Approximate Bayesian Inference for Generalized Linear Models:
A Message Passing Approach

Xiangming Meng
Huawei Technologies Co. Ltd., Shanghai, China
mengxm11@gmail.com

Feb. 27, 2019
RIKEN AIP, Tokyo

Outline

- **Problem Statement**
- **Standard Linear Models (SLM)**
  - **Approximate message passing**
- **Generalized Linear Models (GLM)**
  - **A unified inference framework**
- **Extension of GLM to Bilinear Models**
  - **Bilinear adaptive vector AMP**
- **Conclusion**

Problem Statement

☐ Generalized Linear Models (GLM)

Unknown Signal/parameters		linear mixing		probabilistic mapping		observations
$\mathbf{x} \sim p_0(\mathbf{x})$	$\xrightarrow{\mathbf{x} \in \mathbb{R}^N}$	$\mathbf{A} \in \mathbb{R}^{M \times N}$	$\xrightarrow{\mathbf{z} = \mathbf{A}\mathbf{x}}$	$p(y_a \mid z_a)$	$\longrightarrow$	$\mathbf{y} \in \mathbb{R}^M$

- **Goal**

  To infer the unknown signal/paremeters **x** from **y** and **A**

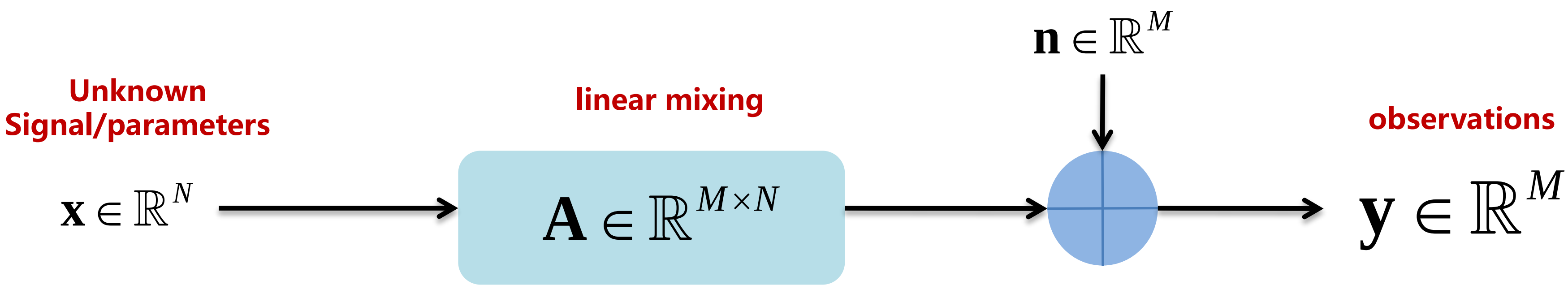
- **Applications**
  - ✓ Information theory: channel estimation, multi-user detection, etc.
  - ✓ Machine learning: linear regression, logistic regression, classification, etc.
  - ✓ Signal processing: compressed sensing, image processing, etc.
  - ✓ Many others…

Problem Statement

☐ Standard Linear Models (SLM)

Special case of GLM:

When the likelihood is Gaussian, GLM reduces to SLM



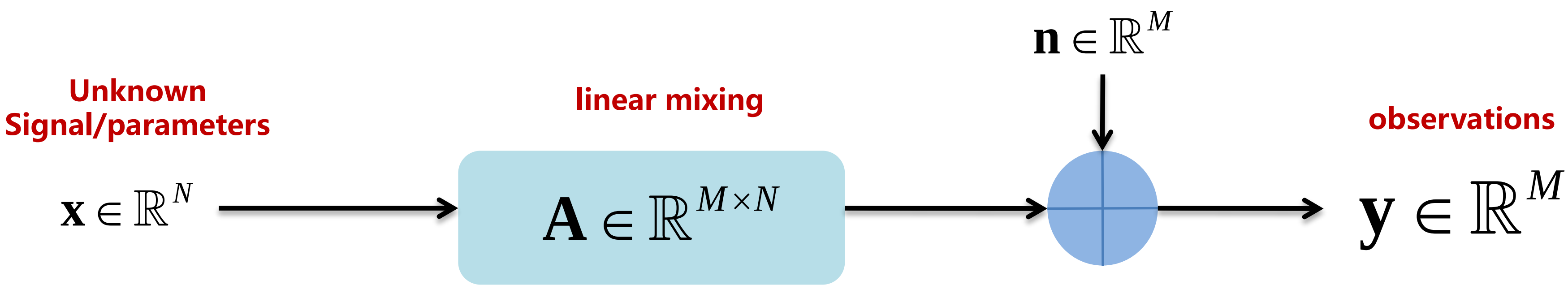
$\mathbf{x} \sim p_0(\mathbf{x})$ $$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$$ $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 I)$

Problem Statement

☐ Standard Linear Models (SLM)

Special case of GLM:

When the likelihood is Gaussian, GLM reduces to SLM



$\mathbf{x} \sim p_0(\mathbf{x})$

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$$

$\mathbf{n} \sim \mathcal{N}(0, \sigma^2 I)$

This is one fundamental model for linear inverse problem in science and engineering

Problem Statement

☐ Generalized Bilinear Models

Extended case of GLM:

The linear matrix A is also unknown or with uncertainty

Unknown Matrix — $\mathbf{X} \sim p_0\left(\mathbf{X};\theta_X\right)$ $\xrightarrow{\mathbf{x} \in \mathbb{R}^{N\times L}}$ **Linear mixing** — $\mathbf{A}$ **(unknown)** $\xrightarrow[\mathbf{Z} = \mathbf{A}\mathbf{X}]{\mathbf{Z} \in \mathbb{R}^{M\times L}}$ **Probabilistic mapping** — $p(\mathbf{Y} \mid \mathbf{Z};\theta_Y)$ $\longrightarrow$ **Observations** — $Y \in \mathbb{R}^{M\times L}$

- **Goal**

  To jointly infer matrix X and A, given Y with unknown parameters θ_X, θ_Y

Problem Statement

Generalized Bilinear Models

Extended case of GLM:

The linear matrix A is also unknown or with uncertainty

Unknown Matrix	Linear mixing		Probabilistic mapping	Observations
$\mathbf{X} \sim p_0(\mathbf{X};\theta_X)$ $\xrightarrow{\mathbf{x} \in \mathbb{R}^{N \times L}}$	$\mathbf{A}$ **(unknown)**	$\xrightarrow[\mathbf{Z} = \mathbf{AX}]{\mathbf{Z} \in \mathbb{R}^{M \times L}}$	$p(\mathbf{Y} \mid \mathbf{Z};\theta_Y)$ $\longrightarrow$	$Y \in \mathbb{R}^{M \times L}$

- **Goal**

  To **jointly infer matrix X and A**, given Y with **unknown parameters** θ_X, θ_Y

Matrix recovery problem

Problem Statement

☐ Generalized Bilinear Models

Extended case of GLM:

The linear matrix A is also unknown or with uncertainty

Unknown Matrix		Linear mixing		Probabilistic mapping		Observations
$\mathbf{X} \sim p_0(\mathbf{X};\theta_X)$	$\xrightarrow{\mathbf{X} \in \mathbb{R}^{N\times L}}$	$\mathbf{A}$ (unknown)	$\xrightarrow[\mathbf{Z}=\mathbf{AX}]{\mathbf{Z} \in \mathbb{R}^{M\times L}}$	$p(\mathbf{Y}\mid\mathbf{Z};\theta_Y)$	$\longrightarrow$	$Y \in \mathbb{R}^{M\times L}$

- **Goal**

  To jointly infer matrix X and A, given Y with unknown parameters θ_X, θ_Y

Matrix recovery problem

- **Applications**

  ✓ Machine learning: Probabilistic PCA, linear factor model, matrix factorization, matrix completion, etc.

  ✓ Signal processing: compressed sensing with matrix uncertainty, dictionary learning, etc.

  ✓ Other matrix recovery problems…

Problem Statement

☐ Generalized Bilinear Models

Extended case of GLM:
The linear matrix A is also unknown or with uncertainty

Unknown Matrix		Linear mixing		Probabilistic mapping		Observations
$\mathbf{X} \sim p_0(\mathbf{X};\theta_X)$	$\xrightarrow{\mathbf{x} \in \mathbb{R}^{N\times L}}$	**A (unknown)**	$\xrightarrow[\mathbf{Z} = \mathbf{AX}]{\mathbf{Z} \in \mathbb{R}^{M\times L}}$	$p(\mathbf{Y} \mid \mathbf{Z};\theta_Y)$	$\longrightarrow$	$Y \in \mathbb{R}^{M\times L}$

Matrix recovery problem

- **Goal**
  To jointly infer matrix X and A, given Y with unknown parameters θ_X, θ_Y

- **Applications**
  ✓ Machine learning: Probabilistic PCA, linear factor model, matrix factorization, matrix completion, etc.
  ✓ Signal processing: compressed sensing with matrix uncertainty, dictionary learning, etc.
  ✓ Other matrix recovery problems…

Bilinear recovery is much more difficult than original GLM since the linear mixing matrix is also unknown

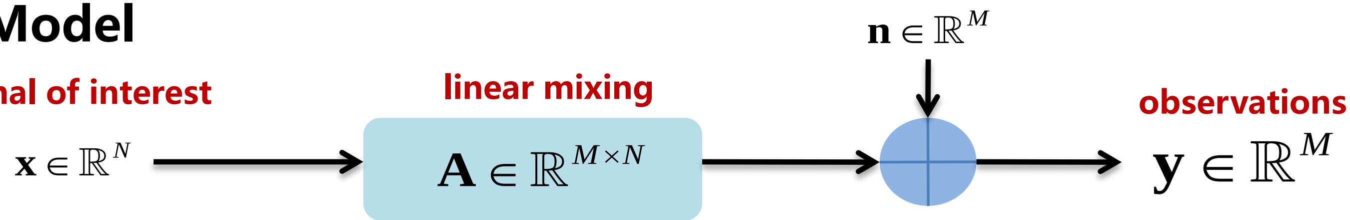
“If you can't solve a problem, then there is an easier problem you can solve: find it.” —George Pólya

(George Pólya: 1887 –1985)

I. Standard Linear Models

Standard Linear Models

☐ System Model



$$\mathbf{x} \sim p_0(\mathbf{x}) \qquad \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \qquad \mathbf{n} \sim \mathcal{N}(0, \sigma^2 I)$$

☐ Classical Methods

- **Least Squares Learning (LS)**

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 \qquad \hat{\mathbf{x}}_{LS} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{y}$$

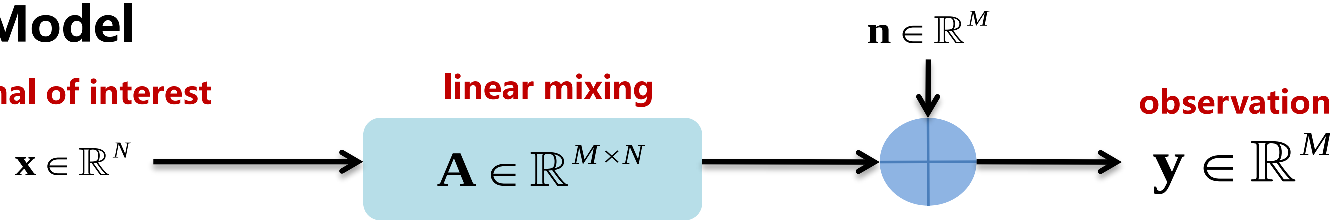
- **Regularized LS Learning**

✓L2 $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 + \frac{\lambda}{2}\|\mathbf{x}\|_2^2$ $\qquad \hat{\mathbf{x}}_{L2} = \left(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}\right)^{-1}\mathbf{A}^T\mathbf{y}$

✓L1 $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 + \lambda\|\mathbf{x}\|_1$ $\qquad$ Iterative soft threshold algorithm (ISTA)

Standard Linear Models

☐ System Model



$\mathbf{x} \sim p_0(\mathbf{x})$ $\qquad$ $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$ $\qquad$ $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 I)$

☐ Classical Methods

- **Least Squares Learning (LS)**

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 \qquad \hat{\mathbf{x}}_{LS} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{y}$$

- **Regularized LS Learning**

✓L2 $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 + \frac{\lambda}{2}\|\mathbf{x}\|_2^2$ $\qquad$ $\hat{\mathbf{x}}_{L2} = \left(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I}\right)^{-1}\mathbf{A}^T\mathbf{y}$

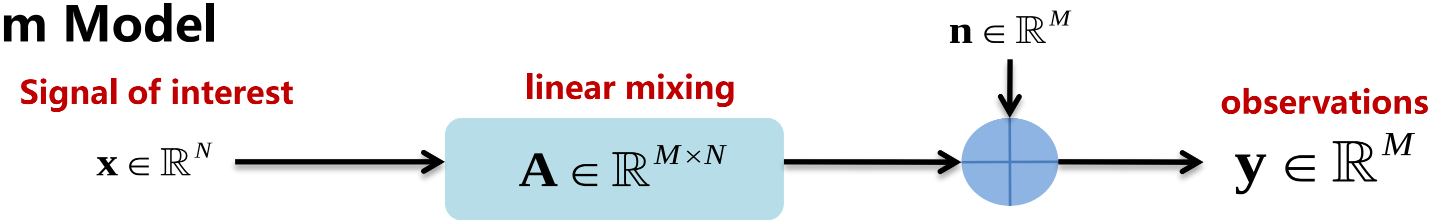
✓L1 $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 + \lambda\|\mathbf{x}\|_1$ $\qquad$ Iterative soft threshold algorithm (ISTA)

☐ Limits

- **Can not provide uncertainty estimates**
- **Poor performance with improper regularization**
- **High complexity even with closed-form solutions**
- **Slow convergence rate with stochastic or iterative methods**

Standard Linear Models

System Model



$\mathbf{x} \sim p_0(\mathbf{x})$ $$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$$ $\mathbf{n} \sim \mathcal{N}(0, \sigma^2 I)$

Classical Methods

- **Least Squares Learning (LS)**

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 \qquad \hat{\mathbf{x}}_{LS} = \left(\mathbf{A}^T \mathbf{A}\right)^{-1} \mathbf{A}^T \mathbf{y}$$

- **Regularized LS Learning**

✓L2 $$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 + \frac{\lambda}{2} \|\mathbf{x}\|_2^2 \qquad \hat{\mathbf{x}}_{L2} = \left(\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I}\right)^{-1} \mathbf{A}^T \mathbf{y}$$

✓L1 $$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2 + \lambda \|\mathbf{x}\|_1$$ Iterative soft threshold algorithm (ISTA)

Limits

- **Can not provide uncertainty estimates**
- **Poor performance with improper regularization**
- **High complexity even with closed-form solutions**
- **Slow convergence rate with stochastic or iterative methods**

Bayesian Solution?

Standard Linear Models

☐ Exact Bayesian Inference

According to the Bayes' rule, the *posterior distribution* can be computed as

$$p(\mathbf{x}\mid\mathbf{y}) = \frac{\underbrace{p_0(\mathbf{x})}_{\text{prior}}\,\underbrace{p(\mathbf{y}\mid\mathbf{x})}_{\text{likelihood}}}{\int p_0(\mathbf{x})\,p(\mathbf{y}\mid\mathbf{x})\,d\mathbf{x}}$$

Standard Linear Models

◻ Exact Bayesian Inference

According to the Bayes' rule, the *posterior distribution* can be computed as

$$p(\mathbf{x}|\mathbf{y}) = \frac{p_0(\mathbf{x})p(\mathbf{y}|\mathbf{x})}{\int p_0(\mathbf{x})p(\mathbf{y}|\mathbf{x})d\mathbf{x}} \xrightarrow{\text{marginalization}} p(x_i|\mathbf{y}) = \int_{\sim x_i} p(\mathbf{x}|\mathbf{y})d\mathbf{x}_{\backslash i}$$

Posterior mean $$\hat{x}_i^{MMSE} = \int x_i p(x_i|\mathbf{y})dx_i$$

Posterior variance $$v_i^{MMSE} = \int x_i^2 p(x_i|\mathbf{y})dx_i - \left(\hat{x}_i^{MMSE}\right)^2$$

Minimum mean square Error (MMSE) estimate

Standard Linear Models

❑ Exact Bayesian Inference

According to the Bayes' rule, the *posterior distribution* can be computed as

$$p(\mathbf{x}|\mathbf{y}) = \frac{p_0(\mathbf{x})p(\mathbf{y}|\mathbf{x})}{\int p_0(\mathbf{x})p(\mathbf{y}|\mathbf{x})d\mathbf{x}}$$

marginalization →

$$p(x_i|\mathbf{y}) = \int_{\sim x_i} p(\mathbf{x}|\mathbf{y})d\mathbf{x}_{\setminus i}$$

Posterior mean

$$\hat{x}_i^{MMSE} = \int x_i p(x_i|\mathbf{y})dx_i$$

Posterior variance

$$v_i^{MMSE} = \int x_i^2 p(x_i|\mathbf{y})dx_i - \left(\hat{x}_i^{MMSE}\right)^2$$

Minimum mean square Error (MMSE) estimate

✓ **No closed-form solutions**
There are no closed-form solutions for general problems

✓ **Curse of Dimensionality:**
Intractable due to high-dimensional integration/summation

Exact inference is intractable!

Standard Linear Models

Exact Bayesian Inference

According to the Bayes' rule, the *posterior distribution* can be computed as

$$p(\mathbf{x}|\mathbf{y}) = \frac{p_0(\mathbf{x})p(\mathbf{y}|\mathbf{x})}{\int p_0(\mathbf{x})p(\mathbf{y}|\mathbf{x})d\mathbf{x}} \quad \xrightarrow{\text{marginalization}} \quad p(x_i|\mathbf{y}) = \int_{\sim x_i} p(\mathbf{x}|\mathbf{y})d\mathbf{x}_{\setminus i}$$

Posterior mean $\qquad \hat{x}_i^{MMSE} = \int x_i p(x_i|\mathbf{y})dx_i$

Posterior variance $\qquad v_i^{MMSE} = \int x_i^2 p(x_i|\mathbf{y})dx_i - \left(\hat{x}_i^{MMSE}\right)^2$

Minimum mean square Error (MMSE) estimate

- ✓ **No closed-form solutions**
  There are no closed-form solutions for general problems

Exact inference is intractable!

- ✓ **Curse of Dimensionality:**
  Intractable due to high-dimensional integration/summation

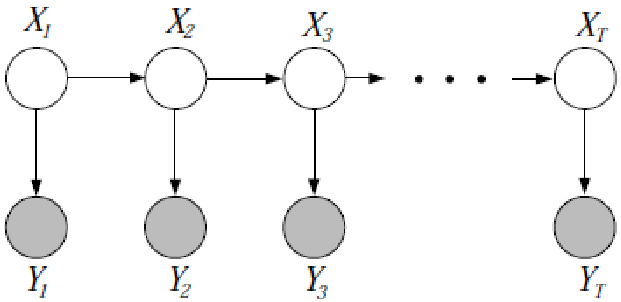
We have to resort to approximate inference methods

Standard Linear Models

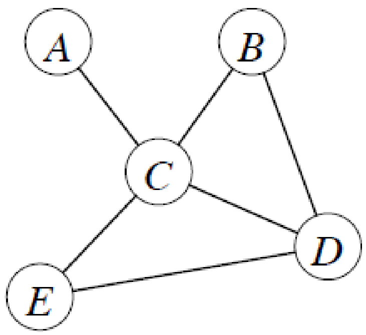
◻ Graphical Models and Message Passing

> **"Graphical Models are a marriage between probability theory and graph theory. "**
> **—Michael I. Jordan**

Intuitively, graphical models expresses the probabilistic relationship, i.e., *conditional dependence* structure between random variables.



HMM (directed models)



MRF(undirected models)

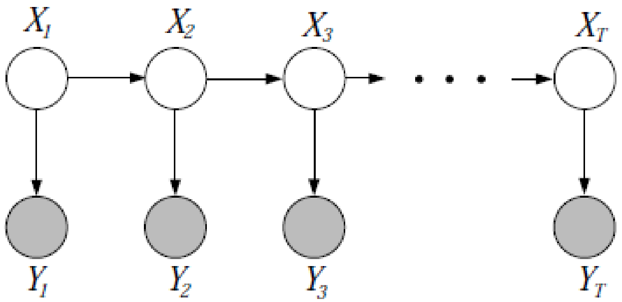
Standard Linear Models

Graphical Models and Message Passing

"Graphical Models are a marriage between probability theory and graph theory. "

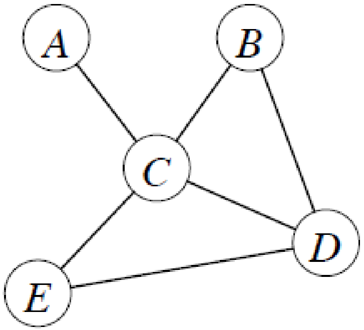
—Michael I. Jordan

Intuitively, graphical models expresses the probabilistic relationship, i.e., ***conditional dependence*** structure between random variables.



HMM (directed models)

Kalman filtering/Viterbi algorithm



MRF(undirected models)

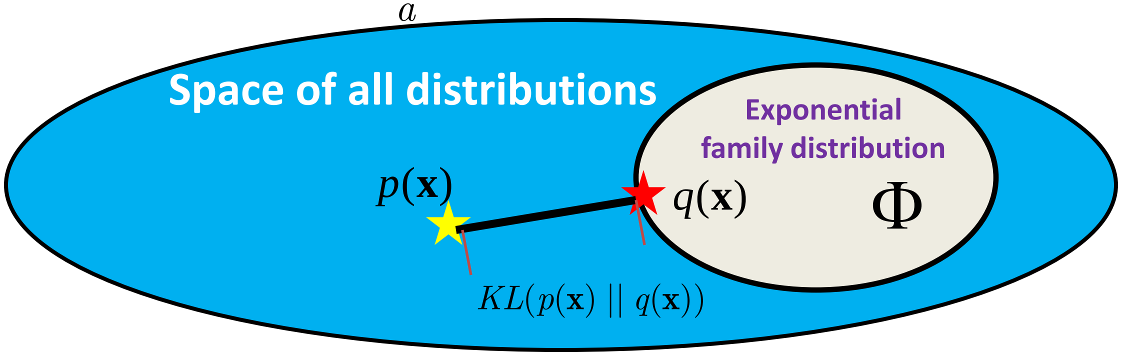
Belief propagation

Graphical Models not only provide a rich framework for representing high-dimensional statistical models, and more importantly, fascinates the design of efficient inference algorithm (e.g., message passing) in a principled manner.

Standard Linear Models

- **Expectation Propagation (EP)** [Minka01] [Opper05]

$$p(\mathbf{x}) = \prod_a f_a(\mathbf{x}) \xrightarrow{\text{approximated as}} q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x})$$

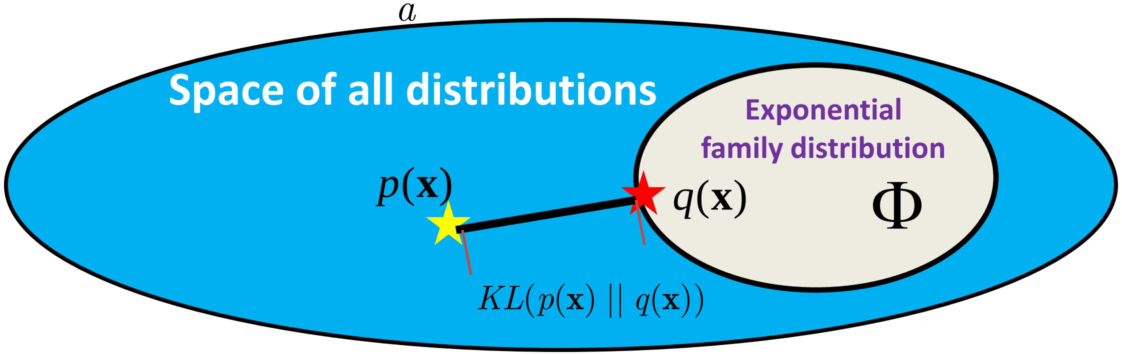


$$q(\mathbf{x}) = \mathrm{Proj}_{\Phi}\left(p(\mathbf{x})\right)$$

Standard Linear Models

- **Expectation Propagation (EP)** [Minka01] [Opper05]

$$p(\mathbf{x}) = \prod_a f_a(\mathbf{x}) \xrightarrow{\text{approximated as}} q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x})$$



$$q(\mathbf{x}) = \mathrm{Proj}_{\Phi}\big(p(\mathbf{x})\big)$$

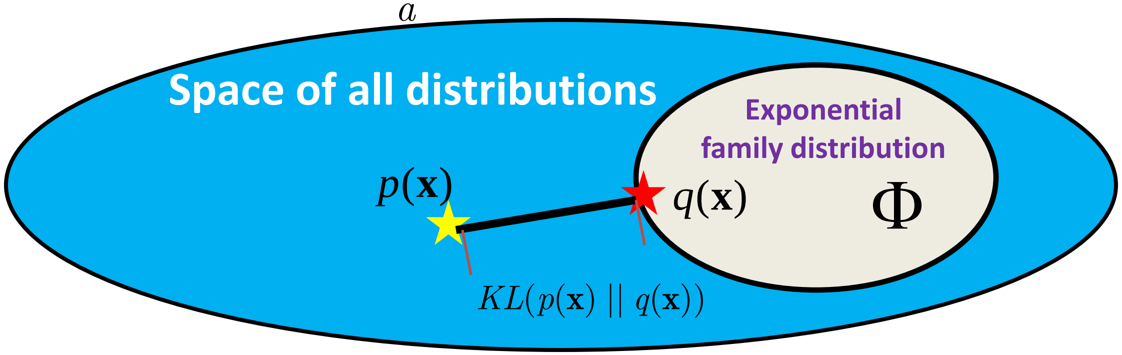
Optimization objective: $\min \; KL(p(\mathbf{x}) \,||\, q(\mathbf{x}))$

$$q(\boldsymbol{x}) = h(\boldsymbol{x}) \exp\left\{\boldsymbol{\theta}^T \phi(\boldsymbol{x}) + g(\boldsymbol{\theta})\right\}$$

Standard Linear Models

- **Expectation Propagation (EP)** [Minka01] [Opper05]

$$p(\mathbf{x}) = \prod_a f_a(\mathbf{x}) \xrightarrow{\text{approximated as}} q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x})$$



$$q(\mathbf{x}) = \mathrm{Proj}_{\Phi}\left(p(\mathbf{x})\right)$$

Optimization objective: $\min \; KL(p(\mathbf{x}) \,||\, q(\mathbf{x}))$

$$q(\boldsymbol{x}) = h(\boldsymbol{x}) \exp\left\{\boldsymbol{\theta}^T \phi(\boldsymbol{x}) + g(\boldsymbol{\theta})\right\}$$

Iterative local optimization

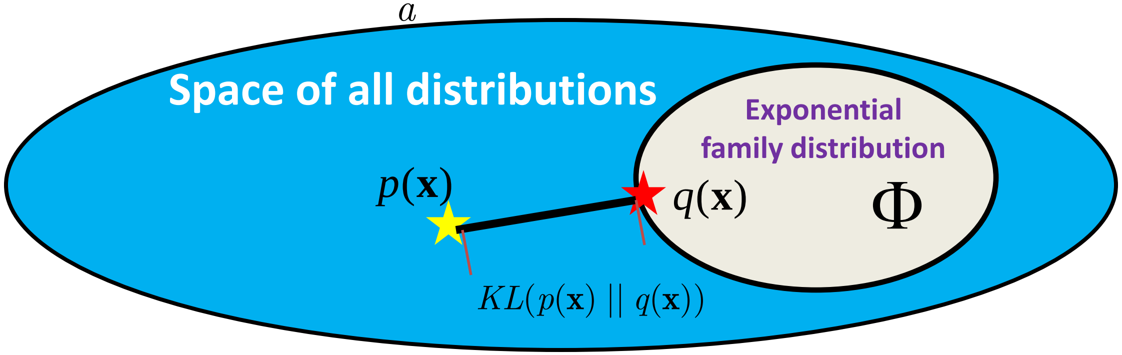
Iteratively refine each factor

$$\tilde{f}_a(\mathbf{x}) = \arg\min_{t(\mathbf{x}) \in \Phi} \; KL\Big(f_a(\mathbf{x}) \prod_{b \neq a} \tilde{f}_b(\mathbf{x}) \,||\, t(\mathbf{x}) \prod_{b \neq a} \tilde{f}_b(\mathbf{x})\Big)$$

Standard Linear Models

- **Expectation Propagation (EP)** [Minka01] [Opper05]

$$p(\mathbf{x}) = \prod_a f_a(\mathbf{x}) \xrightarrow{\text{approximated as}} q(\mathbf{x}) = \prod_a \tilde{f}_a(\mathbf{x})$$



$$q(\mathbf{x}) = \mathrm{Proj}_{\Phi}\left(p(\mathbf{x})\right)$$

Optimization objective: $\min \; KL(p(\mathbf{x}) \,||\, q(\mathbf{x}))$ $\quad q(\boldsymbol{x}) = h(\boldsymbol{x}) \exp\left\{\boldsymbol{\theta}^T \phi(\boldsymbol{x}) + g(\boldsymbol{\theta})\right\}$

Iterative local optimization

Iteratively refine each factor

$$\tilde{f}_a(\mathbf{x}) = \underset{t(\mathbf{x}) \in \Phi}{\arg\min} \; KL\left(f_a(\mathbf{x}) \prod_{b \neq a} \tilde{f}_b(\mathbf{x}) \,||\, t(\mathbf{x}) \prod_{b \neq a} \tilde{f}_b(\mathbf{x})\right)$$

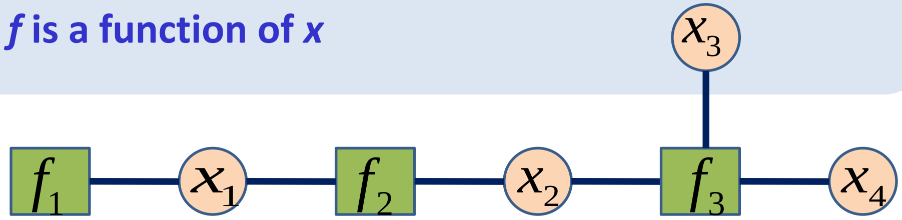
- Variational inference (VI) minimizes ***KL(q||p)*** while EP minimizes ***KL(p||q)***
- EP is one kind of iterative fixed-point algorithm
- EP can be also implemented as message passing on factor graph

Standard Linear Models

□ Expectation propagation [Minka01] [Opper05]

Factor Graph is one kind of bipartite graph which represents the factorization of a distribution where
- **Circles represent random variables**
- **Squares represent compatibility functions**
- **One circle *x* connects one square *f* if and only if *f* is a function of *x***

$$p(\mathbf{x}) = f_1(x_1)\, f_2(x_1, x_2)\, f_3(x_2, x_3, x_4)$$



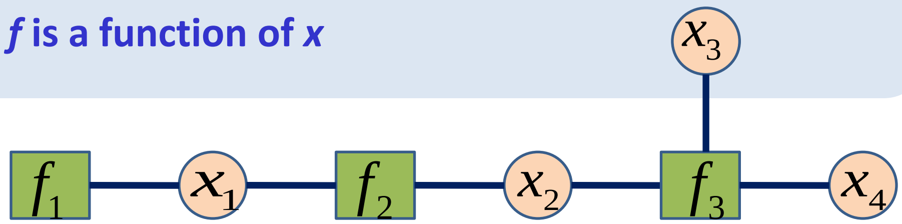
Standard Linear Models

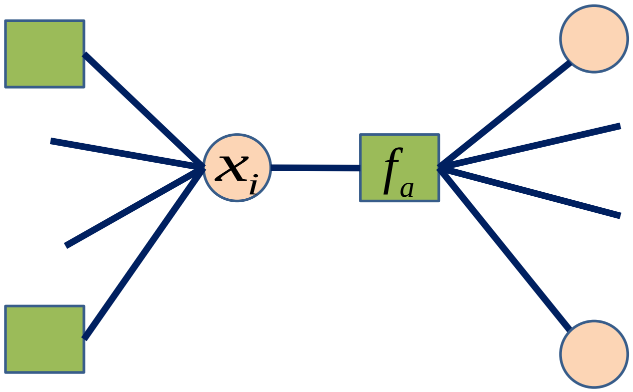
- ☐ **Expectation propagation** [Minka01] [Opper05]

Factor Graph is one kind of bipartite graph which represents the factorization of a distribution where

- **- Circles represent random variables**
- **- Squares represent compatibility functions**
- **- One circle *x* connects one square *f* if and only if *f* is a function of *x***

$$p(\mathbf{x}) = f_1(x_1)\, f_2(x_1, x_2)\, f_3(x_2, x_3, x_4)$$





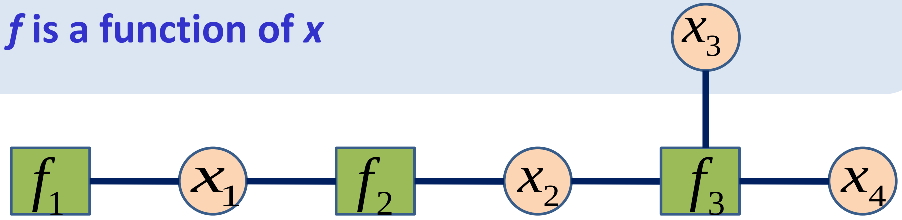
Local message passing for general factor graph

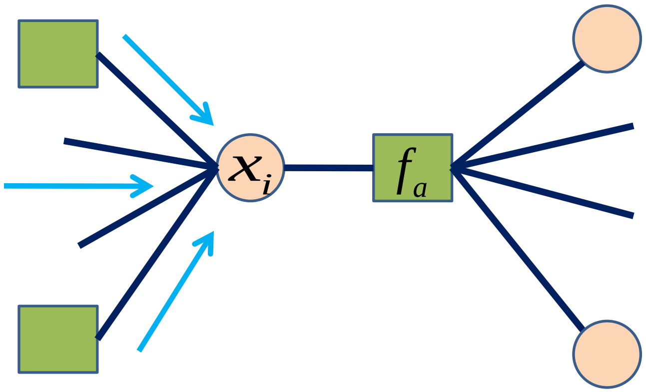
Standard Linear Models

- **Expectation propagation** [Minka01] [Opper05]

Factor Graph is one kind of bipartite graph which represents the factorization of a distribution where
- Circles represent random variables
- Squares represent compatibility functions
- One circle *x* connects one square *f* if and only if *f* is a function of *x*

$$p(\mathbf{x}) = f_1(x_1)\, f_2(x_1, x_2)\, f_3(x_2, x_3, x_4)$$





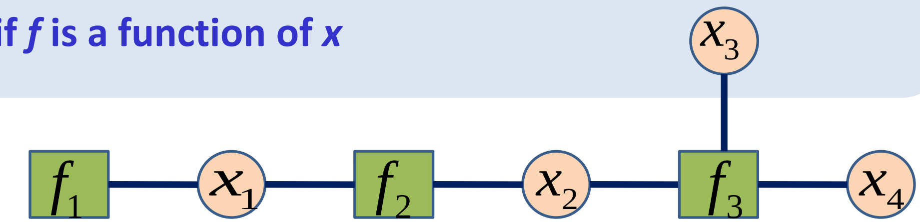
Local message passing for general factor graph

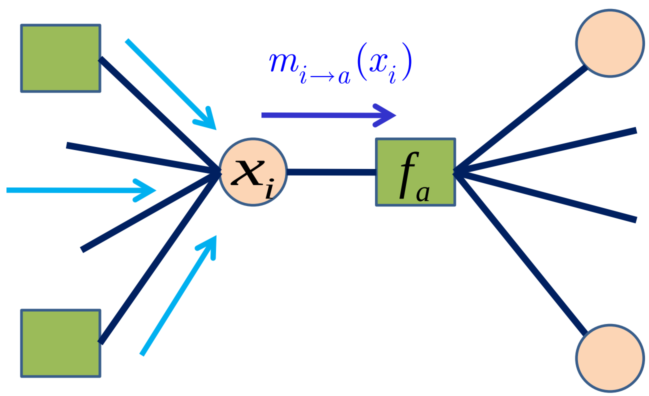
Standard Linear Models

☐ Expectation propagation [Minka01] [Opper05]

Factor Graph is one kind of bipartite graph which represents the factorization of a distribution where
- **Circles represent random variables**
- **Squares represent compatibility functions**
- **One circle *x* connects one square *f* if and only if *f* is a function of *x***

$$p(\mathbf{x}) = f_1(x_1)\, f_2(x_1, x_2)\, f_3(x_2, x_3, x_4)$$





Local message passing for general factor graph

Expectation Propagation (EP)

Factor to node: $$m_{a\to i}(x_i) \propto \frac{\mathrm{Proj}_{\Phi}\left[m_{i\to a}(x_i)\int f_a(\mathbf{x}_a) \prod_{j\in N(a), j\neq i} m_{j\to a}(x_j)\, d\mathbf{x}_{a\setminus i}\right]}{m_{i\to a}(x_i)}$$

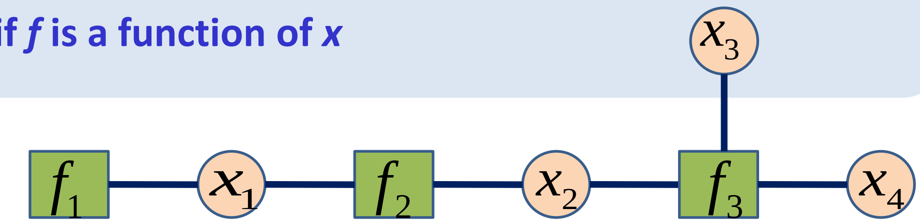
Standard Linear Models

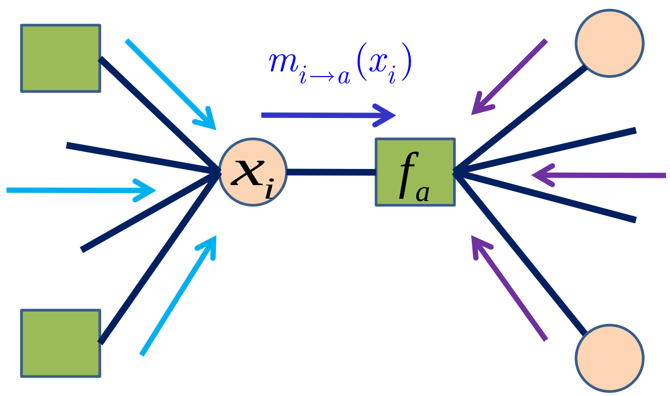
Expectation propagation [Minka01] [Opper05]

Factor Graph is one kind of bipartite graph which represents the factorization of a distribution where

- **Circles represent random variables**
- **Squares represent compatibility functions**
- **One circle *x* connects one square *f* if and only if *f* is a function of *x***

$$p(\mathbf{x}) = f_1(x_1)\, f_2(x_1, x_2)\, f_3(x_2, x_3, x_4)$$





Local message passing for general factor graph

Expectation Propagation (EP)

Factor to node: $$m_{a\to i}(x_i) \propto \frac{\mathrm{Proj}_{\Phi}\left[m_{i\to a}(x_i)\int f_a(\mathbf{x}_a) \prod_{j\in N(a), j\neq i} m_{j\to a}(x_j)\, d\mathbf{x}_{a\setminus i}\right]}{m_{i\to a}(x_i)}$$

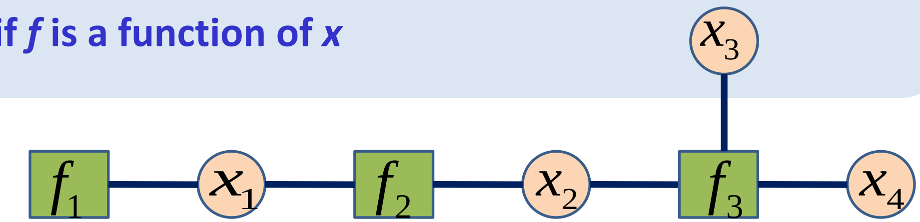
Standard Linear Models

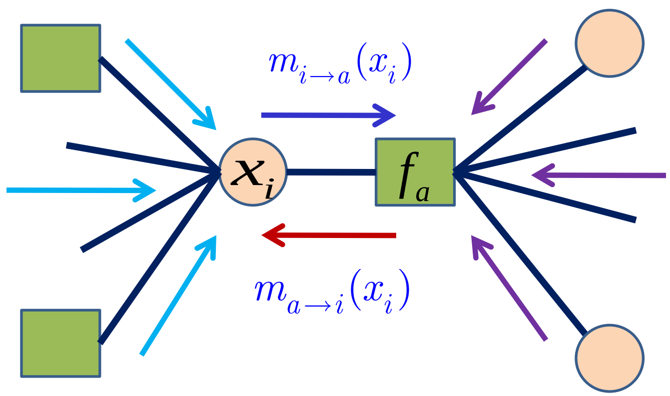
☐ Expectation propagation [Minka01] [Opper05]

Factor Graph is one kind of bipartite graph which represents the factorization of a distribution where

- **Circles represent random variables**
- **Squares represent compatibility functions**
- **One circle *x* connects one square *f* if and only if *f* is a function of *x***

$$p(\mathbf{x}) = f_1(x_1)\, f_2(x_1, x_2)\, f_3(x_2, x_3, x_4)$$





Local message passing for general factor graph

Expectation Propagation (EP)

Factor to node:
$$m_{a\to i}(x_i) \propto \frac{\mathrm{Proj}_{\Phi}\left[m_{i\to a}(x_i)\int f_a(\mathbf{x}_a) \prod_{j\in N(a), j\neq i} m_{j\to a}(x_j)\, d\mathbf{x}_{a\setminus i}\right]}{m_{i\to a}(x_i)}$$

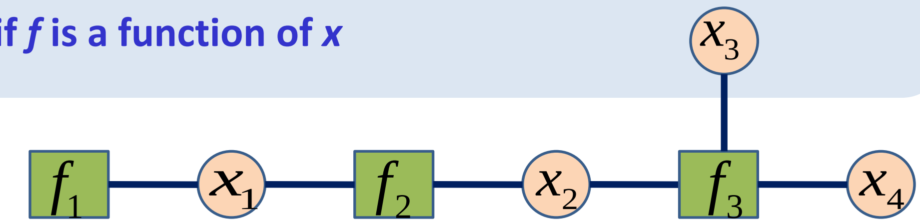
Node to factor:
$$m_{i\to a}(x_i) \propto \prod_{b\in N(i), b\neq a} m_{b\to i}(x_i)$$

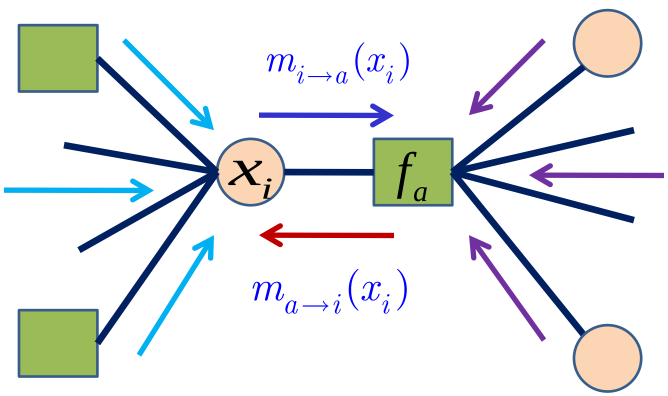
Standard Linear Models

□ Expectation propagation [Minka01] [Opper05]

Factor Graph is one kind of bipartite graph which represents the factorization of a distribution where
- **Circles represent random variables**
- **Squares represent compatibility functions**
- **One circle *x* connects one square *f* if and only if *f* is a function of *x***

$$p(\mathbf{x}) = f_1(x_1)\, f_2(x_1, x_2)\, f_3(x_2, x_3, x_4)$$





Local message passing for general factor graph

Expectation Propagation (EP)

Factor to node:
$$m_{a\to i}(x_i) \propto \frac{\mathrm{Proj}_{\Phi}\left[m_{i\to a}(x_i) \int f_a(\mathbf{x}_a) \prod_{j\in N(a), j\neq i} m_{j\to a}(x_j)\, d\mathbf{x}_{a\setminus i} \right]}{m_{i\to a}(x_i)}$$

Node to factor:
$$m_{i\to a}(x_i) \propto \prod_{b\in N(i), b\neq a} m_{b\to i}(x_i)$$

After convergence or a maximum number of iterations, the marginal distribution is the product of all the incoming messages from neighboring factors

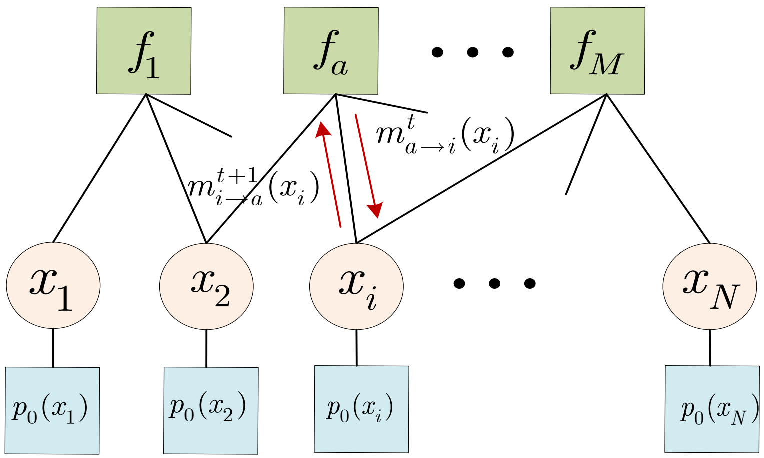
$$m_i(x_i) \propto \prod_{b\in N(i)} m_{b\to i}(x_i)$$

Standard Linear Models

□ Factor Graph of the SLM

For the SLM, the *posterior distribution* can be factorized as follows

$$p(\mathbf{x}\,|\,\mathbf{y}) \propto \prod_{i=1}^{N} p_0(x_i) \prod_{a=1}^{M} \underbrace{\mathcal{N}(y_a; \mathbf{a}^T\mathbf{x}, \sigma^2)}_{f_a}$$

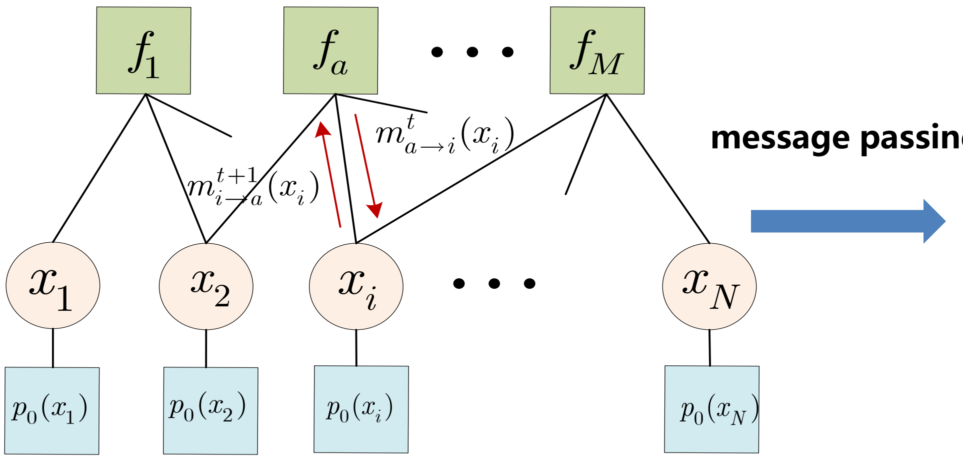


Standard Linear Models

◻ Factor Graph of the SLM

For the SLM, the ***posterior distribution*** can be factorized as follows

$$p(\mathbf{x}\,|\,\mathbf{y}) \propto \prod_{i=1}^{N} p_0(x_i) \prod_{a=1}^{M} \mathcal{N}\underbrace{(y_a; \mathbf{a}^T\mathbf{x}, \sigma^2)}_{f_a}$$



Expectation Propagation (EP)

$$m_{a\to i}^{t}(x_i) \propto \frac{\mathrm{Proj}_{\Phi}\left[m_{i\to a}^{t}(x_i)\int \prod_{j\neq i} m_{j\to a}^{t}(x_j)\, p(y_a|\mathbf{x})\right]}{m_{i\to a}^{t}(x_i)}$$

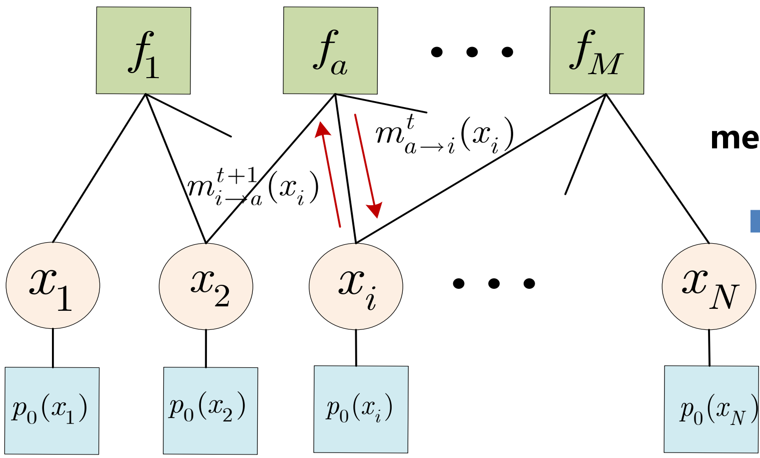
$$m_{i\to a}^{t+1}(x_i) \propto \frac{\mathrm{Proj}_{\Phi}\left[p_0(x_i)\prod_{b} m_{b\to i}^{t}(x_i)\right]}{m_{a\to i}^{t}(x_i)}$$

Standard Linear Models

◻ Factor Graph of the SLM

For the SLM, the *posterior distribution* can be factorized as follows

$$p(\mathbf{x}|\mathbf{y}) \propto \prod_{i=1}^{N} p_0(x_i) \prod_{a=1}^{M} \underbrace{\mathcal{N}(y_a;\mathbf{a}^T\mathbf{x},\sigma^2)}_{f_a}$$



message passing

Expectation Propagation (EP)

$$m^{t}_{a\to i}(x_i) \propto \frac{\mathrm{Proj}_{\Phi}\left[m^{t}_{i\to a}(x_i)\int \prod_{j\neq i} m^{t}_{j\to a}(x_j)\,p(y_a|\mathbf{x})\right]}{m^{t}_{i\to a}(x_i)}$$

$$m^{t+1}_{i\to a}(x_i) \propto \frac{\mathrm{Proj}_{\Phi}\left[p_0(x_i)\prod_{b} m^{t}_{b\to i}(x_i)\right]}{m^{t}_{a\to i}(x_i)}$$

The projection set Φ is chosen to be Gaussian distribution so that the messages become Gaussian distribution

Standard Linear Models

◻ An EP Perspective on AMP

$$m_{a\to i}^{t}(x_i) \propto \mathcal{N}(x_i; \hat{x}_{a\to i}^{t}, v_{a\to i}^{t})$$

$$m_{i\to a}^{t+1}(x_i) \propto \mathcal{N}(x_i; \hat{x}_{i\to a}^{t+1}, v_{i\to a}^{t+1})$$

where

$$V_{a\to i}^{t} = \sum_{j\neq i} |A_{aj}|^2 \nu_{j\to a}^{t} \qquad Z_{a\to i}^{t} = \sum_{j\neq i} A_{aj} \hat{x}_{j\to a}^{t}$$

$$\hat{x}_{a\to i}^{t} = \frac{y_a - Z_{a\to i}^{t}}{A_{ai}}, v_{a\to i}^{t} = \frac{\sigma^2 + V_{a\to i}^{t}}{|A_{ai}|^2}$$

$$\Sigma_i^t = \left[\sum_a \frac{|A_{ai}|^2}{\sigma^2 + V_{a\to i}^{t}}\right]^{-1} \qquad R_i^t = \Sigma_i^t \sum_a \frac{A_{ai}^*(y_a - Z_{a\to i}^{t})}{\sigma^2 + V_{a\to i}^{t}}$$

$$\hat{x}_i^{t+1} = f_a\left(R_i^t, \Sigma_i^t\right) \qquad \hat{\nu}_i^{t+1} = f_c\left(R_i^t, \Sigma_i^t\right)$$

$$\frac{1}{\nu_{i\to a}^{t+1}} = \frac{1}{\nu_i^{t+1}} - \frac{|A_{ai}|^2}{\sigma^2 + V_{a\to i}^{t}},$$

$$\hat{x}_{i\to a}^{t+1} = \nu_{i\to a}^{t+1}\left(\frac{\hat{x}_i^{t+1}}{\nu_i^{t+1}} - \frac{A_{ai}^*(y_a - Z_{a\to i}^{t})}{\sigma^2 + V_{a\to i}^{t}}\right).$$

Standard Linear Models

□ An EP Perspective on AMP

$$m^t_{a\to i}(x_i) \propto \mathcal{N}(x_i; \hat{x}^t_{a\to i}, v^t_{a\to i})$$

$$m^{t+1}_{i\to a}(x_i) \propto \mathcal{N}(x_i; \hat{x}^{t+1}_{i\to a}, v^{t+1}_{i\to a})$$

where

$$V^t_{a\to i} = \sum_{j\neq i} |A_{aj}|^2 \nu^t_{j\to a} \qquad Z^t_{a\to i} = \sum_{j\neq i} A_{aj}\hat{x}^t_{j\to a}$$

$$\hat{x}^t_{a\to i} = \frac{y_a - Z^t_{a\to i}}{A_{ai}}, v^t_{a\to i} = \frac{\sigma^2 + V^t_{a\to i}}{|A_{ai}|^2}$$

$$\Sigma^t_i = \Big[\sum_a \frac{|A_{ai}|^2}{\sigma^2 + V^t_{a\to i}}\Big]^{-1} \qquad R^t_i = \Sigma^t_i \sum_a \frac{A^*_{ai}(y_a - Z^t_{a\to i})}{\sigma^2 + V^t_{a\to i}}$$

$$\hat{x}^{t+1}_i = f_a\left(R^t_i, \Sigma^t_i\right) \qquad \hat{\nu}^{t+1}_i = f_c\left(R^t_i, \Sigma^t_i\right)$$

$$\frac{1}{\nu^{t+1}_{i\to a}} = \frac{1}{\nu^{t+1}_i} - \frac{|A_{ai}|^2}{\sigma^2 + V^t_{a\to i}},$$

$$\hat{x}^{t+1}_{i\to a} = \nu^{t+1}_{i\to a}\Big(\frac{\hat{x}^{t+1}_i}{\nu^{t+1}_i} - \frac{A^*_{ai}(y}{\sigma^2 +}$$

Still Too Complicated!

- **However, the number of messages are *O*(*MN*), which is still intractable for high-dimensional problems**

Standard Linear Models

An EP Perspective on AMP

$$m_{a\to i}^{t}(x_i) \propto \mathcal{N}(x_i; \hat{x}_{a\to i}^{t}, v_{a\to i}^{t})$$

$$m_{i\to a}^{t+1}(x_i) \propto \mathcal{N}(x_i; \hat{x}_{i\to a}^{t+1}, v_{i\to a}^{t+1})$$

where

$$V_{a\to i}^{t} = \sum_{j\neq i} |A_{aj}|^2 \nu_{j\to a}^{t} \qquad Z_{a\to i}^{t} = \sum_{j\neq i} A_{aj}\hat{x}_{j\to a}^{t}$$

$$\hat{x}_{a\to i}^{t} = \frac{y_a - Z_{a\to i}^{t}}{A_{ai}}, v_{a\to i}^{t} = \frac{\sigma^2 + V_{a\to i}^{t}}{|A_{ai}|^2}$$

$$\Sigma_i^t = \left[\sum_a \frac{|A_{ai}|^2}{\sigma^2 + V_{a\to i}^{t}}\right]^{-1} \qquad R_i^t = \Sigma_i^t \sum_a \frac{A_{ai}^{*}(y_a - Z_{a\to i}^{t})}{\sigma^2 + V_{a\to i}^{t}}$$

$$\hat{x}_i^{t+1} = f_a\left(R_i^t, \Sigma_i^t\right) \qquad \hat{\nu}_i^{t+1} = f_c\left(R_i^t, \Sigma_i^t\right)$$

$$\frac{1}{\nu_{i\to a}^{t+1}} = \frac{1}{\nu_i^{t+1}} - \frac{|A_{ai}|^2}{\sigma^2 + V_{a\to i}^{t}},$$

$$\hat{x}_{i\to a}^{t+1} = \nu_{i\to a}^{t+1}\left(\frac{\hat{x}_i^{t+1}}{\nu_i^{t+1}} - \frac{A_{ai}^{*}(y}{\sigma^2 +}\right.$$

Still Too Complicated!

- However, the **number of messages are *O*(*MN*)**, which is still intractable for high-dimensional problems
- To reduce the number of messages, **neglecting the high-order terms** in large system limits

$$Z_a^t = \sum_i A_{ai}\hat{x}_{i\to a}^{t} \qquad V_a^t = \sum_i |A_{ai}|^2 \nu_{i\to a}^{t}$$

$$Z_{a\to i}^{t} = Z_a^t - A_{ai}\hat{x}_{i\to a}^{t},$$ **Be careful!**

$$V_{a\to i}^{t} = V_a^t - |A_{ai}|^2 \nu_{i\to a}^{t} \;\Rightarrow\; V_{a\to i}^{t} \approx V_a^t$$

$$\nu_{i\to a}^{t+1} \approx \nu_i^{t+1} \;\Rightarrow\; V_a^t \approx \sum_i |A_{ai}|^2 \nu_i^t$$

Standard Linear Models

☐ An EP Perspective on AMP

$$m^t_{a\to i}(x_i) \propto \mathcal{N}(x_i; \hat{x}^t_{a\to i}, v^t_{a\to i})$$

$$m^{t+1}_{i\to a}(x_i) \propto \mathcal{N}(x_i; \hat{x}^{t+1}_{i\to a}, v^{t+1}_{i\to a})$$

where

$$V^t_{a\to i} = \sum_{j\neq i} |A_{aj}|^2 \nu^t_{j\to a} \qquad Z^t_{a\to i} = \sum_{j\neq i} A_{aj}\hat{x}^t_{j\to a}$$

$$\hat{x}^t_{a\to i} = \frac{y_a - Z^t_{a\to i}}{A_{ai}}, \quad v^t_{a\to i} = \frac{\sigma^2 + V^t_{a\to i}}{|A_{ai}|^2}$$

$$\Sigma^t_i = \left[\sum_a \frac{|A_{ai}|^2}{\sigma^2 + V^t_{a\to i}}\right]^{-1} \qquad R^t_i = \Sigma^t_i \sum_a \frac{A^*_{ai}(y_a - Z^t_{a\to i})}{\sigma^2 + V^t_{a\to i}}$$

$$\hat{x}^{t+1}_i = f_a(R^t_i, \Sigma^t_i) \qquad \hat{\nu}^{t+1}_i = f_c(R^t_i, \Sigma^t_i)$$

$$\frac{1}{\nu^{t+1}_{i\to a}} = \frac{1}{\nu^{t+1}_i} - \frac{|A_{ai}|^2}{\sigma^2 + V^t_{a\to i}},$$

$$\hat{x}^{t+1}_{i\to a} = \nu^{t+1}_{i\to a}\left(\frac{\hat{x}^{t+1}_i}{\nu^{t+1}_i} - \frac{A^*_{ai}(y}{\sigma^2 +}\right.$$

Still Too Complicated!

- However, the number of messages are $O(MN)$, which is still intractable for high-dimensional problems
- To reduce the number of messages, neglecting the high-order terms in large system limits

$$Z^t_a = \sum_i A_{ai}\hat{x}^t_{i\to a} \qquad V^t_a = \sum_i |A_{ai}|^2 \nu^t_{i\to a}$$

$$Z^t_{a\to i} = Z^t_a - A_{ai}\hat{x}^t_{i\to a}, \Rightarrow \textbf{Be careful!}$$

$$V^t_{a\to i} = V^t_a - |A_{ai}|^2 \nu^t_{i\to a} \Rightarrow V^t_{a\to i} \approx V^t_a$$

$$\nu^{t+1}_{i\to a} \approx \nu^{t+1}_i \Rightarrow V^t_a \approx \sum_i |A_{ai}|^2 \nu^t_i$$

- After some algebra, we obtain the famous approximate message passing (AMP) algorithm.

X. Meng, S. Wu, L. Kuang, and J. Lu, "An expectation propagation perspective on approximate message passing," IEEE Signal Processing Letters, vol. 22, no. 8, pp. 1194-1197, Aug. 2015.

AMP Algorithm

Initialization

Loop: For $t = 1,\dots, T$

Factor node update:

$$V^t_a = \sum_i A^2_{ai}\nu^t_i$$

$$Z^t_a = \sum_i A_{ai}\hat{x}^t_i - \frac{(y_a - Z^{t-1}_a)}{\sigma^2 + V^{t-1}_a} V^t_a$$

Onsager term

Variable node update:

$$\Sigma^t_i = 1 \Big/ \sum_a \frac{A^2_{ai}}{\sigma^2 + V^t_a}$$

$$R^t_i = \hat{x}^t_i + \Sigma^t_i \sum_a \frac{A_{ai}(y_a - Z^t_a)}{\sigma^2 + V^t_a}$$

$$\hat{x}^{t+1}_i = E(x_i \mid R^t_i, \Sigma^t_i), \hat{\nu}^{t+1}_i = Var(x_i \mid R^t_i, \Sigma^t_i)$$

End

Linear Complexity $O(MN)$

Standard Linear Models

An EP Perspective on AMP

AMP iteratively decouples the original vector inference problem to scalar inference problems

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \quad \xrightarrow{\text{decoupled}} \quad \begin{cases} R_1 = x_1 + \tilde{n}_1 \\ \vdots \\ R_N = x_N + \tilde{n}_N \end{cases}$$

decoupling principle

Standard Linear Models

☐ An EP Perspective on AMP

AMP iteratively decouples the original vector inference problem to scalar inference problems

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \quad \xrightarrow{\text{decoupled}} \quad \begin{cases} R_1 = x_1 + \tilde{n}_1 \\ \vdots \\ R_N = x_N + \tilde{n}_N \end{cases}$$

decoupling principle

- **Notes of AMP**
  - ✓ For i.i.d. Gaussian **A**, AMP is proved to be asymptotically Bayesian optimal and rigorously analyzed via state evolution (SE) [BM11]
  - ✓ For general matrices **A**, AMP may diverge [BM11]
  - ✓ Vector AMP (VAMP) converges for right-rotationally invariant matrices [RSF16]

Standard Linear Models

◻ An EP Perspective on AMP

AMP iteratively decouples the original vector inference problem to scalar inference problems

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \xrightarrow{\text{decoupled}} \begin{cases} R_1 = x_1 + \tilde{n}_1 \\ \vdots \\ R_N = x_N + \tilde{n}_N \end{cases}$$

decoupling principle

- **Notes of AMP**
  - ✓For i.i.d. Gaussian **A**, AMP is proved to be asymptotically Bayesian optimal and rigorously analyzed via state evolution (SE) [BM11]
  - ✓For general matrices **A**, AMP may diverge [BM11]
  - ✓Vector AMP (VAMP) converges for right-rotationally invariant matrices [RSF16]

- **The EP perspective of AMP:**
  - ✓ Explicitly establishing the relationship between AMP and EP for the first time
  - ✓ Simplifying the extension of AMP to the complex-valued AMP (simply using circularly-symmetric Gaussian) [MWKL15b]
  - ✓ Providing a unified view of AMP and VAMP (derived from scalar EP [MWKL15a] and vector EP [RSF16], respectively)

X. Meng, S. Wu, L. Kuang, and J. Lu, “An expectation propagation perspective on approximate message passing,” IEEE Signal Processing Letters, vol. 22, no. 8, pp. 1194-1197, Aug. 2015.

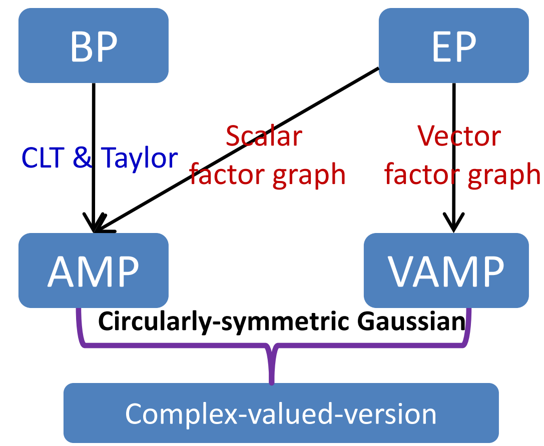
Standard Linear Models

☐ An EP Perspective on AMP

AMP iteratively decouples the original vector inference problem to scalar inference problems

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \xrightarrow{\textbf{decoupled}} \begin{cases} R_1 = x_1 + \tilde{n}_1 \\ \vdots \\ R_N = x_N + \tilde{n}_N \end{cases}$$

decoupling principle



- **Notes of AMP**
  - ✓For i.i.d. Gaussian **A**, AMP is proved to be asymptotically Bayesian optimal and rigorously analyzed via state evolution (SE) [BM11]
  - ✓For general matrices **A**, AMP may diverge [BM11]
  - ✓Vector AMP (VAMP) converges for right-rotationally invariant matrices [RSF16]

- **The EP perspective of AMP:**
  - ✓ Explicitly establishing the relationship between AMP and EP for the first time
  - ✓ Simplifying the extension of AMP to the complex-valued AMP (simply using circularly-symmetric Gaussian) [MWKL15b]
  - ✓ Providing a unified view of AMP and VAMP (derived from scalar EP [MWKL15a] and vector EP [RSF16], respectively)

X. Meng, S. Wu, L. Kuang, and J. Lu, "An expectation propagation perspective on approximate message passing," IEEE Signal Processing Letters, vol. 22, no. 8, pp. 1194-1197, Aug. 2015.

II. Generalized Linear Models

Generalized Linear Models

◻ Motivations

Unknown Signals/parameters | linear mixing | probabilistic mapping | observations

$$\mathbf{x} \sim p_0(\mathbf{x}) \xrightarrow{\mathbf{x} \in \mathbb{R}^N} \boxed{\mathbf{A} \in \mathbb{R}^{M \times N}} \xrightarrow{\mathbf{z} = \mathbf{A}\mathbf{x}} \boxed{p(y_a \mid z_a)} \longrightarrow \mathbf{y} \in \mathbb{R}^M$$

- **GLM is more general: the measurements are often obtained in a nonlinear way**
  - ✓ Difficult to perform inference due to the nonlinearity (non-Gaussian likelihood)
- **SLM inference algorithms have already been extensively studied**
  - ✓ Simple to design and analyze
  - ✓ Various algorithms, e.g., AMP and sparse Bayesian learning (SBL) have already been proposed

Generalized Linear Models

☐ Motivations

Unknown Signals/parameters		linear mixing		probabilistic mapping		observations
$\mathbf{x} \sim p_0(\mathbf{x})$	$\xrightarrow{\mathbf{x} \in \mathbb{R}^N}$	$\mathbf{A} \in \mathbb{R}^{M \times N}$	$\xrightarrow{\mathbf{z} = \mathbf{A}\mathbf{x}}$	$p(y_a \mid z_a)$	$\longrightarrow$	$\mathbf{y} \in \mathbb{R}^M$

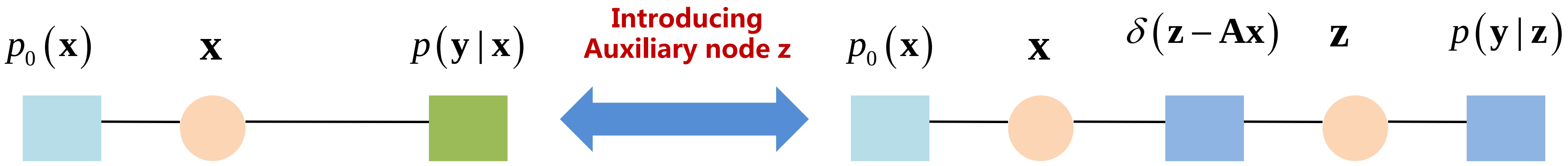
- **GLM is more general: the measurements are often obtained in a nonlinear way**
  - ✓ Difficult to perform inference due to the nonlinearity (non-Gaussian likelihood)
- **SLM inference algorithms have already been extensively studied**
  - ✓ Simple to design and analyze
  - ✓ Various algorithms, e.g., AMP and sparse Bayesian learning (SBL) have already been proposed

Is it possible to perform the GLM inference using the existing SLM inference algorithms?

SLM Inference Algorithms $\xrightarrow{\text{Easily extended?}}$ **GLM Inference Algorithms**

A Unified Inference Framework for GLM

□ Two Equivalent Factor Graphs of GLM

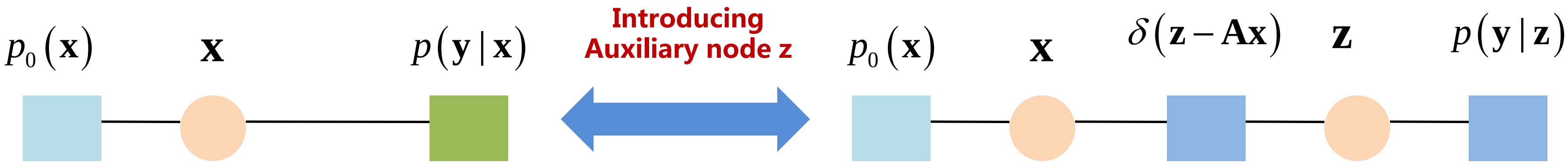


(a) factor graph of GLM

(b) Equivalent factor graph of GLM

A Unified Inference Framework for GLM

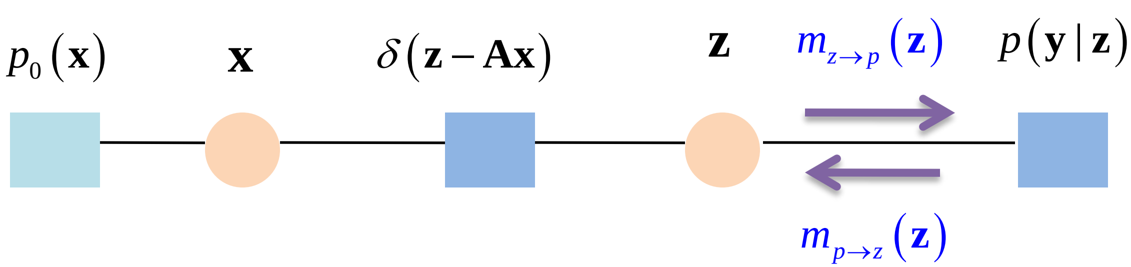
☐ Two Equivalent Factor Graphs of GLM



(a) factor graph of GLM

(b) Equivalent factor graph of GLM

☐ Decoupling GLM into SLM via EP



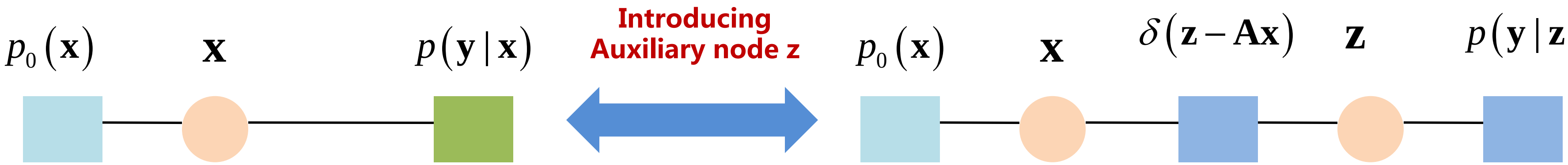
$$m_{z\to p}^{t-1}(\mathbf{z}) \propto \mathcal{N}\left(\mathbf{z}; z_A^{ext}(t-1), v_A^{ext}(t-1)I\right)$$

EP message passing
(t-th iteration)

$$m_{p\to z}^{t}(\mathbf{z}) \propto \frac{\mathrm{Proj}_{\Phi}\left(p(\mathbf{y}|\mathbf{z})\, m_{z\to p}^{t-1}(\mathbf{z})\right)}{m_{z\to p}^{t-1}(\mathbf{z})} \propto \mathcal{N}\left(\mathbf{z}; z_B^{ext}(t), v_B^{ext}(t)I\right)$$

A Unified Inference Framework for GLM

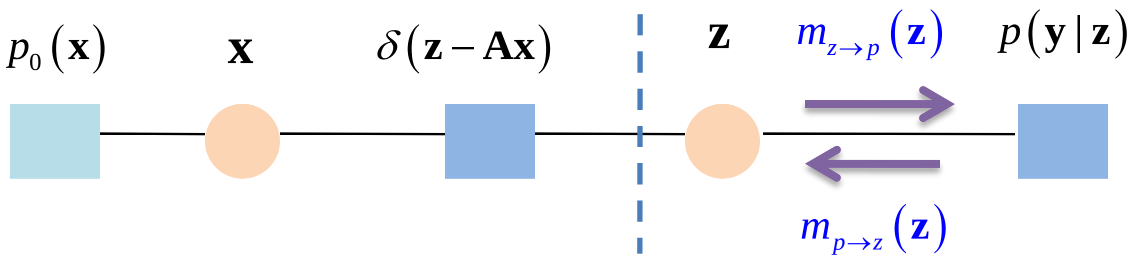
□ Two Equivalent Factor Graphs of GLM



(a) factor graph of GLM

(b) Equivalent factor graph of GLM

□ Decoupling GLM into SLM via EP



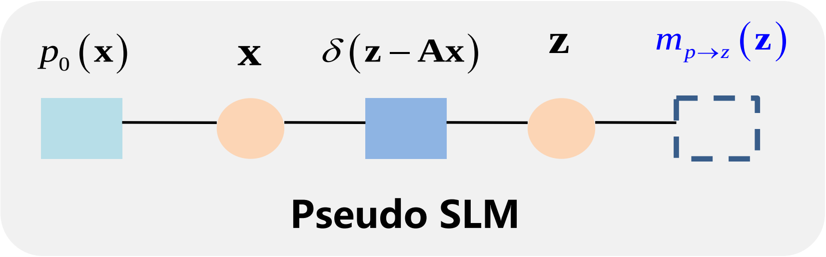
$$m_{z\to p}^{t-1}(\mathbf{z}) \propto \mathcal{N}\left(\mathbf{z}; z_A^{ext}(t-1), v_A^{ext}(t-1)I\right)$$

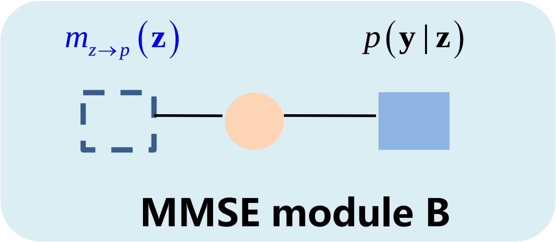
EP message passing
(t-th iteration)

$$m_{p\to z}^{t}(\mathbf{z}) \propto \frac{\mathrm{Proj}_{\Phi}\left(p(\mathbf{y}|\mathbf{z})\, m_{z\to p}^{t-1}(\mathbf{z})\right)}{m_{z\to p}^{t-1}(\mathbf{z})} \propto \mathcal{N}\left(\mathbf{z}; z_B^{ext}(t), v_B^{ext}(t)I\right)$$

A Unified Inference Framework for GLM

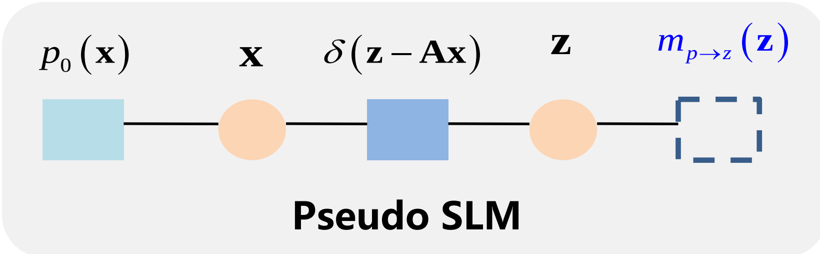
□ Decoupling GLM into SLM via EP

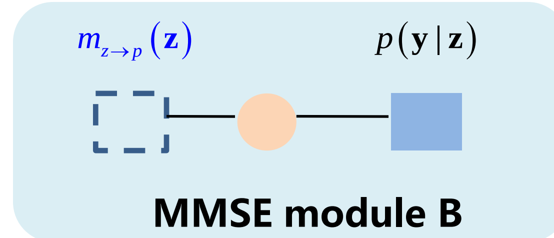




A Unified Inference Framework for GLM

◻ Decoupling GLM into SLM via EP





- **The original GLM is iteratively decoupled into a sequence of simple SLM problems**

Pseudo Linear Models

$$\tilde{\mathbf{y}} = \mathbf{A}\mathbf{x} + \tilde{\mathbf{n}}$$

$$\tilde{\mathbf{n}} \sim \mathcal{N}(0, \tilde{\sigma}^2 I)$$

$$\tilde{\mathbf{y}} = \mathbf{z}_B^{ext}(t), \tilde{\sigma}^2 = v_B^{ext}(t)$$

Module A

$\mathbf{z}_A^{ext}(t-1), v_A^{ext}(t-1)$ →

← $\mathbf{z}_B^{ext}(t), v_B^{ext}(t)$

Component-wise MMSE

$$\mathbf{z}_B^{post} = \mathrm{E}\left[z \mid \mathbf{z}_A^{ext}, v_A^{ext}\right]$$

$$v_B^{post} = \mathrm{Var}\left[z \mid \mathbf{z}_A^{ext}, v_A^{ext}\right]$$

Module B

$$\frac{1}{v_A^{ext}(t)} = \frac{1}{v_A^{post}(t)} - \frac{1}{v_B^{ext}(t)}$$

$$\frac{\mathbf{z}_A^{ext}(t)}{v_A^{ext}(t)} = \frac{\mathbf{z}_A^{post}(t)}{v_A^{post}(t)} - \frac{\mathbf{z}_B^{ext}(t)}{v_B^{ext}(t)}$$

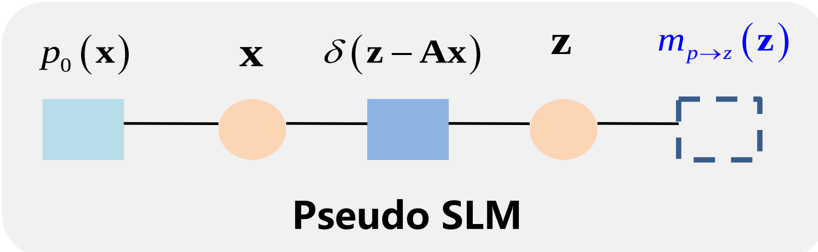
$$\frac{1}{v_B^{ext}(t)} = \frac{1}{v_B^{post}(t)} - \frac{1}{v_A^{ext}(t-1)}$$

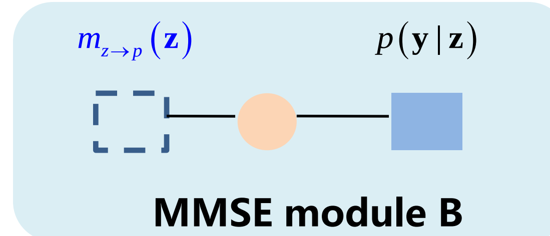
$$\frac{\mathbf{z}_B^{ext}(t)}{v_B^{ext}(t)} = \frac{\mathbf{z}_B^{post}(t)}{v_B^{post}(t)} - \frac{\mathbf{z}_A^{ext}(t-1)}{v_A^{ext}(t-1)}$$

Note: The computation of posterior mean and variance of z in module A may differ for different SLM inference methods.

A Unified Inference Framework for GLM

☐ Decoupling GLM into SLM via EP





- **The original GLM is iteratively decoupled into a sequence of simple SLM problems**

Pseudo Linear Models

$$\tilde{\mathbf{y}} = \mathbf{A}\mathbf{x} + \tilde{\mathbf{n}}$$

$$\tilde{\mathbf{n}} \sim \mathcal{N}(0, \tilde{\sigma}^2 I)$$

$$\tilde{\mathbf{y}} = \mathbf{z}_B^{ext}(t), \tilde{\sigma}^2 = v_B^{ext}(t)$$

Module A

$\mathbf{z}_A^{ext}(t-1), v_A^{ext}(t-1)$ →

← $\mathbf{z}_B^{ext}(t), v_B^{ext}(t)$

Component-wise MMSE

$$\mathbf{z}_B^{post} = \mathrm{E}\left[z \mid \mathbf{z}_A^{ext}, v_A^{ext}\right]$$

$$v_B^{post} = \mathrm{Var}\left[z \mid \mathbf{z}_A^{ext}, v_A^{ext}\right]$$

Module B

$$\frac{1}{v_A^{ext}(t)} = \frac{1}{v_A^{post}(t)} - \frac{1}{v_B^{ext}(t)}$$

$$\frac{\mathbf{z}_A^{ext}(t)}{v_A^{ext}(t)} = \frac{\mathbf{z}_A^{post}(t)}{v_A^{post}(t)} - \frac{\mathbf{z}_B^{ext}(t)}{v_B^{ext}(t)}$$

$$\frac{1}{v_B^{ext}(t)} = \frac{1}{v_B^{post}(t)} - \frac{1}{v_A^{ext}(t-1)}$$

$$\frac{\mathbf{z}_B^{ext}(t)}{v_B^{ext}(t)} = \frac{\mathbf{z}_B^{post}(t)}{v_B^{post}(t)} - \frac{\mathbf{z}_A^{ext}(t-1)}{v_A^{ext}(t-1)}$$

Universal Algorithm Design

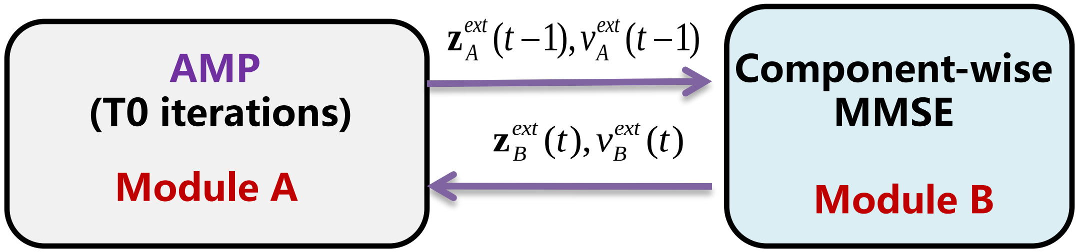
Unified Inference Framework for GLM
- Initialization $\mathbf{z}_A^{ext}(0), v_A^{ext}(0)$
- For t = 1: T, Do
  1. Perform component-wise MMSE
  2. Update $\mathbf{z}_B^{ext}(t), v_B^{ext}(t)$
  3. Perform **SLM inference** one or more iterations
  4. Compute $\mathbf{z}_A^{post}(t), v_A^{post}(t)$ and then update $\mathbf{z}_A^{ext}(t), v_A^{ext}(t)$

Note: The computation of posterior mean and variance of z in module A may differ for different SLM inference methods.

X. Meng, S. Wu and J. Zhu, "A unified Bayesian inference framework for generalized linear model," IEEE Signal Processing Letters, vol. 25, no. 3, Mar. 2018.

A Unified Inference Framework for GLM

☐ From AMP to Gr-AMP

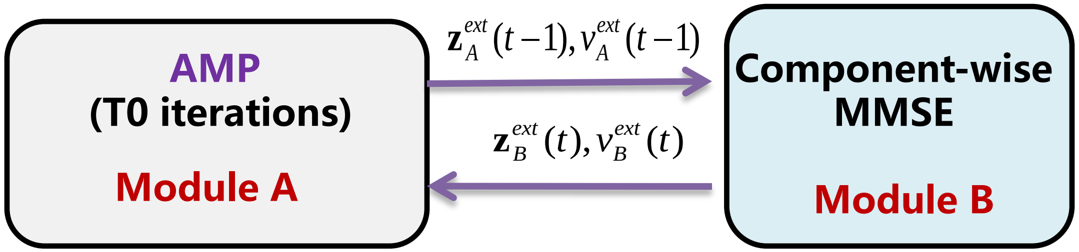


The Gr-AMP Algorithm

- Initialization $\mathbf{z}_A^{ext}(0), v_A^{ext}(0)$
- For t = 1: T, Do
  1. Perform component-wise MMSE
  2. Update $\mathbf{z}_B^{ext}(t), v_B^{ext}(t)$
  3. Perform AMP for T0 iterations
  4. Compute $\mathbf{z}_A^{post}(t), v_A^{post}(t)$ and then update $\mathbf{z}_A^{ext}(t), v_A^{ext}(t)$

A Unified Inference Framework for GLM

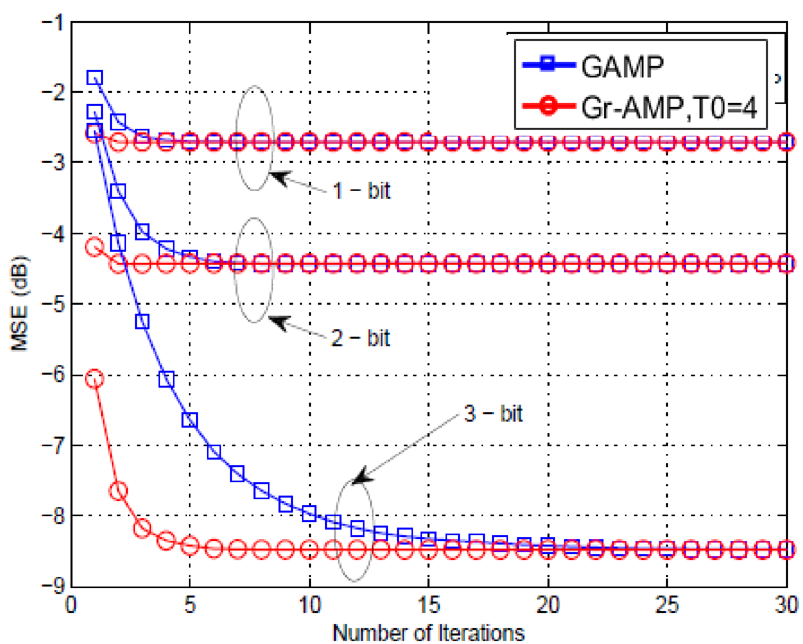
◻ From AMP to Gr-AMP



The Gr-AMP Algorithm

- Initialization $\mathbf{z}_A^{ext}(0), v_A^{ext}(0)$
- For t = 1: T, Do
  1. Perform component-wise MMSE
  2. Update $\mathbf{z}_B^{ext}(t), v_B^{ext}(t)$
  3. Perform AMP for T0 iterations
  4. Compute $\mathbf{z}_A^{post}(t), v_A^{post}(t)$ and then update $\mathbf{z}_A^{ext}(t), v_A^{ext}(t)$

- **Relation of Gr-AMP to GAMP**
  - ✓ Gr-AMP is precisely equivalent to GAMP when T0 = 1 and thus provides an insightful perspective on GAMP: In effect, GAMP performs one iteration of AMP each time after transforming the GLM problem to a pseudo SLM problem.
  - ✓ A more flexible message passing schedule: double-loop implementation

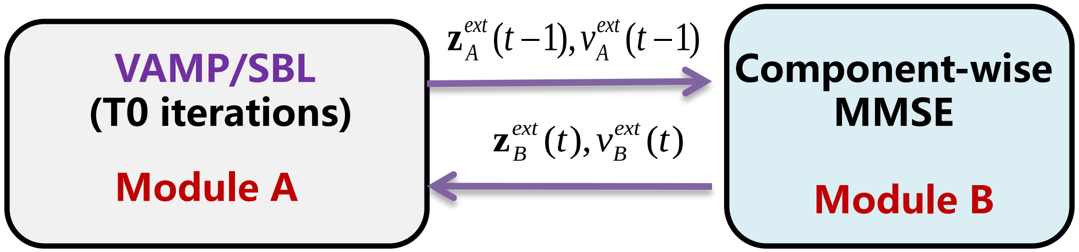


- Quantized CS for 1,2,3-bit cases: N=1024,M=512,SNR=50dB
- Gr-AMP and GAMP converge to the same performance for i.i.d. Gaussian A
- Total number iterations of AMP are about the same while **the number of MMSE operations is reduced** for Gr-AMP.

X. Meng, S. Wu and J. Zhu, "A unified Bayesian inference framework for generalized linear model," IEEE Signal Processing Letters., vol. 25, no. 3, Mar. 2018.

A Unified Inference Framework for GLM

□ From VAMP/SBL to Gr-AMP/Gr-SBL

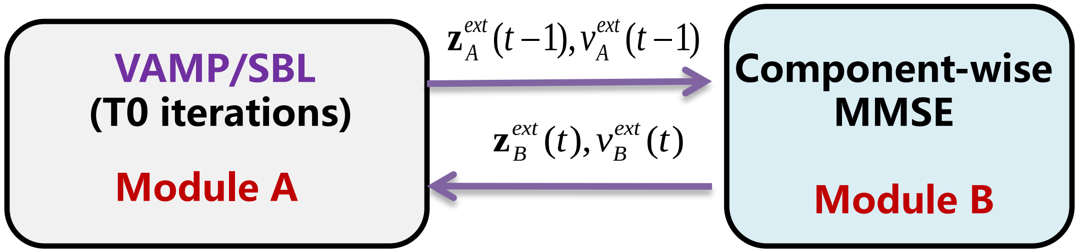


The Gr-VAMP/Gr-SBL Algorithm

- Initialization $\mathbf{z}_A^{ext}(0), v_A^{ext}(0)$
- For t = 1: T, Do
  1. Perform component-wise MMSE
  2. Update $\mathbf{z}_B^{ext}(t), v_B^{ext}(t)$
  3. Perform VAMP/SBL for T0 iterations
  4. Compute $\mathbf{z}_A^{post}(t), v_A^{post}(t)$ and then update $\mathbf{z}_A^{ext}(t), v_A^{ext}(t)$

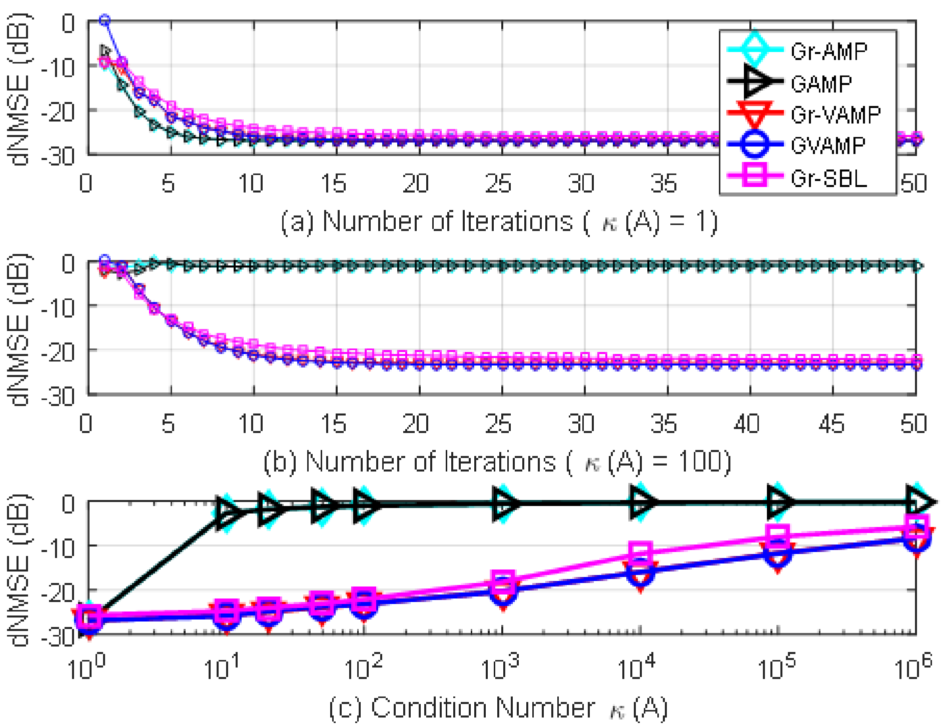
A Unified Inference Framework for GLM

☐ From VAMP/SBL to Gr-AMP/Gr-SBL



The Gr-VAMP/Gr-SBL Algorithm

- Initialization $\mathbf{z}_A^{ext}(0), v_A^{ext}(0)$
- For t = 1: T, Do
  1. Perform component-wise MMSE
  2. Update $\mathbf{z}_B^{ext}(t), v_B^{ext}(t)$
  3. Perform VAMP/SBL for T0 iterations
  4. Compute $\mathbf{z}_A^{post}(t), v_A^{post}(t)$ and then update $\mathbf{z}_A^{ext}(t), v_A^{ext}(t)$



Performance of de-biased NMSE for **1-bit CS**

- ✓ N =512,M=2048,SNR=50dB, sparse ratio 0.1
- ✓ T0 = 1 for both Gr-VAMP and Gr-SBL
- ✓ When conditional number is 1, all kinds of algorithms performs nearly the same.
- ✓ As the condition number increases, the recovery performances degrade smoothly for Gr-VAMP/GVAMP/Gr-SBL while both Gr-AMP and GAMP diverge for even mild condition number, which show the robustness of Gr-VAMP/Gr-SBL/GVAMP for general matrices.

X. Meng, S. Wu and J. Zhu, “A unified Bayesian inference framework for generalized linear model,” IEEE Signal Processing Letters., vol. 25, no. 3, Mar. 2018.

Ⅲ. Extension of GLM to Bilinear Models

Extension of GLM to Bilinear Models

☐ Bilinear GLM Problems

Matrix X		Linear mixing		Probabilistic mapping		Observations
$\mathbf{X} \sim p_0(\mathbf{X};\theta_X)$	$\xrightarrow{\mathbf{x} \in \mathbb{R}^{N\times L}}$	$\mathbf{A}$ (unknown)	$\xrightarrow[\mathbf{Z}=\mathbf{AX}]{\mathbf{Z} \in \mathbb{R}^{M\times L}}$	$p(\mathbf{Y}\mid\mathbf{Z};\theta_Y)$	$\longrightarrow$	$Y \in \mathbb{R}^{M\times L}$

- **Assumptions**

  A(·) is a known affine linear function of unknown vector $\theta_A = \mathbf{b}$,i.e.,

$$\mathbf{A}(\mathbf{b}) = \mathbf{A}_0 + \sum_{q=1}^{Q} b_q \mathbf{A}_q \qquad \boldsymbol{\Theta} \triangleq \{\theta_X, \theta_A, \theta_Y\}$$

- **Goal**

  To jointly infer X and A, given Y with unknown parameters $\boldsymbol{\Theta}$

$$\hat{\boldsymbol{\Theta}}_{\mathrm{ML}} = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\; p_{\mathbf{Y}}(\mathbf{Y};\boldsymbol{\Theta}),$$

$$\hat{\mathbf{X}}_{\mathrm{MMSE}} = \mathrm{E}[\mathbf{X}|\mathbf{Y};\hat{\boldsymbol{\Theta}}_{\mathrm{ML}}],$$

Intractable!

Extension of GLM to Bilinear Models

◻ Bilinear GLM Problems

Matrix X		Linear mixing		Probabilistic mapping		Observations
$\mathbf{X} \sim p_0(\mathbf{X};\theta_X)$	$\xrightarrow{\mathbf{x} \in \mathbb{R}^{N\times L}}$	$\mathbf{A}$ (unknown)	$\xrightarrow[\mathbf{Z}=\mathbf{AX}]{\mathbf{Z} \in \mathbb{R}^{M\times L}}$	$p(\mathbf{Y}\mid\mathbf{Z};\theta_Y)$	$\longrightarrow$	$Y \in \mathbb{R}^{M\times L}$

- **Assumptions**

  A(·) is a known affine linear function of unknown vector $\theta_A = \mathbf{b}$,i.e.,

$$\mathbf{A}(\mathbf{b}) = \mathbf{A}_0 + \sum_{q=1}^{Q} b_q \mathbf{A}_q \qquad \boldsymbol{\Theta} \triangleq \{\theta_X, \theta_A, \theta_Y\}$$

- **Goal**

  To jointly infer X and A, given Y with unknown parameters Θ

$$\hat{\boldsymbol{\Theta}}_{\mathrm{ML}} = \underset{\boldsymbol{\Theta}}{\operatorname{argmax}}\; p_{\mathbf{Y}}(\mathbf{Y};\boldsymbol{\Theta}),$$

$$\hat{\mathbf{X}}_{\mathrm{MMSE}} = \mathrm{E}[\mathbf{X}|\mathbf{Y};\hat{\boldsymbol{\Theta}}_{\mathrm{ML}}],$$

Intractable!

Basic Idea:

✓ **First considering the simple case when the likelihood $p(\mathbf{Y}\mid\mathbf{Z};\theta_Y)$ is Gaussian, i.e.,**

$$\mathbf{Y} = \mathbf{A}(\theta_A)\mathbf{X} + \mathbf{N}$$

✓ **Extending to generalized nonlinear observations using the proposed unified framework for GLM**

Extension of GLM to Bilinear Models

Standard Bilinear Problems

$$\mathbf{Y} = \mathbf{A}(\theta_A)\mathbf{X} + \mathbf{N} \qquad \Theta \triangleq \{\boldsymbol{\theta}_X, \boldsymbol{\theta}_A, \boldsymbol{\theta}_Y\}$$

$$\mathbf{X} \sim p(\mathbf{X}; \boldsymbol{\theta}_X) = \prod_{i,j} p(x_{ij}; \boldsymbol{\theta}_X) = \prod_{l=1}^{L} p(\mathbf{x}_l; \boldsymbol{\theta}_X) \qquad \mathbf{Y} \sim p\left(\mathbf{Y} \mid \mathbf{A}(\theta_A)\mathbf{X}; \theta_Y\right) = \prod_{l=1}^{L} \mathcal{N}\left(\mathbf{y}_l; \mathbf{A}(\theta_A)\mathbf{x}_l, \gamma_w^{-1}\mathbf{I}\right)$$

EM learning framework

E-step:

$$Q\left(\Theta, \Theta^t\right) = \mathrm{E}_{p\left(\mathbf{X}|\mathbf{Y};\Theta^t\right)}\left[\log p\left(\mathbf{X}, \mathbf{Y}; \Theta^t\right)\right]$$

Iterating

M-step:

$$\Theta^{t+1} = \arg\max_{\Theta} Q\left(\Theta, \Theta^t\right)$$

Extension of GLM to Bilinear Models

Standard Bilinear Problems

$$\mathbf{Y} = \mathbf{A}(\theta_A)\mathbf{X} + \mathbf{N} \qquad \Theta \triangleq \{\boldsymbol{\theta}_X, \boldsymbol{\theta}_A, \boldsymbol{\theta}_Y\}$$

$$\mathbf{X} \sim p(\mathbf{X}; \boldsymbol{\theta}_X) = \prod_{i,j} p(x_{ij}; \boldsymbol{\theta}_X) = \prod_{l=1}^{L} p(\mathbf{x}_l; \boldsymbol{\theta}_X) \qquad \mathbf{Y} \sim p\left(\mathbf{Y} \mid \mathbf{A}(\theta_A)\mathbf{X}; \theta_Y\right) = \prod_{l=1}^{L} \mathcal{N}\left(\mathbf{y}_l; \mathbf{A}(\theta_A)\mathbf{x}_l, \gamma_w^{-1}\mathbf{I}\right)$$

EM learning framework

E-step:

$$Q(\Theta, \Theta^t) = \mathrm{E}_{p(\mathbf{X}|\mathbf{Y};\Theta^t)}\left[\log p\left(\mathbf{X}, \mathbf{Y}; \Theta^t\right)\right]$$

Iterating

M-step:

$$\Theta^{t+1} = \arg\max_{\Theta} Q(\Theta, \Theta^t)$$

E-Step Too Complicated!

$$p\left(\mathbf{X}|\mathbf{Y}; \Theta^t\right)$$

Extension of GLM to Bilinear Models

☐ Standard Bilinear Problems

$$\mathbf{Y} = \mathbf{A}(\theta_A)\mathbf{X} + \mathbf{N} \qquad \Theta \triangleq \{\theta_X, \theta_A, \theta_Y\}$$

$$\mathbf{X} \sim p(\mathbf{X}; \boldsymbol{\theta}_X) = \prod_{i,j} p(x_{ij}; \boldsymbol{\theta}_X) = \prod_{l=1}^{L} p(\mathbf{x}_l; \boldsymbol{\theta}_X) \qquad \mathbf{Y} \sim p(\mathbf{Y} \mid \mathbf{A}(\theta_A)\mathbf{X}; \theta_Y) = \prod_{l=1}^{L} \mathcal{N}(\mathbf{y}_l; \mathbf{A}(\theta_A)\mathbf{x}_l, \gamma_w^{-1}\mathbf{I})$$

EM learning framework

E-step:

$$Q(\Theta, \Theta^t) = \mathrm{E}_{p(\mathbf{X}|\mathbf{Y};\Theta^t)}\left[\log p(\mathbf{X}, \mathbf{Y}; \Theta^t)\right]$$

Iterating

M-step:

$$\Theta^{t+1} = \arg\max_{\Theta} Q(\Theta, \Theta^t)$$

E-Step Too Complicated!

$$p(\mathbf{X}|\mathbf{Y};\Theta^t)$$

- **Solution**
  - ✓ **The posterior distribution $p(\mathbf{X}|\mathbf{Y};\Theta^t)$ can be approximated by message passing algorithms, e.g., AMP, and VAMP. In each iteration of EM**

$$p(\mathbf{X}|\mathbf{Y};\Theta^t) \xrightarrow{\text{Replaced by}} q(\mathbf{X}|\mathbf{Y};\Theta^t)$$

message passing result

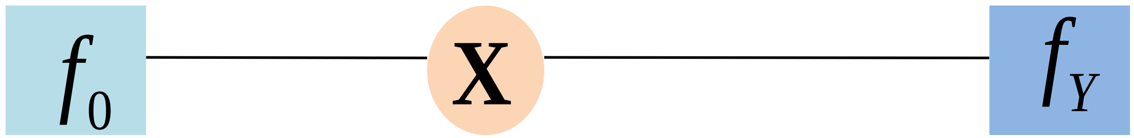
Extension of GLM to Bilinear Models

□ Bilinear Adaptive VAMP

Gaussian likelihood

$p_0\left(\mathbf{X};\theta_X\right)$

$p\left(\mathbf{Y}\,|\,\mathbf{A}\left(\theta_A\right)\mathbf{X};\theta_Y\right)$

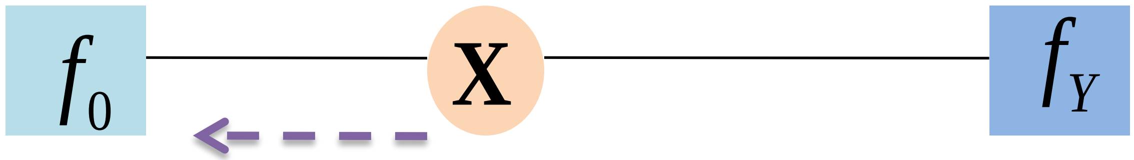


Extension of GLM to Bilinear Models

☐ Bilinear Adaptive VAMP

Gaussian likelihood

$p_0\left(\mathbf{X};\theta_X\right)$ $p\left(\mathbf{Y}\,|\,\mathbf{A}\left(\theta_A\right)\mathbf{X};\theta_Y\right)$



$m_{x\to f_Y}^{t-1}\left(\mathbf{X}\right)$

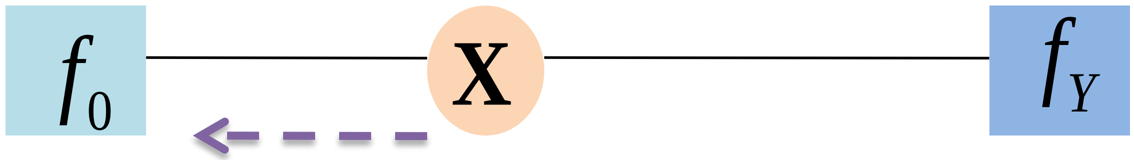
$m_{x\to f_Y}^{t-1}\left(\mathbf{X}\right)$ **message in the last (t-1) iteration**

Extension of GLM to Bilinear Models

◻ Bilinear Adaptive VAMP

Gaussian likelihood

$p_0\left(\mathbf{X};\theta_X\right)$ $p\left(\mathbf{Y}\,|\,\mathbf{A}\left(\theta_A\right)\mathbf{X};\theta_Y\right)$



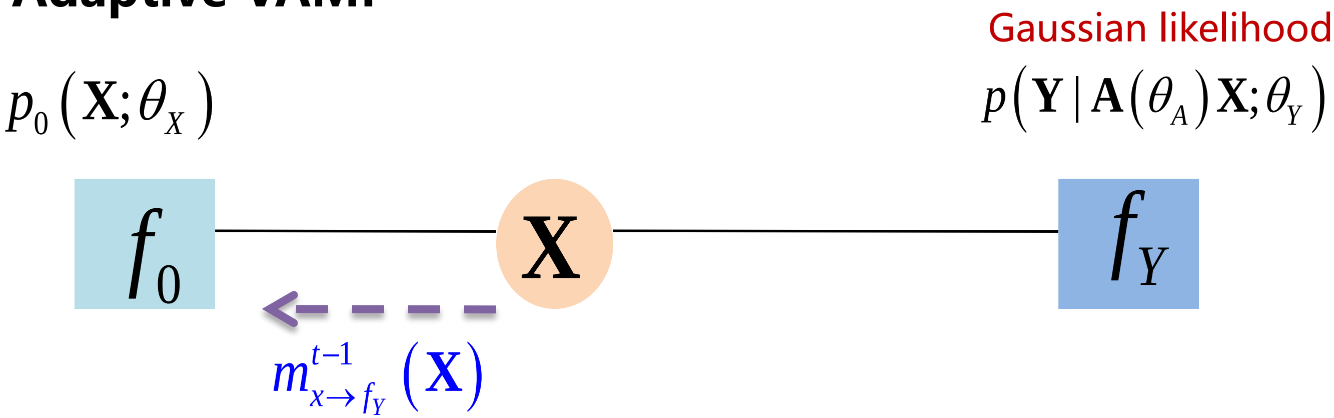
$m_{x\to f_Y}^{t-1}\left(\mathbf{X}\right)$

$m_{x\to f_Y}^{t-1}\left(\mathbf{X}\right)$ message in the last (t-1) iteration

$q_1\left(X\,|Y;\Theta^t\right)$ Update the posterior distribution

Extension of GLM to Bilinear Models

□ Bilinear Adaptive VAMP



$m^{t-1}_{x \to f_Y}(\mathbf{X})$ **message in the last (t-1) iteration**

$q_1\left(X \mid Y; \Theta^t\right)$ **Update the posterior distribution**

EM learning of θ_X

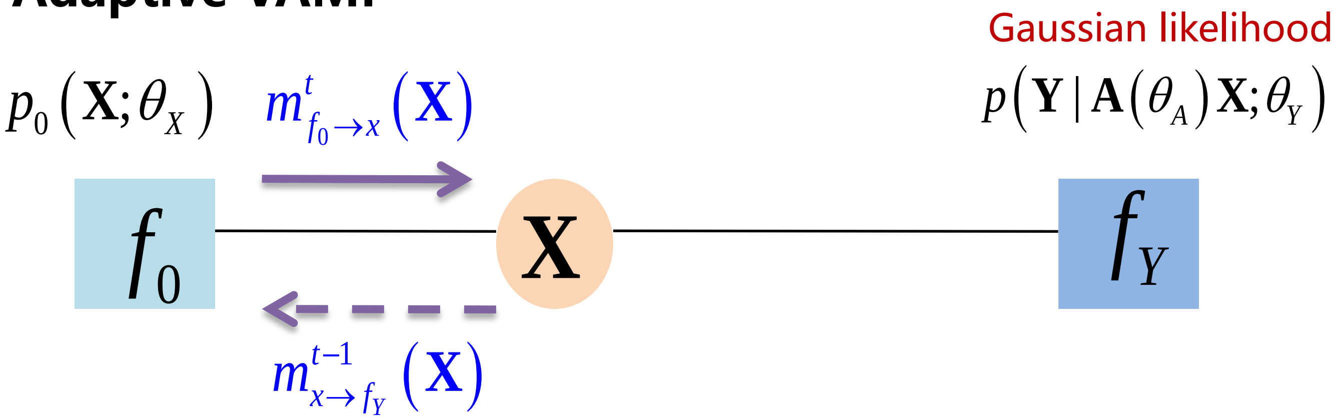
M-step: $\theta_X^{new} = \arg\max_{\theta_X} \mathrm{E}_{q_1\left(X \mid Y; \Theta^t\right)}\left[\log p_0\left(\mathbf{X}; \theta_X\right)\right]$

Iterating

E-step: update $q_1\left(X \mid Y; \Theta^t\right)$

Extension of GLM to Bilinear Models

◻ Bilinear Adaptive VAMP



$m^{t-1}_{x\to f_Y}\left(\mathbf{X}\right)$ **message in the last (t-1) iteration**

$q_1\left(X\,|\,Y;\Theta^t\right)$ **Update the posterior distribution**

EM learning of θ_X

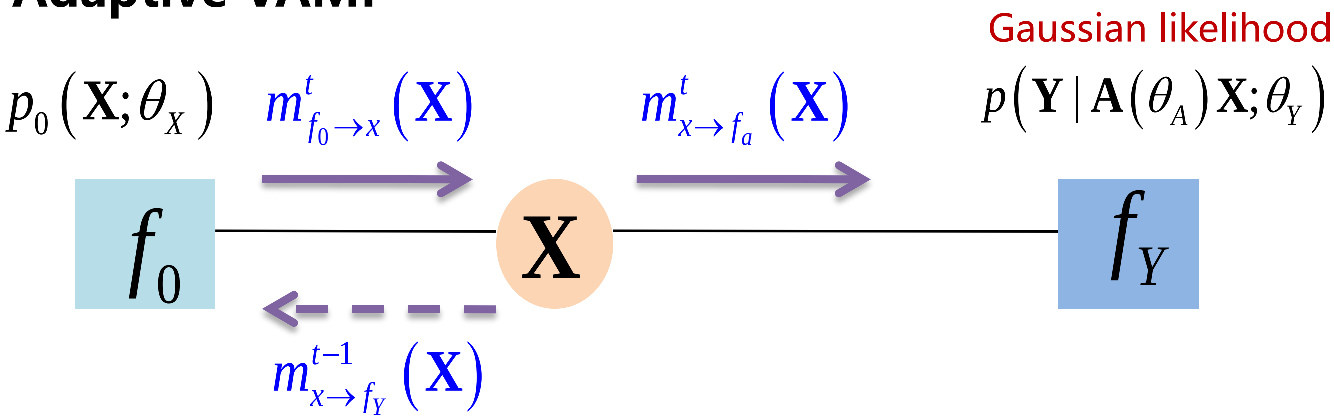
M-step: $\theta_X^{new} = \arg\max_{\theta_X} \mathrm{E}_{q_1\left(X|Y;\Theta^t\right)}\left[\log p_0\left(\mathbf{X};\theta_X\right)\right]$

Iterating

E-step: update $q_1\left(X\,|\,Y;\Theta^t\right)$

Extension of GLM to Bilinear Models

☐ Bilinear Adaptive VAMP



$m^{t-1}_{x\to f_Y}(\mathbf{X})$ **message in the last (t-1) iteration**

$m^{t}_{x\to f_a}(\mathbf{X})$ **update message from variable to f_Y**

$q_1(X\,|\,Y;\Theta^t)$ **Update the posterior distribution**

EM learning of θ_X

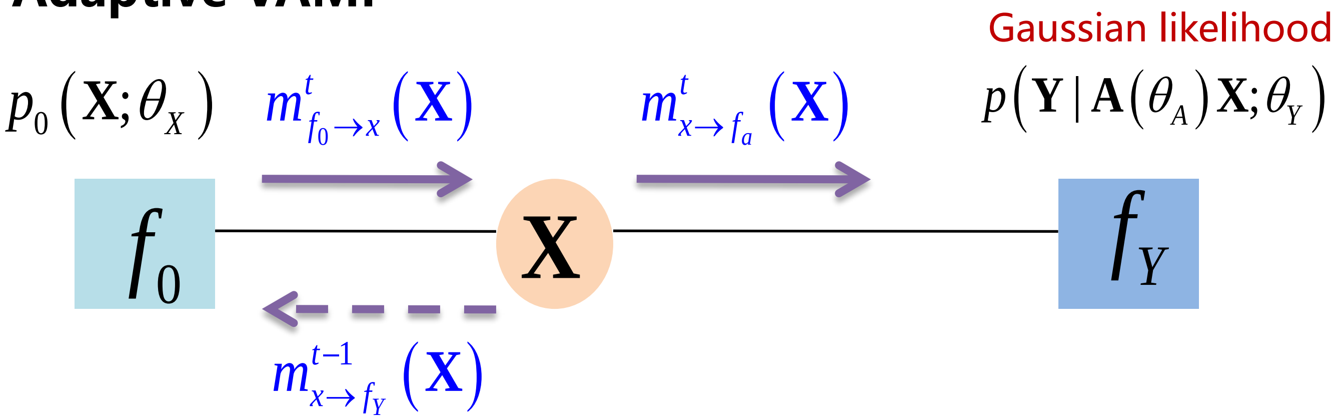
M-step: $\theta_X^{new} = \arg\max_{\theta_X} \mathrm{E}_{q_1(X|Y;\Theta^t)}\left[\log p_0(\mathbf{X};\theta_X)\right]$

Iterating

E-step: update $q_1(X\,|\,Y;\Theta^t)$

Extension of GLM to Bilinear Models

☐ Bilinear Adaptive VAMP



$m^{t-1}_{x\to f_Y}\left(\mathbf{X}\right)$ **message in the last (t-1) iteration**

$q_1\left(X\,|Y;\Theta^t\right)$ **Update the posterior distribution**

EM learning of θ_X

M-step: $\theta_X^{new} = \arg\max_{\theta_X} \mathrm{E}_{q_1\left(X|Y;\Theta^t\right)}\left[\log p_0\left(\mathbf{X};\theta_X\right)\right]$

Iterating

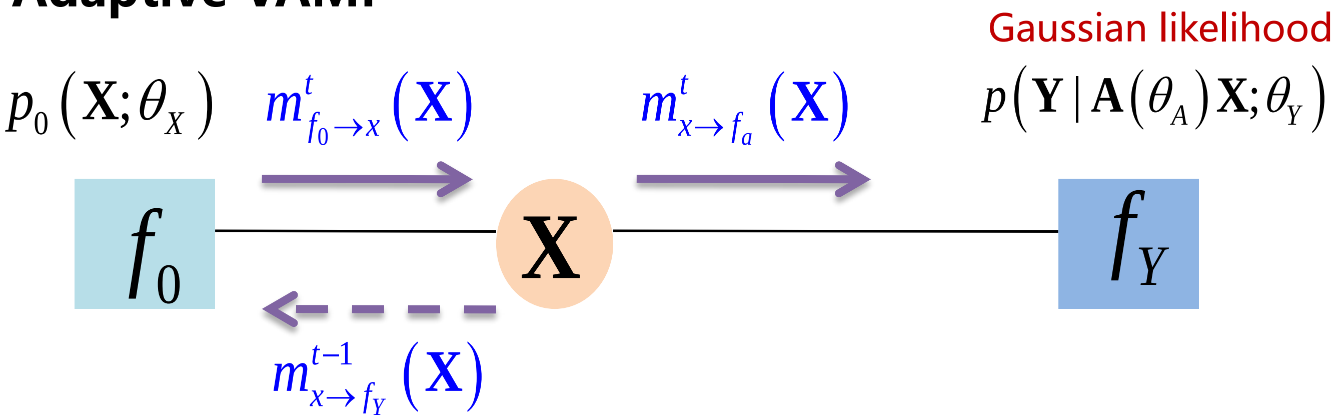
E-step: update $q_1\left(X\,|Y;\Theta^t\right)$

$m^t_{x\to f_a}\left(\mathbf{X}\right)$ **update message from variable to f_Y**

$q_2\left(X\,|Y;\Theta^t\right)$ **Update the posterior distribution**

Extension of GLM to Bilinear Models

☐ Bilinear Adaptive VAMP



$m^{t-1}_{x\to f_Y}(\mathbf{X})$ **message in the last (t-1) iteration**

$q_1(X\,|Y;\Theta^t)$ **Update the posterior distribution**

EM learning of θ_X

M-step: $\theta_X^{new}=\arg\max_{\theta_X}\mathrm{E}_{q_1(X|Y;\Theta^t)}\left[\log p_0(\mathbf{X};\theta_X)\right]$

Iterating

E-step: update $q_1(X\,|Y;\Theta^t)$

$m^t_{x\to f_a}(\mathbf{X})$ **update message from variable to f_Y**

$q_2(X\,|Y;\Theta^t)$ **Update the posterior distribution**

EM learning of θ_A,θ_Y

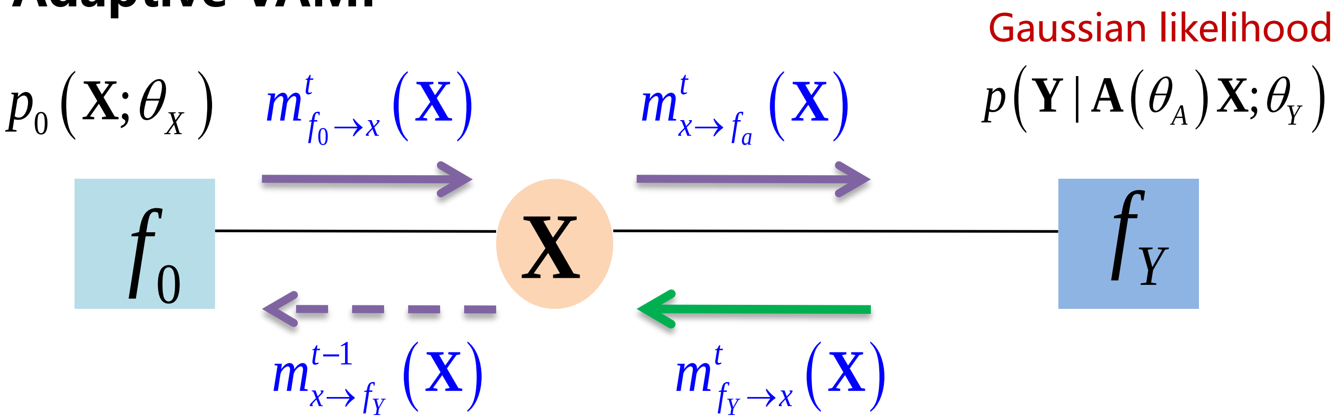
M-step:
$\theta_A^{new}=\arg\max_{\theta_A}\mathrm{E}_{q_2(X|Y;\Theta^t)}\left[p(\mathbf{Y}\,|\,\mathbf{A}(\theta_A)\mathbf{X};\theta_Y)\right]$
$\theta_Y^{new}=\arg\max_{\theta_Y}\mathrm{E}_{q_2(X|Y;\Theta^t)}\left[p(\mathbf{Y}\,|\,\mathbf{A}(\theta_A)\mathbf{X};\theta_Y)\right]$

Iterating

E-step: update $q_2(X\,|Y;\Theta^t)$

Extension of GLM to Bilinear Models

□ Bilinear Adaptive VAMP



$m^{t-1}_{x\to f_Y}\left(\mathbf{X}\right)$ **message in the last (t-1) iteration**

$q_1\left(X|Y;\Theta^t\right)$ **Update the posterior distribution**

EM learning of θ_X

M-step: $\theta_X^{new} = \arg\max_{\theta_X} \mathrm{E}_{q_1\left(X|Y;\Theta^t\right)}\left[\log p_0\left(\mathbf{X};\theta_X\right)\right]$

Iterating

E-step: update $q_1\left(X|Y;\Theta^t\right)$

$m^t_{x\to f_a}\left(\mathbf{X}\right)$ **update message from variable to f_Y**

$q_2\left(X|Y;\Theta^t\right)$ **Update the posterior distribution**

EM learning of θ_A, θ_Y

M-step:

$$\theta_A^{new} = \arg\max_{\theta_A} \mathrm{E}_{q_2\left(X|Y;\Theta^t\right)}\left[p\left(\mathbf{Y}\mid\mathbf{A}\left(\theta_A\right)\mathbf{X};\theta_Y\right)\right]$$

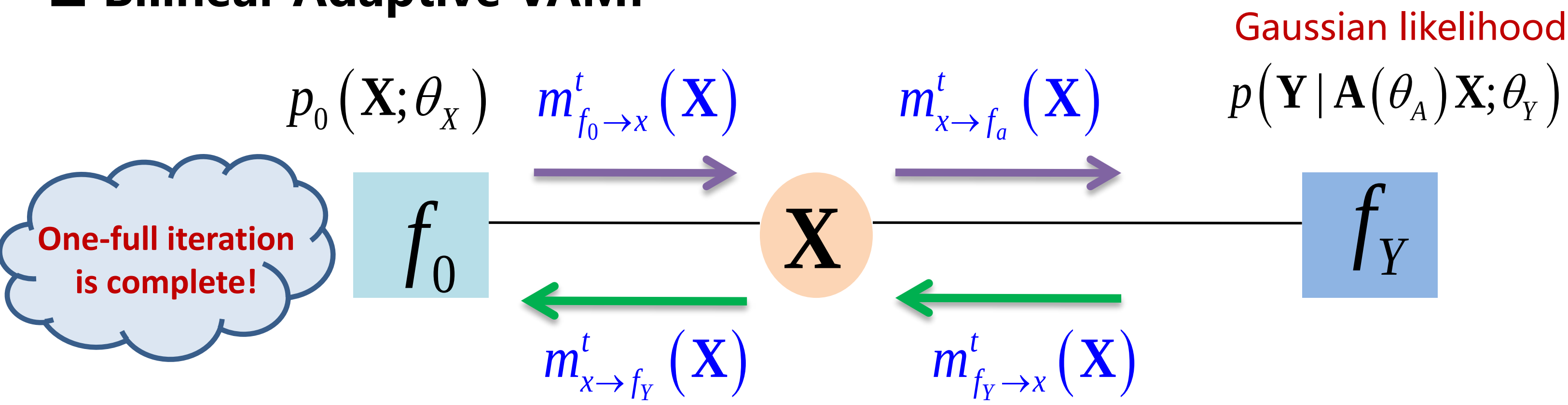
$$\theta_Y^{new} = \arg\max_{\theta_Y} \mathrm{E}_{q_2\left(X|Y;\Theta^t\right)}\left[p\left(\mathbf{Y}\mid\mathbf{A}\left(\theta_A\right)\mathbf{X};\theta_Y\right)\right]$$

Iterating

E-step: update $q_2\left(X|Y;\Theta^t\right)$

Extension of GLM to Bilinear Models

Bilinear Adaptive VAMP



$m^{t-1}_{x\to f_Y}(\mathbf{X})$ **message in the last (t-1) iteration**

$q_1(X|Y;\Theta^t)$ **Update the posterior distribution**

EM learning of θ_X

M-step: $\theta_X^{new} = \arg\max_{\theta_X} \mathrm{E}_{q_1(X|Y;\Theta^t)}\left[\log p_0(\mathbf{X};\theta_X)\right]$

Iterating

E-step: update $q_1(X|Y;\Theta^t)$

$m^t_{x\to f_a}(\mathbf{X})$ **update message from variable to** f_Y

$q_2(X|Y;\Theta^t)$ **Update the posterior distribution**

EM learning of θ_A,θ_Y

M-step:

$\theta_A^{new} = \arg\max_{\theta_A} \mathrm{E}_{q_2(X|Y;\Theta^t)}\left[p(\mathbf{Y}\,|\,\mathbf{A}(\theta_A)\mathbf{X};\theta_Y)\right]$

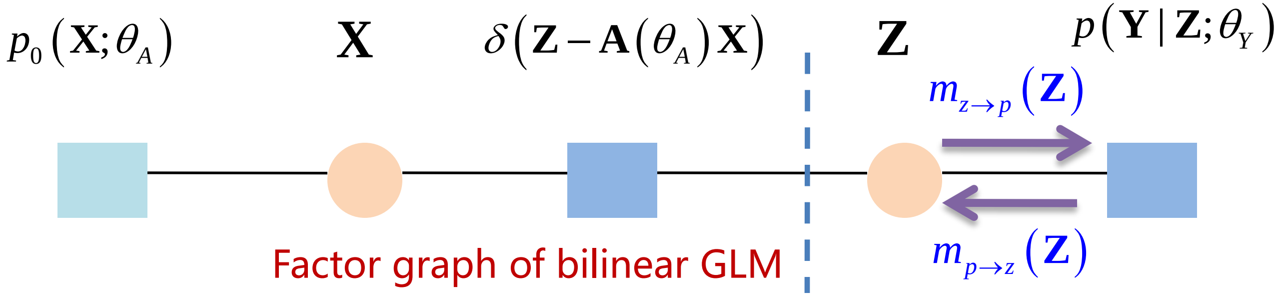
$\theta_Y^{new} = \arg\max_{\theta_Y} \mathrm{E}_{q_2(X|Y;\Theta^t)}\left[p(\mathbf{Y}\,|\,\mathbf{A}(\theta_A)\mathbf{X};\theta_Y)\right]$

Iterating

E-step: update $q_2(X|Y;\Theta^t)$

Extension of GLM to Bilinear Models

- **From Linear to Nonlinear observations**

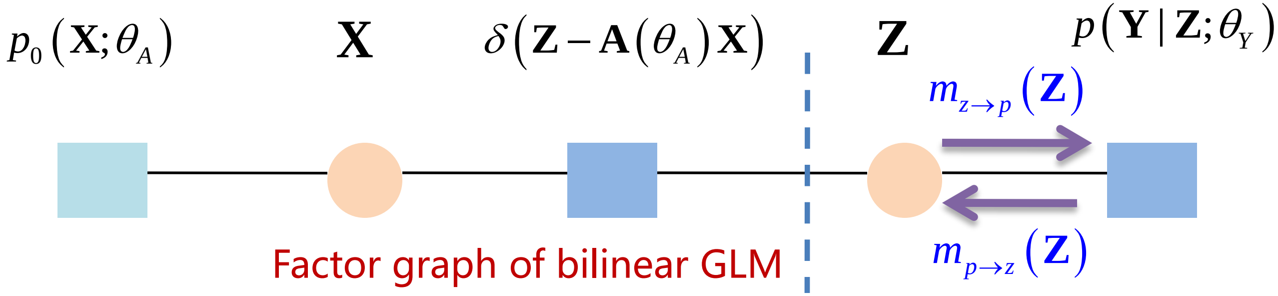


Factor graph of bilinear GLM

Non-Gaussian likelihood!

Extension of GLM to Bilinear Models

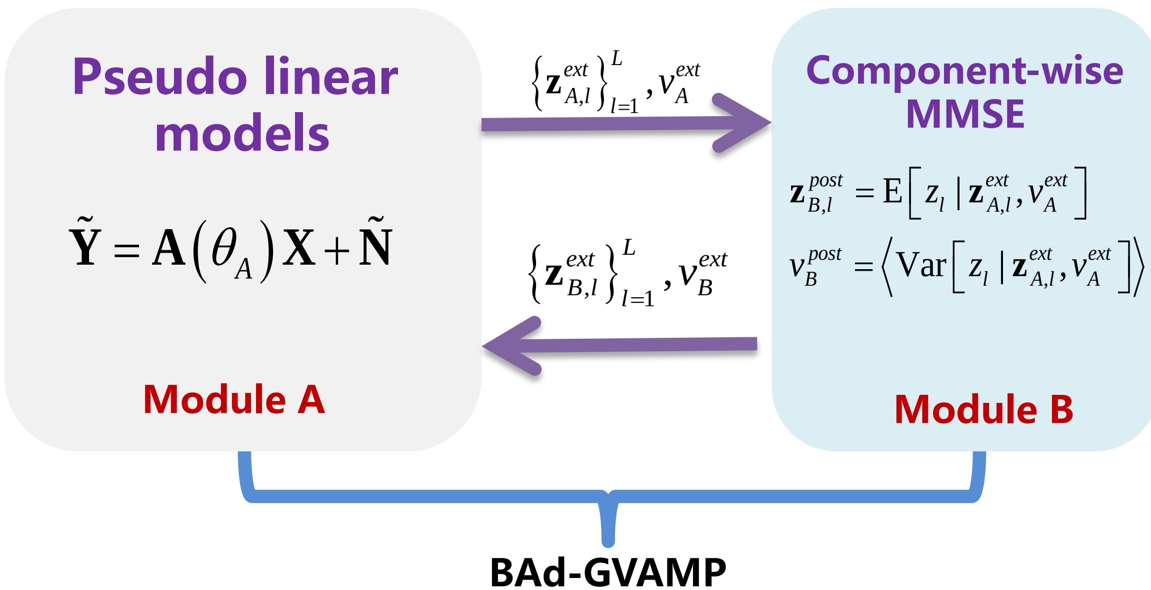
☐ From Linear to Nonlinear observations



Factor graph of bilinear GLM

Non-Gaussian likelihood!

- **Similar to GLM, using EP, it can be iteratively decoupled into two modules**



Bilinear Adaptive Generalized Vector Approximate Message Passing (BAd-GVAMP) Algorithm

X. Meng, and J. Zhu, “Bilinear Adaptive Generalized Vector Approximate Message Passing,” IEEE Access, 2019

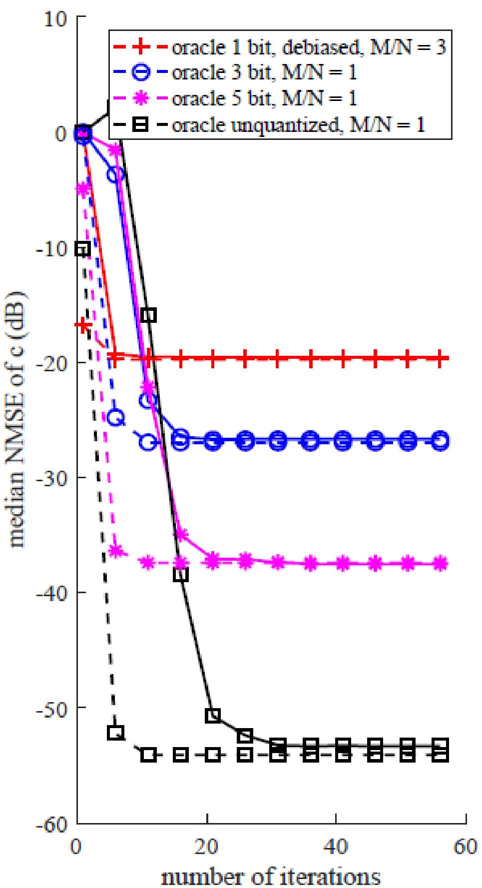
Extension of GLM to Bilinear Models

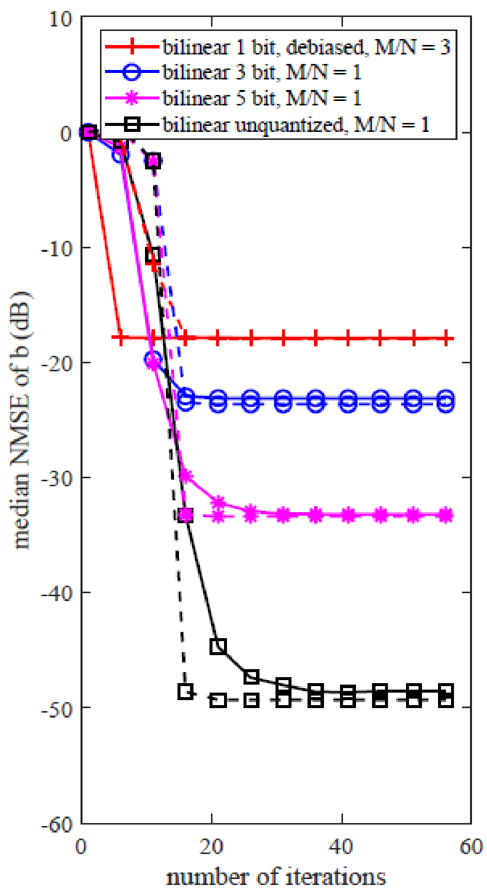
☐ Experimental results of BAd-GVAMP

- **Experiment 1**: **Quantized Compressed Sensing with matrix uncertainty**

$$\mathbf{y} = Q(\mathbf{A}(\mathbf{b})\mathbf{c} + \mathbf{w}) \qquad \mathbf{A}(\mathbf{b}) = \mathbf{A}_0 + \sum_{i=1}^{G} b_i \mathbf{A}_i$$

$\{\mathbf{A}_i\}_{i=0}^{G} \in \mathbb{R}^{M\times N}$ are known, $\mathbf{b}$ are the unknown uncertainty parameters.





$$\mathrm{SNR} \triangleq 10\log\frac{\mathrm{E}\|\mathbf{A}\mathbf{c}\|^2}{\mathrm{E}\|\mathbf{w}\|^2} = 40 \text{ dB}$$

✓ c is generated with uniformly random support with K nonzero elements from i.i.d N(0,1), we set N = 256, G = 10, K = 10

✓ For M/N = 1, the NMSE in dB is shown in left figure:

-- **Converges fast** (20-30 iterations)

-- **Same as the oracle performance**.

X. Meng, and J. Zhu, "Bilinear Adaptive Generalized Vector Approximate Message Passing," IEEE Access, 2019

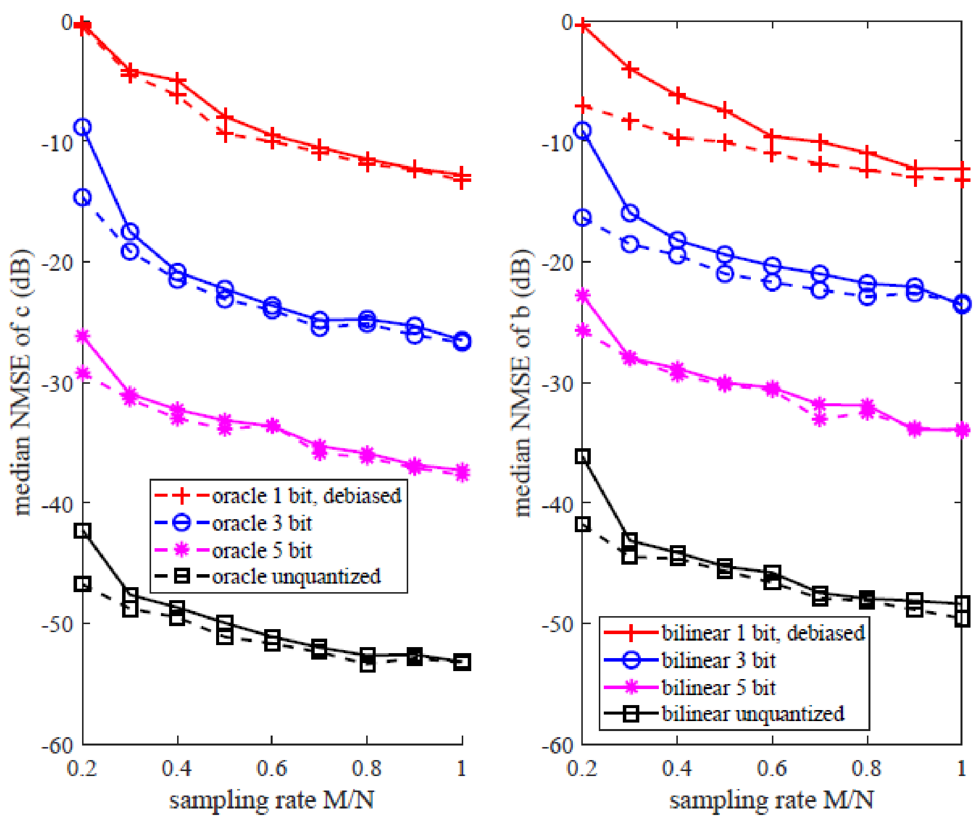
Extension of GLM to Bilinear Models

Experimental results of BAd-GVAMP

- **Experiment 1**: Quantized Compressed Sensing with matrix uncertainty

$$\mathbf{y} = Q(\mathbf{A}(\mathbf{b})\mathbf{c}+\mathbf{w}) \qquad \mathbf{A}(\mathbf{b}) = \mathbf{A}_0 + \sum_{i=1}^{G} b_i \mathbf{A}_i$$

$\{\mathbf{A}_i\}_{i=0}^{G} \in \mathbb{R}^{M\times N}$ are known, $\mathbf{b}$ are the unknown uncertainty parameters.



$$\mathrm{SNR} \triangleq 10\log\frac{\mathrm{E}\|\mathbf{A}\mathbf{c}\|^2}{\mathrm{E}\|\mathbf{w}\|^2} = 40\ \mathrm{dB}$$

✓ c is generated with uniformly random support with K nonzero elements from i.i.d N(0,1), we set N = 256, G = 10, K = 10

✓ Then, the performance vs. ratio M/N is evaluated:

-- As the increase of M/N, the recovery performance improves

-- **Approaching the oracle performance in a wide range of M/N values**

X. Meng, and J. Zhu, “Bilinear Adaptive Generalized Vector Approximate Message Passing,” IEEE Access, 2019

Extension of GLM to Bilinear Models

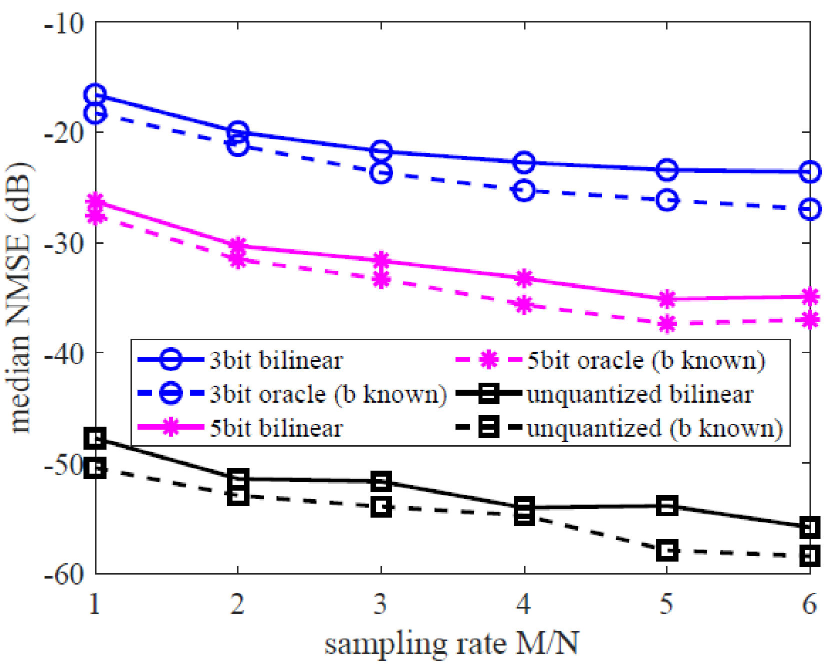
Experimental results of BAd-GVAMP

- **Experiment 2: Self-Calibration from quantized measurements**

$$\mathbf{y} = Q(\mathrm{diag}(\mathbf{Hb})\boldsymbol{\Psi}\mathbf{c} + \mathbf{w}) = Q\left(\left[\sum_{i=1}^{G} b_i \mathrm{diag}(\mathbf{h}_i)\boldsymbol{\Psi}\right]\mathbf{c} + \mathbf{w}\right)$$

$$\text{with known } \mathbf{H} \in \mathbb{R}^{M\times G} \text{ and } \boldsymbol{\Psi} \in \mathbb{R}^{M\times N}$$

Goal: to recover the K-sparse signal vector c and the calibration parameters b



✓ K = 10, G = 8, M = 128 and SNR = 40 dB.
✓ H is constructed using Q randomly selected columns of the Hadamard matrix, the elements of **b** and $\boldsymbol{\Psi}$ are i.i.d. drawn from N(0; 1), and c is generated with *K* nonzero elements i.i.d. drawn from N(0; 1).

$$\mathrm{NMSE} = 10\log\frac{\|\hat{\mathbf{b}}\hat{\mathbf{c}}^{\mathrm{T}} - \mathbf{b}\mathbf{c}^{\mathrm{T}}\|_{\mathrm{F}}^2}{\|\mathbf{b}\mathbf{c}^{\mathrm{T}}\|_{\mathrm{F}}^2}$$

✓ As the sampling rate increases, the median NMSE decreases.
✓ **Near oracle performance.**

X. Meng, and J. Zhu, "Bilinear Adaptive Generalized Vector Approximate Message Passing," IEEE Access, 2019

Extension of GLM to Bilinear Models

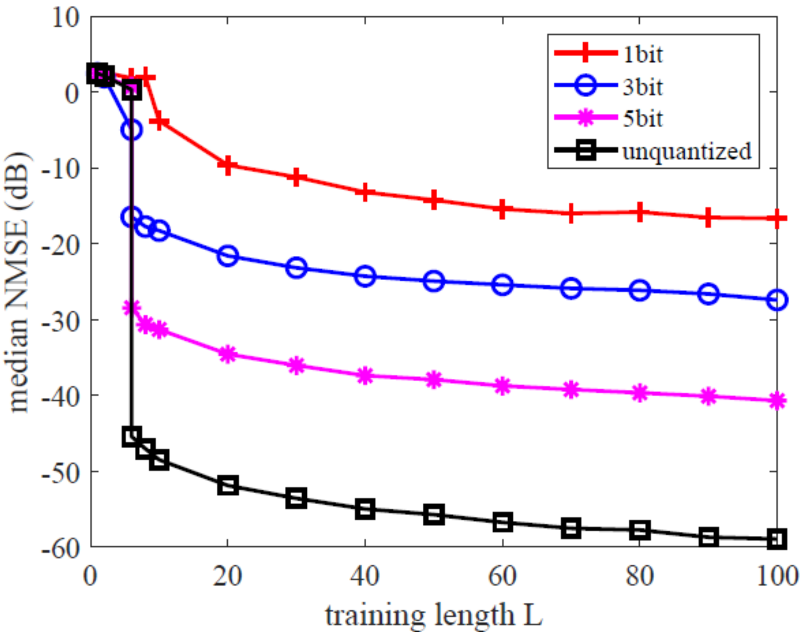
☐ Experimental results of BAd-GVAMP

- **Experiment 3: Structured dictionary learning from quantized measurements**

Goal: Finding a dictionary matrix A and sparse matrix X such that

$$\mathbf{Y} = Q\left(\mathbf{AX} + \mathbf{N}\right) \quad \mathbf{A} \in \mathbb{R}^{M\times N}, \mathbf{X} \in \mathbb{R}^{N\times L}$$

$Q(\bullet)$ **is a quantization function** $\quad \mathbf{A} = \sum_{i=1}^{G} b_i \mathbf{A}_i$



✓ G=M=N=64 and SNR = 40 dB.

$$\mathrm{NMSE}(\hat{\mathbf{A}}) \triangleq \min_{\lambda\in\mathbb{R}} \frac{\|\mathbf{A} - \lambda\hat{\mathbf{A}}\|_F^2}{\|\mathbf{A}\|_F^2}$$

✓ As the training length *L* increases, the NMSE decreases and **dictionary matrix A has been learned** with high performance.

X. Meng, and J. Zhu, "Bilinear Adaptive Generalized Vector Approximate Message Passing," IEEE Access, 2019

Conclusions

- **Considering approximate Bayesian inference methods for generalized linear models (GLM) using the message passing approach**

- **Deriving the approximate message passing (AMP) algorithm from expectation propagation**

- **Proposing a unified approximate inference framework for GLM**
  - **Simplifying the extension of various SLM inference algorithms to GLM inference**
  - **Providing new insights on some existing GLM inference algorithms**

- **Extending the GLM to bilinear matrix recovery problem and proposing one efficient message passing algorithm called BAd-VAMP**

References

- [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.
- [Rangan10] Rangan, Sundeep. "Estimation with random linear mixing, belief propagation and compressed sensing." Information Sciences and Systems (CISS), 2010 44th Annual Conference on. IEEE, 2010.
- [Tanaka 02] Tanaka, Toshiyuki. "A statistical-mechanics approach to large-system analysis of CDMA multiuser detectors." IEEE Transactions on Information theory 48.11 (2002): 2888-2910.
- [Minka01] Minka, Thomas P. "Expectation propagation for approximate Bayesian inference." 2001
- [OW05] Opper, Manfred, and Ole Winther. "Expectation consistent approximate inference." Journal of Machine Learning Research 6.Dec (2005): 2177-2204.
- [Rangan11] Rangan, "Generalized approximate message passing for estimation with random linear mixing." Proc IEEE ISIT 2011
- [RSF16] Rangan, Schniter, Fletcher, “Vector approximate message passing”, 2016
- [SRF16] P. Schniter, S. Rangan, and A. K. Fletcher, “Vector approximate message passing for the generalized linear model,” in Proc. 50th Asilomar Conf. Signals, Syst. Comput., Nov. 2016, pp. 1525-1529.
- [Kabashima 03] Kabashima Y. A , " CDMA multiuser detection algorithm on the basis of belief propagation” , Journal of Physics A: Mathematical and General, 2003, 36(43)
- [KMSSZ12] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, “Probabilistic reconstruction in compressed sensing: algorithms, phase diagrams, and threshold achieving matrices,” Journal of Statistical Mechanics: Theory and Experiment, vol. 2012, no. 08, p. P08009, 2012.
- [MWKL15a] X. Meng, S. Wu, L. Kuang, and J. Lu, “An expectation propagation perspective on approximate message passing,” IEEE Signal Process. Lett., vol. 22, no. 8, pp. 1194-1197, Aug. 2015.
- [MWKL15b] X. Meng, S. Wu, L. Kuang, and J. Lu, “ Concise derivation of complex Bayesian approximate message passing via expectation propagation,” arXiv preprint arXiv:1509.08658, 2015.
- [WKNLHDQ14] S. Wu, L. Kuang, Z. Ni, J. Lu, D. Huang, and Q. Guo, “Low-complexity iterative detection for large-scale multiuser MIMO-OFDM systems using approximate message passing,” *IEEE J. Sel. Topics Signal Process.,* vol. 8, no. 5, pp. 902–915, Oct. 2014.
- [MWZ18]X. Meng, S. Wu and J. Zhu, “A unified Bayesian inference framework for generalized linear model,” IEEE Signal Process. Lett., vol. 25, no. 3, Mar. 2018.

References

- [MZ18] X. Meng, and J. Zhu, “Bilinear Adaptive Generalized Vector Approximate Message Passing,” arXiv preprint arXiv:1810.08129, 2018
- [PSC14a] J. T. Parker, P. Schniter, and V. Cevher, “Bilinear generalized approximate message passing-Part I: Derivation,” IEEE Trans. Signal Process., vol. 62, no. 22, pp. 5839-5853, Nov. 2014.
- [PSC14b] J. T. Parker, P. Schniter, and V. Cevher, “Bilinear generalized approximate message passing-Part II: Applications,” IEEE Trans. Signal Process., vol. 62, no. 22, pp. 5854-5867, Nov. 2014.
- [PS16] J. T. Parker and P. Schniter, “Parametric bilinear generalized approximate message passing,” IEEE J. Sel. Topics Signal Process., vol. 10, no. 4, pp. 795-808, 2016.
- [KKMSZ16] Y. Kabashima, F. Krzakala, M. M´ezard, A. Sakata and L. Zdeborov´a, “Phase transitions and sample complexity in bayesoptimal matrix factorization,” IEEE Trans. Inf. Theory, vol. 62, no. 7, pp. 4228-4265, 2016.
- [SRF] P. Schniter, S. Rangan, and A. K. Fletcher, “Vector approximate message passing for the generalized linear model,” in Proc. 50th Asilomar Conf. Signals, Syst. Comput., Nov. 2016, pp. 1525-1529.
- [ML17] J. Ma and L. Ping, “Orthogonal AMP,” IEEE Access, vol. 5, pp. 2020-2033, 2017.
- [HWJ17] H. He, C. K. Wen, and S. Jin, “Generalized expectation consistent signal recovery for nonlinear measurements,” in Proc. IEEE Int. Symp. Inf. Theory, Jun. 2017, pp. 2333-2337.
- [QZW18] Zou Q, Zhang H, Wen C K, et al. , “ Concise Derivation for Generalized Approximate Message Passing Using Expectation Propagation ,” IEEE Signal Processing Letters, 2018.
- [SFRS18] S. Sarkar, A. K. Fletcher, S. Rangan and P. Schniter, “Bilinear recovery using adaptive vector-AMP,” avaliable at
- https://arxiv.org/pdf/1809.00024.pdf.
- [VS13] J. P. Vila and P. Schniter, “Expectation-maximization Gaussian-mixture approximate message passing,” IEEE Trans. Signal Process., vol. 61, no. 19, pp. 4658-4672, Oct. 2013.
- [KMZ13] F. Krzakala, M. Mezard, and L. Zdeborova, “Compressed sensing under matrix uncertainty: Optimum thresholds and robust approximate message passing,” ICASSP, 2013.

Thank You！

ありがとう

Q&A