Diverse Worlds of Belief Propagation

Michael Chertkov

Center for Nonlinear Studies & Theory Division, LANL

October 8, 2009 Santa Fe Institute Colloquium

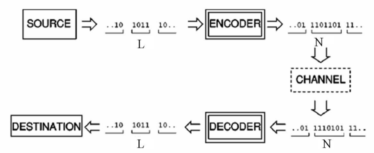
Outline

1. Three Problems
   - Error Correction
   - Particle Tracking
   - Power Grid
2. One Method
   - Common Language (Graphical Models) & Common Questions
   - Message Passing/ Belief Propagation
   - ... and beyond ... (theory)
3. Results
   - Error Correction
   - Particle Tracking
   - Power Grid

Error Correction

Cell phone Hard disk Optical disk Fiber

Scheme:



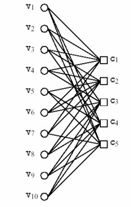
Example of Additive White Gaussian Channel:

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

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

- **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

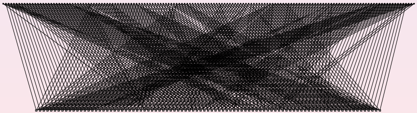
Low Density Parity Check Codes



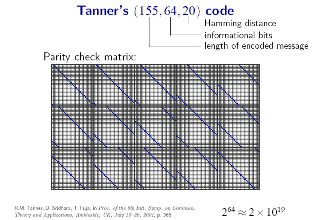
- N bits, M checks, $L = N - M$ information bits
  example: $N = 10$, $M = 5$, $L = 5$
- 2^L codewords of 2^N possible patterns
- Parity check: $\hat{H}\mathbf{v} = \mathbf{c} = \mathbf{0}$
  example:

$$\hat{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

- LDPC = graph (parity check matrix) is sparse



Almost a tree! [Sparse Graph/Code]



Decoding as Inference

Statistical Inference

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

Maximum Likelihood

Marginal Probability

$$\arg\max_{\boldsymbol{\sigma}} \mathcal{P}(\boldsymbol{\sigma}|\mathbf{x}) \qquad \arg\max_{\sigma_i} \sum_{\boldsymbol{\sigma}\setminus\sigma_i} \mathcal{P}(\mathbf{x}|\boldsymbol{\sigma})$$

Exhaustive search is generally expensive:
complexity of the algorithm $\sim 2^N$

Decoding as Inference

Statistical Inference

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

Maximum Likelihood

Marginal Probability

$$\arg\max_{\boldsymbol{\sigma}} \mathcal{P}(\boldsymbol{\sigma}|\mathbf{x}) \qquad \arg\max_{\sigma_i} \sum_{\boldsymbol{\sigma}\setminus\sigma_i} \mathcal{P}(\mathbf{x}|\boldsymbol{\sigma})$$

Exhaustive search is generally expensive:
complexity of the algorithm $\sim 2^N$

Decoding as Inference

Statistical Inference

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

Maximum Likelihood

Marginal Probability

$$\arg\max_{\boldsymbol{\sigma}} \mathcal{P}(\boldsymbol{\sigma}|\mathbf{x}) \qquad \arg\max_{\sigma_i} \sum_{\boldsymbol{\sigma}\setminus\sigma_i} \mathcal{P}(\mathbf{x}|\boldsymbol{\sigma})$$

Exhaustive search is generally expensive:
complexity of the algorithm $\sim 2^N$

Decoding as Inference

Statistical Inference

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

$$\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_N), \quad N \text{ finite}, \quad \sigma_i = \pm 1 \text{ (example)}$$

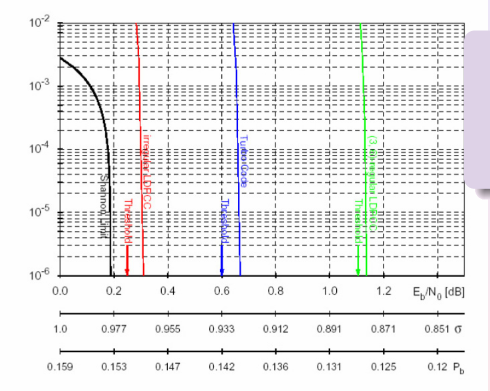
Maximum Likelihood

Marginal Probability

$$\arg\max_{\boldsymbol{\sigma}} \mathcal{P}(\boldsymbol{\sigma}|\mathbf{x}) \qquad\qquad \arg\max_{\sigma_i} \sum_{\boldsymbol{\sigma}\setminus\sigma_i} \mathcal{P}(\mathbf{x}|\boldsymbol{\sigma})$$

Exhaustive search is generally expensive:
complexity of the algorithm $\sim 2^N$

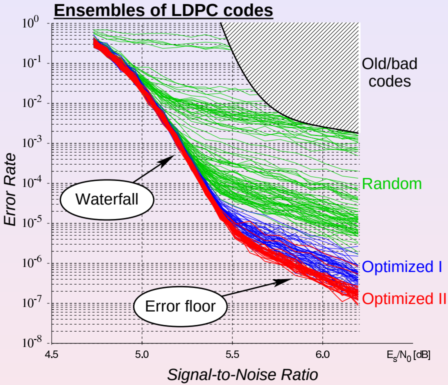
Shannon Transition



Existence of an efficient
MESSAGE PASSING
[belief propagation] decoding
makes LDPC codes special!

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

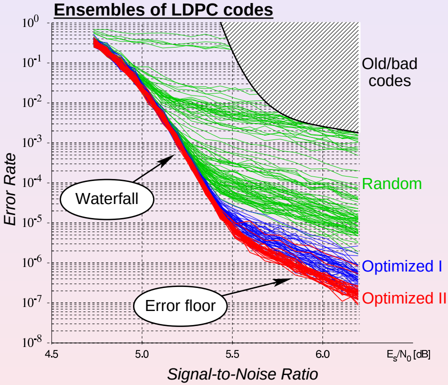
Error-Floor



- T. Richardson '03 (EF)
- Density evolution does not apply (to EF)

- 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 $\lesssim 10^{-8}$

Error-floor Challenges



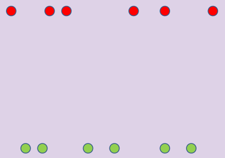
- Understanding the Error Floor (Inflection point, Asymptotics), Need an efficient method to analyze error-floor
- Improving Decoding
- Constructing New Codes

$\boxed{\text{Dance in Turbulence [movie]}}$

Learn the flow from tracking particles

Learning via Statistical Inference

Two images



Particle Image Velocimetry & Lagrangian Particle Tracking [standard solution]

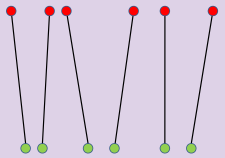
- Take snapshots often = Avoid trajectory overlap
- Consequence = A lot of data
- Gigabit/s to monitor a two-dimensional slice of a $10cm^3$ experimental cell with a pixel size of $0.1mm$ and exposition time of $1ms$
- Still need to “learn” velocity (diffusion) from matching

New twist [MC, L.Kroc, F. Krzakala, L. Zdeborova, M. Vergassola '09]

- Take fewer snapshots = Let particles overlap
- Put extra efforts into Learning/Inference
- Use our (turbulence community) knowledge of Lagrangian evolution
- Focus on learning (rather than matching)

Learning via Statistical Inference

Two images



Particle Image Velocimetry & Lagrangian Particle Tracking [standard solution]

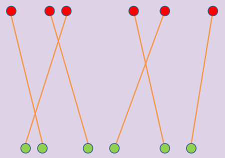
- Take snapshots often = Avoid trajectory overlap
- Consequence = A lot of data
- Gigabit/s to monitor a two-dimensional slice of a $10cm^3$ experimental cell with a pixel size of $0.1mm$ and exposition time of $1ms$
- Still need to “learn” velocity (diffusion) from matching

New twist [MC, L.Kroc, F. Krzakala, L. Zdeborova, M. Vergassola '09]

- Take fewer snapshots = Let particles overlap
- Put extra efforts into Learning/Inference
- Use our (turbulence community) knowledge of Lagrangian evolution
- Focus on learning (rather than matching)

Learning via Statistical Inference

Two images



Particle Image Velocimetry & Lagrangian Particle Tracking [standard solution]

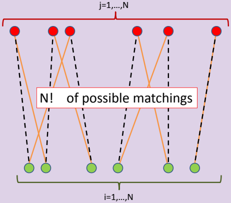
- Take snapshots often = Avoid trajectory overlap
- Consequence = A lot of data
- Gigabit/s to monitor a two-dimensional slice of a $10cm^3$ experimental cell with a pixel size of $0.1mm$ and exposition time of $1ms$
- Still need to "learn" velocity (diffusion) from matching

New twist [MC, L.Kroc, F. Krzakala, L. Zdeborova, M. Vergassola '09]

- Take fewer snapshots = Let particles overlap
- Put extra efforts into Learning/Inference
- Use our (turbulence community) knowledge of Lagrangian evolution
- Focus on learning (rather than matching)

Learning via Statistical Inference

Two images



And after all we actually don't need matching. Our goal is to LEARN THE FLOW.

Particle Image Velocimetry & Lagrangian Particle Tracking [standard solution]

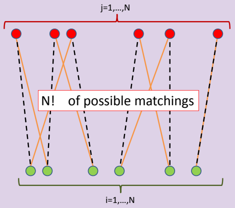
- Take snapshots often = Avoid trajectory overlap
- Consequence = A lot of data
- Gigabit/s to monitor a two-dimensional slice of a $10cm^3$ experimental cell with a pixel size of $0.1mm$ and exposition time of $1ms$
- Still need to "learn" velocity (diffusion) from matching

New twist [MC, L.Kroc, F. Krzakala, L. Zdeborova, M. Vergassola '09]

- Take fewer snapshots = Let particles overlap
- Put extra efforts into Learning/Inference
- Use our (turbulence community) knowledge of Lagrangian evolution
- Focus on learning (rather than matching)

Learning via Statistical Inference

Two images



And after all we actually don't need matching. Our goal is to LEARN THE FLOW.

Particle Image Velocimetry & Lagrangian Particle Tracking [standard solution]

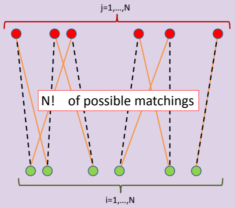
- Take snapshots often = Avoid trajectory overlap
- Consequence = A lot of data
- Gigabit/s to monitor a two-dimensional slice of a $10cm^3$ experimental cell with a pixel size of $0.1mm$ and exposition time of $1ms$
- Still need to "learn" velocity (diffusion) from matching

New twist [MC, L.Kroc, F. Krzakala, L. Zdeborova, M. Vergassola '09]

- Take fewer snapshots = Let particles overlap
- Put extra efforts into Learning/Inference
- Use our (turbulence community) knowledge of Lagrangian evolution
- Focus on learning (rather than matching)

Learning via Statistical Inference

Two images



And after all we actually don't need matching. Our goal is to LEARN THE FLOW.

Particle Image Velocimetry & Lagrangian Particle Tracking [standard solution]

- Take snapshots often = Avoid trajectory overlap
- Consequence = A lot of data
- Gigabit/s to monitor a two-dimensional slice of a $10cm^3$ experimental cell with a pixel size of $0.1mm$ and exposition time of $1ms$
- Still need to "learn" velocity (diffusion) from matching

New twist [MC, L.Kroc, F. Krzakala, L. Zdeborova, M. Vergassola '09]

- Take fewer snapshots = Let particles overlap
- Put extra efforts into Learning/Inference
- Use our (turbulence community) knowledge of Lagrangian evolution
- Focus on learning (rather than matching)

Lagrangian Dynamics under the Viscous Scale

Plausible (for PIV) Modeling Assumptions

- Particles are normally seed with mean separation few times smaller than the viscous scale.
- The Lagrangian velocity at these scales is spatially smooth.
- Moreover the velocity gradient, $\hat{s}$, at these scales and times is frozen (time independent).

Batchelor (diffusion + smooth advection) Model

- Trajectory of i's particles obeys: $d\mathbf{r}_i(t)/dt = \hat{s}\mathbf{r}_i(t) + \boldsymbol{\xi}_i(t)$
- $\mathrm{tr}(\hat{s}) = 0$ - incompressible flow
- $\langle \xi_i^\alpha(t_1)\xi_j^\beta(t_2)\rangle = \kappa\delta_{ij}\delta^{\alpha\beta}\delta(t_1 - t_2)$

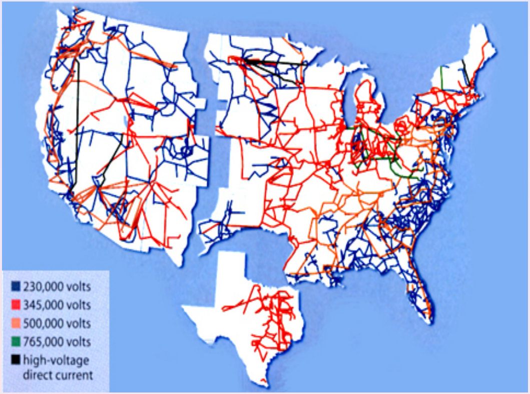
Inference & Learning

Main Task: Learning parameters of the flow and of the medium

- Given positions of N identical particles at $t=0$ and $t=1$: $\forall i,j=1,\cdots,N,\quad \mathbf{x}_i=\mathbf{r}_i(0)$ and $\mathbf{y}^j=\mathbf{r}_j(1)$
- To output MOST PROBABLE values of the flow, $\hat{s}$, and the medium, κ, characterizing the inter-snapshot span: $\boldsymbol{\theta}=(\hat{s};\kappa)$. [Matchings are hidden variables.]

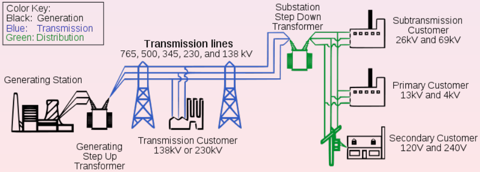
Sub-task: Inference [reconstruction] of Matchings

- Given parameters of the medium and the flow, $\boldsymbol{\theta}$
- To reconstruct Most Probable matching between identical particles in the two snapshots ["ground state"]
- Even more generally - Probabilistic Reconstruction: to assign probability to each matchings and evaluate marginal probabilities ["magnetizations"]



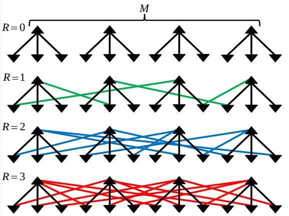
US power grid

The greatest Engineering Achievement of the 20th century

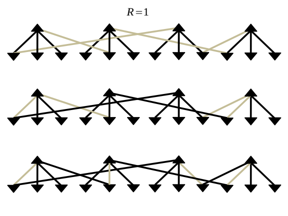


will require smart revolution in the 21st century

Optimization & Control of Power Grid [L. Zdeborova, A. Decelle, MC '09]



A: $R = 0; 1; 2; 3$. Graph samples. Ancillary connections to foreign generators/consumers are shown in color.



B: $R = 1$. Three valid (SAT) configurations (shown in black, the rest is in gray) for a sample graph shown in Fig. A.

- Can the anchillary lines (redundancy) help?
- Design and efficient switching algorithm for finding SAT solution.

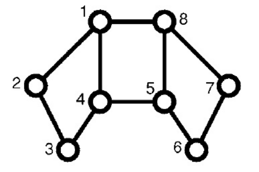
Outline

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

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

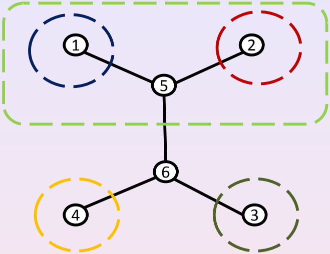
Easy & Difficult Boolean Problems

EASY

- Any graphical problems on a tree (Bethe-Peierls, Dynamical Programming, BP, TAP 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, Peierls '36



$$Z_{51}(\sigma_{51}) = f_1(\sigma_{51}), \quad Z_{52}(\sigma_{52}) = f_2(\sigma_{52}),$$
$$Z_{63}(\sigma_{63}) = f_3(\sigma_{63}), \quad Z_{64}(\sigma_{64}) = f_4(\sigma_{64})$$
$$Z_{65}(\sigma_{56}) = \sum_{\vec{\sigma}_5 \setminus \sigma_{56}} f_5(\vec{\sigma}_5) Z_{51}(\sigma_{51}) Z_{52}(\sigma_{52})$$
$$Z = \sum_{\vec{\sigma}_6} f_6(\vec{\sigma}_6) Z_{63}(\sigma_{63}) Z_{64}(\sigma_{64}) Z_{65}(\sigma_{65})$$

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

Belief Propagation Equations

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

e.g. Thouless-Anderson-Palmer (1977) Eqs.

Belief Propagation (BP) and Message Passing

- Apply what is exact on a tree (the equation) to other problems on graphs with loops [heuristics ... but a good one]
- To solve the system of N equations is EASIER then to count (or to choose one of) 2^N states.

Bethe Free Energy formulation of BP [Yedidia, Freeman, Weiss '01]

Minimize the Kubblack-Leibler functional

$$\mathcal{F}\{b(\{\sigma\})\} \equiv \sum_{\{\sigma\}} b(\{\sigma\}) \ln \frac{b(\{\sigma\})}{\mathcal{L}(\{\sigma\})}$$

Difficult/Exact

under the following "almost variational" substitution" for beliefs:

$$b(\{\sigma\}) \approx \frac{\prod_i b_i(\boldsymbol{\sigma}_i) \prod_j b^j(\boldsymbol{\sigma}^j)}{\prod_{(i,j)} b_i^j(\sigma_i^j)} \quad \text{[tracking]}$$

Easy/Approximate

- Message Passing is a (graph) Distributed Implementation of BP
- Graphical Models = the language

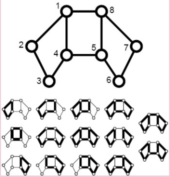
Beyond BP [MC, V. Chernyak '06-'09 + J. Johnson '09]

Only mentioning briefly today

Loop Calculus/Series:

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

each r_C is expressed solely in terms of BP marginals

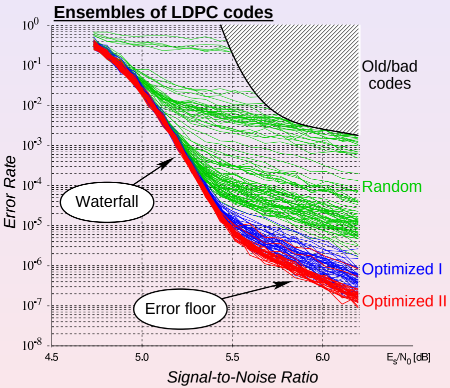


- BP is a Gauge. There are other interesting choices of the Gauges.
- Loop Series for Gaussian Integrals, Fermions, etc.
- Planar and Surface Graphical Models which are Easy [alas dimer]. Holographic Computations. Matchgates. Quantum Theory of Computations.
- Orbit product for Gaussian GM [J. Johnson's SFI coll in three week]

Outline

1. Three Problems
   - Error Correction
   - Particle Tracking
   - Power Grid
2. One Method
   - Common Language (Graphical Models) & Common Questions
   - Message Passing/ Belief Propagation
   - ... and beyond ... (theory)
3. Results
   - Error Correction
   - Particle Tracking
   - Power Grid

Error-floor Challenges



- Understanding the Error Floor (Inflection point, Asymptotics), Need an efficient method to analyze error-floor
- ... i.e. an efficient method to analyze **rare-events [BP failures]** $\Rightarrow$

Optimal Fluctuation (Instanton) Approach for Extracting Rare but Dominant Events

Ed was unlucky enough to find the needle in the haystack!

Optimal Fluctuation (Instanton) Approach for Extracting Rare but Dominant Events

Ed was unlucky enough to find the needle in the haystack!

You were right: There's a needle in this haystack...

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

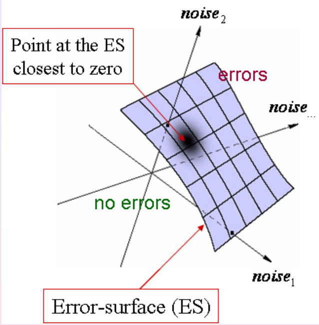
Instanton = optimal conf of the noise

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

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

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

Instantons are decoded to Pseudo-Codewords



Instanton-amoeba

= optimization algorithm

Stepanov, et.al '04,'05

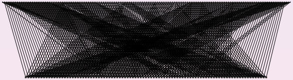
Stepanov, Chertkov '06

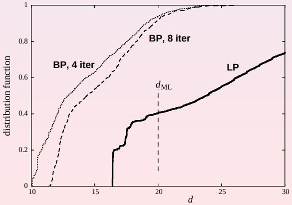
Efficeint Instanton Search Algorithm

[MC, M. Stepanov '07; MC,MC, S. Chillapagari, B. Vasic '08-'09]

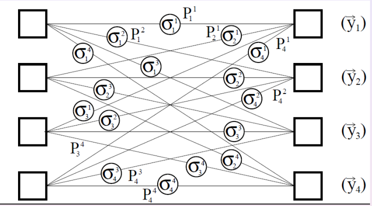
$$\text{BER} \approx \max_{\text{noise}} \overbrace{\min_{\text{output}} \text{Weight(noise;output)}}^{\text{decoding=BP,LP}} \Bigg|_{\text{Error Surface}}$$

- Developed Efficient Alg. for LP-Instanton Search. The output is the spectra of the dangerous pseudo-codewords
- Started to design Better Decoding = Improved LP/BP +
- Started to design new codes





Tracking Particles as a Graphical Model



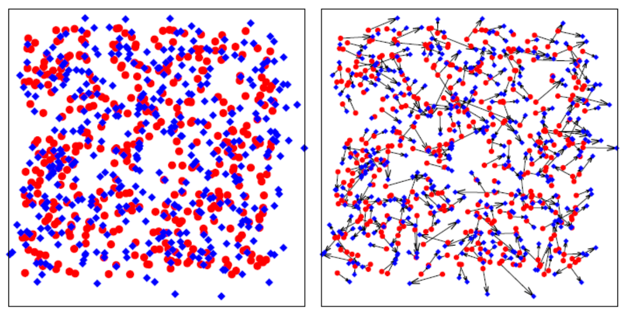
$$\mathcal{L}(\{\sigma\}|\boldsymbol{\theta}) = C\left(\{\sigma\}\right) \prod_{(i,j)} \left[P_i^j\left(\mathbf{x}_i, \mathbf{y}^j|\boldsymbol{\theta}\right)\right]^{\sigma_i^j}$$

$$C\left(\{\sigma\}\right) \equiv \prod_j \delta\left(\sum_i \sigma_i^j, 1\right) \prod_i \delta\left(\sum_j \sigma_i^j, 1\right)$$

Surprising Exactness of BP for ML-assignement

- Exact Polynomial Algorithms (auction, Hungarian) are available for the problem
- Generally BP is exact only on a graph without loops [tree]
- In this [Perfect Matching on Bipartite Graph] case it is still exact in spite of many loops!! [Bayati, Shah, Sharma '08], also Linear Programming/TUM interpretation [MC '08]

Can you guess who went where?



- N particles are placed uniformly at random in a d-dimensional box of size $N^{1/d}$
- Choose $\boldsymbol{\theta} = (\kappa, \mathbf{s})$ in such a way that after rescaling, $\hat{s}^* = \hat{s}N^{1/d}$, $\kappa^* = \kappa$, all the rescaled parameters are $O(1)$.
- Produce a stochastic map for the N particles from the original image to respective positions in the consecutive image.

- $N = 400$ particles. 2D.
- $\hat{s} = \begin{pmatrix} a & b-c \\ b+c & a \end{pmatrix}$
- Actual values: $\kappa = 1.05$, $a^* = 0.28$, $b^* = 0.54$, $c^* = 0.24$
- Output of OUR LEARNING algorithm: [accounts for multiple matchings !!] $\kappa_{BP} = 1$, $a_{BP} = 0.32$, $b_{BP} = 0.55$, $c_{BP} = 0.19$ [within the "finite size" error]

Combined Message Passing with Parameters' Update

Fixed Point Equations for Messages

- BP equations: $\overline{h}^{i\to j}=-\frac{1}{\beta}\ln\sum_{k\neq j}P_i^k e^{\beta \underline{h}^{k\to i}}$; $\underline{h}^{j\to i}=-\frac{1}{\beta}\ln\sum_{k\neq i}P_k^j e^{\beta \overline{h}^{k\to j}}$
- BP estimation for $Z_{BP}(\boldsymbol{\theta})=Z(\boldsymbol{\theta}|\mathbf{h}$ solves BP eqs. at $\beta=1)$
- MPA estimation for $Z_{MPA}(\boldsymbol{\theta})=Z(\boldsymbol{\theta}|\mathbf{h}$ solves BP eqs. at $\beta=\infty)$

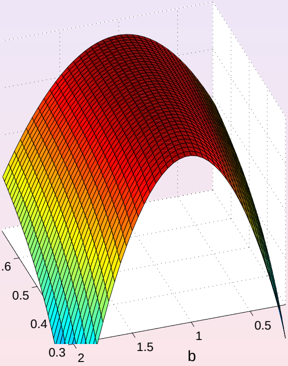
$$Z(\boldsymbol{\theta}|\mathbf{h};\beta)=\sum_{(ij)}\ln\left(1+P_i^j e^{\beta\overline{h}^{i\to j}+\beta\underline{h}^{j\to i}}\right)-\sum_i\ln\left(\sum_j P_i^j e^{\beta\underline{h}^{j\to i}}\right)-\sum_j\ln\left(\sum_i P_i^j e^{\beta\overline{h}^{i\to j}}\right)$$

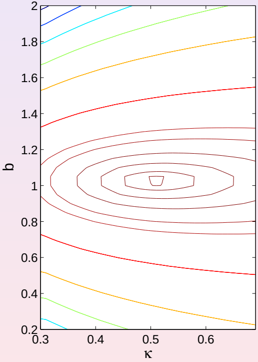
Learning: $\text{argmin}_{\boldsymbol{\theta}} Z(\boldsymbol{\theta})$

- Solved using Newton's method in combination with message-passing: after each Newton step, we update the messages
- Even though (theoretically) the convergence is not guaranteed, the scheme always converges
- Complexity [in our implementation] is $O(N^2)$, even though reduction to $O(N)$ is straightforward

Quality of the Predicition [is good]

2D. $a^* = b^* = c^* = 1$, $\kappa^* = 0.5$. $N = 200$.





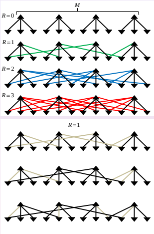
- The BP Bethe free energy vs κ and b. Every point is obtained by minimizing wrt a, c
- Perfect maximum at $b = 1$ and $\kappa = 0.5$ achieved at
  $a_{BP} = 1.148(1)$,
  $b_{BP} = 1.026(1)$,
  $c_{BP} = 0.945(1)$,
  $\kappa_{BP} = 0.509(1)$.

We also have a "random distance" model [ala random matching of Mezard, Parisi '86-'01] providing a theory support for using BP in the reconstruction/learning algorithms.

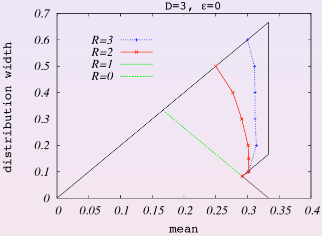
We are working on

- Applying the algorithm to real particle tracking in turbulence experiments
- Extending the approach to learning multi-scale velocity field and possibly from multiple consequential images
- Going beyond BP [improving the quality of tracking]

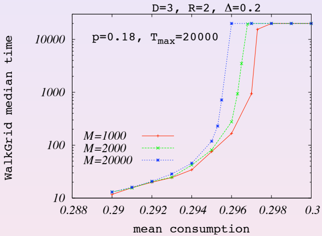
Message-Passing Switching Algorithm for the grid model with ancillary lines



SAT-UNSAT transition



Algorithm Performance [in SAT phase]



- To analyze the SAT-UNSAT transition We solved Cavity Equations (averaged BP) with Population Dynamics Algorithm
- We developed WalkGrid (greedy search) algorithm which finds SAT-switching efficiently

We are working on

- Application of the approach to more realistic grids
- Extending the story beyond "the commodity flow" approach towards accounting for AC/DC specifics of the power flows
- Switching vs Contingency. Off-line games. Control Algorithms.
- ... this research is a part of a new DR project at LANL on "Optimization and Control Theory for Smart Grids"

Bottom Line

- Applications of Belief Propagation (and its distributed iterative realization, Message Passing) are diverse and abundant
- BP/MP is also advantageous, thanks to existence of very reach and powerful tree-like, sparse analysis techniques [physics, CS, statistics]
- BP/MP has great theory and application potential for improvements [account for loops]
- BP/MP can be combined with other techniques (e.g. Markov Chain, planar inference, etc) and in this regards it represents the tip of the iceberg called "Science of Algorithms"

References

http://cnls.lanl.gov/~chertkov/pub.htm