$

$$r(C)=\frac{\prod_{a\in C}\mu_a}{\prod_{(ab)\in C}(1-m_{ab}^2)}=\prod_{a\in C}\tilde{\mu}_a$$

$C\in$ Generalized Loops $=$ Loops without loose ends

$$m_{ab}=\sum_{\vec{\sigma}_a} b_a^{(bp)}(\vec{\sigma}_a)\sigma_{ab}$$

$$\mu_a=\sum_{\vec{\sigma}_a} b_a^{(bp)}(\vec{\sigma}_a)\prod_{b\in a,C}(\sigma_{ab}-m_{ab})$$



- The Loop Series is finite
- All terms in the series are calculated within BP
- BP is exact on a tree
- BP is a Gauge fixing condition. Other choices of Gauges would lead to different representation.

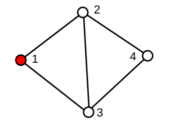
Summary (Loop Calculus)

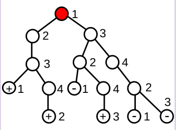
- BP eqs. solve Gauge fixing conditions
- BP eqs also explains no-loose-end coloring constraints
- BP minimizes gauge dependence in the ground state
- Loop series expresses partition function in terms of a sum of terms, each associated with a generalized loop of the graph
- Each term in the Loop Series depends explicitly on the BP solution

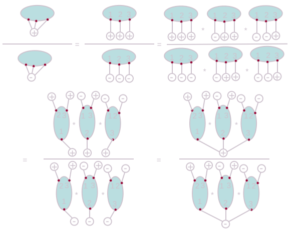
Self-avoiding Tree — Weitz '06

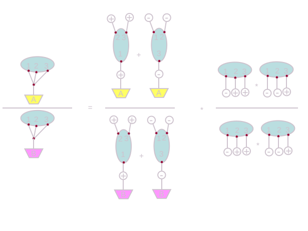
Bipartite. Binary.

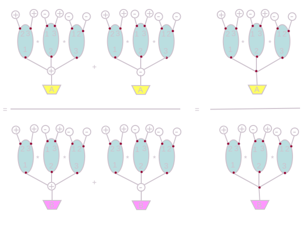
$$\left.\begin{aligned}
\mathcal{P}(\vec{\sigma}) &= Z^{-1}\prod_{(i,j)} f(\sigma_i,\sigma_j)\\
p_i(\sigma_i) &= \sum_{\vec{\sigma}\setminus\sigma_i}\mathcal{P}(\vec{\sigma})\\
\frac{p_i(+)}{p_i(-)} &= \frac{\sum_{\vec{\sigma}\setminus\sigma_i}\left(\prod_{(k,j)} f(\sigma_k,\sigma_j)\right)\Big|_{\sigma_i=+}}{\sum_{\vec{\sigma}\setminus\sigma_i}\left(\prod_{(k,j)} f(\sigma_k,\sigma_j)\right)\Big|_{\sigma_i=-}}
\end{aligned}\right\} \Rightarrow$$











Magnify Magnify Magnify

Self-avoiding Tree — Weitz '06

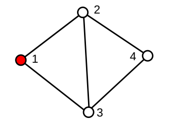
Bipartite. Binary.

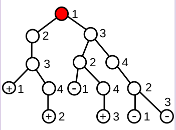
$$\mathcal{P}(\vec{\sigma}) = Z^{-1} \prod_{(i,j)} f(\sigma_i, \sigma_j)$$

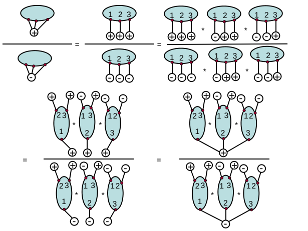
$$p_i(\sigma_i) = \sum_{\vec{\sigma}\setminus\sigma_i} \mathcal{P}(\vec{\sigma})$$

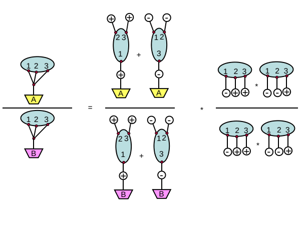
$$\frac{p_i(+)}{p_i(-)} = \frac{\sum_{\vec{\sigma}\setminus\sigma_i}\left(\prod_{(k,j)} f(\sigma_k,\sigma_j)\right)\Big|_{\sigma_i=+}}{\sum_{\vec{\sigma}\setminus\sigma_i}\left(\prod_{(k,j)} f(\sigma_k,\sigma_j)\right)\Big|_{\sigma_i=-}}$$

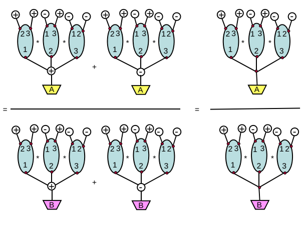
$\Rightarrow$











Magnify Magnify Magnify

Complementarity of Loop Calculus & Graphical Transformations

Speculations

- Loop Calculus is built on Gauge Transformations. Gauge Transformations do not change the graph but reparametrize factor functions.
- Graphical Transformations keep factor functions but modify the graph.
- Loop Calculus & Graphical Transformations are complementary.
- It may be advantageous to build efficient optimality achieving algorithms on the combination of the two: the Loop Calculus and the Graphical Transformations.

Complementarity of Loop Calculus & Graphical Transformations

Speculations

- Loop Calculus is built on Gauge Transformations. Gauge Transformations do not change the graph but reparametrize factor functions.
- Graphical Transformations keep factor functions but modify the graph.
- Loop Calculus & Graphical Transformations are complementary.
- It may be advantageous to build efficient optimality achieving algorithms on the combination of the two: the Loop Calculus and the Graphical Transformations.

Complementarity of Loop Calculus & Graphical Transformations

Speculations

- Loop Calculus is built on Gauge Transformations. Gauge Transformations do not change the graph but reparametrize factor functions.
- Graphical Transformations keep factor functions but modify the graph.
- Loop Calculus & Graphical Transformations are complementary.
- It may be advantageous to build efficient optimality achieving algorithms on the combination of the two: the Loop Calculus and the Graphical Transformations.

Complementarity of Loop Calculus & Graphical Transformations

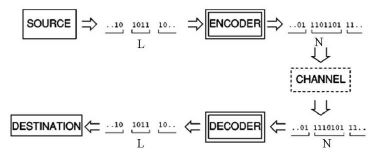
Speculations

- Loop Calculus is built on Gauge Transformations. Gauge Transformations do not change the graph but reparametrize factor functions.
- Graphical Transformations keep factor functions but modify the graph.
- Loop Calculus & Graphical Transformations are complementary.
- It may be advantageous to build efficient optimality achieving algorithms on the combination of the two: the Loop Calculus and the Graphical Transformations.

Error Correction

Cell phone Hard disk Optical disk Fiber

Scheme:



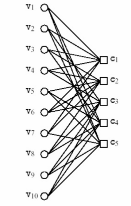
Example of Additive White Gaussian Channel:

$$P(\mathbf{x}_{out}|\mathbf{x}_{in}) = \prod_{i=bits} p(x_{out;i}|x_{in;i})$$

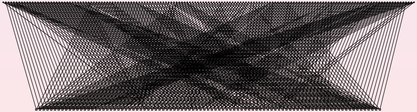
$$p(x|y) \sim \exp(-s^2(x-y)^2/2)$$

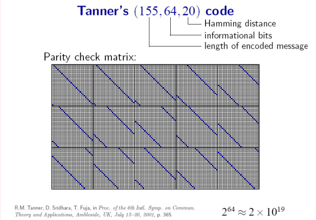
- Channel
  is noisy "black box" with only statistical information available
- Encoding:
  use redundancy to redistribute damaging effect of the noise
- Decoding [Algorithm]:
  reconstruct most probable codeword by noisy (polluted) channel

Low Density Parity Check Codes



- N bits, M checks, $L = N - M$ information bits
  example: $N = 10$, $M = 5$, $L = 5$
- 2^L codewords of 2^N possible patterns
- Parity check: $\hat{H}\mathbf{v} = \mathbf{c} = \mathbf{0}$
  example:
$$\hat{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$
- LDPC = graph (parity check matrix) is sparse





Decoding as Statistical Inference

Decoding $\sigma_i = \pm 1$

$$\mathcal{P}(\boldsymbol{\sigma}|\mathbf{x}) = Z^{-1}(\mathbf{x}) \prod_{\alpha} \delta\left(\prod_{i\in\alpha} \sigma_i, +1 \right) \prod_i p(x_i|\sigma_i)$$

Hard (check) constraints define the graph/code

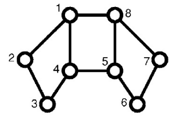
N.Sourlas '89 – Stat Phys & Error-correction

Graphical models

Factorization (Forney '01, Loeliger '01)

$$\mathcal{P}(\boldsymbol{\sigma}|\mathbf{x}) = Z^{-1}\prod_a f_a(\mathbf{x}_a|\boldsymbol{\sigma}_a)$$

$$\underbrace{Z(\mathbf{x}) = \sum_{\boldsymbol{\sigma}}\prod_a f_a(\mathbf{x}_a|\boldsymbol{\sigma}_a))}_{\text{partition function}}$$

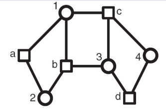


$$f_a \geq 0$$
$$\sigma_{ab} = \sigma_{ba} = \pm 1$$
$$\boldsymbol{\sigma}_1 = (\sigma_{12}, \sigma_{14}, \sigma_{18})$$
$$\boldsymbol{\sigma}_2 = (\sigma_{12}, \sigma_{13})$$

Example: Error-Correction (linear code, bipartite Tanner graph)

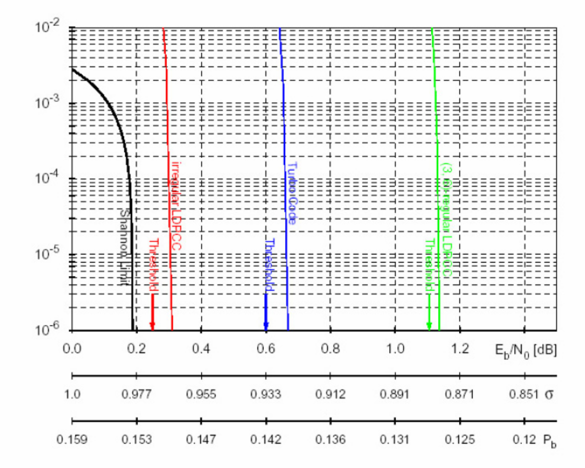
$$f_i(h_i|\boldsymbol{\sigma}_i) = \exp(\sigma_i h_i)\cdot\begin{cases} 1, & \forall\alpha,\beta \ni i,\quad \sigma_{i\alpha} = \sigma_{i\beta} \\ 0, & \text{otherwise}\end{cases}$$

$$f_\alpha(\boldsymbol{\sigma}_\alpha) = \delta\left(\prod_{i\in\alpha}\sigma_i, +1\right)$$



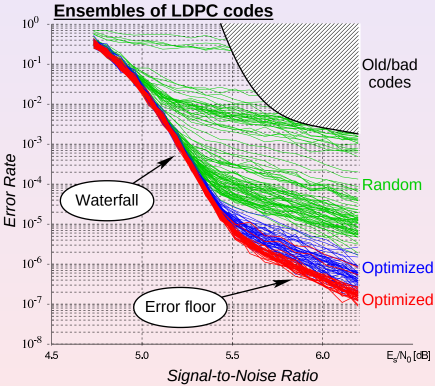
h_i - log-likelihoods

Shannon Transition



- Phase Transition
- Ensemble of Codes [analysis & design]
- Thermodynamic limit but ...

Error-Floor



- BER vs SNR = measure of performance
- Finite size effects
- Waterfall $\leftrightarrow$ Error-floor
- Error-floor typically emerges due to sub-optimality of decoding, i.e. due to unaccounted loops
- Monte-Carlo is useless at FER $\lesssim 10^{-8}$
- Need an efficient method to analyze error-floor

Pseudo-codewords and Instantons

Error-floor is caused by Pseudo-codewords:

Wiberg '96; Forney et.al'99; Frey et.al '01; Richardson '03; Vontobel, Koetter '04-'06

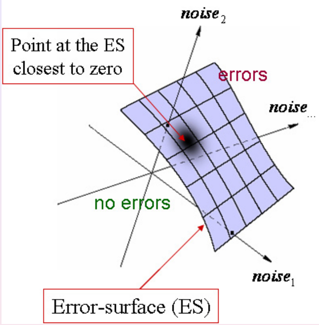
Instanton = optimal conf of the noise

$$BER = \int d(noise)\ WEIGHT(noise)$$

$$BER \sim WEIGHT\left(\begin{array}{c} optimal\ conf \\ of\ the\ noise \end{array}\right)$$

$$\begin{array}{c} optimal\ conf \\ of\ the\ noise \end{array} = \begin{array}{c} Point\ at\ the\ ES \\ closest\ to\ "0" \end{array}$$

Instantons are decoded to Pseudo-Codewords



Instanton-amoeba

= optimization algorithm

Stepanov, et.al '04,'05

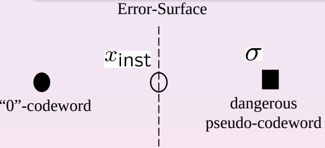
Stepanov, Chertkov '06

Pseudo-Codeword Search Algorithm

LP decoding $(\sigma_i = 0, 1 \quad \text{AWGN channel})$

Minimize, $E = \sum_\alpha \sum_{\sigma_\alpha} b_\alpha(\sigma_\alpha) \sum_{i\in\alpha} \sigma_i(1-2x_i)/q_i$, under $0 \le b_i(\sigma_i), b_\alpha(\sigma_\alpha) \le 1$

$\forall\ \alpha :\ \sum_{\sigma_\alpha} b_\alpha(\sigma_\alpha) = 1, \quad \& \quad \forall\ i\ \forall\ \alpha \ni i :\ b_i(\sigma_i) = \sum_{\sigma_\alpha \setminus \sigma_i} b_\alpha(\sigma_\alpha)$



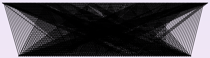
Weighted Median:

$$\mathbf{x}_{\text{inst}} = \frac{\boldsymbol{\sigma}}{2}\frac{\sum_i \sigma_i}{\sum_i \sigma_i^2}, \quad d = \frac{(\sum_i \sigma_i)^2}{\sum_i \sigma_i^2}$$

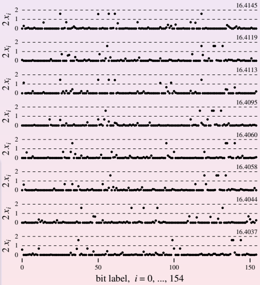
$$\text{FER} \sim \exp(-d \cdot s^2/2)$$

Wiberg '96; Forney et.al '01
Vontobel, Koetter '03,'05

$(155, 64, 20)$, AWGN test:

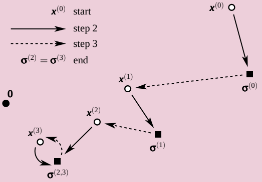


- Fast Convergence



$\sim$ 200 pseudo-codewords within $16.4037 < d < 20$

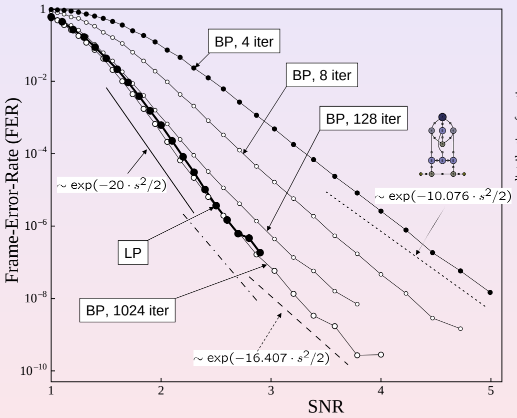
PCS Algorithm Chertkov, Stepanov '06

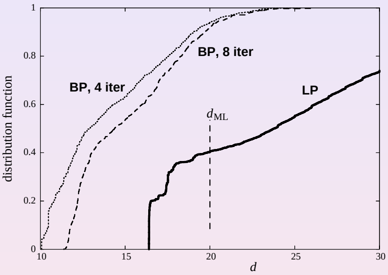


- **Start:** Initiate $\mathbf{x}^{(0)}$.
- **Step 1:** $\mathbf{x}^{(k)}$ is decoded to $\boldsymbol{\sigma}^{(k)}$.
- **Step 2:** Find $\mathbf{y}^{(k)}$ - weighted median between $\boldsymbol{\sigma}^{(k)}$, and "0"
- **Step 3:** If $\mathbf{y}^{(k)} = \mathbf{y}^{(k-1)}$, $k_* = k$ End. Otherwise go to **Step 2** with $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k)} + 0$.

Reducing Complexity of LP

Frame Error-Rate vs Signal-to-Noise-Ratio

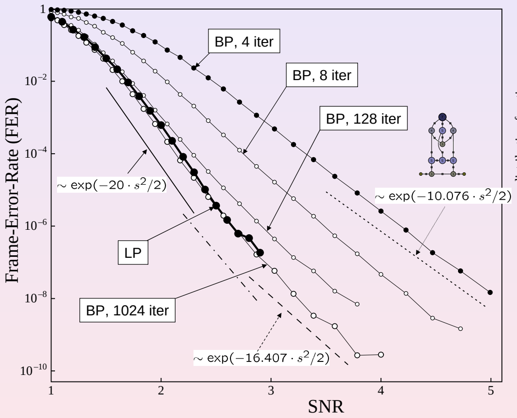


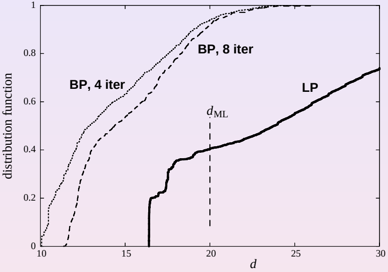


Instanton-amoeba: Stepanov, et.al '04,'05,'06
LP-based PC-search: Chertkov, Stepanov '06,'07

What does Loop Calculus show for dangerous Pseudo-codewords?

Frame Error-Rate vs Signal-to-Noise-Ratio





Instanton-amoeba:
Stepanov, et.al '04,'05,'06
LP-based PC-search:
Chertkov, Stepanov '06,'07

What does Loop Calculus show for dangerous Pseudo-codewords?

Loop Calculus & Pseudo-Codeword Analysis

Chertkov,Chernyak '06

Single loop truncation

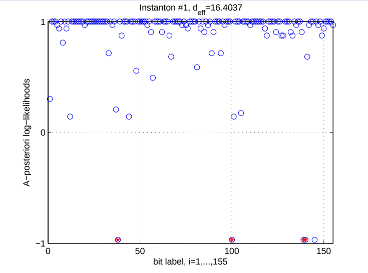
$$Z = Z_0(1+\sum_C r_C) \approx Z_0(1+r(\Gamma))$$

Synthesis of Pseudo-Codeword Search Algorithm (Chertkov, Stepanov '06) & Loop Calculus

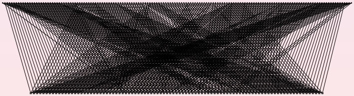
- Consider pseudo-codewords one after other
- For an individual pseudo-codeword/instanton identify a critical loop, Γ, giving major contribution to the loop series.
- Hint: look for single connected loops and use local "triad" contributions as a tester: $r(\Gamma)=\prod_{\alpha\in\Gamma}\tilde{\mu}_\alpha^{(bp)}$

Proof-of-Concept test [(155, 64, 20) code over AWGN]

- $\forall$ pseudo-codewords with $16.4037 < d < 20$ (~ 200 found) there always exists a simple single-connected critical loop(s) with $r(\Gamma) \sim 1$.
- Pseudo-codewords with the lowest d show $r(\Gamma) = 1$
- Invariant with respect to other choices of the original codeword



Bigger Set



Extended Variational Principle & Loop-Corrected BP

Bare BP Variational Principle: $$\left.\frac{\partial Z_0}{\partial \eta_{ab}}\right|_{\eta^{(bp)}} = 0$$

New choice of Gauges guided by the knowledge of the critical loop Γ

$$\left.\frac{\partial \exp(-\mathcal{F})}{\partial \eta_{ab}}\right|_{\eta_{\rm eff}} = 0, \quad \mathcal{F} \equiv -\ln(Z_0 + Z_\Gamma)$$

BP-equations are modified along the critical loop Γ

$$\left.\frac{\sum_{\boldsymbol{\sigma}_a}(\tanh(\eta_{ab}+\eta_{ba})-\sigma_{ab})P_a(\boldsymbol{\sigma}_a)}{\sum_{\boldsymbol{\sigma}_a} P_a(\boldsymbol{\sigma}_a)}\right|_{\eta_{\rm eff}} = \text{explicitly known contribution}\Big|_{\eta_{\rm eff}} \neq 0 \quad [\text{along } \Gamma]$$

Loop-Corrected BP Algorithm

- **1.** Run bare BP algorithm. Terminate if BP succeeds (i.e. a valid code word is found).
- **2.** If BP fails find the most relevant loop Γ that corresponds to the maximal $|r_\Gamma|$. Triad search is helping.
- **3.** Solve the modified-BP equations for the given Γ. Terminate if the improved-BP succeeds.
- **4.** Return to **Step 2** with an improved Γ-loop selection.

LP-erasure = simple heuristics

- **1.** Run LP algorithm. Terminate if LP succeeds (i.e. a valid code word is found).
- **2.** If LP fails, find the most relevant loop Γ that corresponds to the maximal amplitude $r(\Gamma)$.
- **3.** Modify the log-likelihoods along the loop Γ introducing a shift towards zero, i.e. introduce a complete or partial erasure of the log-likelihoods at the bits. Run LP with modified log-likelihoods. Terminate if the modified LP succeeds.
- **4.** Return to **Step 2** with an improved selection principle for the critical loop.

$(155, 64, 20)$ Test

- **IT WORKS!**
  All troublemakers ($\sim$ 200 of them) previously found by LP-based Pseudo-Codeword-Search Algorithm method were successfully corrected by the LP-erasure algorithm.
- Method is invariant with respect the choice of the codeword (used to generate pseudo-codewords).

General Conjecture:

- Loop-erasure algorithm is capable of reducing the error-floor
- Local adjustment of the algorithm, anywhere along the critical loop, in the spirit of the Facet Guessing (Dimakis, Wainwright '06), may be sufficient $\Rightarrow$

Breaking the critical loop locally

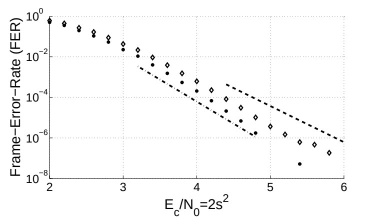
Chertkov '07

- Exhaustive Bit Guessing (simplified version of [Dimakis, Wainwright '06]) corrects all the $\sim$ 200 dangerous pseudo-codewords !!
- Set of "successful" bits correlates strongly with the set of bits forming the critical loop

Loop Guided Guessing (LGG)

- **1.** Run the LP algorithm. Terminate if LP succeeds.
- **2.** If LP fails, find the critical loop, Γ.
- **3.** Pick any bit along the critical loop and "fix the bit" running two two corrected LP schemes. Terminate if any of LPs succeeds.
- **4.** If not return to **Step 3** selecting another bit along the critical loop or to **Step 2** for an improved selection principle for Γ.

$[155, 64, 20]$ test of LGG



- Complexity of LGG is the same as of LP
- LGG corrects 9 out of 10 errors at $E_b/N_0 = 4.8$!!

What to do with the remaining 1/10 ?

- Draper, Yedidia, Wang ISIT'07: Fixing $1, 2, ..., k$ bits $= 2^k$ LPs till decode to a codeword (ML certificate enforced).
- Weiss, Yanover, Meltzer '07: Sufficient condition for bits decoded by the bare LP to integers to show the right values.

Our further strategy:

- Use Loop Calculus in sequential selection of the fixed bits
- Longer codes
- Back to iterative BP

Summary (LDPC Decoding)

- Error floor is due to low-weight (dangerous) pseudo-codewords
- Instanton-amoeba & Pseudo-codeword search algorithms allows to find the dangerous pseudo-codewords efficiently
- Critical loops in the Loop Series signify wrong decoding
- Loop Series based analysis offers efficient guiding principle for decoding improvement
- Reducing the error floor may be not that difficult ... after all [N.B. We are discussing Average Case Complexity]

All papers are available at http://cnls.lanl.gov/∼chertkov/pub.htm

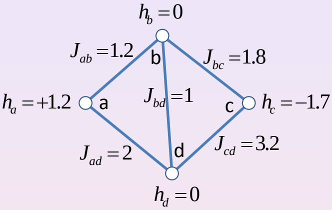
Bibliography

Ferromagnetic Random-Field Ising Model

$$p(\vec{\sigma}) = Z^{-1} \exp\left(\frac{1}{2T}\sum_{(i,j)} J_{ij}\sigma_i\sigma_j + \frac{1}{T}\sum_i h_i\sigma_i\right)$$

$J_{ij} \geq 0, \;\; h_i \gtrless 0$

(i,j) are edges on an undirected graph $\mathcal{G}$

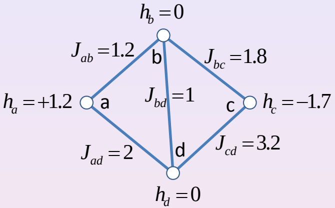


Ground State, $T \to 0$

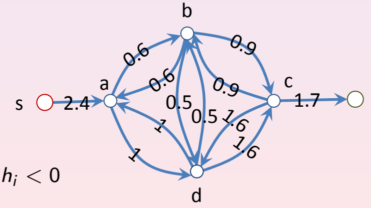
$$\min_{\boldsymbol{\sigma}}\left(-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}} J_{ij}\sigma_i\sigma_j - \sum_{i\in\mathcal{G}} h_i\sigma_i\right)\Bigg|_{\forall\; i\in\mathcal{G}:\;\; \sigma_i=\pm 1}$$

Undirected $\Rightarrow$ Directed $\Rightarrow$ (s-t)-Extended

$$\min_{\sigma}\left(-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}} J_{ij}\sigma_i\sigma_j - \sum_{i\in\mathcal{G}} h_i\sigma_i\right)\Bigg|_{\forall\ i\in\mathcal{G}:\quad \sigma_i=\pm 1}$$



$$\min_{\sigma}\left(-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}'_d} J_{i\to j}\sigma_i\sigma_j\right)\Bigg|_{\forall\ i\in\mathcal{G}'_d:\quad \sigma_i=\pm 1;\sigma_s=+1;\sigma_t=-1}$$

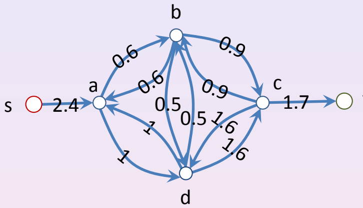


directed: $J_{i\to j}=J_{j\to i}=J_{ij}/2$

(s-t) extended: $J_{s\to i}=2h_i,$ if $h_i>0$ $\quad J_{i\to t}=2|h_i|,$ if $h_i<0$

From Vertexes to Edges

$$\min_{\boldsymbol{\sigma}}\left(-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}'_d} J_{i\to j}\sigma_i\sigma_j\right)\Bigg|_{\forall\ i\in\mathcal{G}'_d:\quad \sigma_i=\pm1;\sigma_s=+1;\sigma_t=-1}$$



Integer Linear Programming

$$\eta_{i\to j}=\begin{cases}1, & \sigma_i=1,\sigma_j=-1\\ 0, & \text{otherwise}\end{cases}\qquad p_i=(1-\sigma_i)/2=0,1$$

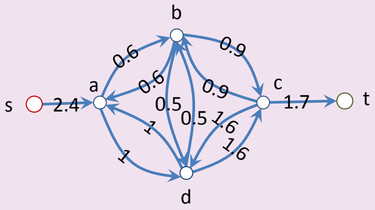
$$\sigma_i\sigma_j+\sigma_j\sigma_i=2-4(\eta_{i\to j}+\eta_{j\to i}),\quad \sigma_s\sigma_i=1-2\eta_{s\to i},\quad \sigma_i\sigma_t=1-2\eta_{i\to t}$$

$$-\frac{1}{2}\sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j}+\min_{\{\eta,p\}}\sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j}\eta_{i\to j}\Bigg|\begin{array}{c}\forall i\in\mathcal{G}'_d, p_i=0,1;\ p_s=0,\ p_t=1\\ \forall(i\to j)\in\mathcal{G}'_d:\\ p_i-p_j+\eta_{i\to j}=0,1\end{array}$$

FRFI=Min-Cut=Max-Flow

$$-\frac{1}{2}\sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j} + \min_{\{\eta,p\}} \sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j}\eta_{i\to j}\Bigg| \begin{array}{c} \forall i\in\mathcal{G}'_d, p_i=0,1;\ \ p_s=0,\ \ p_t=1 \\ \forall (i\to j)\in\mathcal{G}'_d: \\ p_i-p_j+\eta_{i\to j}=0,1 \end{array}$$

Min-Cut



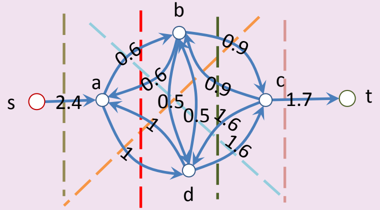
Max-Flow

A.K. Hartman & H. Rieger, *Optimization Algorithms in Physics*, Wiley-VCH, 2002, and references therein

FRFI=Min-Cut=Max-Flow

$$-\frac{1}{2}\sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j}+\min_{\{\eta,p\}}\sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j}\eta_{i\to j}\Bigg|\begin{array}{c}\forall i\in\mathcal{G}'_d,\, p_i=0,1;\;\; p_s=0,\;\; p_t=1\\ \forall (i\to j)\in\mathcal{G}'_d:\\ p_i-p_j+\eta_{i\to j}=0,1\end{array}$$

Min-Cut



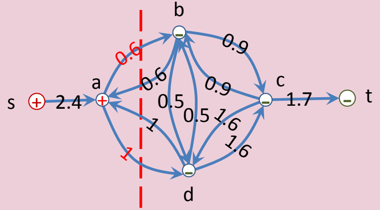
Max-Flow

A.K. Hartman & H. Rieger, *Optimization Algorithms in Physics*, Wiley-VCH, 2002, and references therein

FRFI=Min-Cut=Max-Flow

$$-\frac{1}{2}\sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j} + \min_{\{\eta,p\}} \sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j}\eta_{i\to j}\Bigg| \begin{array}{c} \forall i\in\mathcal{G}'_d,\, p_i=0,1;\;\; p_s=0,\;\; p_t=1 \\ \forall(i\to j)\in\mathcal{G}'_d: \\ p_i-p_j+\eta_{i\to j}=0,1 \end{array}$$

Min-Cut



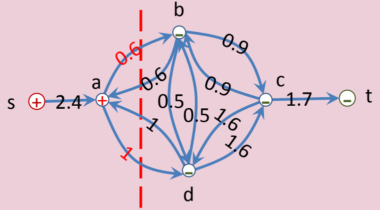
Max-Flow

A.K. Hartman & H. Rieger, *Optimization Algorithms in Physics*, Wiley-VCH, 2002, and references therein

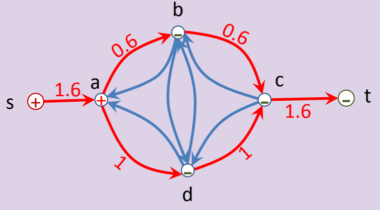
FRFI=Min-Cut=Max-Flow

$$-\frac{1}{2}\sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j}+\min_{\{\eta,p\}}\sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j}\eta_{i\to j}\Bigg|\begin{array}{c}\forall i\in\mathcal{G}'_d,\ p_i=0,1;\ \ p_s=0,\ \ p_t=1\\ \forall(i\to j)\in\mathcal{G}'_d:\\ p_i-p_j+\eta_{i\to j}=0,1\end{array}$$

Min-Cut



Max-Flow

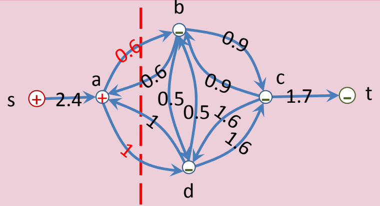


A.K. Hartman & H. Rieger, *Optimization Algorithms in Physics*, Wiley-VCH, 2002, and references therein

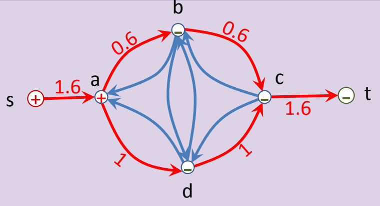
Back to Undirected Graph

$$-\frac{1}{2}\sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j}+\min_{\{\eta,p\}}\sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j}\eta_{i\to j}\Big|\quad \begin{array}{c}\forall i\in\mathcal{G}'_d,\, p_i=0,1;\;\; p_s=0,\;\; p_t=1\\ \forall(i\to j)\in\mathcal{G}'_d:\;\; p_i-p_j+\eta_{i\to j}=0,1\end{array}$$

Min-Cut



Max-Flow



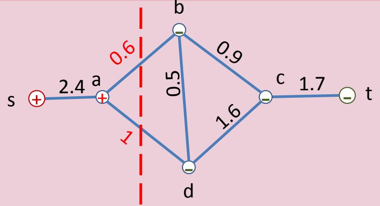
$$-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}'} J_{ij}+\min_{\{\eta,p\}}\sum_{(ij)\in\mathcal{G}'_d} J_{ij}\eta_{ij}\Big|\quad \begin{array}{c}\forall i\in\mathcal{G}',\, p_i=0,1;\;\; p_s=0,\;\; p_t=1\\ \forall(i,j)\in\mathcal{G}':\;\; p_i-p_j+\eta_{ij}=0,1\end{array}$$

$$J_{si}=2h_i,\quad \text{if}\quad h_i>0\qquad J_{it}=2|h_i|,\quad \text{if}\quad h_i<0$$

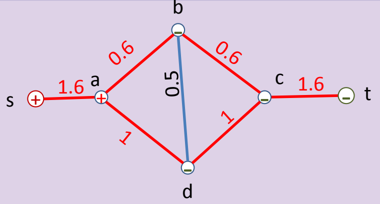
Back to Undirected Graph

$$-\frac{1}{2}\sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j} + \min_{\{\eta,p\}} \sum_{(i\to j)\in\mathcal{G}'_d} J_{i\to j}\eta_{i\to j}\Big| \quad \begin{array}{c}\forall i\in\mathcal{G}'_d,\, p_i=0,1;\ \ p_s=0,\ \ p_t=1\\ \forall(i\to j)\in\mathcal{G}'_d:\ \ p_i-p_j+\eta_{i\to j}=0,1\end{array}$$

Min-Cut



Max-Flow



$$-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}'} J_{ij} + \min_{\{\eta,p\}} \sum_{(ij)\in\mathcal{G}'_d} J_{ij}\eta_{ij}\Big| \quad \begin{array}{c}\forall i\in\mathcal{G}',\, p_i=0,1;\ \ p_s=0,\ \ p_t=1\\ \forall(i,j)\in\mathcal{G}':\ \ p_i-p_j+\eta_{ij}=0,1\end{array}$$

$$J_{si}=2h_i,\quad \text{if}\quad h_i>0 \qquad J_{it}=2|h_i|,\quad \text{if}\quad h_i<0$$

FRFI/Min-Cut/Max-Flow is EASY

- Many network algorithms. See e.g. T.H. Cormen, et al, *Introduction to Algorithms*, MIT-Press (2001)
- Reduction to Linear Programming. See e.g. H. Papadimitriou, I. Steiglitz, *Combinatorial Optimization: Alg. and Complexity*, Dover (1998)

Relaxation of Min-Cut Integer LP to respective LP is exact

$$-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}'} J_{ij} + \min_{\{\eta,p\}} \sum_{(ij)\in\mathcal{G}'_d} J_{ij}\eta_{ij}\Big| \begin{array}{l} p_s=0,\ p_t=1;\ \forall i\in\mathcal{G}', p_i=0,1 \\ \forall(i,j)\in\mathcal{G}':\ p_i-p_j+\eta_{ij}=0,1 \end{array}$$

- Matrix of LP constraints is Totally Uni-Modular (TUM)
- Min-Cut LP and Max-Flow LP are Dual

FRFI/Min-Cut/Max-Flow is EASY

- Many network algorithms. See e.g. T.H. Cormen, et al, *Introduction to Algorithms*, MIT-Press (2001)
- Reduction to Linear Programming. See e.g. H. Papadimitriou, I. Steiglitz, *Combinatorial Optimization: Alg. and Complexity*, Dover (1998)

Relaxation of Min-Cut Integer LP to respective LP is exact

$$-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}'} J_{ij} + \min_{\{\eta,p\}} \sum_{(ij)\in\mathcal{G}'_d} J_{ij}\eta_{ij}\Big| \begin{array}{c} p_s=0,\ p_t=1;\ \forall i\in\mathcal{G}', p_i=[0,1] \\ \forall(i,j)\in\mathcal{G}':\ p_i-p_j+\eta_{ij}=[0,1] \end{array}$$

- Matrix of LP constraints is Totally Uni-Modular (TUM)
- Min-Cut LP and Max-Flow LP are Dual

How about using BP for FRFI?

First Impression:

Should not work for arbitrary graph because of Loops

On Second Thought:

May be the $T \to 0$ limit is not that hopeless? After all we know that the problem is easy!

Tree reweighted BP of Kolmogorov & Wainwright '05

At $T \to 0$ BP solves the FRFI model exactly on any graph!

Another Easy Example with Loops: Bayati, Shah and Sharma '06

Maximum Weight Matching of a Bi-partite graph

How about using BP for FRFI?

First Impression:

Should not work for arbitrary graph because of Loops

On Second Thought:

May be the $T \to 0$ limit is not that hopeless? After all we know that the problem is easy!

Tree reweighted BP of Kolmogorov & Wainwright '05

At $T \to 0$ BP solves the FRFI model exactly on any graph!

Another Easy Example with Loops: Bayati, Shah and Sharma '06

Maximum Weight Matching of a Bi-partite graph

How about using BP for FRFI?

First Impression:

Should not work for arbitrary graph because of Loops

On Second Thought:

May be the $T \to 0$ limit is not that hopeless? After all we know that the problem is easy!

Tree reweighted BP of Kolmogorov & Wainwright '05

At $T \to 0$ BP solves the FRFI model exactly on any graph!

Another Easy Example with Loops: Bayati, Shah and Sharma '06

Maximum Weight Matching of a Bi-partite graph

How about using BP for FRFI?

First Impression:

Should not work for arbitrary graph because of Loops

On Second Thought:

May be the $T \to 0$ limit is not that hopeless? After all we know that the problem is easy!

Tree reweighted BP of Kolmogorov & Wainwright '05

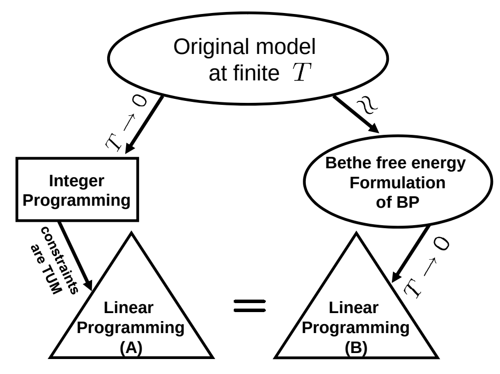
At $T \to 0$ BP solves the FRFI model exactly on any graph!

Another Easy Example with Loops: Bayati, Shah and Sharma '06

Maximum Weight Matching of a Bi-partite graph

Easy Problems with Loops and Bethe Free energy

Proof of the BP-exactness via the Bethe Free energy approach



Chertkov '08

Bethe Free Energy for FRFI

At any Temperature

Minimize the Free Energy :

$$\mathcal{F} = E - TS, \quad E = -\sum_{(i,j)}\sum_{\sigma_i,\sigma_j} b_{ij}(\sigma_i,\sigma_j)\frac{J_{ij}}{2}\sigma_i\sigma_j - \sum_i\sum_{\sigma_i} b_i(\sigma_i)h_i\sigma_i$$

$$S = \sum_{(i,j)}\sum_{\sigma_i,\sigma_j} b_{ij}(\sigma_i,\sigma_j)\ln b_{(i,j)}(\sigma_i,\sigma_j) - \sum_i\sum_{\sigma_i} b_i(\sigma_i)\ln b_i(\sigma_i)$$

$$\forall i \ \ \& \ \ \forall j\in i: \quad b_i(\sigma_i) = \sum_{\sigma_j} b_{ij}(\sigma_i,\sigma_j), \quad \forall i: \quad \sum_{\sigma_i} b_i(\sigma_i) = 1$$

$T\to 0 \quad \Rightarrow \quad$ Linear Programming

Minimize the Self Energy :

$$E = -\sum_{(i,j)}\sum_{\sigma_i,\sigma_j} b_{ij}(\sigma_i,\sigma_j)\frac{J_{ij}}{2}\sigma_i\sigma_j - \sum_i\sum_{\sigma_i} b_i(\sigma_i)h_i\sigma_i$$

$$\forall i \ \ \& \ \ \forall j\in i: \quad b_i(\sigma_i) = \sum_{\sigma_j} b_{ij}(\sigma_i,\sigma_j), \quad \forall i: \quad \sum_{\sigma_i} b_i(\sigma_i) = 1$$

Bethe Free Energy for FRFI

At any Temperature

Minimize the Free Energy :

$$\mathcal{F} = E - TS, \quad E = -\sum_{(i,j)} \sum_{\sigma_i,\sigma_j} b_{ij}(\sigma_i,\sigma_j) \frac{J_{ij}}{2} \sigma_i \sigma_j - \sum_i \sum_{\sigma_i} b_i(\sigma_i) h_i \sigma_i$$

$$S = \sum_{(i,j)} \sum_{\sigma_i,\sigma_j} b_{ij}(\sigma_i,\sigma_j) \ln b_{(i,j)}(\sigma_i,\sigma_j) - \sum_i \sum_{\sigma_i} b_i(\sigma_i) \ln b_i(\sigma_i)$$

$$\forall i \ \ \& \ \ \forall j \in i : \quad b_i(\sigma_i) = \sum_{\sigma_j} b_{ij}(\sigma_i,\sigma_j), \quad \forall i : \quad \sum_{\sigma_i} b_i(\sigma_i) = 1$$

$T \to 0 \quad \Rightarrow \quad$ Linear Programming

Minimize the Self Energy :

$$E = -\sum_{(i,j)} \sum_{\sigma_i,\sigma_j} b_{ij}(\sigma_i,\sigma_j) \frac{J_{ij}}{2} \sigma_i \sigma_j - \sum_i \sum_{\sigma_i} b_i(\sigma_i) h_i \sigma_i$$

$$\forall i \ \ \& \ \ \forall j \in i : \quad b_i(\sigma_i) = \sum_{\sigma_j} b_{ij}(\sigma_i,\sigma_j), \quad \forall i : \quad \sum_{\sigma_i} b_i(\sigma_i) = 1$$

Linear Programming (B) for FRFI

$$\text{(s-t) modification:}\quad \left\{\begin{array}{lll} J_{si}=2h_i & b_{si}(\sigma_s,\sigma_i)=b_i(\sigma_i)\delta(\sigma_s,+1) & h_i>0 \\ J_{it}=2|h_i| & b_{it}(\sigma_i,\sigma_t)=b_i(\sigma_i)\delta(\sigma_t,-1) & h_i<0 \end{array}\right.$$

$$\min_{\{b_i;b_{ij}\}}\left(-\sum_{(i,j)\in\mathcal{G}'}\sum_{\sigma_i,\sigma_j} b_{ij}(\sigma_i,\sigma_j)\frac{J_{ij}}{2}\sigma_i\sigma_j\right)\left|\begin{array}{c} \forall i\in\mathcal{G}' \;\;\&\;\; \forall j\in i:\quad b_i(\sigma_i)=\sum_{\sigma_j} b_{ij}(\sigma_i,\sigma_j) \\ \forall i\in\mathcal{G}':\quad \sum_{\sigma_i} b_i(\sigma_i)=1 \\ b_s(+)=1 \;\;\&\;\; b_d(-)=1 \end{array}\right. .$$

$$-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}'} J_{ij}+\min_{\{\mu,\pi\}}\sum_{(i,j)\in\mathcal{G}'} J_{ij}\mu_{ij}\left|\begin{array}{c} \forall (i,j)\in\mathcal{G}':\quad \pi_i-\pi_j+\mu_{ij}\geq 0 \\ \forall (i,j)\in\mathcal{G}':\quad 1\geq\pi_i,\mu_{ij}\geq 0 \\ \pi_s=0,\quad \pi_t=1 \end{array}\right.$$

Linear Programming (B) for FRFI

(s-t) modification: $\begin{cases} J_{si} = 2h_i & b_{si}(\sigma_s,\sigma_i) = b_i(\sigma_i)\delta(\sigma_s,+1) & h_i > 0 \\ J_{it} = 2|h_i| & b_{it}(\sigma_i,\sigma_t) = b_i(\sigma_i)\delta(\sigma_t,-1) & h_i < 0 \end{cases}$

$$\min_{\{b_i;b_{ij}\}}\left(-\sum_{(i,j)\in\mathcal{G}'}\sum_{\sigma_i,\sigma_j} b_{ij}(\sigma_i,\sigma_j)\frac{J_{ij}}{2}\sigma_i\sigma_j\right)\Bigg| \begin{array}{c} \forall i\in\mathcal{G}' \;\&\; \forall j\in i: \quad b_i(\sigma_i)=\sum_{\sigma_j} b_{ij}(\sigma_i,\sigma_j) \\ \forall i\in\mathcal{G}': \quad \sum_{\sigma_i} b_i(\sigma_i)=1 \\ b_s(+)=1 \;\&\; b_d(-)=1 \end{array}.$$

$$\{b\}\to\{\mu,\pi\}: \quad \begin{cases} \mu_{ij}\equiv b_{ij}(+,-)+b_{ij}(-,+)=1-b_{ij}(+,+)-b_{ij}(-,-), & \forall(i,j)\in\mathcal{G}' \\ \pi_i = b_i(-)=b_{ij}(-,+)+b_{ij}(-,-), & \forall i\in\mathcal{G}' \end{cases}$$

$$-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}'} J_{ij} + \min_{\{\mu,\pi\}}\sum_{(i,j)\in\mathcal{G}'} J_{ij}\mu_{ij}\Bigg| \begin{array}{c} \forall(i,j)\in\mathcal{G}': \quad \pi_i-\pi_j+\mu_{ij}\ge 0 \\ \forall(i,j)\in\mathcal{G}': \quad 1\ge\pi_i,\mu_{ij}\ge 0 \\ \pi_s=0, \quad \pi_t=1 \end{array}$$

Linear Programming (B) for FRFI

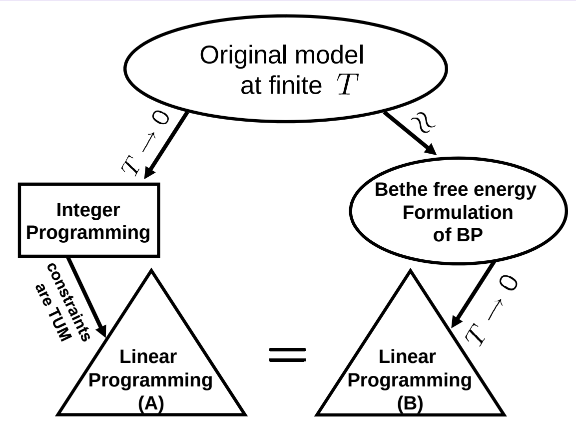
(s-t) modification: $\left\{\begin{array}{lll} J_{si}=2h_i & b_{si}(\sigma_s,\sigma_i)=b_i(\sigma_i)\delta(\sigma_s,+1) & h_i>0 \\ J_{it}=2|h_i| & b_{it}(\sigma_i,\sigma_t)=b_i(\sigma_i)\delta(\sigma_t,-1) & h_i<0 \end{array}\right.$

$$\min_{\{b_i;b_{ij}\}}\left(-\sum_{(i,j)\in\mathcal{G}'}\sum_{\sigma_i,\sigma_j} b_{ij}(\sigma_i,\sigma_j)\frac{J_{ij}}{2}\sigma_i\sigma_j\right)\Bigg| \begin{array}{c} \forall i\in\mathcal{G}' \ \ \& \ \ \forall j\in i: \quad b_i(\sigma_i)=\sum_{\sigma_j} b_{ij}(\sigma_i,\sigma_j) \\ \forall i\in\mathcal{G}': \quad \sum_{\sigma_i} b_i(\sigma_i)=1 \\ b_s(+)=1 \ \ \& \ \ b_d(-)=1 \end{array}.$$

$$\{b\}\to\{\mu,\pi\}: \quad \left\{\begin{array}{lr} \mu_{ij}\equiv b_{ij}(+,-)+b_{ij}(-,+)=1-b_{ij}(+,+)-b_{ij}(-,-), & \forall(i,j)\in\mathcal{G}' \\ \pi_i=b_i(-)=b_{ij}(-,+)+b_{ij}(-,-), & \forall i\in\mathcal{G}' \end{array}\right.$$

$$-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}'}J_{ij}+\min_{\{\mu,\pi\}}\sum_{(i,j)\in\mathcal{G}'}J_{ij}\mu_{ij}\Bigg| \begin{array}{c} \forall(i,j)\in\mathcal{G}': \quad \pi_i-\pi_j+\mu_{ij}\geq 0 \\ \forall(i,j)\in\mathcal{G}': \quad 1\geq\pi_i,\mu_{ij}\geq 0 \\ \pi_s=0, \quad \pi_t=1 \end{array}$$

FRFI at $T = 0$ is solved exactly by BP



FRFI at $T=0$ is solved exactly by BP

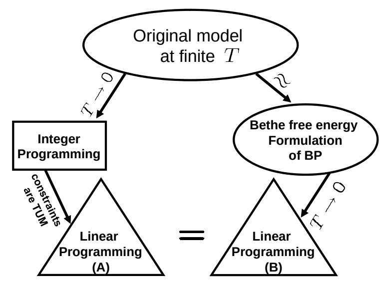
LP(A)

$$-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}'} J_{ij} + \min_{\{\eta,p\}} \sum_{(ij)\in\mathcal{G}'_d} J_{ij}\eta_{ij}\Big| \begin{array}{c} p_s=0,\ p_t=1;\ \forall i\in\mathcal{G}', p_i=[0,1] \\ \forall (i,j)\in\mathcal{G}':\ p_i-p_j+\eta_{ij}=[0,1] \end{array}$$

LP(B)

$$-\frac{1}{2}\sum_{(i,j)\in\mathcal{G}'} J_{ij} + \min_{\{\mu,\pi\}} \sum_{(i,j)\in\mathcal{G}'} J_{ij}\mu_{ij}\Bigg| \begin{array}{cc} \forall (i,j)\in\mathcal{G}': & \pi_i-\pi_j+\mu_{ij}\geq 0 \\ \forall (i,j)\in\mathcal{G}': & 1\geq \pi_i,\mu_{ij}\geq 0 \\ \pi_s=0, & \pi_t=1 \end{array}$$

LP(A)=LP(B)



The scheme also works for $T \to 0$ of

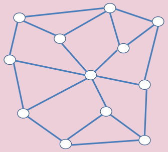
$$p(\boldsymbol{\sigma}) = Z^{-1} \exp\left(-T^{-1}\sum_i h_i\sigma_i\right)\prod_\alpha \delta\left(\sum_i J_{\alpha i}\sigma_i, m_\alpha\right).$$

where $\hat{J}$ is a Totally Uni-Modular matrix

Glassy Ising & Dimer Models on a Planar Graph

Partition Function of $J_{ij} \gtrless 0$ Ising Model, $\sigma_i = \pm 1$

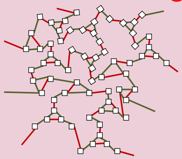
$$Z = \sum_{\vec{\sigma}} \exp\left(\frac{\sum_{(i,j)\in\Gamma} J_{ij}\sigma_i\sigma_j}{T}\right)$$



Partition Function of Dimer Model, $\pi_{ij} = 0, 1$

$$Z = \sum_{\vec{\pi}} \prod_{(i,j)\in\Gamma} (z_{ij})^{\pi_{ij}} \prod_{i\in\Gamma} \delta\left(\sum_{j\in i} \pi_{ij}, 1\right)$$

perfect matching

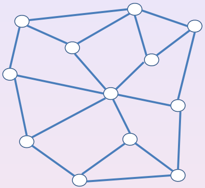


Ising & Dimer Classics

- L. Onsager, *Crystal Statistics*, Phys.Rev. **65**, 117 (1944)
- M. Kac, J.C. Ward, *A combinatorial solution of the Two-dimensional Ising Model*, Phys. Rev. **88**, 1332 (1952)
- C.A. Hurst and H.S. Green, *New Solution of the Ising Problem for a Rectangular Lattice*, J.of Chem.Phys. **33**, 1059 (1960)
- M.E. Fisher, *Statistical Mechanics on a Plane Lattice*, Phys.Rev **124**, 1664 (1961)
- P.W. Kasteleyn, *The statistics of dimers on a lattice*, Physics **27**, 1209 (1961)
- P.W. Kasteleyn, *Dimer Statistics and Phase Transitions*, J. Math. Phys. **4**, 287 (1963)
- M.E. Fisher, *On the dimer solution of planar Ising models*, J. Math. Phys. **7**, 1776 (1966)
- F. Barahona, *On the computational complexity of Ising spin glass models*, J.Phys. A **15**, 3241 (1982)

From Ising to Dimer (I)

$$Z = \sum_{\vec{\sigma}} \exp\left(\frac{\sum_{(i,j)\in\Gamma} J_{ij}\sigma_i\sigma_j}{T}\right)$$

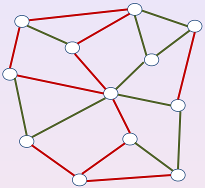


- For a given $\vec{\sigma}$ an edge is sat:
  if $J_{ij} > 0$ & $\sigma_i\sigma_j = 1$ of $J_{ij} < 0$ & $\sigma_i\sigma_j = -1$
- Circle is frustrated if the number of negative edges is odd. (N.B. Frustration of a circle is invariant wrt $\vec{\sigma}$.)
- Equivalent configurations, $\vec{\sigma}$ and $-\vec{\sigma}$, have the same weight
- Introduce dual graph, Γ^*. A vertex of Γ^* correspondent to a frustrated (unfrustrated) face is odd (even).

$$E = -\sum_{(ij)} J_{ij}\sigma_i\sigma_j = -\sum_{(ij)} |J_{ij}| + 2 \sum_{\text{unsat edges}} |J_{ij}|$$

From Ising to Dimer (I)

$$Z = \sum_{\vec{\sigma}} \exp\left(\frac{\sum_{(i,j)\in\Gamma} J_{ij}\sigma_i\sigma_j}{T}\right)$$

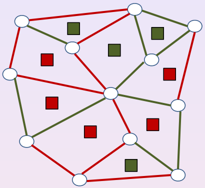


- For a given $\vec{\sigma}$ an edge is sat: if $J_{ij} > 0 \;\; \& \;\; \sigma_i\sigma_j = 1$ of $J_{ij} < 0 \;\; \& \;\; \sigma_i\sigma_j = -1$
- Circle is frustrated if the number of negative edges is odd. (N.B. Frustration of a circle is invariant wrt $\vec{\sigma}$.)
- Equivalent configurations, $\vec{\sigma}$ and $-\vec{\sigma}$, have the same weight
- Introduce dual graph, Γ^*. A vertex of Γ^* correspondent to a frustrated (unfrustrated) face is odd (even).

$$E = -\sum_{(ij)} J_{ij}\sigma_i\sigma_j = -\sum_{(ij)} |J_{ij}| + 2 \sum_{\text{unsat edges}} |J_{ij}|$$

From Ising to Dimer (I)

$$Z = \sum_{\vec{\sigma}} \exp\left(\frac{\sum_{(i,j)\in\Gamma} J_{ij}\sigma_i\sigma_j}{T}\right)$$

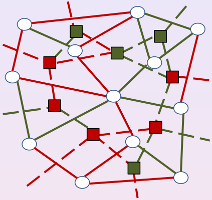


- For a given $\vec{\sigma}$ an edge is sat:
  if $J_{ij} > 0 \;\; \& \;\; \sigma_i\sigma_j = 1$ of $J_{ij} < 0 \;\; \& \;\; \sigma_i\sigma_j = -1$
- Circle is frustrated if the number of negative edges is odd. (N.B. Frustration of a circle is invariant wrt $\vec{\sigma}$.)
- Equivalent configurations, $\vec{\sigma}$ and $-\vec{\sigma}$, have the same weight
- Introduce dual graph, Γ^*. A vertex of Γ^* correspondent to a frustrated (unfrustrated) face is odd (even).

$$E = -\sum_{(ij)} J_{ij}\sigma_i\sigma_j = -\sum_{(ij)} |J_{ij}| + 2 \sum_{\text{unsat edges}} |J_{ij}|$$

From Ising to Dimer (I)

$$Z = \sum_{\vec{\sigma}} \exp\left(\frac{\sum_{(i,j)\in\Gamma} J_{ij}\sigma_i\sigma_j}{T}\right)$$

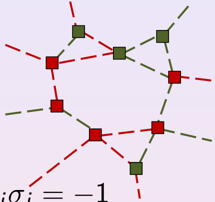


- For a given $\vec{\sigma}$ an edge is sat:
  if $J_{ij} > 0$ & $\sigma_i\sigma_j = 1$ of $J_{ij} < 0$ & $\sigma_i\sigma_j = -1$
- Circle is frustrated if the number of negative edges is odd. (N.B. Frustration of a circle is invariant wrt $\vec{\sigma}$.)
- Equivalent configurations, $\vec{\sigma}$ and $-\vec{\sigma}$, have the same weight
- Introduce dual graph, Γ^*. A vertex of Γ^* correspondent to a frustrated (unfrustrated) face is odd (even).

$$E = -\sum_{(ij)} J_{ij}\sigma_i\sigma_j = -\sum_{(ij)} |J_{ij}| + 2 \sum_{\text{unsat edges}} |J_{ij}|$$

From Ising to Dimer (I)

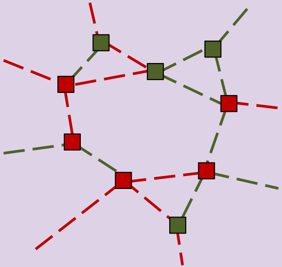
$$Z = \sum_{\vec{\sigma}} \exp\left(\frac{\sum_{(i,j)\in\Gamma} J_{ij}\sigma_i\sigma_j}{T}\right)$$



- For a given $\vec{\sigma}$ an edge is sat:
  if $J_{ij} > 0$ & $\sigma_i\sigma_j = 1$ of $J_{ij} < 0$ & $\sigma_i\sigma_j = -1$
- Circle is frustrated if the number of negative edges is odd.
  (N.B. Frustration of a circle is invariant wrt $\vec{\sigma}$.)
- Equivalent configurations, $\vec{\sigma}$ and $-\vec{\sigma}$, have the same weight
- Introduce dual graph, Γ^*. A vertex of Γ^* correspondent to a frustrated (unfrustrated) face is odd (even).

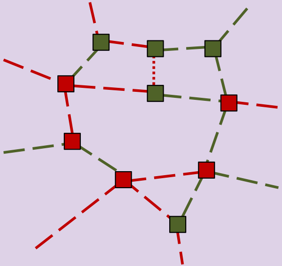
$$E = -\sum_{(ij)} J_{ij}\sigma_i\sigma_j = -\sum_{(ij)} |J_{ij}| + 2 \sum_{\text{unsat edges}} |J_{ij}|$$

From Ising to Dimer (II)



- The graphical transformations are invariant, i.e. they do not depend on the original configuration of $\vec{\sigma}$ (colors of vertexes/edges of the dual lattice stay/change)
- Spin glass Ising model on a planar graph is reduced to the Dimer Matching model on an auxiliary planar graph with all nodes of the connectivity three or smaller (graph. transformations in two steps)

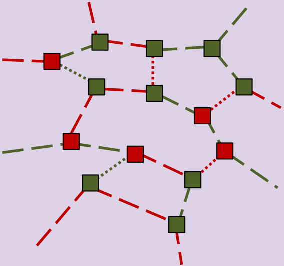
From Ising to Dimer (II)



- New edges (dotted) have zero energy
- Color of a new edge is fixed by colors of the vertexes it neighbors

- The graphical transformations are invariant, i.e. they do not depend on the original configuration of $\vec{\sigma}$ (colors of vertexes/edges of the dual lattice stay/change)
- Spin glass Ising model on a planar graph is reduced to the Dimer Matching model on an auxiliary planar graph with all nodes of the connectivity three or smaller (graph. transformations in two steps)

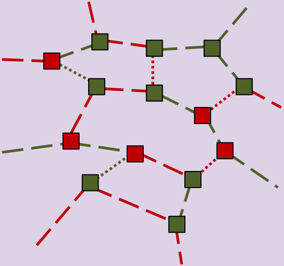
From Ising to Dimer (II)



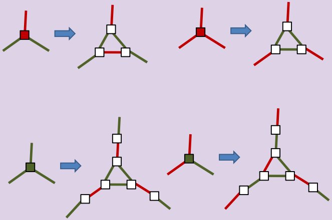
- New edges (dotted) have zero energy
- Color of a new edge is fixed by colors of the vertexes it neighbors

- The graphical transformations are invariant, i.e. they do not depend on the original configuration of $\vec{\sigma}$ (colors of vertexes/edges of the dual lattice stay/change)
- Spin glass Ising model on a planar graph is reduced to the Dimer Matching model on an auxiliary planar graph with all nodes of the connectivity three or smaller (graph. transformations in two steps)

From Ising to Dimer (II)



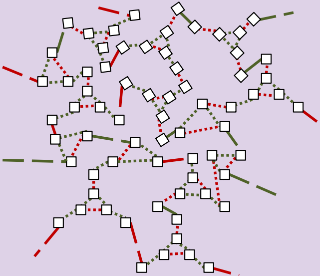
- New edges (dotted) have zero energy
- Color of a new edge is fixed by colors of the vertexes it neighbors



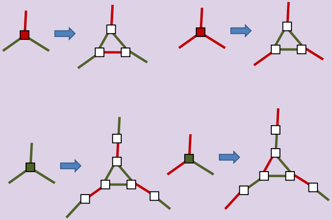
- All copies of an even vertex are even, one copy of an odd vertex is odd and the others are even
- Infinite node should also be 3-plicated (not shown)

- The graphical transformations are invariant, i.e. they do not depend on the original configuration of $\vec{\sigma}$ (colors of vertexes/edges of the dual lattice stay/change)
- Spin glass Ising model on a planar graph is reduced to the Dimer Matching model on an auxiliary planar graph with all nodes of the connectivity three or smaller (graph. transformations in two steps)

From Ising to Dimer (II)



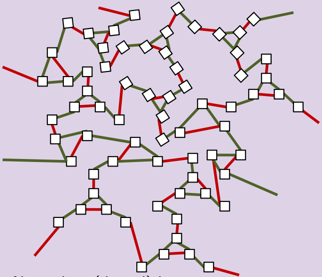
- New edges (dotted) have zero energy
- Color of a new edge is fixed by colors of the vertexes it neighbors



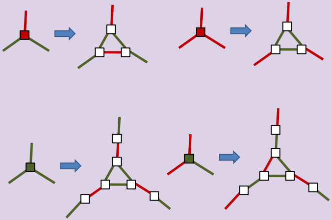
- All copies of an even vertex are even, one copy of an odd vertex is odd and the others are even
- Infinite node should also be 3-plicated (not shown)

- The graphical transformations are invariant, i.e. they do not depend on the original configuration of $\vec{\sigma}$ (colors of vertexes/edges of the dual lattice stay/change)
- Spin glass Ising model on a planar graph is reduced to the Dimer Matching model on an auxiliary planar graph with all nodes of the connectivity three or smaller (graph. transformations in two steps)

From Ising to Dimer (II)



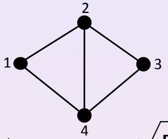
- New edges (dotted) have zero energy
- Color of a new edge is fixed by colors of the vertexes it neighbors



- All copies of an even vertex are even, one copy of an odd vertex is odd and the others are even
- Infinite node should also be 3-plicated (not shown)

- The graphical transformations are invariant, i.e. they do not depend on the original configuration of $\vec{\sigma}$ (colors of vertexes/edges of the dual lattice stay/change)
- Spin glass Ising model on a planar graph is reduced to the Dimer Matching model on an auxiliary planar graph with all nodes of the connectivity three or smaller (graph. transformations in two steps)

Pfaffian solution of the Matching problem

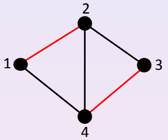


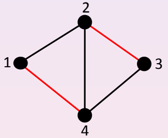
$$Z = z_{12}z_{34}+z_{14}z_{23} = \sqrt{\mathrm{Det}\hat{A}} = \mathrm{Pf}[\hat{A}]$$

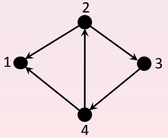
$$\hat{A} = \begin{pmatrix} 0 & -z_{12} & 0 & -z_{14} \\ z_{12} & 0 & z_{23} & -z_{24} \\ 0 & -z_{23} & 0 & z_{34} \\ z_{14} & z_{24} & -z_{34} & 0 \end{pmatrix}$$

Odd-face rule

Direct edges of the graph such that for every internal face the number of edges oriented clockwise is odd







Fermion/Grassman Representation

Planar Spin Glass and Dimer Matching Problems

The Pfaffian formula with the "odd-face" orientation rule extends to any planar graph thus proving constructively that

- Counting weighted number of dimer matchings on a planar graph is easy
- Calculating partition function of the spin glass Ising model on a planar graph is easy

N.B.

- Adding magnetic field to planar, non-planar geometry, or non-binary alphabet makes the spin-glass problem difficult
- Dimer-monomer matching is difficult even in the planar case
- Planar-Graph Decomposition [*Globerson, Jaakola '06*] is an example of an approximate algorithm that could be constructed for "nearly" planar problems

Single-connected Partition

Chertkov, Chernyak, Teodorescu '08

- Functions are on vertexes; variables (binary) are on edges
- Vertexes are of degree three (not restrictive)

Loop Series = BP + sum over generalized loops

$$Z = Z_0 \cdot z, \;\; z \equiv \left(1 + \sum_C \prod_{a \in C} \mu_{a,\bar{a}_C}\right), \;\; \mu_{a,\bar{a}_C} \equiv \frac{\tilde{\mu}_{a,\bar{a}_C}}{\prod_{b \in C}^{(a,b)\in C} \sqrt{1 - m_{ab}(C)}}$$

$$m_{ab} = \sum_{\sigma_{ab}} \sigma_{ab} b_{ab}(\sigma_{ab}), \;\; \tilde{\mu}_{a,\bar{a}_C} = \sum_{\vec{\sigma}_a} \prod_{b \in \bar{a}_C} (\sigma_{ab} - m_{ab}) b_a(\vec{\sigma}_a),$$

Single-Connected Partition

$$Z_s = Z_0 \cdot z_s, \quad z_s = 1 + \sum_{C \in \mathcal{G}}^{\forall a \in C, \; |\delta(a)|_C = 2} r_C,$$

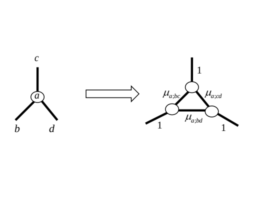
Is the Single-Connected Partition on a planar graph summable (easy)?

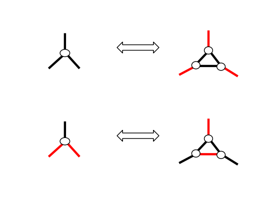
Single-connected Partition (II)

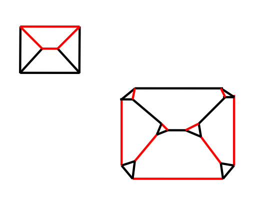
Chertkov, Chernyak, Teodorescu '08

Reduction to the dimer-matching model on an auxiliary graph

- reminiscent of the Fisher's transformation







$$z_s = \sum_{\vec{\pi}} \prod_{(a,b)\in\mathcal{G}_e} (\mu_{ab})^{\pi_{ab}} \prod_a \delta\left(\sum_b^{(a,b)\in\mathcal{G}_e} \pi_{ab}, 1\right)$$

- z_s is a Pfaffian on a planar graph [Kasteleyn] → EASY !

Problems Reducible to Single-Connected Partition

Generic planar problem is difficult

A planar problem is easy if

the factor functions satisfy

$$\forall\ a \in \mathcal{G}: \quad \sum_{\vec{\sigma}_a} f_a(\vec{\sigma}_a) \prod_{b}^{(a,b)\in\mathcal{E}} \left(\exp\left(\eta_{ab}\sigma_{ab}\right)\left(\sigma_{ab} - \tanh\left(\eta_{ab}+\eta_{ba}\right)\right)\right) = 0.$$

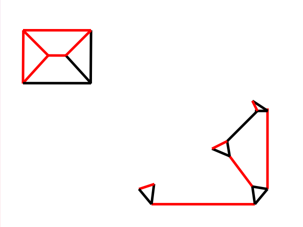
where η are messages from a BP solution for the model

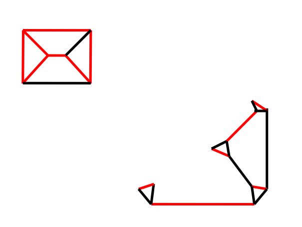
Loop Series as a Pfaffian Series

$$z=\sum_{\Psi} z_{\Psi}\prod_{a\in\Psi}^{|\bar{a}|=3}\mu_{a;\bar{a}},\quad z_{\Psi}=\text{Pf}\left(\hat{A}_{\Psi}\right)=\sqrt{\text{Det}\left(\hat{A}_{\Psi}\right)}$$

All z_Ψ are computationally tractable (Pfaffians)

- "Exclude" the fully connected part (vertexes of degree three within the generalized loop and adjusted edges)
- "Extend" the remaining graph (part of the generalized loop)





- N.B. $\hat{A}_\Psi$ is not simply a minor of $\hat{A}$ (exclusion changes signs)

Some Future Challenges

- Search for new approximate schemes for intractable planar problems
- Perturbative exploration of a larger set of intractable non-planar problems which are close, in some sense, to planar problems (e.g. in the spirit of Globerson, Jaakkola '06)
- Extension to other Graph Minor excluded families of graphs, e.g. only K_5 excluded, or only $K_{3,3}$ excluded
- Extension to q-ary case. Loop Tower. Potts model, etc.
- Possible Relation to Integrable Hierarchies and Quantum Computations
- Disorder-averaged planar problems

Thank You!

All papers are available at http://cnls.lanl.gov/∼chertkov/pub.htm

Bibliography List

Gauge Fixing & Bethe Free Energy

in the spirit of Yedidia, Freeman, Weiss '01

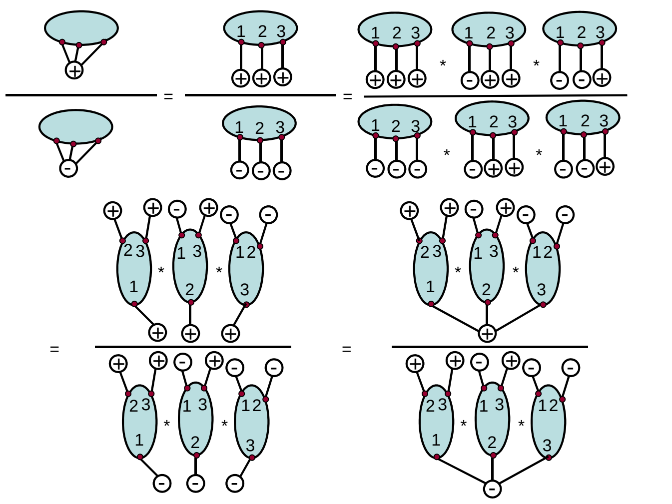
Minimize: $\Phi_B = \sum_a \sum_{\vec\sigma_a} b_a(\vec\sigma_a) \ln\left(\frac{b_a(\vec\sigma_a)}{f_a(\vec\sigma_a)}\right) - \sum_{(ab)} \sum_{\sigma_{ab}} b_{ab}(\sigma_{ab}) \ln b_{ab}(\sigma_{ab})$

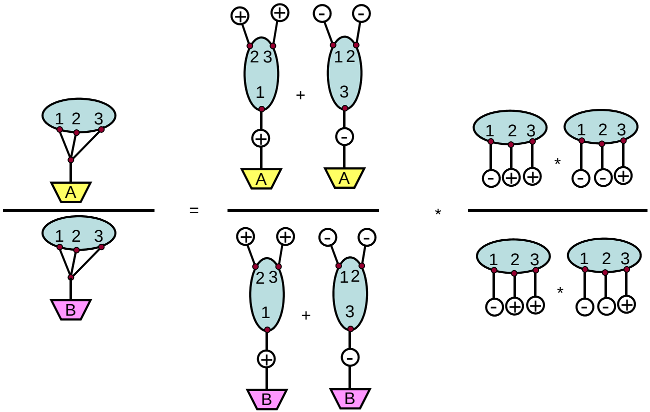
under the conditions: $\forall a$ & $\forall c \in a$

$$0 \le b_a(\vec\sigma_a), b_{ac}(\sigma_{ac}) \le 1$$
$$\sum_{\vec\sigma_a} b_a(\vec\sigma_a) = 1$$
$$b_{ac}(\sigma_{ac}) = \sum_{\vec\sigma_a \setminus \sigma_{ac}} b_a(\vec\sigma_a)$$

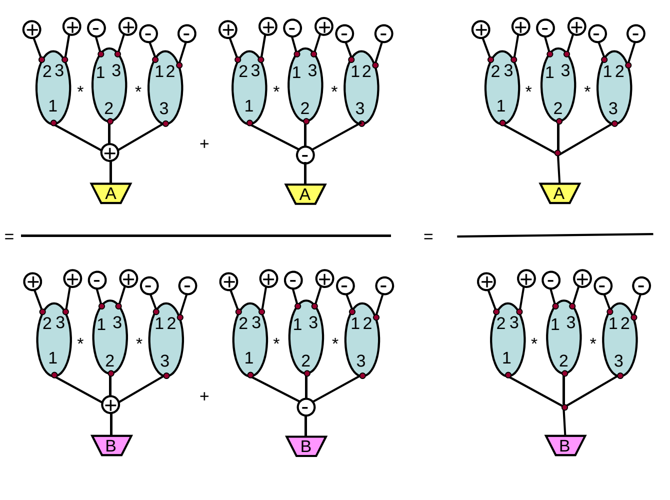
- $\mathcal{L}_B = \Phi_B + \sum\limits_{(ab)} [\sum\limits_{\sigma_{ab}} \ln(\epsilon_{ab}(\sigma_{ab}))(b_{ab}(\sigma_{ab}) - \sum\limits_{\vec\sigma_a \setminus \sigma_{ab}} b_a(\vec\sigma_a)) + \sum\limits_{\sigma_{ba}} \ln(\epsilon_{ba}(\sigma_{ba}))(b_{ab}(\sigma_{ba}) - \sum\limits_{\vec\sigma_b \setminus \sigma_{ba}} b_b(\vec\sigma_b))]$
- Finding extremum of the Bethe Lagrangian with respect to beliefs, b_{ab} and b_a and expressing the result in terms of ϵ: $\mathcal{L}_B(b,\epsilon) \Rightarrow \mathcal{F}_B(\epsilon)$
- $\mathcal{F}_B(\epsilon)\Big|_{\{\forall (a,b):\ \sum_{\sigma_{ab}} \epsilon_{ab}(\sigma_{ab})\epsilon_{ba}(\sigma_{ab})=1\}} = \mathcal{F}_0(\epsilon) = -\ln(Z(\epsilon))$

Variational approach

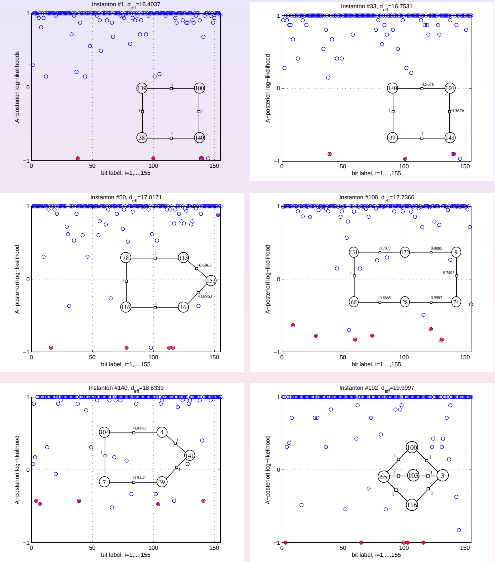




1 2 3
A
1 2 3
B
=
2 3
1
+
1 2
3
A
A
2 3
1
+
1 2
3
B
B
*
1 2 3
*
1 2 3
1 2 3
*
1 2 3



2 3
1
1 3
2
1 2
3
*
+
=
A
B


Instanton #1, d_eff=16.4037
Instanton #33, d_eff=16.7531
Instanton #50, d_eff=17.0171
Instanton #100, d_eff=17.7366
Instanton #140, d_eff=18.8339
Instanton #192, d_eff=19.9997
A−posteriori log−likelihood
A−posteriori log−likelihoods
bit label, i=1,...,155

Back

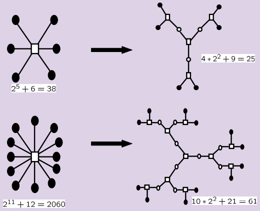
Reducing complexity of LP

Complexity of the bare LP grows exponentially with check degree

Current solutions:

- Adaptive LP (Taghavi, Siegel '06)
- BP-style relaxation of LP (Vontobel, Koetter '06)

Dendro-trick = Graph Modification (our solution) Chertkov,Stepanov'07



- MAP solutions are identical
- Set of Pseudo-codewords are identical
- Instanton spectra are very alike, $\approx$

Grassmann (fermion) Calculus for Pfaffians

Grassman Variables on Vertexes

$$\forall(a,b) \in \mathcal{G}_e: \quad \theta_a\theta_b + \theta_b\theta_a = 0 \quad \int d\theta = 0, \quad \int \theta d\theta = 1$$

Pfaffian as a Gaussian Berezin Integral over the Fermions

$$\int \exp\left(-\frac{1}{2}\vec{\theta}^t \hat{A}\vec{\theta}\right) d\vec{\theta} = \mathrm{Pf}(\hat{A}) = \sqrt{\det(\hat{A})}$$

◂ Pfaffian Formula

My papers on ... (A)

Loop Calculus, Series, Tower

- M. Chertkov, V. Chernyak, R. Teodorescu, "Belief Propagation and Loop Series on Planar Graphs", submitted to JSTAT, `arxiv.org/abs/0802.3950`.
- M. Chertkov, *Exactness of Belief Propagation for Some Graphical Models with Loops*, submitted to the Journal of Machine Learning, `arxiv.org/abs/0801.0341`
- V. Chernyak and M. Chertkov, "Loop Calculus and Belief Propagation for q-ary Alphabet: Loop Tower," *Proceedings of IEEE ISIT 2007*, June 2007, Nice, `arXiv:cs.IT/0701086`.
- M. Chertkov and V. Chernyak, "Loop series for discrete statistical models on graphs," JSTAT/2006/P06009, `arXiv:cond-mat/0603189`.
- M. Chertkov and V. Chernyak, "Loop Calculus in Statistical Physics and Information Science," *Phys. Rev. E*, **73**, 065102(R) (2006), `arXiv:cond-mat/0601487`.

Reducing the Error Floor

- M. Chertkov, "Reducing the Error Floor", invited talk at the *Information Theory Workshop '07 on "Frontiers in Coding"*, September 2-6, 2007.
- M. Chertkov and V. Chernyak, "Loop Calculus Helps to Improve Belief Propagation and Linear Programming Decodings of Low-Density-Parity-Check Codes," invited talk at 44^{th} *Allerton Conference*, September 27-29, 2006, Allerton, IL, `arXiv:cs.IT/0609154`.

All papers are available at http://cnls.lanl.gov/∼chertkov/pub.htm

My papers on ... (B)

Instantons & Pseudo-Codewords. Analyzing the Error-Floor.

- M. Chertkov and M. Stepanov, "Searching for low weight pseudo-codewords," invited talk at the *2007 Information Theory and Application Workshop, proceedings*, ITA CALIT2, UCSD, `arXiv:cs.IT/0702024`.
- M. Chertkov and M. Stepanov, "Pseudo-codeword Landscape," *Proceedings of IEEE ISIT 2007*, June 2007, Nice, `arXiv:cs.IT/0701084`.
- M. Chertkov and M. Stepanov, "An Efficient Pseudo-Codeword-Search Algorithm for Linear Programming Decoding of LDPC Codes," to appear in *IEEE Transactions on Information Theory*, `arXiv:cs.IT/0601113`.
- M. Stepanov and M. Chertkov, "Instanton analysis of Low-Density-Parity-Check codes in the error-floor regime," *Proceeding of IEEE ISIT 2006*, July 2006 Seattle, `arXiv:cs.IT/0601070`.
- M. Stepanov and M. Chertkov, "The error-floor of LDPC codes in the Laplacian channel," Proceedings of 43rd Allerton Conference (September 28-30, 2005, Allerton, IL), `arXiv:cs.IT/0507031`.
- M. Stepanov, V. Chernyak, M. Chertkov and B. Vasic, "Diagnosis of weakness in error correction: a physics approach to error floor analysis," Phys. Rev. Lett. **95**, 228701 (2005),`cond-mat/0506037`.
- V. Chernyak, M. Chertkov, M. Stepanov and B. Vasic, "Error correction on a tree: An instanton approach" , Phys. Rev. Lett. **93**, 198702-1 (2004).

Other subjects related to LDPC+ decoding

- J. A. Anguita, M. Chertkov, B. Vasic and M. A. Neifeld, "Bethe-Free-Energy Based Decoding of Low-Density Parity-Check Codes on Partial Response Channels," submitted to *IEEE Journal of Selected Areas in Communications*.
- M. Stepanov and M. Chertkov, "Improving convergence of belief propagation decoding," *Proceedings of 44th Allerton Conference*, September 27-29, 2006, Allerton, IL, `arXiv:cs.IT/0607112`.