Approximate Message Passing Tutorial

SUNDEEP RANGAN, ALYSON K. FLETCHER, PHIL SCHNITER

STATISTICAL PHYSICS AND MACHINE LEARNING WORKSHOP,
CARGESE, CORSICA, FRANCE, 28 AUGUST 2018

NYU TANDON SCHOOL OF ENGINEERING

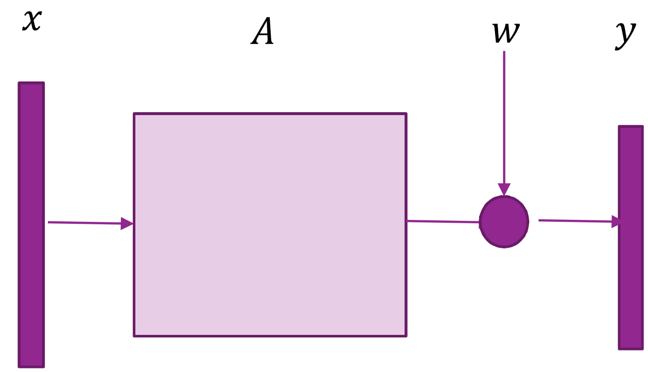
NYU WIRELESS

Outline

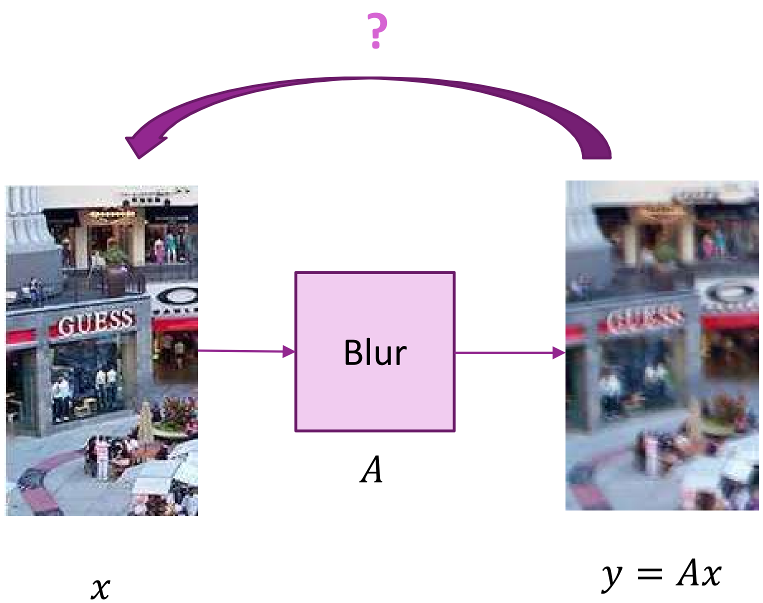
- AMP and Compressed Sensing
- Proximal Operators and ISTA
- State Evolution for AMP
- Bayes Denoising, Optimality and the Replica Method
- Belief Propagation and Factor Graphs
- AMP Derivation from Belief Propagation
- Convergence, Fixed Points and Stability
- Extensions: Vector AMP
- Thoughts on What is Next

Linear Inverse Problems

- Model:
$$y = Ax + w$$
  - x = unknown vector
  - w = "noise"
- Problem: Estimate x from A and y
- Many applications:
  - Linear regression with prior on weights
  - Compressed sensing
  - Image processing
  - …
- Will look at this problem and more complex variants



Example 1: Image Reconstruction



http://www.digitalphotopix.com/unbelievable/photo-deblur/
Article on Photoshop

- Recover original image x
- y = degraded / transformed image
- Operator A represents
  - Blurring
  - Measurement distortion
  - …
- Problem: Recover original x from y

Example 2: Multiple Linear Regression

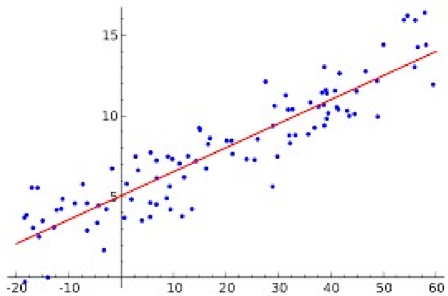
- Given data samples $(x_i, y_i), i = 1, \ldots, N$
  - Vector data: $x_i = (x_{i1}, \ldots, x_{id}), d =$number of features
- Problem: Fit a linear model

$$y_i = w_0 + w_1 x_{i1} + \cdots + w_d x_{id} + \epsilon_i$$

- Write in matrix form

$$y = Aw + \epsilon, \qquad A = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{Nd} & \cdots & x_{Nd} \end{bmatrix}$$

- Estimate weight vector w from data



Unconstrained Least Squares Estimation

- Most common method for linear inverse problems is least squares estimation:

$$\hat{x} = \arg\min_{x} \frac{1}{2} \|y - Ax\|^2 = (A^T A)^{-1} A^T y$$

  - Computationally simple, easy to analyze, interpretable results
- Standard LS is unconstrained: Minimization above is all possible x
- But, in many problems, we have prior knowledge on x
  - Example: x is a natural image
- How do we incorporate prior knowledge?

Regularized Least Squares Estimation

- Regularized LS: Add a penalty term:

$$\hat{x} = \arg\min_x \frac{1}{2}\|y - Ax\|^2 + \phi(x)$$

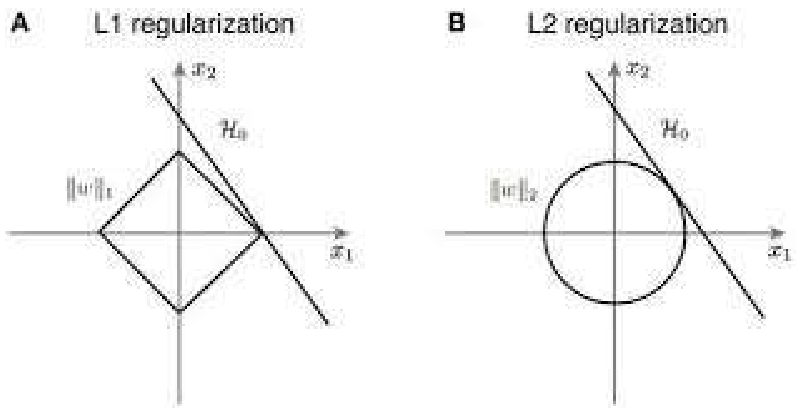
- $\phi(x)$ = Regularization function:
  - Penalizes values that are less likely or desirable as solutions
- Two common simple regularization functions:
  - L2 (called ridge regression in statistics): $\phi(x) = \lambda\|x\|_2^2 = \lambda\sum\left|x_j\right|^2$
  - L1 (called LASSO in statistics): $\phi(x) = \lambda\|x\|_1 = \lambda\sum|x_j|$
- Both functions force x to be close to zero (or some mean value if known)

L1 Regularization and Sparsity

- L1 regularized least-squares:

$$\hat{x} = \arg\min_{x} \frac{1}{2} \|y - Ax\|^2 + \lambda \|x\|_1$$

- L1 regularization favors sparse x
- Makes many coefficients exactly zero



L1 Regularization for Model Selection

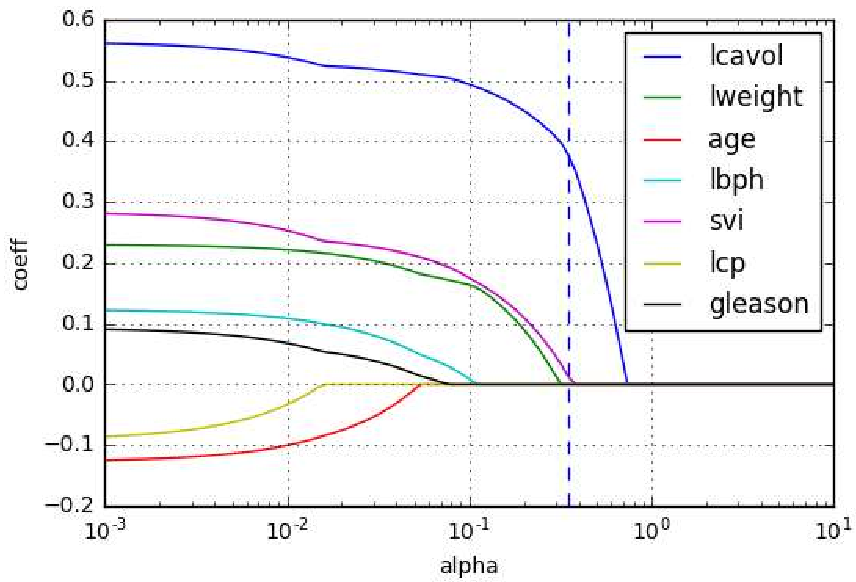
- Linear regression: Find weights w for

$$y_i = w_0 + w_1 x_{i1} + \cdots + w_d x_{id} + \epsilon_i$$

- Often need to perform model selection:
  - Number of features may be large
  - Only a few features are likely to be relevant
  - But, don't know which ones a priori
- Use LASSO estimation to find relevant features

$$\hat{w} = \arg\min_{x} \frac{1}{2}\|y - Aw\|^2 + \lambda \|w\|_1$$

  - Regularization tries to find a sparse weight vector
  - Many coefficients can be set to zero



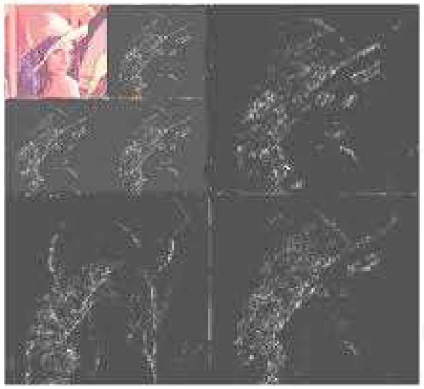
LASSO solutions for PSA level prediction
Path as a function of $\alpha = \lambda N$

L1 Estimation in Image Recovery

- Image x is often sparse in a transform domain
  - $u = Tx$ is sparse
  - Many transforms: Gradient operators, wavelet, …
  - All exploit that edges are sparse in natural images
- Use L1-regularization on transform components:

$$\hat{x} = \arg\min_{x} \frac{1}{2} \|y - Ax\|^2 + \lambda \|Tx\|_1$$

- Example: Total variation (TV) denoising
  - T= horizontal and vertical difference in pixels



Wavelet transform

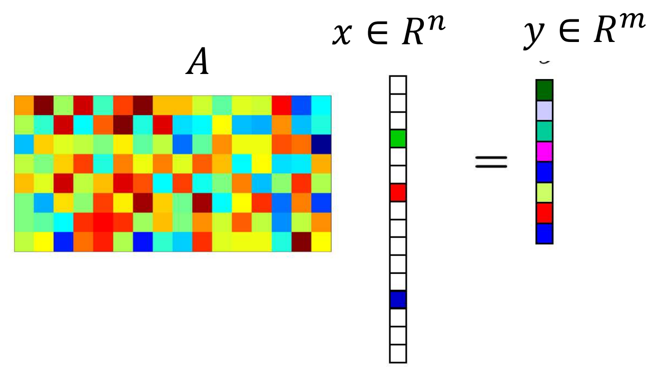
Transform is sparse

Compressed Sensing

- Revival of interest in L1 methods
  - [Donoho 06], [Candes, Romberg, Tao 06]
- Key observation:
  - Suppose x is sparse (e.g. $\|x\|_0 \leq k \ll n$)
  - Can recover x from underdetermined $y = Ax$
  - More unknowns than measurements (i.e. A is fat)
- Typical scaling of measurements:

$$m \geq Ck \log n$$

  - Requires incoherence of matrix with vector
  - Satisfied with random matrices with high probability



k non-zero elements

Compressed Sensing and its Challenges

- Significant work beginning 2006:
  - Scaling laws on measurements to recover “true” sparse vector
  - Fast algorithms
  - Successful applications, esp. in MRI

- Challenges:
  - Most analyses could only provide bounds. May be conservative.
  - Methods were often specific to L1 recovery. Difficult to generalized to other inverse problems
  - Hard to find theoretical optimal estimates
  - Ex: What is the minimum MSE: $E||\hat{x} - x_0||^2$ for a true vector x_0 with some noise model?
  - When can we achieve the optimal estimate and how?

Approximate Message Passing

- Benefits: For certain random matrices:
  - Very fast convergence
  - Can be precisely analyzed
  - Testable conditions for optimality
  - Can be extended to more complex models (parametric uncertainty, multilayer models, …)
- Problems / research questions: For general problems
  - Algorithm can diverge
  - Requires significant tuning
  - Requires precise specification of problem (partially solved)
- Arose out of study in compressed sensing
  - But, theory may provide insights to more complex models in inference today

AMP History

- Early work in CDMA detection for wireless
  - Boutros, Caire 02; Kabashima 02,03; Montanari & Tse (06), Guo & Wang (06), Tanaka & Okada (06)
- AMP re-discovered for compressed sensing
  - Donoho, Maleki, Montanari 09; Bayati-Montanari 10
- Connections to the replica method in statistical physics
  - Tanaka 04; Guo-Verdu 05;
  - Krzakala, F., Mézard, M., Sausset, F., Sun, Y. F., & Zdeborová, L. (2012)
  - Rangan, Fletcher, Goyal 09

AMP Extensions

- Since original paper, there have vast number of extensions
- GAMP: Generalized AMP for GLMs
- EM-(G)AMP: AMP with EM for parameter estimation
- BiGAMP: Bilinear AMP
- VAMP: Vector AMP that provides better convergence (end of this tutorial)
- ML-AMP and ML-VAMP: Multi-layer models
- Many more…

Outline

- AMP and Compressed Sensing
- Proximal Operators and ISTA
- State Evolution for AMP
- Bayes Denoising, Optimality and the Replica Method
- Belief Propagation and Factor Graphs
- AMP Derivation from Belief Propagation
- Convergence, Fixed Points and Stability
- Extensions: Vector AMP
- Thoughts on What is Next

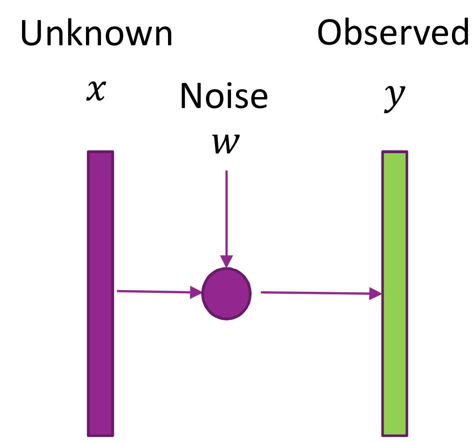
Proximal Operators

- Denoising problem: Given measurement $y = x + w$ estimate x
- Suppose we use a regularized estimator:

$$\hat{x} = \arg\min_x \frac{1}{2\tau}\|y - x\|^2 + \phi(x)$$

- Special case of regularized LS with $A = I$
  - We will look at adding general A matrix later
- The solution to this denoising problem is called the proximal operator

$$\text{Prox}_\phi(y,\tau) \coloneqq \arg\min_x \left[\frac{1}{2\tau}\|x - y\|^2 + \phi(x)\right]$$



Ex 1: Projections

- Proximal operators are generalizations of projections
- Suppose for some set C:

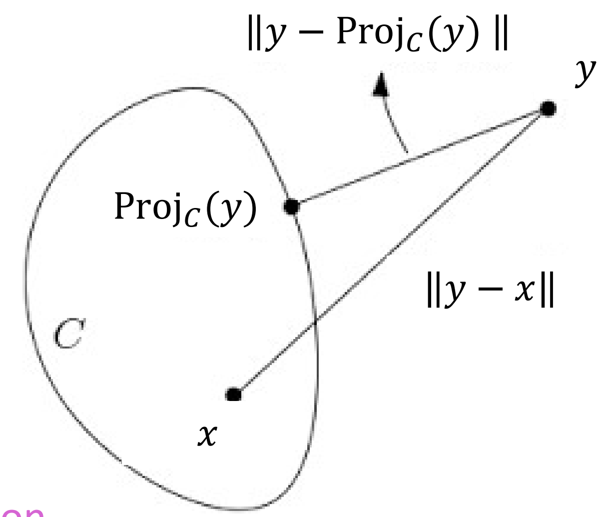
$$\phi(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C \end{cases}$$

- Then, proximal operator is a projection

$$\mathrm{Prox}_\phi(y,\tau) := \arg\min_x \left[\phi(x) + \frac{1}{2\tau}\|x-y\|^2\right]$$

$$= \arg\min_{x\in C}\|x-y\|^2 = \mathrm{Proj}_C(y)$$

- For general $\phi(\cdot)$, proximal operator $\mathrm{Prox}_\phi(y,\tau)$ is a "soft" projection
  - Finds points close to y where $\phi(x)$ is small



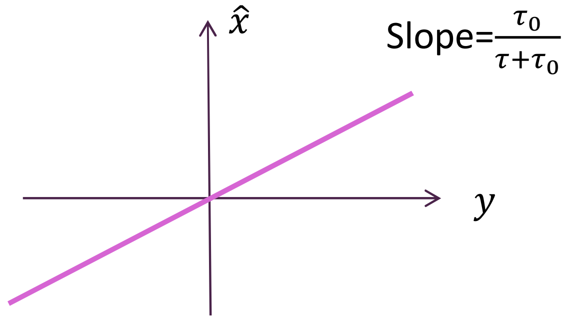
Ex 2: Scalar L2 / Quadratic Penalty

- Consider a quadratic penalty $\phi(x) := \frac{1}{2\tau_0}(x - x_0)^2$
  - Denoising is equivalent to a Gaussian prior
- Then, proximal operator is a linear function:

$$\text{Prox}_\phi(y,\tau) := \arg\min_x \left[\phi(x) + \frac{1}{2\tau}(x-y)^2\right]$$
$$= \frac{1}{\tau+\tau_0}[\tau x_0 + \tau_0 y]$$

  - Convex combination of observed y and prior x_0
- When $x_0 = 0$, get a shrinkage to zero:

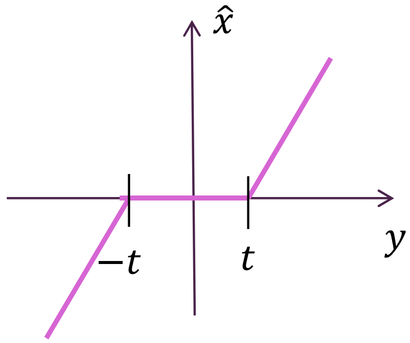
$$\text{Prox}_\phi(y,\tau) = \frac{\tau_0}{\tau+\tau_0} y$$



Ex 3: Scalar L1 Penalty

- L1 penalty: $\phi(x) = \lambda|x|$
  - Scalar LASSO problem
- Proximal operator is a soft-threshold

$$\mathrm{Prox}_\phi(y,\tau) = \begin{cases} y - t & y > t \\ 0 & |y| \le t \\ y + t & y < -t \end{cases}, \qquad t = \tau\lambda$$



- Can result in exactly zero solutions

Ex 4: Separable Penalties

- Suppose $\phi(x)$ is separable:

$$\phi(x) = \sum_{i=1}^{N} \phi_i(x_i)$$

- Example: L1 and L2 penalties
  - $\phi(x) = \|x\|_2^2 = \sum |x_i|^2$
  - $\phi(x) = \|x\|_1 = \sum |x_i|$
- Then, proximal operator applies componentwise

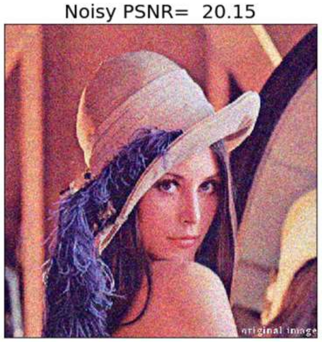
$$\hat{x} = \mathrm{Prox}_{\phi}(y, \tau) \Leftrightarrow \hat{x}_i = \mathrm{Prox}_{\phi_i}(y_i, \tau)$$

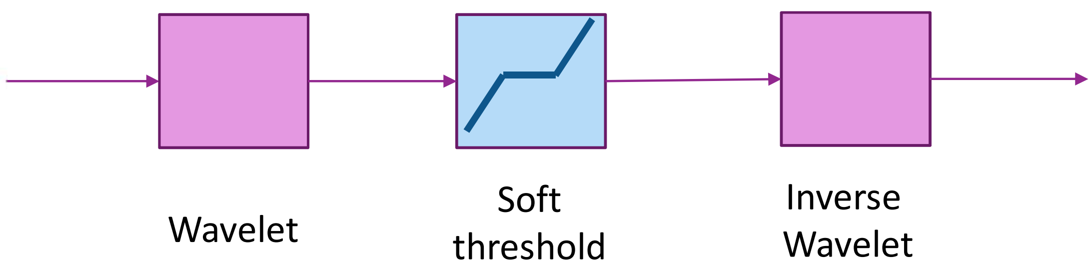
  - Apply proximal operator on each component

Ex 5: Wavelet Image Soft-Thresholding

- Suppose penalty is applied in wavelet domain: $\phi(x) = h(Wx)$,
  - W = orthogonal wavelet transform
- Often use L1 penalty: $h(z) = \begin{cases} 0 & \text{if } z_n \text{ is a scale coefficient} \\ \lambda|z_n| & \text{if } z_n \text{ is a detail coefficient} \end{cases}$
- Proximal operator can be applied in wavelet domain: $\mathrm{Prox}_\phi(y,\tau) = W^{-1}\mathrm{Prox}_h(Wy,\tau)$
  - Use fact that W is orthogonal

Noisy PSNR= 20.15





Soft-Thresh PSNR= 27.35

Regularized LS Estimation

- Now, return to regularized LS problem:

$$\hat{x} = \arg\min_x \frac{1}{2}\|y - Ax\|^2 + \phi(x)$$

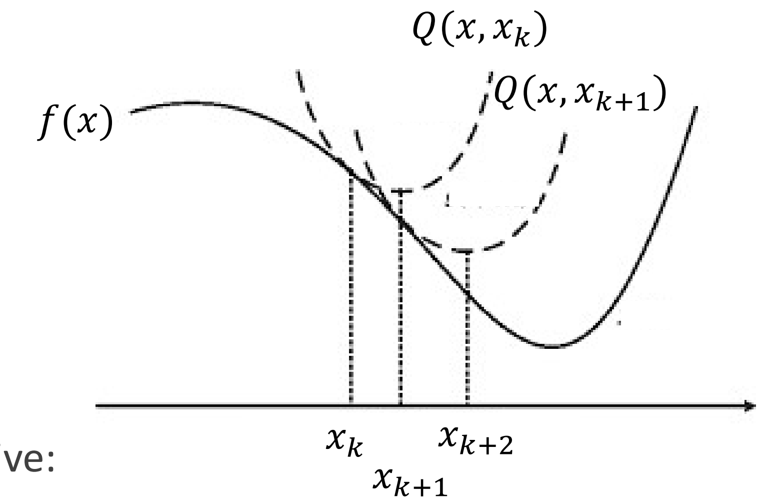
  - For LASSO: $\phi(x) = \lambda\|x\|_1$

- Challenges:
  - No closed form solution when $A \neq I$
  - Objective is non-smooth
  - Cannot directly apply gradient descent

- ISTA: Iterative Soft Thresholding Algorithm
  - Key idea: Break problems into sequence of proximal problems
  - Based on majorization minimization (next slide)

Majorization-Minimization

- Suppose minimizing $f(x)$ is hard to minimize directly
- At each x_k, find a majorizing function $Q(x, x_k)$:
  - $Q(x, x_k) \geq f(x)$ for all x
  - $Q(x_k, x_k) = f(x_k)$
- Majorization-Minimization algorithm:
  Iteratively minimize the majorizing function:
  $$x_{k+1} = \arg\min_{x} Q(x, x_k)$$
- Theorem: MM monotonically decreases the true objective:
  $$f(x_k) = Q(x_k, x_k) \geq Q(x_{k+1}, x_k) \geq f(x_{k+1})$$



MM for Regularized LS

- Rewrite regularized LS with two components:

$$\hat{x} = \arg\min_{x}[g(x) + \phi(x)]$$

  - Smooth component: $g(x) \coloneqq \frac{1}{2}\|y - Ax\|^2$
  - Non-smooth but separable component: $\phi(x) = \lambda\|x\|_1$

- Define majorizing function:

$$Q(x, x_k) \coloneqq g(x_k) + \nabla g(x_k) \cdot (x - x_k) + \frac{1}{2\alpha}\|x - x_k\|^2 + \phi(x)$$

- Easy to verify two properties:
  - $Q(x_k, x_k) = g(x_k) + h(x_k) = f(x_k)$
  - If α is sufficiently small, $Q(x, x_k) \geq F(x_k)$ for all x

- MM Algorithm:

$$x_{k+1} = \arg\min_{x} Q(x, x_k) = \arg\min_{x}\left[\phi(x) + \nabla g(x_k) \cdot (x - x_k) + \frac{1}{2\alpha}\|x - x_k\|^2\right]$$

ISTA Algorithm

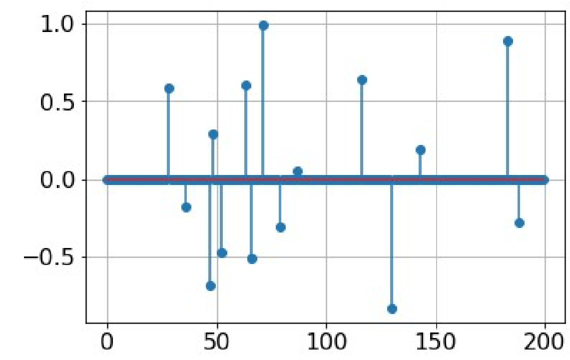
- From previous slide, MM algorithm is:

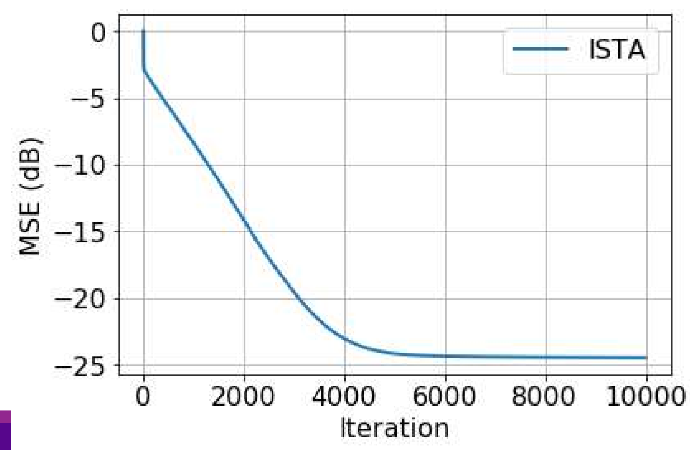
$$x_{k+1} = \arg\min_x Q(x, x_k) = \arg\min_x \left[\phi(x) + \nabla g(x_k) \cdot (x - x_k) + \frac{1}{2\alpha}\|x - x_k\|^2\right]$$

- Completing squares of MM Algorithm, we obtain two step algorithm
- Iterative Soft Threshold Algorithm:
  - Gradient step: $r_k = x_k - \alpha \nabla g(x_k) = r_k - \alpha A^T(Ax_k - y)$
  - Proximal step: $x_{k+1} = \mathrm{Prox}_\phi(r_k, \alpha)$
- Estimation is performed by sequence of proximal operators
  - For L1 / LASSO minimization, proximal operators are soft thresholds

Simple Compressed Sensing Example

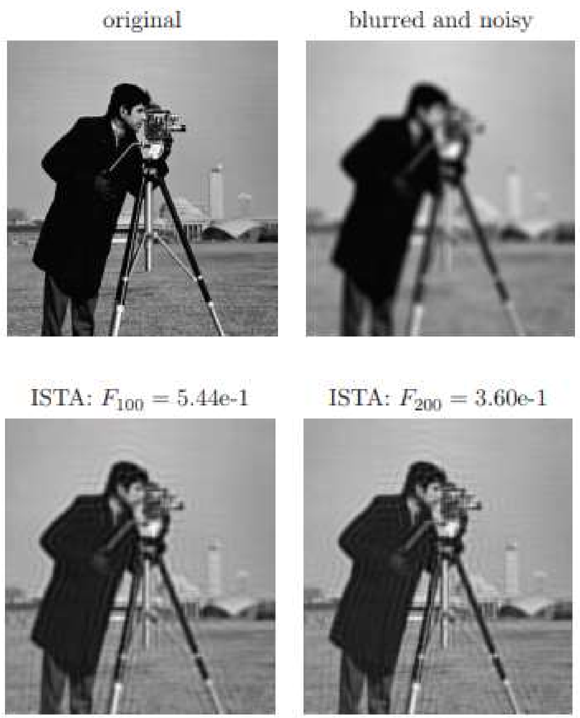
- Synthetic sparse signal:
  - $x_i = \begin{cases} 0 & \text{with prob } 1-\rho = 0.9 \\ N(0,1) & \text{with prob } \rho = 0.1 \end{cases}$
- Random measurement matrix $A \in R^{100\times200}$
  - $A_{ij} \sim N\left(0, \frac{1}{N}\right), \; N = 200$
  - Underdetermined
  - SNR = 30 dB noise
- Use LASSO estimate with $\lambda = 10$





Wavelet Image Deblurring with ISTA

- Measurements: $y = Ax + w$
  - $A =$ Gaussian blur
  - $w =$ Gaussian noise
- Denoiser uses 3 level Haar wavelet
- Decent results after 200 iterations



Beck, Amir, and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems." *SIAM journal on imaging sciences* 2.1 (2009): 183-202.

ADMM

- Proximal operators also appear in Alternating Direction Method of Multipliers (ADMM)
- Consider cost function: $f(x) = g(x) + h(x)$
- Variable splitting: Equivalent constrained minimization:

$$\min_{x,y} g(x) + h(y) \quad s.t.\ x = y$$

- Define augmented Lagrangian:

$$L(x, y, s) \coloneqq g(x) + h(y) + \alpha s^T(x - y) + \frac{\alpha}{2}\|x - y\|^2$$

- ADMM algorithm:
  - $\hat{x} = \arg\min_x L(x, \hat{y}, s)$
  - $\hat{y} = \arg\min_y L(\hat{x}, y, s)$
  - $s = s + x - y$

ADMM with Proximal Operators

- Complete squares as before
- ADMM can be rewritten with two proximal operators
  - $\hat{x} = \arg\min_x L(x, \hat{y}, s) = \mathrm{Prox}_g(\hat{y} - s; \alpha^{-1})$
  - $\hat{y} = \arg\min_y L(\hat{x}, y, s) = \mathrm{Prox}_h(\hat{x} + s; \alpha^{-1})$
  - $s = s + x - y$
- For LASSO problem:
  - $g(x) \coloneqq \frac{1}{2}\|b - Ax\|^2 \Rightarrow \mathrm{Prox}_g(u; \alpha^{-1}) = (A^T A + \alpha I)^{-1}(A^T b + \alpha u)$
  - $h(y) \coloneqq \lambda\|y\|_1 \Rightarrow \mathrm{Prox}_h(u; \alpha^{-1}) = T_t(u)$

Questions

- Proximal operator methods can guarantee convergence to a local minima
  - Appears to work in well in several key problems, esp. L1 regularized LS
  - Can be applied to non-smooth optimization
- Tremendous additional work (not covered here)
  - Rates of convergence
  - Interesting denoisers (low rank matrix recovery)
  - …
- But, several open questions:
  - How close are the resulting solutions to the correct value?
  - What is the “optimal” estimate?
  - Can we converge faster?
  - Describe this more…

Outline

- AMP and Compressed Sensing
- Proximal Operators and ISTA
- AMP and its State Evolution
- Bayes Denoising, Optimality and the Replica Method
- Belief Propagation and Factor Graphs
- AMP Derivation from Belief Propagation
- Convergence, Fixed Points and Stability
- Extensions: Vector AMP
- Thoughts on What is Next

ISTA

- Consider regularized LS problem:

$$\hat{x} = \arg\min_x \left[\frac{1}{2}\|y - Ax\|^2 + \phi(x)\right]$$

  - For LASSO, $\phi(x) = \lambda_1 \|x\|_1$

- ISTA algorithm from previous section:
  - $d_k = y - A\hat{x}_k$
  - $r_k = \hat{x}_k + \tau A^T d_k$
  - $\hat{x}_{k+1} = \text{Prox}_\phi(r_k, \tau)$

- Can we do better?

Approximate Message Passing

AMP

$$d_k = y - A\hat{x}_k + \frac{N}{M}\alpha_k d_{k-1}$$

$$r_k = \hat{x}_k + A^T d_k$$

$$\hat{x}_{k+1} = g_{in}(r_k, \theta^k)$$

$$\alpha_{k+1} = \langle g'_{in}(r_k, \theta^k)\rangle = \frac{1}{N}\sum_i \frac{\partial g_{in}(r_k, \theta^k)}{\partial r_{ki}}$$

Note: A and y must be scaled such that:

$$||A||_F^2 = N$$

ISTA

$$d_k = y - A\hat{x}_k$$
$$r_k = \hat{x}_k + \tau A^T d_k$$
$$\hat{x}_{k+1} = \mathrm{Prox}_\phi(r_k, \tau)$$

Key modifications for AMP

- Memory term in the residual update $\alpha_k d_{k-1}$
  - Acts as a momentum
  - Called the “Onsager term” in statistical physics
  - More on this soon
- Proximal operator replaced by general estimator
  - θ^k is an arbitrary parameter of the estimator

Fixed Points

- Theorem: At any fixed point of AMP with a proximal denoiser $\hat{x} = \mathrm{Prox}_{\phi}(r,\tau)$

$$\hat{x} = \arg\min_{x}[\lambda\phi(x) + g(x)], \qquad g(x) = \frac{1}{2}\|y - Ax\|^2, \qquad \lambda = (1-\alpha)\tau$$

- Proof: For any fixed point:
  - $d = y - A\hat{x} + \alpha d \Rightarrow d = \frac{y - A\hat{x}}{1-\alpha}$
  - $r = \hat{x} + A^T d = \hat{x} - \frac{1}{1-\alpha}\nabla g(\hat{x})$

- If we use a proximal denoiser: $\hat{x} = \mathrm{Prox}_{\phi}(r,\tau)$

$$\begin{aligned}\hat{x} &= \arg\min_{x}\left[\phi(x) + \frac{1}{2\tau}\|r - x\|^2\right] \\ &= \arg\min_{x}\left[\phi(x) + \frac{1}{(1-\alpha)\tau}\nabla g(\hat{x})^T(x - \hat{x}) + \frac{1}{2\tau}\|\hat{x} - x\|^2\right]\end{aligned}$$

Selecting the AMP step size

- AMP "solves" the regularized LS problem (when it converges):

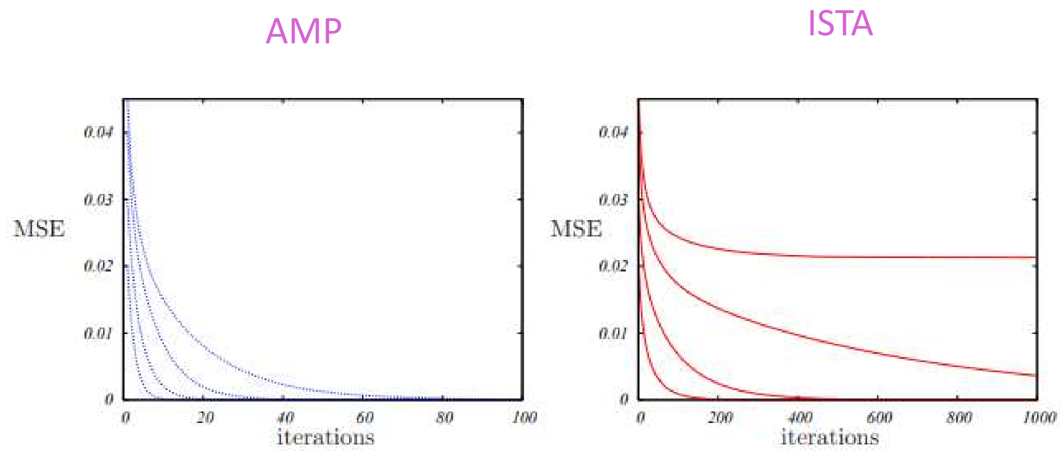
$$\hat{x} = \arg\min_x \left[\lambda\phi(x) + \frac{1}{2}\|y - Ax\|^2 \right]$$

- But, regularization parameter is computed implicitly: $\lambda = (1-\alpha)\theta$
- For LASSO problem:
  - Typically select threshold directly $t = c\frac{\|d_k\|}{\sqrt{N}}$ for some constant c
  - Constant c is larger when sparsity is greater
  - Tune c instead of λ

Compressed Sensing Example

- Sparse random vector example
  - $N = 8000, M = 1600$
  - $\|x\|_0 = \{800,1200,1600,1800\}$
  - $A_{ij} \sim N(0, \frac{1}{M})$

- AMP: Much faster convergence
  - Look at axes!



Montanari, "Graphical models concepts in compressed sensing." *Compressed Sensing: Theory and Applications* (2012)

Large System Limit

- When does AMP work so well? Why?
- Analysis of AMP is generally in a large system limit
- Sequence of problems $N \to \infty$ with $\lim_{N\to\infty} \frac{M}{N} = \beta$
- Assumptions: For every N:
  - There is a true vector x_0 and noise w with $y = Ax_0 + w$
  - Random matrix: $A \in R^{M\times N}$, $A_{ij} \sim N\left(0, \frac{1}{M}\right)$
  - Estimator $g(r,\tau)$ is Lipschitz in r and acts componentwise: $g(r,\tau)_i = g(r_i,\tau)$
  - Vectors x_0, $\hat{x}_j^0$ and w are i.i.d. and independent of A: $x_{0j} \sim X_0$, $w_i \sim W$, $\hat{x}_j^0 \sim \hat{X}^0$
- Note that due to normalization: $||A||_F^2 \approx N$

Scalar MSE Function

- Want to predict the asymptotic MSE in each iteration:

$$\eta^k := \lim_{N \to \infty} \frac{1}{N} \left\| \hat{x}^k - x_0 \right\|^2$$

- Define the scalar MSE function:

$$MSE(\nu, \theta) := E|g(X_0 + N(0, \nu), \theta) - X_0|^2,$$

  - Average error on a single component of the estimate $\hat{X} = g(R, \theta)$ with $R = X_0 + N(0, \nu)$
  - Estimation error under Gaussian noise

- Can be computed via scalar integrals $\Rightarrow$ Easy!
- We will show: $\eta^k = MSE(\nu^k, \theta^k)$ where ν^k can be computed recursively

Ex 1: MSE Function for an L2 Penalty

- Suppose that estimator is linear: $g(r,\theta) = \theta r$
  - Proximal estimator corresponding to a L2 / quadratic penalty
- Then MSE is given by:

$$\begin{aligned}\text{MSE}(\nu,\theta) &= E|g(X_0 + N(0,\nu),\theta) - X_0|^2 \\ &= E|\theta(X_0 + V) - X_0|^2 = (1-\theta)^2 E|X_0|^2 + \theta^2 \nu\end{aligned}$$

- If $E|X_0|^2$ and ν were known, we could optimize θ:

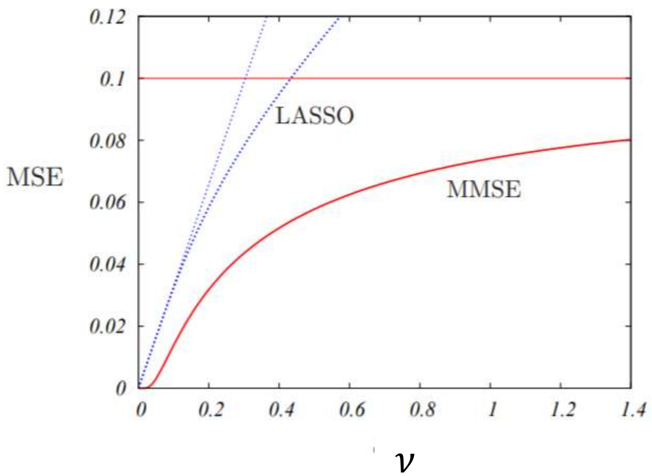
$$\hat{\theta} = \frac{E|X_0|^2}{E|X_0|^2 + \nu}, \qquad \min_{\theta} MSE\ (\nu,\theta) = \frac{\nu E|X_0|^2}{E|X_0|^2 + \nu}$$

  - Called the linear minimum MSE estimator (LMMSE) in signal processing

Ex 2: MSE Function for an L1 Penalty

- Suppose true signal is 3-point sparse $\mathrm{P}(X_0 = x) = \begin{cases} \rho/2 & x = 1 \\ 1-\rho & x = 0 \\ \rho/2 & x = -1 \end{cases}$ $\rho = 0.1$

- Two estimators:
  - LASSO with optimized λ
  - MMSE: $\hat{X} = \mathrm{E}(\mathrm{X}_0 | \mathrm{R} = \mathrm{X}_0 + \mathrm{N}(0, \nu))$



Montanari, "Graphical models concepts in compressed sensing." *Compressed Sensing: Theory and Applications* (2012)

State Evolution

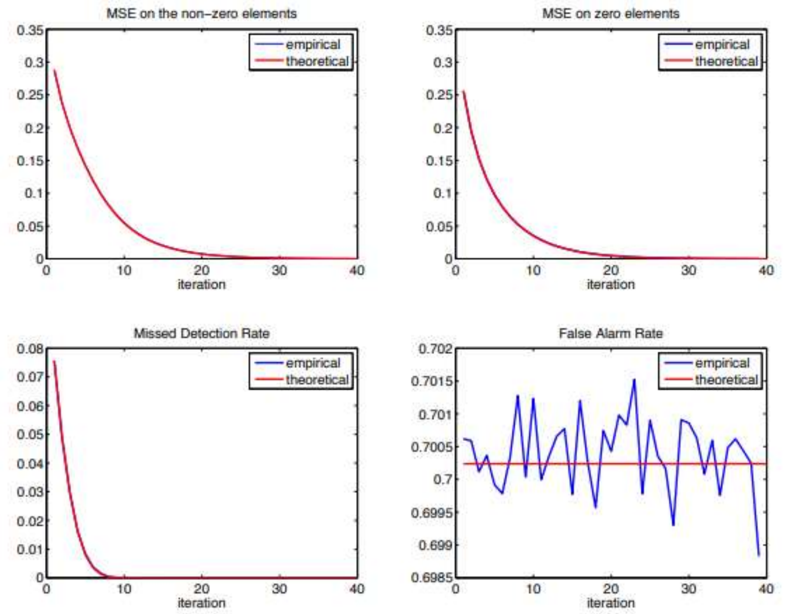
- Theorem: In the large system limit, the asymptotic MSE evolves as:

$$\eta^{k+1} = MSE\left(\frac{N}{M}\eta^k + E|W|^2, \theta^k\right), \qquad \eta^0 = E\left|\hat{X}^0 - X_0\right|^2$$

- In the LSL, MSE can be exactly predicted!
- Result is very general:
  - All i.i.d. densities on the true signal x_0
  - Separable Lipschitz estimators $\hat{x}^k = g(r^k, \theta^k)$ with any parameters θ^k
  - Can account for estimators of non-convex functions

State Evolution Example 1

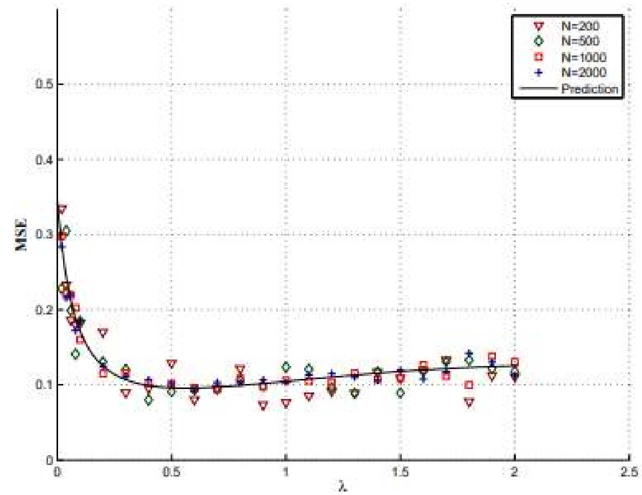
- Can predict the MSE per component exactly!
- Can also predict various statistics
  - eg. Missed detection, false alarm rate
  - Will show how to do this later
- Simulation parameters:
  - $x_{0j} = \begin{cases} 1 & \text{Prob} = \rho \\ 0 & \text{Prob} = 1-\rho \end{cases}, \ \rho = 0.045$
  - $N = 5000, \frac{\mathrm{M}}{N} = 0.3$
  - No noise



Donoho, Maleki, Montanari. "Message passing algorithms for compressed sensing: I. motivation and construction." *ITW 2010*

State Evolution Example 2

- Comparison of predicted limit of SE
  - Predicted = $\lim_{k\to\infty} \eta^k$
  - Simulated = measured MSE
- We obtain close to an exact match
  - Match improves as N increases
- Simulation parameters
  - x iid three point density
  - N varies, $\frac{M}{N} = 0.64$



Error Vectors

- Proof of the SE requires that we track certain error quantities
- AMP algorithm:
  - $d_k = y - A\hat{x}_k + \frac{N}{M}\alpha_k d_{k-1}$
  - $r_k = \hat{x}_k + A^T d_k$
  - $\hat{x}_{k+1} = g_{in}(r_k, \theta^k), \;\; \alpha_{k+1} = \langle g'_{in}(r_k, \theta^k) \rangle$
- Define the error vectors:
  - $u^k = \hat{x}^k - x_0$: Error after estimator
  - $q^k = r^k - x_0$: : Error before estimator
  - $v^k = d^k$: Output error with noise
  - $p^k = d^k - w = A(\hat{x}^k - x_0) - \alpha^k d^{k-1}$: Output prediction without noise
- Scalar error functions:
  - Output: $G_p(p, w, \tau) \coloneqq w - p$
  - Input: $G_q(q, w, \tau) \coloneqq g(q + x_0, \tau) - x_0$

Error System

- Simple algebra shows that error vectors evolve as the general recursion:
  - $p^k = Au^k - \lambda_u^k v^{k-1}$
  - $v^k = G_p(p^k, w_p, \theta_p^k),\ \lambda_v^k = \langle G_p'(p^k, w_p, \theta_p^k)\rangle$
  - $q^k = A^T v^k - \lambda_v^k u^k$
  - $u^{k+1} = G_q(q^k, w_q, \theta_q^k), \quad \lambda_u^k = \langle G_q'(q^k, w_q, \theta_q^k)\rangle$

- Simple structure: Each iteration involves:
  - Multiplication by A and A^T
  - Componentwise nonlinearities

Main Result: Scalar Equivalence

Error system:

- $p^k = Au^k - \lambda_u^k v^{k-1}$,
- $v^k = G_p\left(p^k, w_p, \theta_p^k\right)$, $\lambda_v^k = \left\langle G_p'\left(p^k, w_p, \theta_p^k\right)\right\rangle$
- $q^k = A^T v^k - \lambda_v^k u^k$,
- $u^{k+1} = G_q\left(q^k, w_q, \theta_q^k\right)$, $\lambda_u^k = \left\langle G_q'\left(q^k, w_q, \theta_q^k\right)\right\rangle$

=

Scalar system

- $P^k \sim N\left(0, \tau_p^k\right)$, $\tau_p^k = \frac{N}{M} E\left|U^k\right|^2$
- $V^k = G_p(P^k, W_p, \theta_p^k)$
- $Q^k \sim N\left(0, \tau_q^k\right)$, $\tau_q^k = E\left|V^k\right|^2$
- $U^{k+1} = G_q(Q^k, W_q, \theta_q^k)$

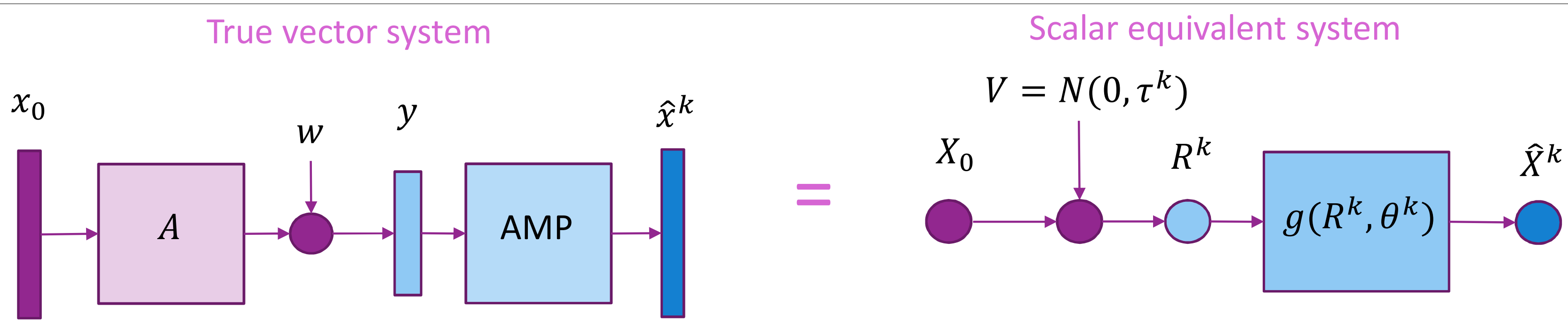
❑ Theorem [Bayati-Montanari 2010]: In the large system limit:

- Distribution of components error system vectors = distribution of scalar random variables

$$\lim_{N\to\infty} p^k = P^k, \lim_{N\to\infty} q^k = Q^k, \ldots$$

- Formal definition of convergence given below
- Shows errors are Gaussian
- AMP SE follows as special case

Scalar Equivalent Model



- ❑Each component of the vector system behaves like a simple scalar model
- ❑Equivalent to estimating component in Gaussian noise
- ❑Level of Gaussian noise accounts for "interference" between components from A
- ❑Also called a decoupling principle or single letter model

Convergence Formalities

- In what sense do vectors "converge empirically" in Bayati-Monanari result?
- Consider a sequence of vectors $x = x(N) \in R^N$
  - Dimension of vector grows with N. Vector can be deterministic
- Definition: A function $\phi \in PL(k)$ if:
$$\|\phi(x) - \phi(y)\| \leq L\|x - y\|\left[1 + \|x - y\|^{k-1}\right]$$
  - For $k = 1$, this is the standard Lipschitz continuity
- Definition: Sequence $x = x(N)$ converges empirically $PL(k)$ to a scalar X if:
$$\lim_{N\to\infty} \frac{1}{N} \sum\nolimits_n \phi(x_n) = E\big(\phi(X)\big) < \infty \text{ for all } \phi \in PL(k)$$
  - Satisfied for $x_n \sim X$ i.i.d. if $E|X|^k < \infty$

Metrics

- Scalar equivalent model can be used to measure any separable metric
- Ex: MSE. Take $\phi(x_0, \hat{x}^k) := \left|\hat{x}^k - x_0\right|^2$. Then,

$$\text{MSE} = \frac{1}{N}\left\|x_0 - \hat{x}^k\right\|^2 = \frac{1}{N}\sum_{j=1}^{N}\phi(x_{0j}, \hat{x}_j^k) = E\phi\left(X_0, \hat{X}^k\right) = \left|\hat{X}^k - X_0\right|^2$$

- Can also other separable metrics:
  - False alarm, missed detection
  - Error thresholds …

Proof of SE: First Half Iteration

- For $k = 0$: $p^0 = Au^0$

$$p_i^0 = \sum_{j=1}^{N} A_{ij} u_j^0$$

- Since A is i.i.d. $A_{ij} = N(0, \frac{1}{M})$, and u_j^0 are i.i.d. independent of one another:

$$p_i^0 \to N\left(0, \frac{N}{M} E|U^0|^2\right)$$

- Components are Gaussian
- But, this argument doesn't work for $k > 0$
  - A_{ij} becomes dependent on u_j^k

Intuition for $k > 0$. Part I

- Now consider subsequent iteration:

$$q_j^k = \sum_{i=1}^{M} A_{ij} G_q(p_i^k, w_{pi}) - \lambda_u^k u_j^k, \qquad p_i^k = \sum A_{i\ell}\, u_\ell^k$$

- Problems for analysis:
  - Variables are no longer independent
- Idea: Remove dependence between A_{ij} and $A_{ij} u_j^k$
- Define $p_{i\backslash j}^k = p_i^k - A_{ij} u_j^k = \sum_{\ell \neq j} A_{i\ell}\, u_\ell^k$
- Then:

$$q_j^k \approx \sum_{i=1}^{M} \left[A_{ij} G_q\left(p_{i\backslash j}^k, w_{pi}\right) + A_{ij}^2 u_j^k G_q'(p_{i\backslash j}, w_{pi}) \right] - \lambda_u^k u_j^k$$

Intuition for $k > 0$. Part 2

- From before: $q_j^k \approx \sum_{i=1}^{M} \left[A_{ij} G_p\left(p_{i\backslash j}^k, w_{pi}\right) + A_{ij}^2 u_j^k G_p'(p_{i\backslash j}, w_{pi}) \right] - \lambda_u^k u_j^k$
- Now assume A_{ij} is independent of $p_{i\backslash j}^k$
  - We have subtracted the term $A_{ij} u_j^k$
- Since $A_{ij} = N(0, \frac{1}{M})$:

$$\sum_i A_{ij}^2 G_p'(p_{i\backslash j}, w_{pi}) \approx \frac{1}{M} \sum_i G_q'\left(p_{i\backslash j}, w_{pi}\right) \approx \frac{1}{M} \sum_i G_q'\left(p_i, w_{pi}\right) = \lambda_u^k$$

- Also, by CLT: $\sum_i A_{ij} G_q\left(p_{i\backslash j}^k, w_{pi}\right) \approx N\left(0, \tau_q^k\right)$

$$\tau_p^k = \frac{1}{M} \sum_i G_p^2\left(p_{i\backslash j}, w_{pi}\right) \approx \frac{1}{M} \sum_i G_p^2\left(p_i^k, w_{pi}\right) = E G_p^2\left(P^k, W_p\right)$$

Bolthausen Conditioning 1

- How do we make above argument rigorous?
- Key idea of Bayati-Montanari 10
  - Credited to Erwin Bolthausen 14
- Consider conditional distribution of A after k iterations
- After k iterations, we know:

$$p^j = Au^j - \lambda_u^j v^{j-1}, \qquad q^j = A^T v^j - \lambda_u^j u^j, \qquad j = 0,1,\ldots,k$$

- Each iteration reveals actions on vectors u^k and v^k

Bolthausen Conditioning 2

- Want the conditional distribution of A subject to linear constraints
  - $X = AU, \quad U = \left[u^0\, u^1 \cdots u^k\right], \quad X = [p^0, p^1 + \lambda_v^0 u^0, \cdots, p^k + \lambda_v^{k-1} u^{k-1}]$
  - $Y = A^T V, \quad V = \left[v^0\, v^1 \cdots v^k\right], \quad Y = [q^0, q^1 + \lambda_u^0 v^0, \cdots, q^{k-1} + \lambda_u^{k-2} v^{k-2}]$
- This is just conditional distribution of a Gaussian subject to linear constraints
- Can show, conditional distribution is:

$$A = E + P_V^{\perp}\, \tilde{A} P_U^{\perp}$$

  - E is a deterministic matrix with from U, X, V, Y
  - $P_U^{\perp}, P_V^{\perp}$ are projection operators
  - $\tilde{A}$ is independent of U, X, V, Y

Bolthausen Conditioning 3

- We have $A = E + P_V^{\perp}\,\tilde{A}P_U^{\perp}$
- Consider action: $Av^k = Ev^k + P_V^{\perp}\,\tilde{A}P_U^{\perp}v^k$
- Second term: $P_V^{\perp}\,\tilde{A}P_U^{\perp}v^k \approx \tilde{A}v^k$= i.i.d. Gaussian
  - Uses independence of $\tilde{A}$
  - Projections remove only k of N components. So, their effect is small as $N \to \infty$
- First term: Ev^k
  - Can write in terms of inner products $\langle p^j, v^k \rangle = \frac{1}{M}\sum_i p_i^j\, v_i^k = \frac{1}{M}\sum_i p_i^j\, G_p(p_i^j, w_{pi})$
  - Induction hypothesis: $\langle p^j, v^k \rangle = E\left(P^j G_p(P^k, W_p)\right)$
  - By Gaussianity and Stein's Lemma: $\langle p^j, v^k \rangle = E\left(G_p'(P^k, W_p)\right) E(P^j P^k)$
  - With lots of algebra, this shows: : $Ev^k \approx \lambda_u^k \mathrm{u}^{k-1}$

Outline

- AMP and Compressed Sensing
- Proximal Operators and ISTA
- State Evolution for AMP
- Bayes Denoising, Optimality and the Replica Method
- Belief Propagation and Factor Graphs
- AMP Derivation from Belief Propagation
- Convergence, Fixed Points and Stability
- Extensions: Vector AMP
- Thoughts on What is Next

Optimizing the MSE

- SE shows: $\eta^{k+1} = MSE\left(\frac{N}{M}\eta^k + E|W|^2, \theta^k\right)$
- MSE function depends on the estimator:
$$MSE(\nu, \theta) \coloneqq E|g(X_0 + N(0,\nu), \theta) - X_0|^2$$
- Suppose that distribution on X_0 is known
  - Equivalent to known the statistics on the unknown vector exactly
- Idea: Select $g(\cdot)$ to minimize the MSE.
- Optimal estimator is: $g(r) = E(X_0 | R = r,\ R = X_0 + N(0,\nu))$
- Minimum MSE is:
$$MSE^*(\nu) \coloneqq Var(X_0 | X_0 + N(0,\nu))$$

Implementing the MMSE Estimator

- Implementation is possible if:
  - x_0 is well-modeled as i.i.d. and
  - Distribution of components $x_{0j} \sim X_0$ is known
- MMSE estimator can often be analytically for many densities
  - Ex: X_0 is a Gaussian-Mixture Model (GMM):

$$p(X_0|Z=i) = N(\mu_i, \tau_i), \qquad P(Z=i) = q_i, \qquad i = 1,..,L$$

  - Then

$$g(r) = E(X_0|R = X_0 + N(0,v)) = \sum_i E(X_0|R, Z=i)P(Z=i|R)$$

- General densities can be done with numerical integration
  - Or some approximation

Optimality

- With MMSE estimator, SE is: $\eta^{k+1} = MSE^*\left(\frac{N}{M}\eta^k + E|W|^2\right)$
- Can show this converges to a fixed point: $\eta = MSE^*\left(\frac{N}{M}\eta + E|W|^2\right)$
- Optimal MMSE for the original vector problem: $\eta_{opt} = \lim_{N\to\infty} \frac{1}{N}\|x_0 - E(x_0|y)\|^2$
- Theorem: In the Large System Limit, the true optimal MMSE satisfies:

$$\eta_{opt} = MSE^*\left(\frac{N}{M}\eta_{opt} + E|W|^2\right)$$

  - Conjectured originally by the replica method in statistical physics [Guo-Verdu 05]
  - Proven rigorously by [Reeves, Pfister 16], [Barbier, Dia, Macris, Krzakala 16]
- Conclusion: If the fixed point is unique, MMSE-AMP is optimal!

Story So Far

- AMP is computationally simple:
  - Multiplication by A and A^T and scalar estimators
- Applies to general class of problem: Any i.i.d. prior
- For large i.i.d. Gaussian matrix A:
  - Can be exactly analyzed via state evolution
  - Gives optimal performance when SE equations have unique fixed point
  - Holds true even in non-convex multi-modal problems
- Up soon: What happens outside the i.i.d. Gaussian matrix A case?

Outline

- AMP and Compressed Sensing
- Proximal Operators and ISTA
- State Evolution for AMP
- Bayes Denoising, Optimality and the Replica Method
- Belief Propagation and Factor Graphs
- AMP Derivation from Belief Propagation
- Convergence, Fixed Points and Stability
- Extensions: Vector AMP
- Thoughts on What is Next

Belief Propagation and AMP

Problem → AMP-like algorithm → Proof that works (or is optimal)

BP: "Derive" an algorithm

Bolthausen conditioning, Replica, …

- AMP was originally "derived" as an approximation to Belief Propagation
- But, BP proof techniques do not generally apply to AMP problems
- So we use BP to derive algorithms
  - Use other methods to analyze them
- Here, we present BP to derive a generalization of AMP called GAMP

Estimation in High Dimensions

- Belief propagation is for problems in high dimensions
- Consider random vector $x = (x_1, \ldots, x_N)$ with posterior density $p(y|x)$ and
- Want to estimate x from y:
  - ML: $\hat{x} = \arg\max_x p(y|x)$
  - MAP: $\hat{x} = \arg\max_x p(x|y)$
  - Posterior mean / MMSE: $\hat{x} = E(x|y)$
- Curse of Dimensionality: Estimate complexity grows exponentially in N
  - Brute force summation / maximization are not possible at moderate N
  - Need approximate methods or some other structure
- Explain a little more

Belief Propagation: Divide and Conquer

- AMP methods are based on belief propagation (next section)
- Key idea in BP: Many densities have a factorizable structure
- Posterior density $p(x|y)$ on vector $x = (x_1, \dots, x_N)$ can be written as:

$$p(x|y) = \frac{1}{Z(y)} \exp[-H(x,y)], \qquad H(x,y) = \sum_i f_i(x,y)$$

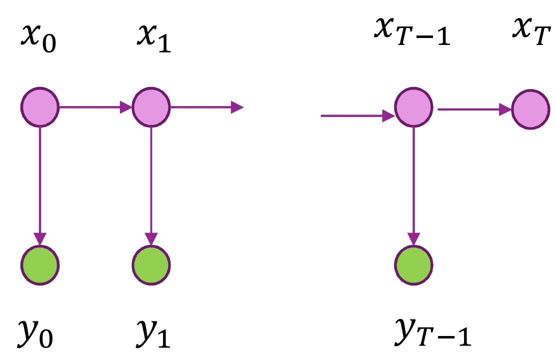
  - $H(x,y)$ is called the energy function
  - Each factor $f_i(x,y)$ assumed to depend only small number of components of x and y
- Belief propagation: Reduces estimation problem on x onto sub-problems of each factor
  - If factors have small numbers of components, estimation is tractable
  - May be approximate…

Ex 1: Estimation in a Hidden Markov Chain

- Markov chain: $x = (x_0, \ldots, x_T)$, $x_t \in R^d$
- Observations $y = (y_0, \ldots, y_{T-1})$
- Problem: Estimate sequence x from y
  - Applications: Dynamical systems, control, time series, …
- By Markov property:

$$\ln p(x, y) = \sum_{t=0}^{T-1} \ln p(x_{t+1}|x_t) + \sum_{t=0}^{T-1} \ln p(y_t|x_t)$$

- Energy function $H(x, y) = -\ln p(x, y)$ factorizes:
  - Total dimension of $x = d(T+1)$
  - T factors of dimension $2d$ and T factors of dimension d



Ex 2: TV Image Denoising

- Image denoising: Estimate image $x = (x_{ij})$ from noisy version $y = x + w$
- TV denoising: Minimize prediction + gradients

$$\hat{x} = \arg\min_x H(x,y), \qquad H(x,y) = \frac{1}{2}\|y-x\|^2 + \lambda\|G_1 x\|_1 + \lambda\|G_2 x\|_1$$

- Cost function is factorizable:

$$H(x,y) = \frac{1}{2}\sum_{ij} |y_{ij} - x_{ij}|^2 + \lambda \sum_{ij} |x_{i+1,j} - x_{ij}|^2 + \lambda \sum_{ij} |x_{i,j+1} - x_{ij}|^2$$

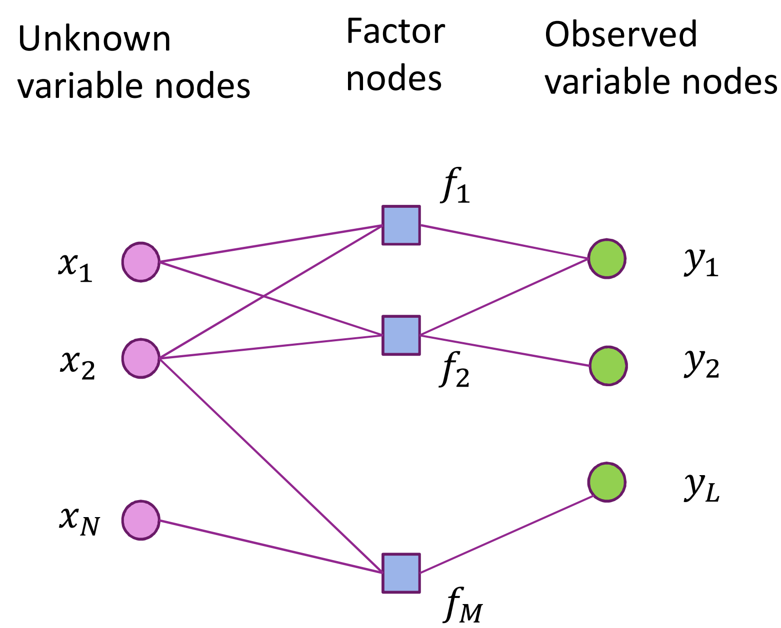
  - Unknown x is typically high-dimensional (e.g. 512 x 512 = 2^{18} components)
  - But, each factor involves only 1 or 2 pixels
  - Differences between neighboring pixels
  - Differences with true pixel value x_{ij} and observed y_{ij}

Factor Graphs

- Assume $x \in R^N, y \in R^L$
- Assume energy function factorizes:

$$H(x, y) = \sum_{i=1}^{M} f_i(x_{a(i)}, y_{b(i)})$$

  - Factors $f_i(\cdot), i = 1, \ldots, M$
  - $a(i) \subset \{1, \ldots, N\}, b(i) \subset \{1, \ldots, L\}$ components in factor i



- Factor graph:
  - Graphical representation of dependencies
  - Undirected, bipartite graph
  - Edge (i, j) in graph $\Leftrightarrow j \in a(i)$, (i, ℓ) in graph $\Leftrightarrow \ell \in b(i)$
  - Let $d(j)$ = neighbors of x_j
- Note: Add example on next slide

Max-Sum and Sum-Product BP

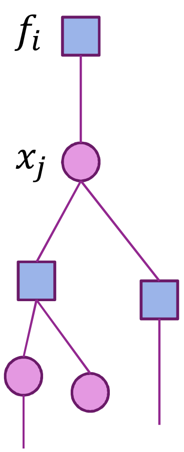
- Consider factorizable posterior density:

$$p(x|y) = \frac{1}{Z(y)} \exp[-H(x)], \qquad H(x) = \sum_{i=1}^{M} f_i(x_{a(i)}), \qquad \psi_i\left(x_{a(i)}\right) := e^{-f_i\left(x_{a(i)}\right)}$$

  - Suppressed dependence on observed variables y
- Two variants of BP, depending on problem
- Sum-product: Estimates posterior marginals $p(x_j|y)$
  - Can compute posterior mean / MMSE estimate $E(x|y)$ from the marginals
- Max-sum: MAP estimation $\hat{x} = \arg\max_x p(x|y) = \arg\min_x H(x)$
  - Can also be viewed as function minimization with no probabilistic interpretation
- Will focus on sum-product

Acyclic Factor Graphs

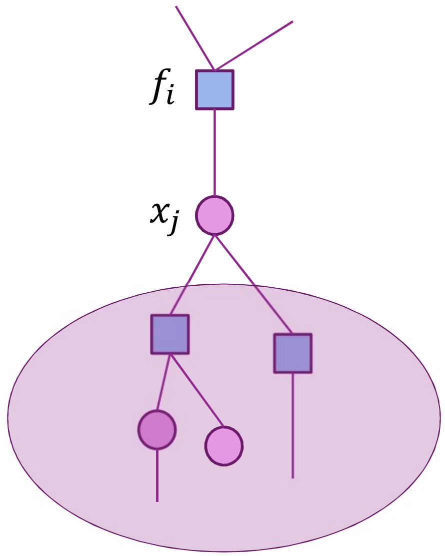
- Suppose factor graph is acyclic (i.e. no loops) and connected
- Acyclic, connected graph can be written as a tree.
  - Can select any node as root.
- Belief propagation on a tree: Max-sum and sum-product are exact!
- Based on message passing
  - Messages from variables to factor nodes
  - Messages from factor to variable nodes
  - Each message is a partial MAP or density estimate
- Will illustrate for sum-product



Sum-Product Messages on the Factor Graph

- Variable x_j to factor f_i:

$$b_{i\leftarrow j}(x_j) \propto \prod_{k \in T_{i\leftarrow j}} \psi_k(x_{a(k)})$$

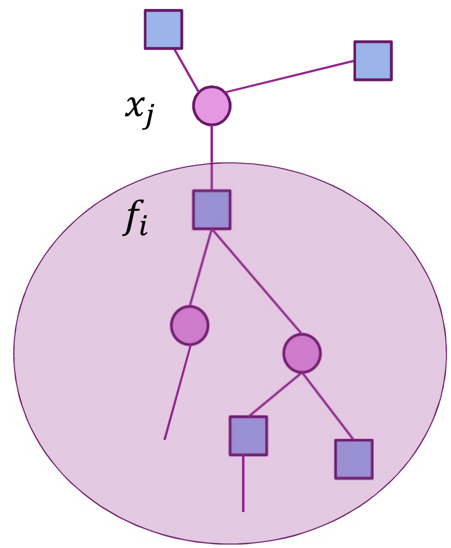


$T_{i\leftarrow j}$ Subtree:

Nodes connected to x_j w/o going via f_i

- Factor f_i to variable x_j:

$$b_{i\leftarrow j}(x_j) \propto \prod_{k \in T_{i\rightarrow j}} \psi_k(x_{a(k)})$$



$T_{i\rightarrow j}$ Subtree:

Nodes connected to f_i w/o going via x_j

Recursive Formula for Sum-Product

- Easy to verify properties on a tree:
  - For any leaf node: $b_{i\leftarrow j}(x_j) = 1$ and $b_{i\rightarrow j}(x_j) = \psi_i(x_j)$
  - For all messages from variable nodes:

$$b_{i\leftarrow j}(x_j) \propto \prod_{k\in d(j)\backslash \mathrm{i}} b_{k\rightarrow j}(x_j)$$

  - For all messages from factor nodes:

$$b_{i\rightarrow j}(x_j) \propto \sum_{x_k, k\in a(i)\backslash \mathrm{j}} \psi_i(x_{a(i)}) \prod_{k\in a(i)\backslash \mathrm{j}} b_{i\leftarrow k}(x_k)$$

  - Final posterior marginal can be computed from:

$$b(x_j) \propto \prod_{k\in d(j)} b_{k\rightarrow j}(x_j)$$

BP on a Tree

- Recursive equations enables simple algorithm for exact inference
- Select any root node and form a tree
  - For all leaf nodes, set messages: $b_{i\leftarrow j}(x_j) = 1$ and $b_{i\rightarrow j}(x_j) = \psi_i(x_j)$
- Compute messages recursively from leaf nodes to root:

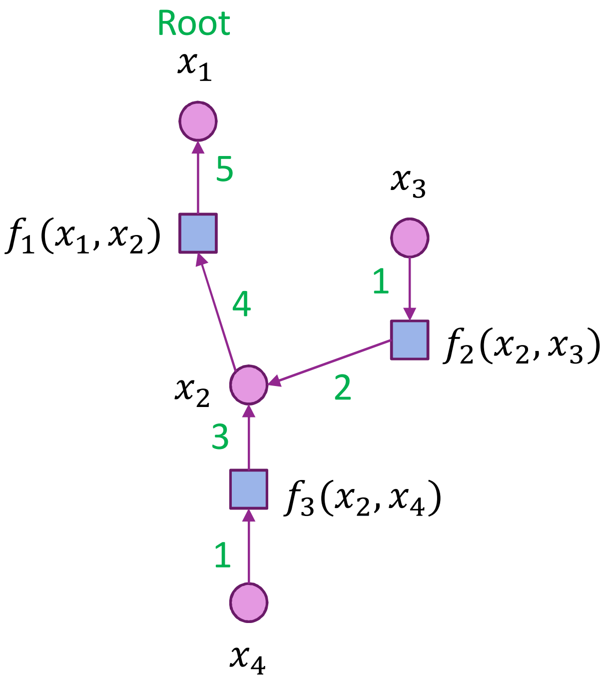
$$b_{i\rightarrow j}(x_j) \propto \sum_{x_k, k\in a(i)\setminus j} \psi_i(x_{a(i)}) \prod_{k\in a(i)\setminus j} b_{i\leftarrow k}(x_k), \qquad b_{i\leftarrow j}(x_j) \propto \prod_{k\in d(j)\setminus i} b_{k\rightarrow j}(x_j)$$

- Then, compute messages from root back to leaves
- Compute final estimates

$$b(x_j) \propto \prod_{k\in d(j)} b_{k\rightarrow j}(x_j)$$

Example: Messages to Root Node

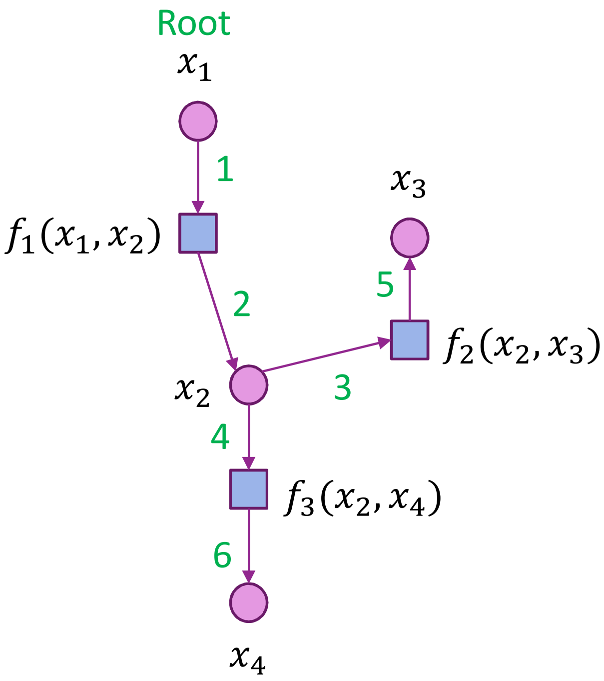
- Pick root node x_1 (you can pick any node)
- Recursively compute message to root:
  1. Initialize $\mu_{3\leftarrow 4}(x_4) = 0, \mu_{2\leftarrow 3}(x_3) = 0$
  2. Compute $\mu_{2\rightarrow 2}(x_2)$ from $\mu_{2\leftarrow 3}(x_3)$ and $f_2(x_2, x_3)$
  3. Compute $\mu_{3\rightarrow 2}(x_2)$ from $\mu_{3\leftarrow 4}(x_4)$ and $f_3(x_2, x_4)$
  4. Compute $\mu_{1\leftarrow 2}(x_2) = \mu_{2\rightarrow 2}(x_2)+ \mu_{3\rightarrow 2}(x_2)$
  5. Compute $\mu_{1\rightarrow 1}(x_1)$ from $\mu_{1\leftarrow 2}(x_2)$ and $f_1(x_1, x_2)$



Example: Messages to Leaf Nodes

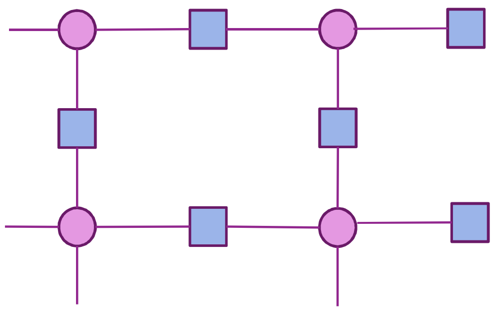
- Recursively compute messages to leaf nodes:
  1. Initialize $\mu_{1\leftarrow 1}(x_1) = 0$,
  2. Compute $\mu_{1\to 2}(x_2)$ from $\mu_{1\leftarrow 1}(x_1)$ and $f_1(x_1, x_2)$
  3. Compute $\mu_{2\leftarrow 2}(x_2) = \mu_{1\to 2}(x_2) + \mu_{3\to 2}(x_2)$
  4. Compute $\mu_{3\leftarrow 2}(x_2) = \mu_{1\to 2}(x_2) + \mu_{2\to 2}(x_2)$
  5. Compute $\mu_{2\to 3}(x_3)$ from $\mu_{2\leftarrow 2}(x_2)$ and $f_2(x_2, x_3)$
  6. Compute $\mu_{3\to 4}(x_4)$ from $\mu_{3\leftarrow 2}(x_2)$ and $f_3(x_2, x_4)$

- Compute estimates:
  - $\hat{x}_1 = \arg\min_{x_1} \mu_{1\to 1}(x_1)$
  - $\hat{x}_2 = \arg\min \mu_{1\to 2}(x_2) + \mu_{2\to 2}(x_2) + \mu_{3\to 2}(x_2)$
  - $\hat{x}_3 = \arg\min \mu_{2\to 3}(x_3)$
  - $\hat{x}_4 = \arg\min \mu_{3\to 4}(x_4)$



Graphs with Loops

- Problem: In many problems, graph has loops
  - Ex: For TV denoising, graph is a lattice structure
- Loopy belief propagation: Approximate solution
  - Apply same recursions as BP with trees for messages
- Typically iterations:
  1. Initialize $b_{i\leftarrow j}(x_j) = 1$
  2. Factor node update:
  $$b_{i\rightarrow j}(x_j) \propto \sum_{x_k, k\in a(i)\backslash \mathrm{j}} \psi_i(x_{a(i)}) \prod_{k\in a(i)\backslash \mathrm{j}} b_{i\leftarrow k}(x_k)$$
  3. Variable node update:
  $$b_{i\leftarrow j}(x_j) \propto \prod_{k\in d(j)\backslash \mathrm{i}} b_{k\rightarrow j}(x_j)$$
  4. Repeat Step 2 and 3 until convergence



Loopy Belief Propagation Issues

- Potential shortcomings:
  - Loopy BP may diverge
  - When it converges, no guarantee that estimate is correct
- Considerable work to find convergence guarantees / approximation bounds
  - Locally tree like conditions
  - Dorbrushin condition (weak coupling)
- AMP will be derived from loopy BP
  - But, we prove its convergence via state evolution

Outline

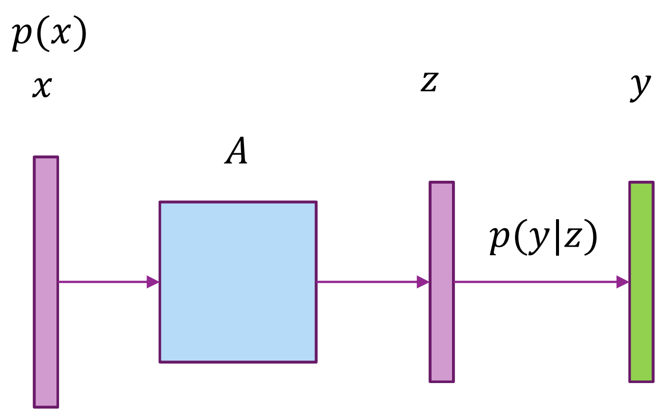
- AMP and Compressed Sensing
- Proximal Operators and ISTA
- State Evolution for AMP
- Bayes Denoising, Optimality and the Replica Method
- Belief Propagation and Factor Graphs
- AMP Derivation from Belief Propagation
- Convergence, Fixed Points and Stability
- Extensions: Vector AMP
- Thoughts on What is Next

GLM Estimation

- Will look at AMP in a more general setting
- Bayesian Generalized linear model (GLM):
  - IID prior $p(x) = \prod_j p(x_j)$
  - Linear transform: $z = Ax$
  - Componentwise likelihood: $p(y|z) = \prod_i p(y_i|z_i)$
- Problem: Estimate x and z from A and y
- Linear inverse problem is a special case:

$$y = z + w, \qquad w \sim N(0, \sigma^2 I)$$

- But, GLM can incorporate:
  - Nonlinearities in outputs
  - Outputs can be discrete
  - Non-Gaussian noise

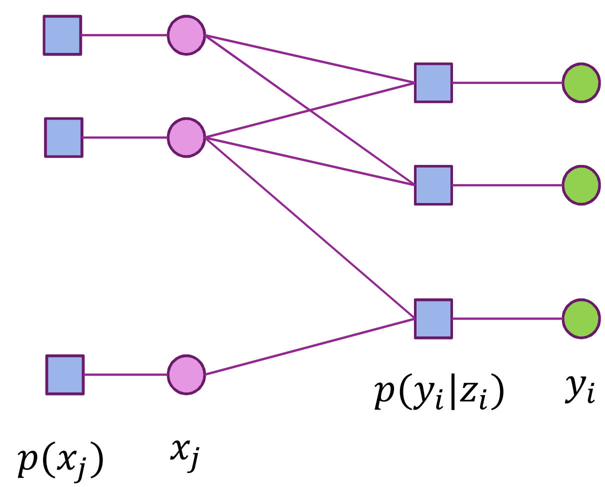


Factor Graph for a GLM

- Posterior density factors as:

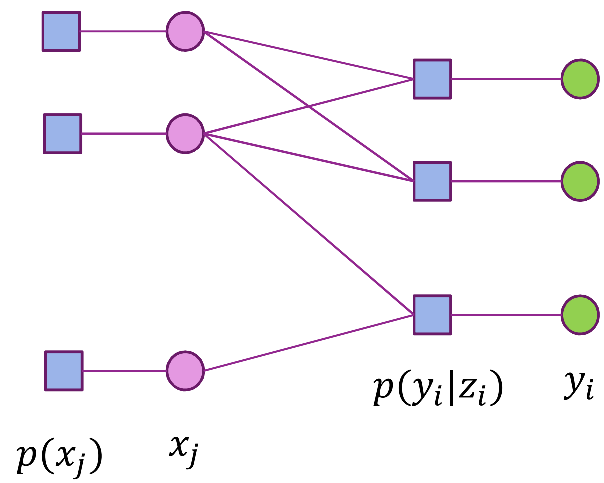
$$p(x|y) = \frac{1}{Z(y)} \prod_j p(x_j) \prod_i p(y_i|z_i), \qquad z_i = A_i^T x = \sum_j A_{ij} x_j$$

- Problem applying BP directly:
  - Factor graph has loops
  - Graph is dense if A is dense
  - Messages must be over variables x_j
- Can we still get approximate inference using BP?
  - Will it converge?
  - Can it be optimal?



Sum-Product BP for GLM

- Consider sum-product (Loopy) BP for the GLM problem
- With slight rearrangement, updates can be written in two stages



- Output node updates:

$$b_{i\to j}(x_j) \propto E\{p(y_i|z_i)|x_j,\ x_k \sim b_{i\leftarrow k}(x_k)\}, \qquad z_i = \sum_j A_{ij} x_j$$

  - Message from factors $p(y_i|z_i)$ to variables x_j

- Input node updates:

$$b_{i\leftarrow j}(x_j) \propto p(x_j) \prod_i b_{i\to j}(x_j)$$

  - Message from variables x_j to factors $p(y_i|z_i)$

GAMP Approximations

- Assume that each A_{ij} is relatively small

$$\frac{|A_{ij}|^2}{\sum_k |A_{ik}|^2} = O\left(\frac{1}{N}\right), \qquad \frac{|A_{ij}|^2}{\sum_k |A_{kj}|^2} = O\left(\frac{1}{N}\right)$$

  - Applies to dense matrices where all components are roughly same value
- Under this assumption:
  - Apply a Central Limit Theorem approximation in the output update
  - A second order approximation of the messages in the input update
- The approximation is heuristic
  - No rigorous bound for discrepancy between full BP and AMP
  - Full BP is approximate anyway (since graph has loops)
- Used as motivation for the algorithm
- Can analyze rigorously via state evolution

Output Node Approximation

- Output update

$$b_{i\to j}(x_j) \propto E\{p(y_i|z_i)\big|x_j,\ x_k \sim b_{i\leftarrow k}(x_k)\}, \qquad z_i = \sum_j A_{ij}x_j$$

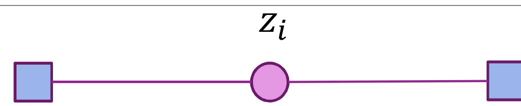
- Define mean and variance of each incoming msg: $\hat{x}_{i\leftarrow j} = E\left(x_j \middle| b_{i\leftarrow j}\right),\ \tau^x_{i\leftarrow j} := var\left(x_j \middle| b_{i\leftarrow j}\right)$
- Apply Central Limit Theorem: Since incoming messages are independent:

$$b_{i\to j}(x_j) \approx \frac{1}{Z} E\left\{p(y_i|z_i)\middle| z_i \sim N(p_{i\to j}, \tau^p_{i\to j})\right\}$$

- Mean and variance on z_i given by:
  - $p_{i\to j} := p_i + A_{ij}\left(x_j - \hat{x}_{i\leftarrow j}\right),\ p_i := \sum_k A_{ik}\,\hat{x}_{i\leftarrow k},$
  - $\tau^p_{i\to j} = \sum_{k\neq j} A^2_{ik}\tau^p_{i\leftarrow k}$

Scalar Output Channel

Gaussian Prior $z_i \sim N(z_i|p_i, \tau_i^p)$



Measurement $y_i \sim P(y_i|z_i)$

- Scalar output estimation problem: Estimate z_i, Gaussian prior on z_i, measurement y_i
- Posterior density is:

$$b_i^z\left(z_i \middle| p_i, \tau_i^p\right) := \frac{1}{Z(p_i, \tau_i^p, y_i)} p(y_i|z_i) N(z_i|p_i, \tau_i^p)$$

- Log likelihood of y_i is $G_i\left(p_i, \tau_i^p, y_i\right) := \ln E\left\{p(y_i|z_i) \middle| z_i \sim N\left(p_i, \tau_i^p\right)\right\} = \ln Z_i(p_i, \tau_i^p, y_i)$
- Using properties of exponential families, can show:

$$G_i'\left(p_i, \tau_i^p\right) = E\left[z_i \middle| p_i, \tau_i^p\right], \qquad G_i''\left(p_i, \tau_i^p\right) = -\frac{1}{\tau_i^p}\left[1 - \frac{var\left[z_i \middle| p_i, \tau_i^p\right]}{\tau_i^p}\right]$$

  - Derivatives are with respect to p_i

Input Node Approximation

- Consider logarithmic message: $\mu_{i\to j}(x_j) = -\ln b_{i\to j}(x_j)$
- By output update:

$$\mu_{i\to j}(x_j) = -\ln E\left\{p(y_i|z_i)\middle|z_i \sim N\left(p_{i\to j}, \tau^p_{i\to j}\right)\right\} \approx -G_i\left(p_i + A_{ij}(x_j - \hat{x}_{i\leftarrow j}), \tau^p_{i\to j}, y_i\right)$$

- If A_{ij} is small:

$$\mu_{i\to j}(x_j) \approx -G_i\left(p_i, \tau^p_{i\to j}\right) - G_i'\left(p_i, \tau^p_{i\to j}\right)A_{ij}(x_j - \hat{x}_{i\leftarrow j}) - \frac{1}{2}G_i''\left(p_{i\to j}, \tau^p_{i\to j}\right)A_{ij}^2(x_j - \hat{x}_{i\leftarrow j})^2$$

- Input node update:

$$b_{i\leftarrow j}(x_j) = \frac{1}{Z}p(x_j)\exp\left[-\sum_{k\neq i}\mu_{i\to k}(x_j)\right] \approx \frac{1}{Z}p(x_j)\exp\left[-\frac{(x_j - r_{i\leftarrow j})^2}{2\tau^r_{i\leftarrow j}}\right]$$

Scalar Input Channel

- Scalar input estimation problem: Estimate x_j with prior on $p(x_j)$ and Gaussian measurement r_j
- Posterior density is:

$$b_j^x\left(x_j \middle| r_j, \tau_j^r\right) := \frac{1}{Z\left(r_j, \tau_j^r\right)} p(x_j) N(r_j | x_j, \tau_j^r)$$

- Message to output nodes are: $b_{i \leftarrow j}\left(x_j\right) = b_j^x\left(x_j \middle| r_{i \leftarrow j}, \tau_{i \leftarrow j}^r\right)$
- For the output node update, we need to compute mean and variance under this density:

$$\hat{x}_{i \leftarrow j} = E\left(x_j \middle| r_{i \leftarrow j}, \tau_{i \leftarrow j}^r\right), \qquad \tau_{i \leftarrow j}^x \coloneqq var\left(x_j \middle| r_{i \leftarrow j}, \tau_{i \leftarrow j}^r\right)$$

Sum-Product GAMP Algorithm

- Combining terms with some more algebra results in simple algorithm

GAMP (Generalized AMP) [Rangan 10]

- Initialization: $\hat{x}_j = E(x_j),\ \tau_j^x = var(x_j)$ from prior $p(x_j)$
- Repeat until convergence:
  - Forward linear transform: $p_i = \sum_j A_{ij}\hat{x}_j - \tau_i^p s_i,\ \ \tau_i^p = \sum_j |A_{ij}|^2 \tau_j^x$
  - Output estimation: $\hat{z}_i = E\left(z_i \middle| p_i, \tau_i^p\right),\ \ \tau_i^z = var\left(z_i \middle| p_i, \tau_i^p\right)$
  - Reverse nonlinear transform : $s_i = \frac{1}{\tau_i^p}[\hat{z}_i - p_i],\ \ \tau_i^s = \frac{1}{\tau_i^p}\left[1 - \frac{\tau_i^z}{\tau_i^p}\right]$
  - Reverse linear transform: $r_j = \hat{x}_j - \tau_j^r \sum_i A_{ij} s_i\ ,\ \ \tau_j^r = \left(\sum_i |A_{ij}|^2 \tau_j^s\right)^{-1}$
  - Input node estimation: $\hat{x}_j = E\left(x_j \middle| r_j, \tau_j^r\right),\ \ \tau_j^x = var\left(x_j \middle| r_j, \tau_j^r\right)$

Scalar Variance GAMP

- Make approximation that all variances are constant: e.g. $\tau_j^x \approx \tau^x$
- Resulting algorithm:
  - Forward linear transform: $p = A\hat{x} - \tau^p s, \quad \tau^p = \frac{1}{M}\|A\|_F^2 \tau^x$
  - Output estimation: $\hat{z} = g_{out}(p, \tau_p), \quad \tau^z = \tau^p \langle g_{out}'(p, \tau_p) \rangle$
  - Reverse nonlinear transform : $s = \frac{1}{\tau^p}[\hat{z} - p], \quad \tau^s = \frac{1}{\tau^p}\left[1 - \frac{\tau^z}{\tau^p}\right]$
  - Reverse linear transform: $r = \hat{x} - \tau^r A^T s, \quad \tau^r = N\left(\|A\|_F^2 \tau^s\right)^{-1}$
  - Input node estimation: $\hat{x} = g_{in}(r, \tau_r), \quad \tau^r = \tau^r \langle g_{in}'(r, \tau_r) \rangle$

From GAMP to AMP

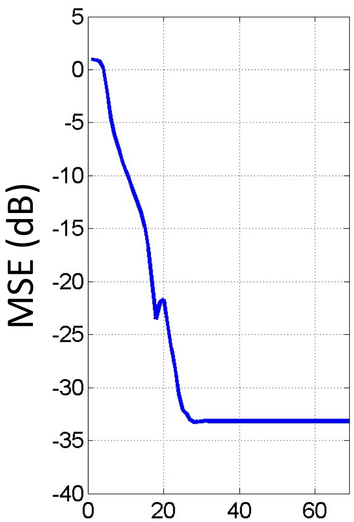
- Special case of an AWGN output: $y = z + w, \quad w \sim N(0, \tau_w I)$
- Output estimator:
  - $g_{out}(p, \tau^p) = \frac{1}{\tau_p + \tau_w}(\tau_w p + \tau_p y), \; g'_{out}(p, \tau^p) = \frac{\tau_p \tau_w}{\tau_p + \tau_w}$
- Then, we obtain AMP with specific choice of thresholding
  - Forward linear transform: $p = A\hat{x} - \tau^p s, \quad \tau^p = \frac{1}{M}\|A\|_F^2 \tau^x$
  - Residual: $s = \frac{1}{\tau^p + \tau^w}(y - p), \quad \tau^s = \frac{1}{\tau^p + \tau^w}$
  - Reverse linear transform: $r = \hat{x} - \frac{N}{\|A\|_F^2} A^T (y - p), \quad \tau^r = \frac{N}{\|A\|_F^2}(\tau^p + \tau^w)$
  - Input node estimation: $\hat{x} = g_{in}(r, \tau_r), \quad \tau^r = \tau^r \langle g'_{in}(r, \tau_r) \rangle$

Outline

- ❑AMP and Compressed Sensing
- ❑Proximal Operators and ISTA
- ❑State Evolution for AMP
- ❑Bayes Denoising, Optimality and the Replica Method
- ❑Belief Propagation and Factor Graphs
- ❑AMP Derivation from Belief Propagation
- ❑Convergence, Fixed Points and Stability
- ❑Extensions: Vector AMP
- ❑Thoughts on What is Next

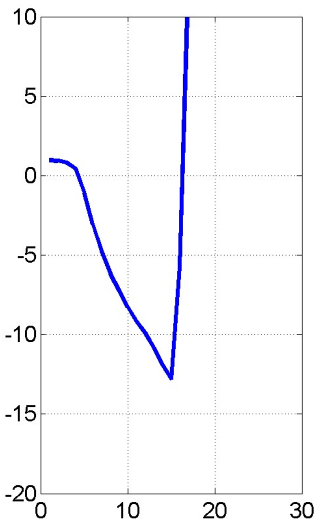
Problems with AMP for Non-IID A

A = iid, N(0,1)



Converges rapidly

A = iid N(0.5,1)

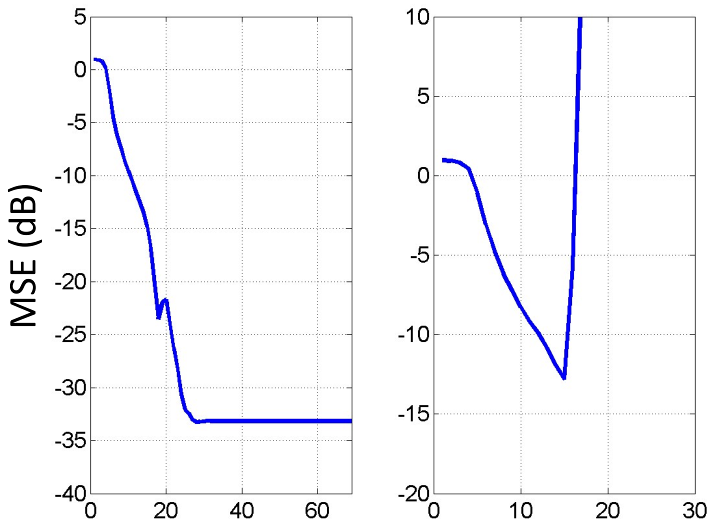


Diverges

- *Evidently, this promise comes with the caveat that message-passing algorithms are specifically designed to solve sparse-recovery problems for Gaussian matrices…”,* Felix Herman, Nuit Blanche blog, 2012
- AMP can diverge for non-iid A
- Even non-pathological matrices
- See [FZK14]

Questions for This Section

A = iid, N(0,1) **A** = iid N(0.5,1)



Converges rapidly

Diverges

- ❑When exactly does AMP converge?
- ❑What does it converge to, if it does?
- ❑Can convergence be improved?

GAMP on a Gaussian Problem

- Consider simple Gaussian problem:
  - $x \sim N(0, \tau_x I),\ y = Ax + w,\ w \sim N(0, \tau_w I)$
- Question: When does AMP/GAMP converge for this problem?
  - Convergence of second-order / variance terms
  - Convergence of first-order / mean terms
- GAMP is not the best solution for Gaussian problem
  - MMSE solution has explicit solution: $\hat{x} = \tau_x(\tau_x A^T A + \tau_w I)^{-1} A^T y$
- Look at Gaussian problem since:
  - Can derive exact conditions for convergence
  - Convergence conditions are easy to interpret

Rangan, Schniter, Fletcher. "On the convergence of approximate message passing with arbitrary matrices." *Proc IEEE ISIT 2014*

Variance Convergence

- AWGN vector-valued variance updates:

$$\tau_p^t = S\tau_x^t, \qquad \tau_s^t = \frac{1}{\tau_p^t + \tau_w},$$

$$\tau_r^t = \frac{1}{S^* \tau_s^t}, \qquad \tau_x^{t+1} = \frac{\tau_r^t \tau_0}{\tau_r^t + \tau_0}$$

  - $S = |A|^2$ = componentwise magnitude squared

- Theorem: For any τ_w and τ_0, the AWGN variance updates converge to unique fixed points

- Subsequent results will consider algorithm with fixed variance vectors.

Proof of the Variance Convergence

- Define vector valued functions:

$$g_s: \tau_x^t \mapsto \tau_s^t, \qquad g_x: \tau_s^t \mapsto \tau_x^{t+1}, \qquad g = g_x \circ g_s$$

- Verify g satisfies:
  - Monotonically increasing
  - $g(\alpha \tau_s) \leq \alpha g(\tau_s)$ for $\alpha \geq 1$.

- Convergence now follows from R. D. Yates, "A framework for uplink power control in cellular radio systems", 1995
  - Used for convergence of power control loops

Convergence of the Means
Uniform Variance Update

- Consider constant case:
  - Constant variances: $\tau_{0j} = \tau_0,\ \tau_{wi} = \tau_w$.
  - Uniform variance updates in GAMP

- Theorem: The means of the AWGN GAMP will converge for all τ_0 and τ_w if and only if

$$\sigma_{max}^2(A) < \frac{2(m+n)}{mn}\|A\|_F^2$$

  - $\sigma_{max}(A)$: maximum singular value
  - $\|A\|_F^2$ = Frobenius norm = sum of singular values

Some Matrices Work…

$$\sigma_{max}^2(A) < \frac{2(m+n)}{mn} \|A\|_F^2$$

- Convergence depends on bounded spread of singular values.

- Examples of convergent matrices:
  - Random iid: Converges due to Marcenko-Pastur
  - Subsampled unitary: $\sigma_{max}^2(A)$=1, $\|A\|_F^2 = \min(m,n)$
  - Total variation operator: $(Ax)_i = x_i - x_{i-1}$
  - Walk summable matrices: Generalizes result by Maliutov, Johnson and Willsky (2006)

But, Many Matrices Diverge

$$\sigma_{max}^2(A) < \frac{2(m+n)}{mn} \|A\|_F^2$$

- Examples of matrices that do not converge:
  - Low rank: If A has r equal singular values and other are zero:
  $$2r(m+n) > mn \Rightarrow r > \min(m,n)/2$$
  - $A \in R^{m\times m}$ is a linear filter: $Ax = h * x$ for some filter h
  $$\sup_\theta \left|H\left(e^{i\theta}\right)\right| < \frac{1}{2}\ \frac{1}{2\pi}\int \left|H\left(e^{i\theta}\right)\right|^2 d\theta$$
  - Some matrices with large non-zero means: $A = A_0 + \mu 1^T$

Proof of Convergence

- With constant variances system is linear:

$$\begin{bmatrix} s^t \\ x^{t+1} \end{bmatrix} = G \begin{bmatrix} s^{t-1} \\ x^t \end{bmatrix} + b$$

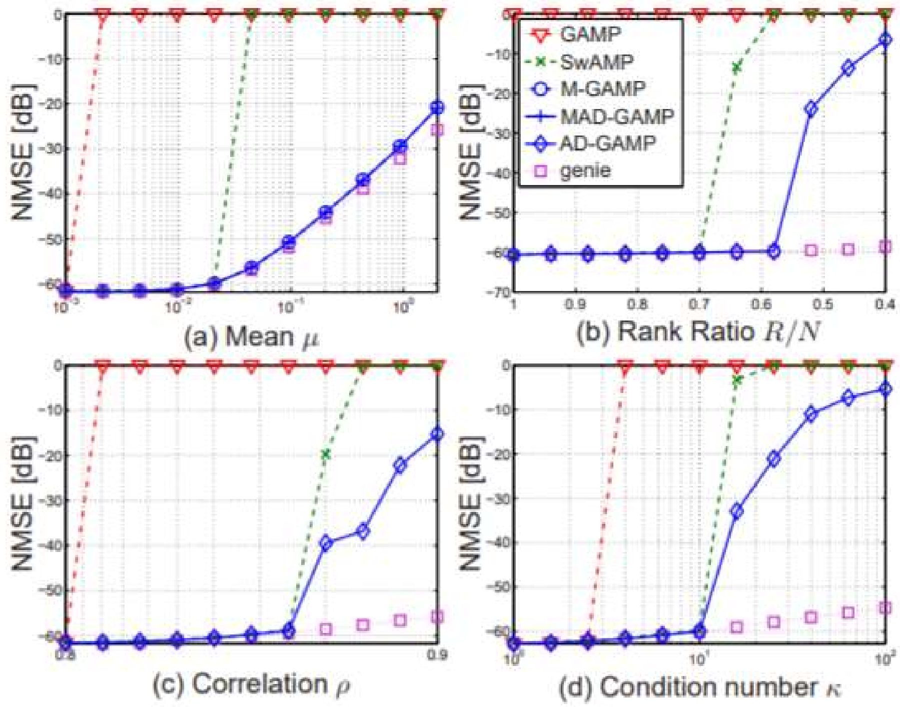
  - $G = \begin{bmatrix} I & 0 \\ D(\tau_x)A^* & D(\tau_x \tau_r^{-1}) \end{bmatrix} \begin{bmatrix} D(\tau_p \tau_s) & -D(\tau_s)A \\ 0 & I \end{bmatrix}$
  - $D(\tau) = diag(\tau)$

- System is stable if and only if $\lambda_{max}(G) < 1$
- Eigenvalue condition related to singular values of

$$F = D\left(\tau_s^{1/2}\right) AD\left(\tau_x^{1/2}\right)$$

Improving Stability

- Many methods
  - Coordinate-wise descent [MKTZ14,CKZ14]
  - Damping [VSR+14, JBD09]
  - Double loop / ADMM [RSF17]
- Slow rate with improved robustness
- But:
  - May still fail
  - Often needs tuning



Outline

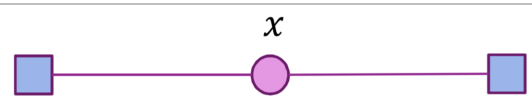
- ❑AMP and Compressed Sensing
- ❑Proximal Operators and ISTA
- ❑State Evolution for AMP
- ❑Bayes Denoising, Optimality and the Replica Method
- ❑Belief Propagation and Factor Graphs
- ❑AMP Derivation from Belief Propagation
- ❑Convergence, Fixed Points and Stability
- ❑Extensions: Vector AMP
- ❑Thoughts on What is Next

Story So Far

- Benefits of AMP: For large Gaussian i.i.d. A
  - Fast convergence
  - Can be analyzed rigorously via state evolution
  - Testable conditions for optimality
- But, outside Gaussian i.i.d. A:
  - Can diverge.
  - Stability techniques are only partially successful
  - Loses key properties
- Is there a better way?

A Vector Valued Factor Graph

Prior $p(x)$



Measurement $y = Ax + w,\ w \sim N(0, \gamma_w^{-1} I)$

- Consider simpler factor graph for linear inverse problem:
  - Single vector variable node for x
  - One factor for prior $\psi_1(x) \coloneqq p(x)$ (separable)
  - One factor for likelihood $\psi_2(x) \coloneqq p(y|x)$ (Gaussian)
- Posterior density factors as:

$$p(x|y) = \frac{1}{Z(y)} \psi_1(x) \psi_2(x)$$

- Insight due to [Cakmak, Winther, Fleury, 14, 15]

Variational Inference

- Write posterior as:

$$p(x|y) = \frac{1}{Z(y)}\psi_1(x)\psi_2(x), \qquad \psi_1(x) = p(x), \qquad \psi_2(x) = p(y|x)$$

- Variational inference:

$$\hat{b} = \arg\min_b D(b(\cdot)||p(\cdot\,|y)) = \arg\min_b [D(b\|\psi_1) + D(b\|\psi_2) + H(b)]$$

- Apply variable splitting:

$$\hat{b}_1, \hat{b}_2 = \arg\min_{b_1,b_2}\max_q J(b_1,b_2,q)\,, \qquad J(b_1,b_2,q) = D(b_1\|\psi_2) + D(b_2\|\psi_1) + H(q)$$

  - Subject to constraints $b_1 = b_2 = q$

- Problem is intractable.
  - Must optimize over N dimensional densities

EC Inference

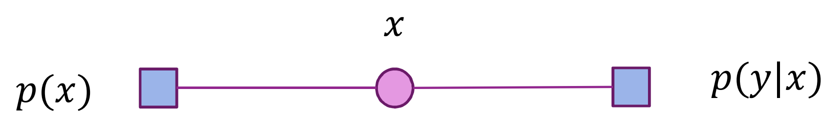
- Desired optimization is too hard:

$$\hat{b}_1, \hat{b}_2 = \arg\min_{b_1,b_2} \max_q J(b_1, b_2, q)\,, \qquad J(b_1, b_2, q) = D(\psi_1 \| b) + D(\psi_2 \| b) + H(b)$$

  - Subject to constraints $b_1 = b_2 = q$
- Expectation consistent inference: Replace constraints by moment matching conditions:
  - $E(x|b_1) = E(x|b_2) = E(x|q)$
  - $E(\|x\|^2|b_1) = E(\|x\|^2|b_2) = E(\|x\|^2|q)$
  - Proposed by Opper-Winther 04, 05
- At EC stationary points:
  - $b_i(x) \propto \psi_i(x) N\left(x \middle| r_i, \gamma_i^{-1} I\right)$, $\quad q(x) = N(x|\hat{x}, \eta^{-1} I)$
  - $E(x|b_1) = E(x|b_2) = E(x|q) = \hat{x}$
  - $E(\|x\|^2|b_1) = E(\|x\|^2|b_2) = E(\|x\|^2|q) = \frac{N}{\eta}$

Vector AMP

- Use Expectation Propagation to find stationary points
- Input Denoising
  - $\hat{x}_1 = g_1(r_1, \gamma_1)$
  - $\eta_1 = \gamma_1/\alpha_1\,,\ \alpha_1 = \langle g_1'(r_1, \gamma_1)\rangle$
  - $\gamma_2 = \eta_1 - \gamma_1,\ r_2 = (\eta_1 \hat{x}_1 - \gamma_1 r_1)/\gamma_2$



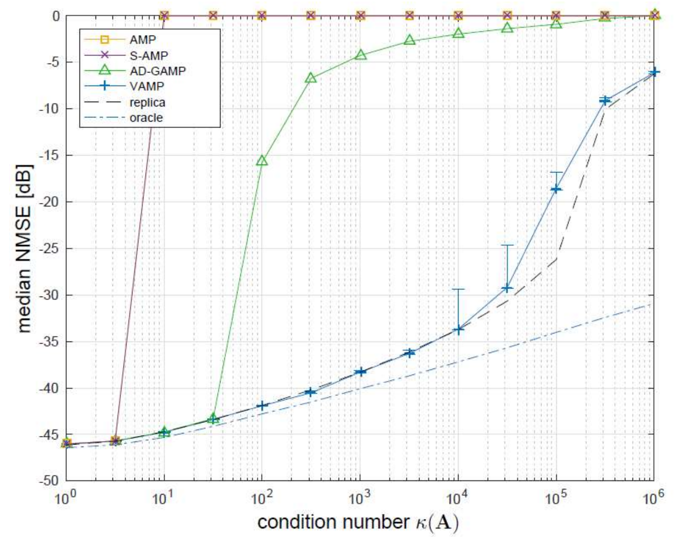
- Denoisers:
  - $g_1(r_1, \gamma_1) = E\left(x \middle| r_1 = x + N\left(0, \gamma_1^{-1}\right),\ x \sim p(x)\right)$
  - $g_2(r_2, \gamma_2) = E\left(x \middle| r_2 = Ax + w,\ x \sim N\left(0, \gamma_2^{-1} I\right)\right)$

- Output Denoising
  - $\hat{x}_2 = g_2(r_2, \gamma_2)$
  - $\eta_2 = \gamma_2/\alpha_2\,,\ \alpha_2 = \langle g_2'(r_2, \gamma_2)\rangle$
  - $\gamma_1 = \eta_2 - \gamma_2,\ r_1 = (\eta_2 \hat{x}_2 - \gamma_2 r_2)/\gamma_1$

Rangan, Schniter, Fletcher, “Vector Approximate Message Passing”, Proc IEEE ISIT 2017

Why Use VAMP?

- Computationally efficient
  - Though harder than AMP.
  - Requires SVD or matrix inverse
- Numerically stable over ill-conditioned matrices
  - Overcomes major problem with AMP
- Performance matches state evolution
  - Achieves replica prediction for optimality
- Extensions: EM, image processing, …



Right-Rotationally Invariant Matrices

- Measurement Model: $y = Ax^0 + w,\ w \sim N(0, \gamma_w^{-1} I)$
- Take SVD: $A = U\mathrm{diag}(s)V^T \in R^{N\times N},\ \ s = (s_1, \ldots, s_N)$
  - WLOG assume A is square (otherwise add zero singular values)
- Left factor U is arbitrary
  - Will assume $U = I$ (Otherwise, look at $U^T y$).
- Right rotationally invariant A :
  - V Haar, uniform on the orthogonal matrices
  - S has limiting distribution
- Includes A Gaussian iid. But, much more general
- New model: $y = \mathrm{diag}(s)V^T x^0 + w,\ w \sim N(0, \gamma_w^{-1} I)$

State Evolution

- Two key error quantities:
  - $p_k = r_{1k} - x^0$: Error on the input to the input denoiser
  - $v_k = r_{2k} - x^0$: Error on the input to the LMMSE denoiser
- Transformed errors: $u_k = V^T p_k, \;\; q_k = V^T v_k$
- Theorem: In large system limit:

$$p_k \to P_k \sim N(0, \tau_k^p), \qquad q_k \to Q_k \sim N(0, \tau_k^q),$$

  - Variances τ_k^p, τ_k^q can be calculated via state evolution
- Shows errors are Gaussian
- Fixed points of SE equations match predictions for optimality from the replica method
  - [Takeda, Uda, Kabishma 06], [Tulino, Caire, Verdu, Shamai 13]
  - Rigorously shown for large subclass in [Barbier, Macris, Maillard, Krzakala 17]

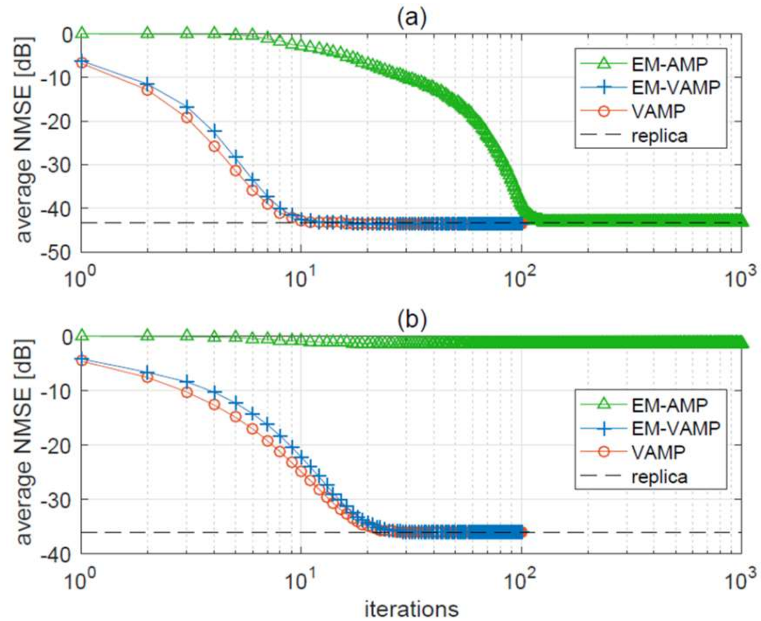
Proof of the SE

- Error recursion
  - $p_k = Vu_k$
  - $v_k = C_1(\alpha_{1k})\left[f_p(p_k, w^p, \gamma_{1k}) - \alpha_{1k}p_k\right],\ \alpha_{1k} = \left\langle f_p'(p_k, w^p, \gamma_{1k})\right\rangle$
  - $q_k = V^T v_k$
  - $u_{k+1} = C_2(\alpha_{2k})\left[f_q(q_k, w^q, \gamma_{2k}) - \alpha_{2k}q_k\right],\ \alpha_{2k} = \left\langle f_q'(q_k, w^q, \gamma_{2k})\right\rangle$
- Similar to Bayati-Montanari recursion but:
  - Gaussian iid A replaced by Haar matrix V
- Can apply Bolthausen conditioning
  - Conditional distribution of Haar matrix V subject to linear constraints $A = VB$

$$V_{|G} = A(A^T A)^{-1}B^T + U_{A^\perp}\tilde{V}U_{B^\perp}$$

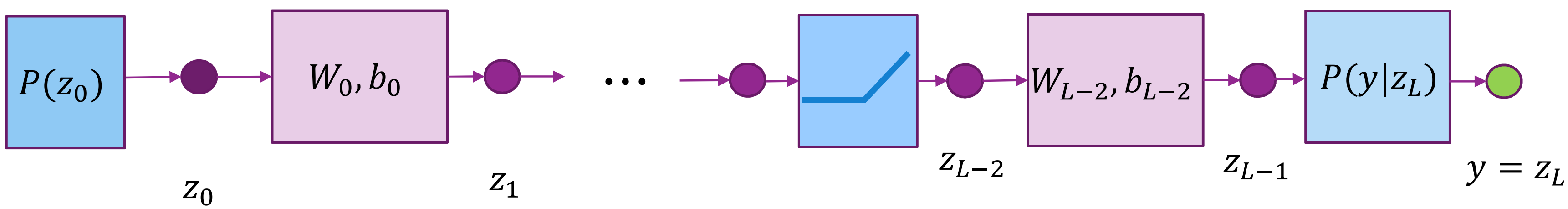
EM VAMP

- Suppose densities have unknown parameters:
  - $x \sim p(x|\theta_1)$
  - $y = Ax + w,\ w \sim N(0, \theta_2^{-1} I)$
- Problem: Estimate x and $\theta = (\theta_1, \theta_2)$
- EM-VAMP:
  - E-Step: Use VAMP to estimate $p\left(x\middle|y, \hat{\theta}\right)$
  - M-Step: $\hat{\theta}^{new} = \arg\max_{\theta} E\left[p(x, y|\theta)|\hat{\theta}\right]$
- Similar ideas in AMP [KrzMSSZ12, VS12]



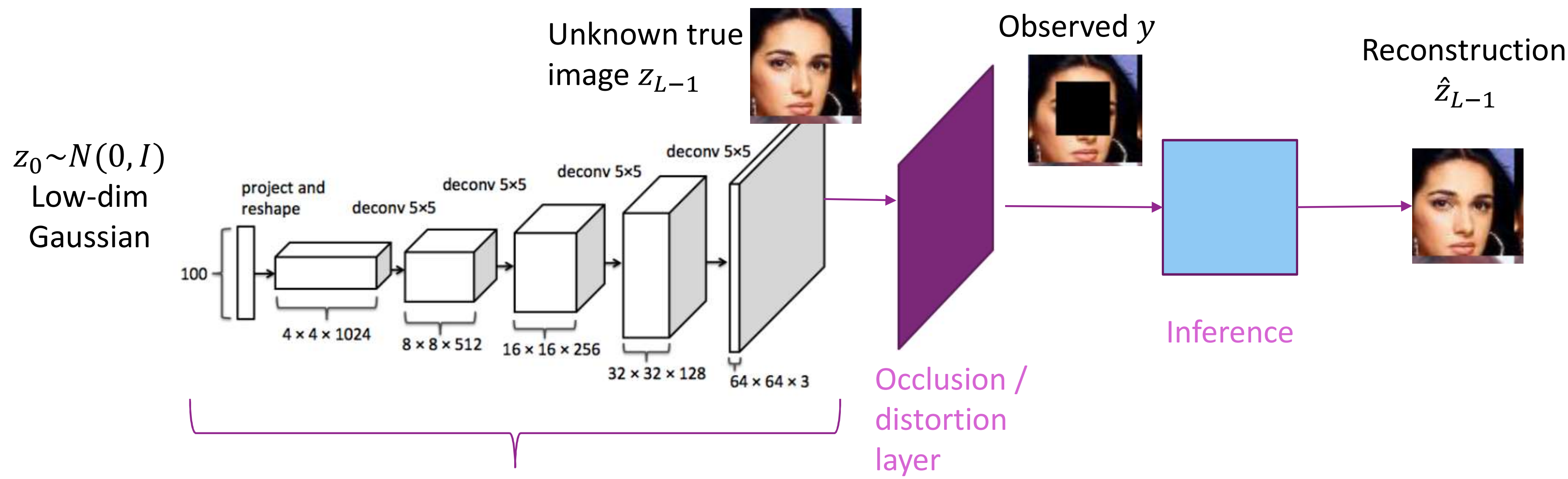
Fletcher, Schniter. "Learning and free energies for vector approximate message passing." *Proc. IEEE ICASSP 2017*

Inference in Deep Networks



- Model:
  - A multi-layer neural network with known weights and biases, activations
  - Distribution on network input $P(z_0)$
  - Network is already trained. Not a learning problem!
- Inference problem: Given observed output $z_L = y$ and network, estimate:
  - Input z_0 and hidden layers $z_1, \ldots, z_{L-1}$
- Network is already trained. Not a learning problem

Motivation: Image Reconstruction



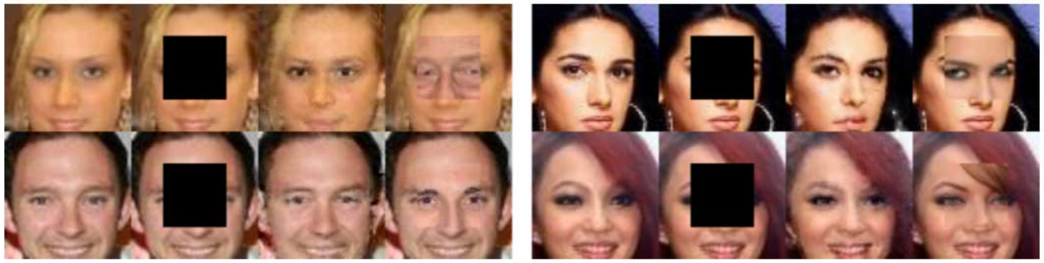
Trained on ensemble of images

Variational autoencoder, Kingma & Welling (2016)

Generative adversarial nets, Goodfellow et al (2014)

Deep convolutional GAN, Radford et al (2015)

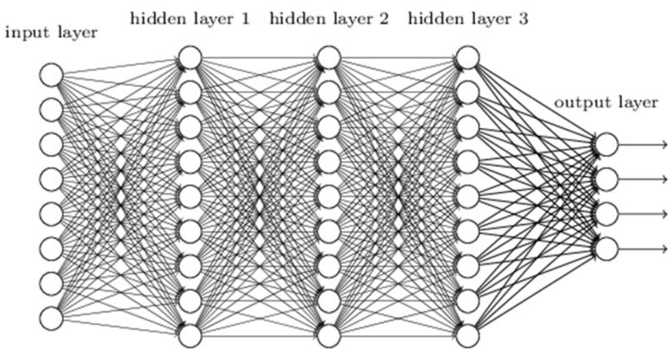
Example Results

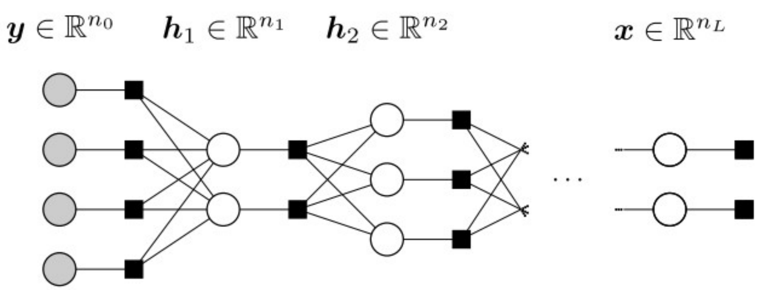


Yeh et al, Semantic Image Inpainting with Perceptual and Contextual Losses, 2016

- Example above:
  - Use DCGAN to train generative model: $x = G(z_0)$ and a discriminator cost $\log(1 - D(G(z_0))$
  - Loss minimized via gradient descent
- Works well in practice, but…
  - Difficult to analyze rigorously
  - Few theoretical guarantees
  - What are the limits on which this works?

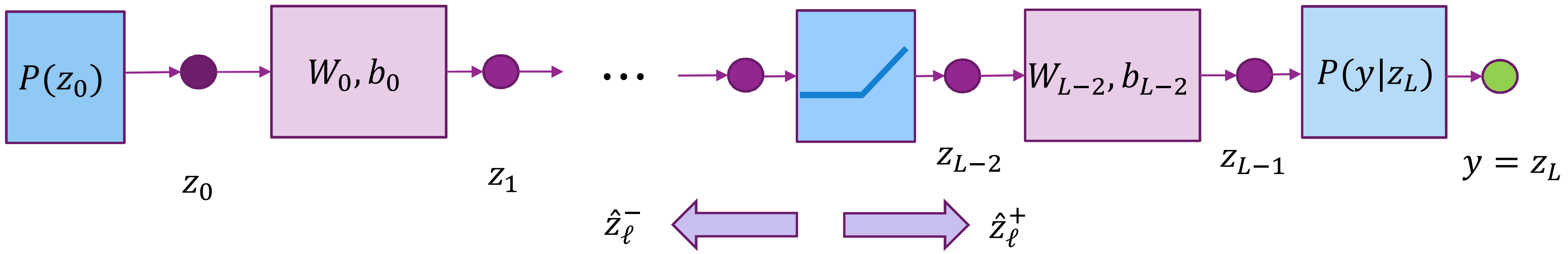
AMP for Multi-Layer Inference





- Use AMP for multi-layer inference
- Proposed by [Manoel, Krzakala, Mezard, Zdeborova, 2017]:
  - Derives simple AMP algorithm
  - Postulates state evolution, free energy, …
  - See also [Gabrié et al, 2016]
- But, limited to Gaussian iid
- Can VAMP do better?

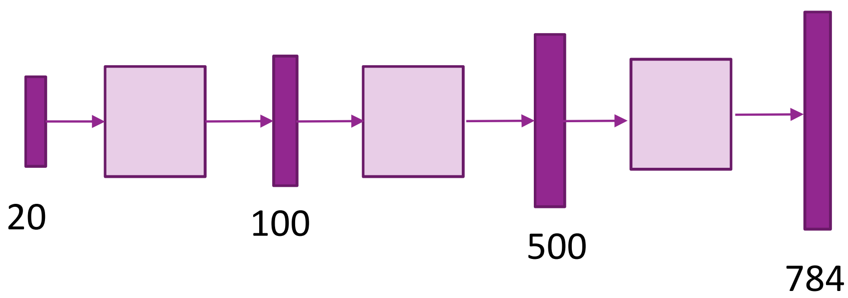
Multi-Layer VAMP



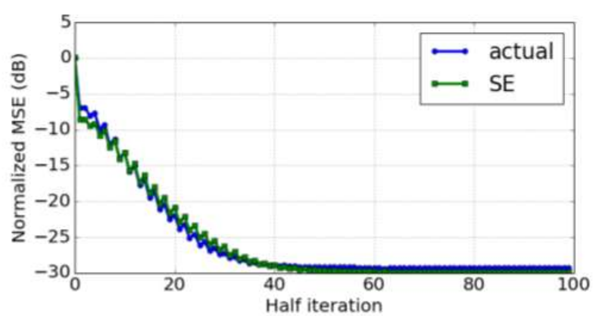
- Multi-layer VAMP: message passing method for multi-layer model
  - Derive with similar EC method as VAMP
  - Extension of [HeWenJin 2017] in GLM model
  - Updates have forward-backward iterations
- Applies to rotationally invariant weight matrices & separable activations
- Can rigorously prove state evolution

Fletcher, Rangan, Schniter. "Inference in deep networks in high dimensions." *Proc IEEE ISIT 2018*

Synthetic Data Example



MSE (dB)

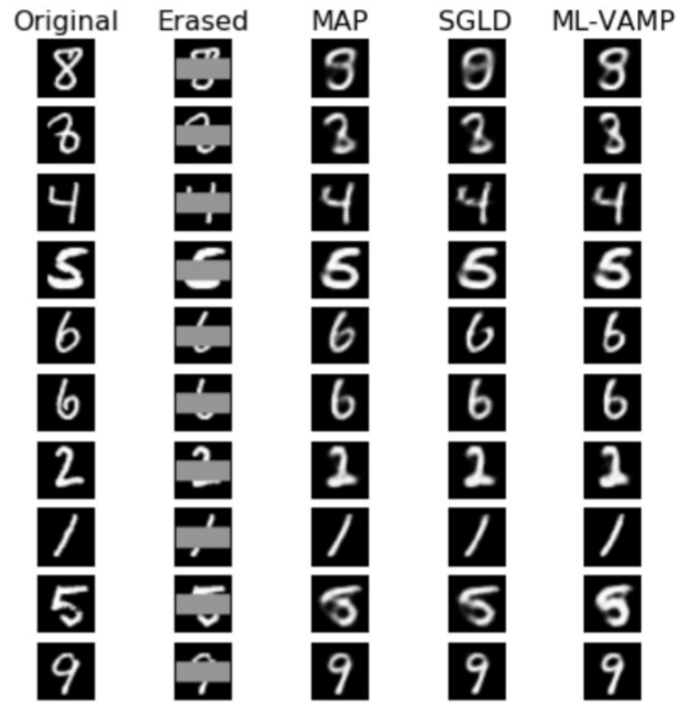


Half Iteration

- Simple network
  - $M = 3$ fully-connected layers
  - ReLU activations $z_{2m+1} = \max(z_{2m}, 0)$
- Random parameters
  - Gaussian iid W_0, W_1. Rotationally invariant W_2
  - Biases selected for sparsity at ReLU output
- Output AWGN at SNR=20 dB
- Good final estimate of posterior variance

Toy MNIST Inpainting

- Train network via VAE
  - Fully connected layers + ReLUs
  - 20 input variables
  - 400 hidden units
  - 784 dim output
- Perform ML-VAMP for inference
  - Need damping for stability
- Faster convergence:
  - MAP with ADAM optimizer: ~400 iterations
  - SGLD: ~20000 iterations
  - ML-VAMP: ~20 iterations



What is Known

Model	Iid Gaussian		Orthogonally Invariant	
	Algorithmic SE	Fundamental limit	Algorithmic SE	Fundamental limit
Linear	[DMM10,BM09]	[GV05,**RP16, BDMK16**]	[CVF14,RSF16]	[TUK06,TCVS13, BMMK17]
GLM	[Ran10,JavMon13]	[MKMZ17, **BKMMZ17**]	[SRF16,HWJ17]	[Reeves17,GML18+]
Multi-layer	[MKMZ17]	[MKMZ17]	[FRS18]	[Reeves17,GML18+]

- [Reeves17]: Reeves, "Additivity of Information in Multilayer Networks via Additive Gaussian Noise Transforms"
  - Postulates SE for ML-VAMP and rigorously proves this for Gaussian case
- [GML18+] Gabrié, Manoel, Luneau, Barbier, Macris, Krzakala, Zdeborová. "Entropy and mutual information in models of deep neural networks." *2018:*
  - Proves multi-layer model rigorously for large class of matrices and L=2 layers

Outline

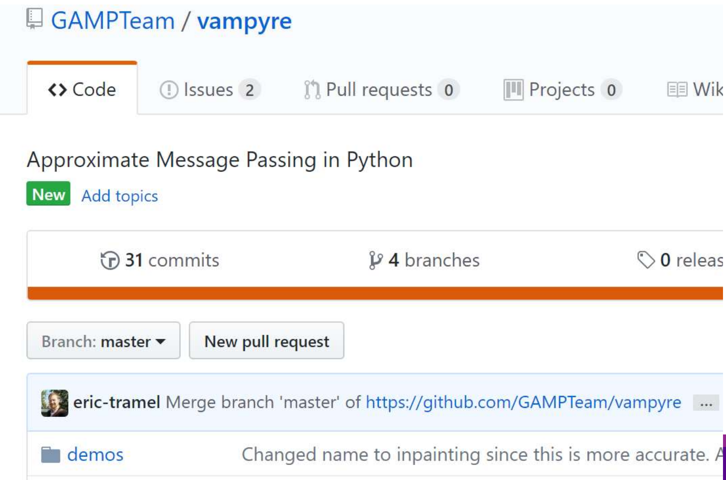
- AMP and Compressed Sensing
- Proximal Operators and ISTA
- State Evolution for AMP
- Bayes Denoising, Optimality and the Replica Method
- Belief Propagation and Factor Graphs
- AMP Derivation from Belief Propagation
- Convergence, Fixed Points and Stability
- Extensions: Vector AMP
- Thoughts on What is Next

Summary and Challenges

- Benefits of AMP
  - Enables rigorous analysis of complex inference problems in random settings
  - Computationally tractable
  - Identifies hard and easy regimes. Optimality guarantees
  - Can be extended to many complex problems
- Algorithmic issues: Still unstable. Requires damping, other tweaks
- Many models unsolved
  - Deep network model today covers inference, not learning
  - Can AMP understand learning multi-layer networks
  - Other key algorithms: VAE, GANs, …

Vampyre

- A new python package
- Thanks to Eric Tramel and others
- Modular, flexible, …
- Try it out!



Multi-Layer Perceptron Inpainting with MNIST

In the MLP demo, we saw how to use the multi-layer VAMP (ML-VAMP) method for der perceptron. We illustrated the method on synthetic data generated from a random MLP r with the MNIST data. Specifically, we consider the problem of estimating an MNIST digit ir form,

$$y = Ax,$$

where A is a sub-sampling operation. The sub-sampling operation outputs a subset of th occuluded area. This problem of reconstructing an image x with a portion of the imag requires a prior on the image. In this demo, we will use an MLP generative model for that p

Importing the Package

We first import the `vampyre` and other packages as in the sparse linear inverse demo.

```
# Add the vampyre path to the system path
import os
import sys
vp_path = os.path.abspath('../../')
if not vp_path in sys.path:
    sys.path.append(vp_path)
import vampyre as vp

# Load the other packages
import numpy as np
import matplotlib
```

Selected References 1

- L1 methods and ISTA
  - [Tib96] Tibshirani, Robert. "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society. Series B (Methodological)* (1996): 267-288.
  - [Mal08] Mallat, *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.
  - [ROF92] Rudin, Osher, Fatemi. "Nonlinear total variation based noise removal algorithms." *Physica D: nonlinear phenomena* 60.1-4 (1992): 259-268.
  - [BT09] Beck, Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems." *SIAM journal on imaging sciences* 2.1 (2009): 183-202.
- Compressed sensing early papers
  - [Donoho06] Donoho, "Compressed sensing." *IEEE Trans. Information theory* 52.4 (2006)
  - [CRT06] Candes, Romberg, Tao, "Stable signal recovery from incomplete and inaccurate measurements." *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 59.8 (2006): 1207-1223.
  - [LDSP08] Lustig, Donoho, Santos, Pauly, J. M. (2008). Compressed sensing MRI. *IEEE signal processing magazine*
- Website with exhaustive references / code / tutorials from Rice University: http://dsp.rice.edu/cs/

Selected References 2

- Early AMP papers

  - [DMM09] Donoho, Maleki, Montanari. "Message-passing algorithms for compressed sensing." *Proceedings of the National Academy of Sciences* 106.45 (2009): 18914-18919.
  - [DMM10] Donoho, Maleki, Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction." *Proc IEEE ITW, 20110*
  - [BM11] Bayati, Montanari. "The dynamics of message passing on dense graphs, with applications to compressed sensing." *IEEE Transactions on Information Theory* 57.2 (2011): 764-785.
  - [Mon12] Montanari, "Graphical models concepts in compressed sensing." *Compressed Sensing: Theory and Applications* (2012)
  - [Ran10] Rangan, Sundeep. "Estimation with random linear mixing, belief propagation and compressed sensing." *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*. IEEE, 2010.

Selected References 3

- AMP Connections to Statistical Physics and CDMA Detection

  - [BouCai02] Boutros, Caire. "Iterative multiuser joint decoding: Unified framework and asymptotic analysis." *IEEE Transactions on Information Theory* 48.7 (2002): 1772-1793.
  - [Tan02] Tanaka, Toshiyuki. "A statistical-mechanics approach to large-system analysis of CDMA multiuser detectors." *IEEE Transactions on Information theory* 48.11 (2002): 2888-2910.
  - [GuoV05] Guo, Dongning, and Sergio Verdú. "Randomly spread CDMA: Asymptotics via statistical physics." *IEEE Transactions on Information Theory* 51.6 (2005): 1983-2010.
  - [FMSSZ12] Krzakala, Mézard, Sausset, Y. F. Sun, Zdeborová. "Statistical-physics-based reconstruction in compressed sensing." *Physical Review X* 2, no. 2 (2012): 021005.
  - [RGF09] Rangan, Goyal, Fletcher. "Asymptotic analysis of map estimation via the replica method and compressed sensing." *Advances in Neural Information Processing Systems*. 2009.

Selected References 4

- Belief propagation, Expectation Consistent Approximate Inference, Expectation Propagation
  - [WJ08] Wainwright, Martin J., and Michael I. Jordan. "Graphical models, exponential families, and variational inference." *Foundations and Trends® in Machine Learning* 1.1–2 (2008): 1-305.
  - [YFW01] Yedidia, Freeman, Weiss. "Bethe free energy, Kikuchi approximations, and belief propagation algorithms." *Advances in neural information processing systems* 13 (2001).
  - [Minka01] Minka, Thomas P. "Expectation propagation for approximate Bayesian inference." *2001*
  - [OppWin04] Opper, Winther. "Expectation consistent free energies for approximate inference." *Advances in Neural Information Processing Systems*. 2004
  - [OppWin05] Opper, Manfred, and Ole Winther. "Expectation consistent approximate inference." *Journal of Machine Learning Research* 6.Dec (2005): 2177-2204.
- Generalized AMP
  - [Rangan11] Rangan, "Generalized approximate message passing for estimation with random linear mixing." *Proc IEEE ISIT 2011*
  - [JavMon13] Javanmard, Montanari. "State evolution for general approximate message passing algorithms, with applications to spatial coupling." *Information and Inference: A Journal of the IMA* 2.2 (2013): 115-144.

Selected References 5

- Vector Approximate Message Passing (VAMP) and other recent work

  - [RSF16] Rangan, Schniter, Fletcher, “Vector approximate message passing”, 2016
  - [CWF14] Cakmak, O. Winther, and B. H. Fleury, “S-AMP: Approximate message passing for general matrix ensembles,” IEEE ITW 14
  - [SRF16] Schniter, Rangan, Fletcher. "Vector approximate message passing for the generalized linear model." *Signals, Systems and Computers, 2016 50th Asilomar Conference on*. IEEE, 2016.
  - [MKMZ17] Manoel, A., Krzakala, F., Mézard, M., & Zdeborová, L. (2017, June). Multi-layer generalized linear estimation. Proc IEEE ISIT 2017
  - [FS17] Fletcher, Alyson K., and Philip Schniter. "Learning and free energies for vector approximate message passing." *Proc. IEEE ICASSP 2017*
  - [FRS18] Fletcher, Alyson K., Sundeep Rangan, and Philip Schniter. "Inference in deep networks in high dimensions." *Proc IEEE ISIT 2018*
  - [HWJ17] H. He, C.-K. Wen, and S. Jin, “Generalized expectation consistent signal recovery for nonlinear measurements,” arXiv:1701.04301, 2017.

Selected References 6

- Recent works on free energies, replica and optimality

  - [RP16] Reeves, Pfister. "The replica-symmetric prediction for compressed sensing with Gaussian matrices is exact." *Information Theory (ISIT), 2016 IEEE International Symposium on*. IEEE, 2016.
  - [BDMK+16] Barbier, J., Dia, M., Macris, N., Krzakala, F., Lesieur, T., & Zdeborová, L. (2016). Mutual information for symmetric rank-one matrix estimation: A proof of the replica formula. Proc NIPS 16
  - [Reeves17] Reeves "Additivity of information in multilayer networks via additive Gaussian noise transforms." *Allerton 2017*
  - [GML18+] Gabrié, Manoel, Luneau, Barbier, Macris, Krzakala, Zdeborová. "Entropy and mutual information in models of deep neural networks." *2018*
  - [BKMMZ17] Barbier, J., Krzakala, F., Macris, N., Miolane, L., & Zdeborová, L. (2017). Phase transitions, optimal errors and optimality of message-passing in generalized linear models.