A High-bias Low-variance Introduction to Approximate Bayesian Inference

Xiangming Meng
Postdoctoral Researcher
Approximate Bayesian Inference (ABI) Team
RIKEN Center for Advanced Intelligence Project

Oct. 11, 2019
Tokyo Institute of Technology

Physics Reports 810 (2019) 1–124

Contents lists available at ScienceDirect

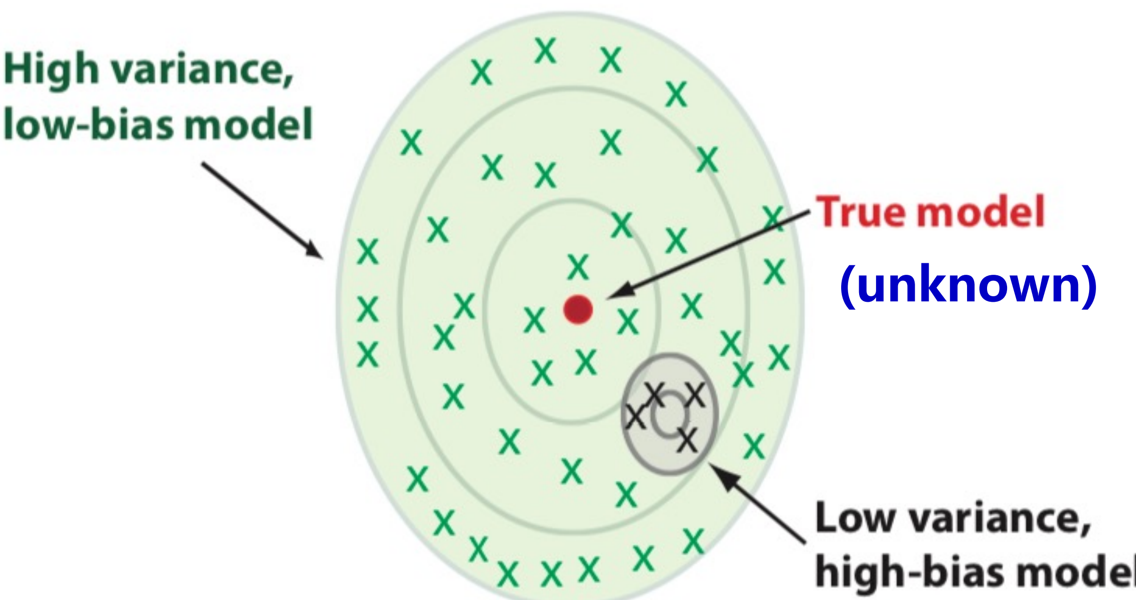
Physics Reports

journal homepage: www.elsevier.com/locate/physrep

A high-bias, low-variance introduction to Machine Learning for physicists

Pankaj Mehta [a], Marin Bukov [b,*], Ching-Hao Wang [a], Alexandre G.R. Day [a], Clint Richardson [a], Charles K. Fisher [c], David J. Schwab [d]

100 pages !



Physics Reports 810 (2019) 1–124

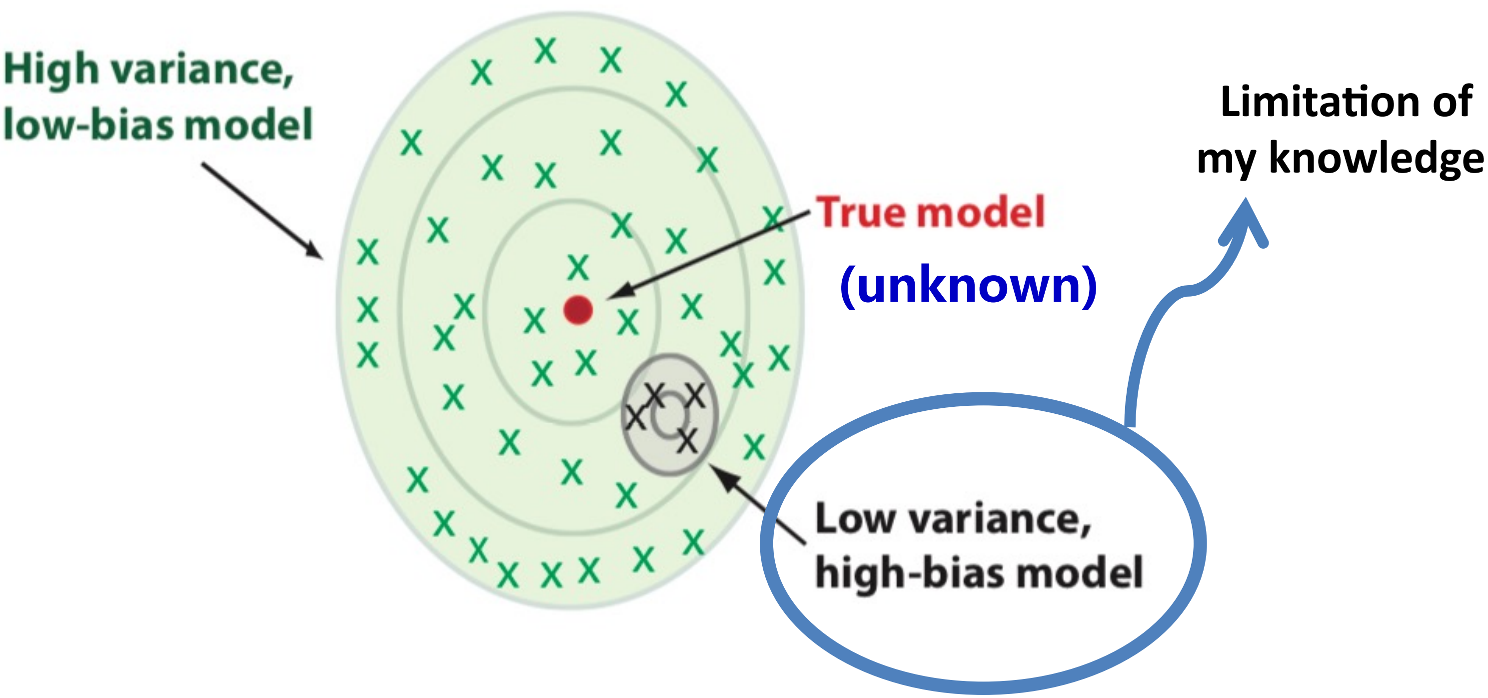
Contents lists available at ScienceDirect

Physics Reports

journal homepage: www.elsevier.com/locate/physrep

A high-bias, low-variance introduction to Machine Learning for physicists

Pankaj Mehta [a], Marin Bukov [b,*], Ching-Hao Wang [a], Alexandre G.R. Day [a], Clint Richardson [a], Charles K. Fisher [c], David J. Schwab [d]

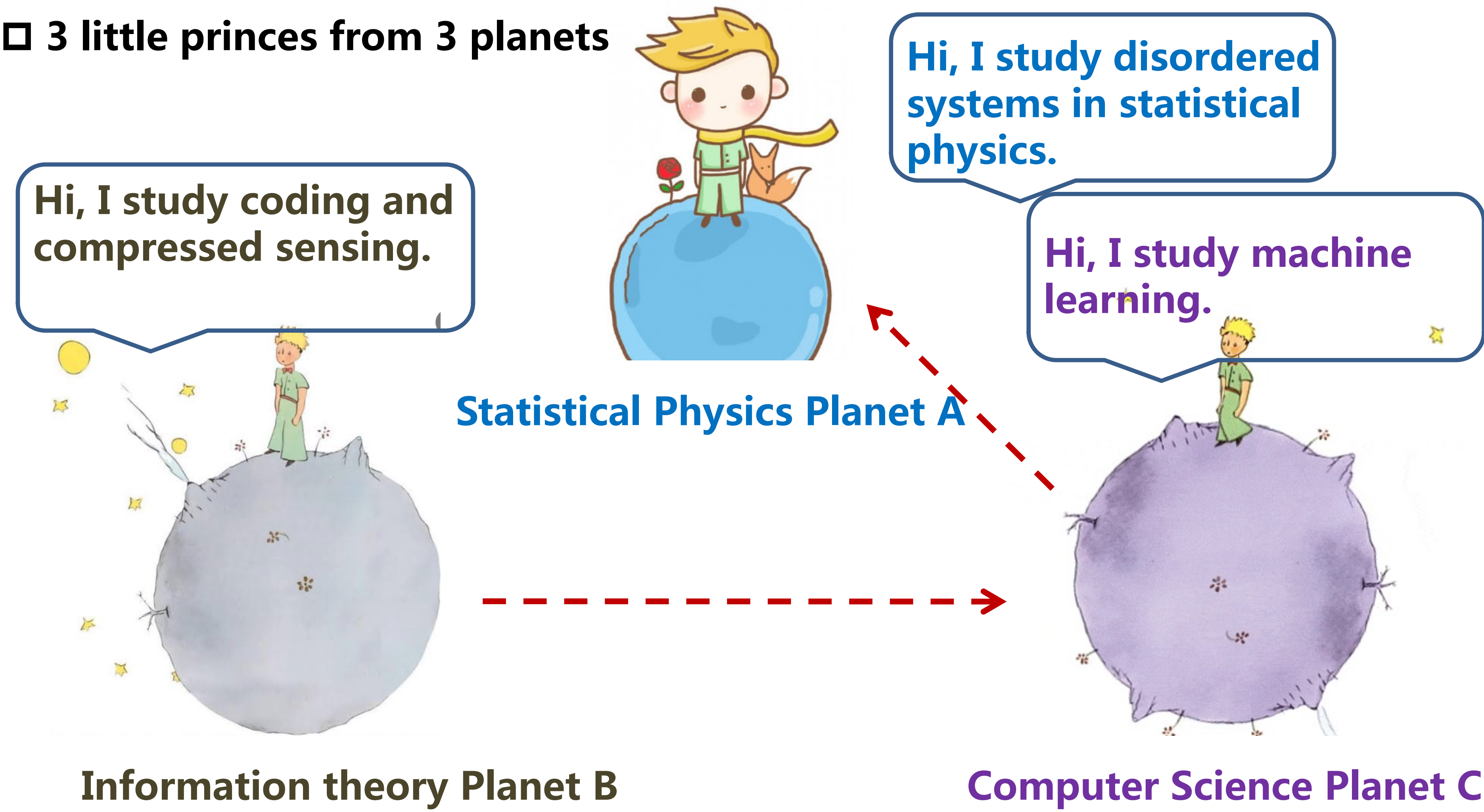


Outline

- Background
- Variational Inference
- Expectation Propagation
- A Unified EP Perspective on AMP and its extensions
- Conclusion

Background

- 3 little princes from 3 planets



After a long time discusstion, it turns out that
they are studying similar problems using different langurages

Background

◻ Communication

> *"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point"*
>
> —Shannon (1948)

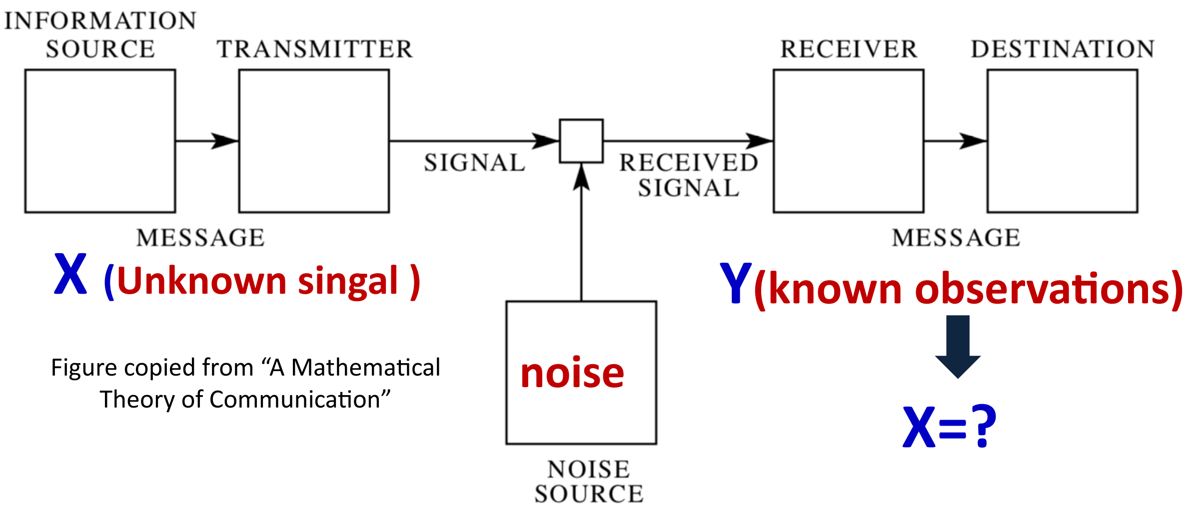
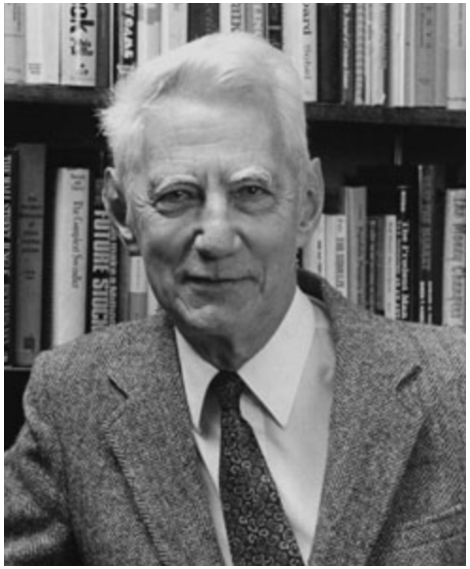


Figure copied from "A Mathematical Theory of Communication"

Fig 1. Schematic diagram of a general communication system



Claude Elwood Shannon (1916-2001)

Background

◻ Communication

> *"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point"*
>
> —Shannon (1948)

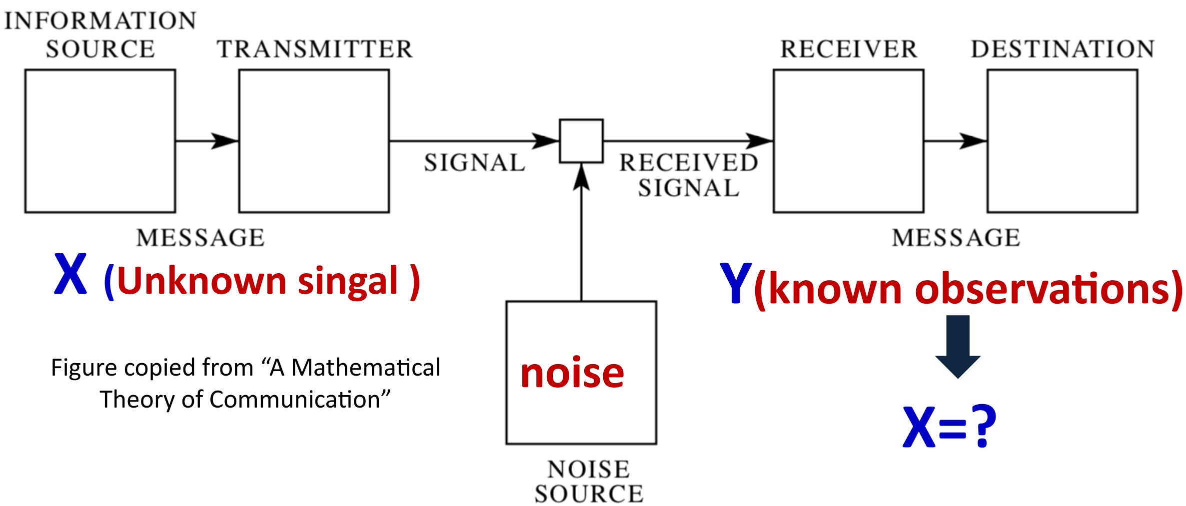
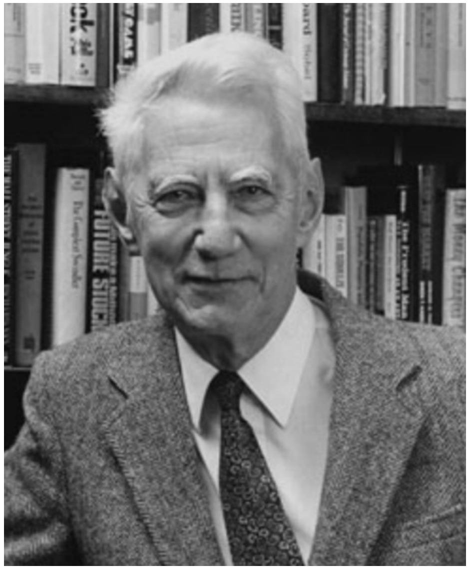


Figure copied from "A Mathematical Theory of Communication"

Fig 1. Schematic diagram of a general communication system

- **Q1: How to quantize information?**
  Entropy $H = -\sum_k p_k \log p_k$
- **Q2: What is the capacity of a communication system?**
  Shannon Formula: C = W*log(1+S/N) **maxmimum rate**
- **Q3: How to approach the capacity?**
  Channel coding (Turbo code, LDPC code, Polar code in 5G)



Claude Elwood Shannon (1916-2001)

> 'You should call it entropy, for two reasons. In the first place your uncertainty function has been used in statistical mechanics under that name, so it already has a name. In the second place, and more important, no one really knows what entropy really is, so in a debate you will always have the advantage.'
>
> —John von Neumann

Background

□ Communication

Received Message y

Tokye Institote of Technalogy

?

Background

◻ Communication

Received Message y

Tokye Institote of Technalogy

Corrected Message x

Tokyo Institute of Technology

Background

- Communication

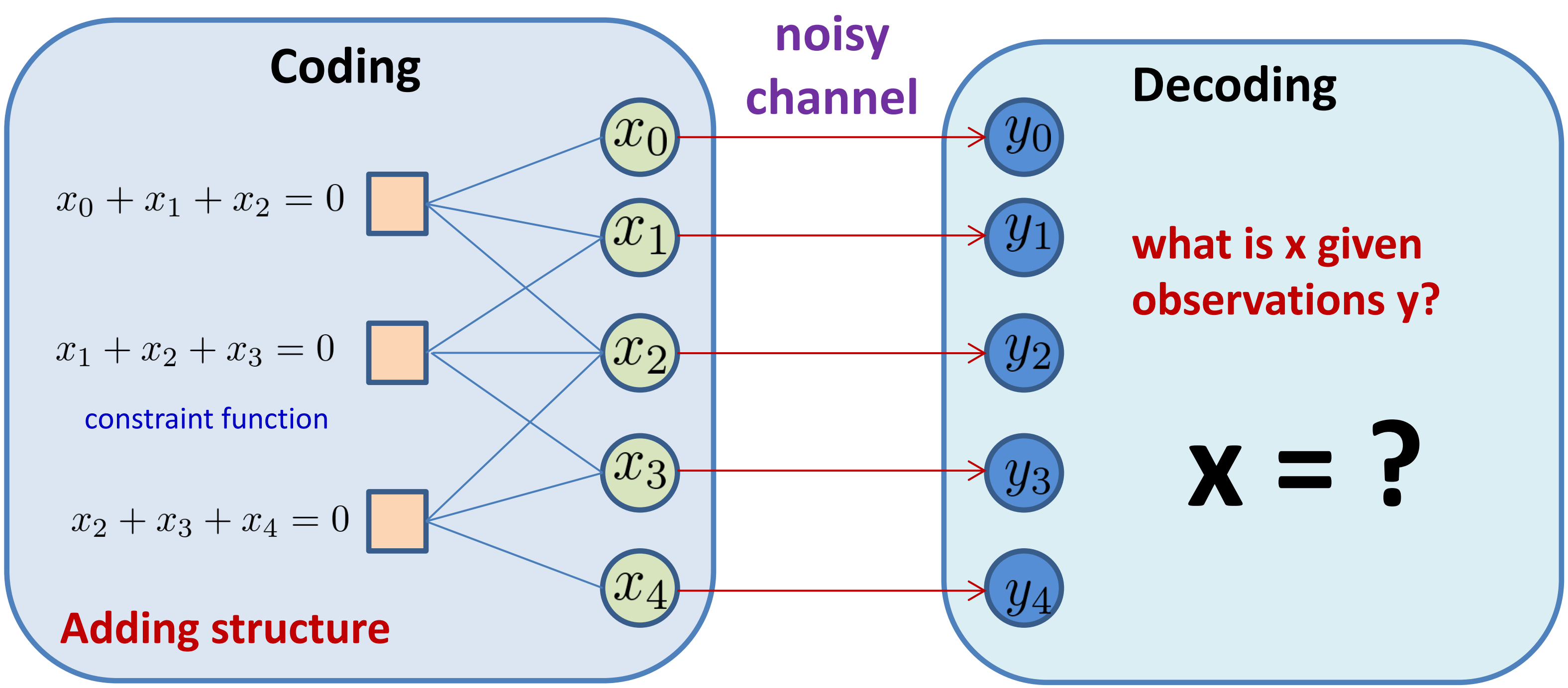
Received Message y

Tokye Institote of Technalogy

Corrected Message x

Tokyo Institute of Technology

There is structure within the transmitted codes.



Background

□ Compressed Sensing

Raw: 15MB

JPEG: 150KB

- **Massive data acuisition**
- **Most of the data is reduntant**
- **Wasteful meaurements**

- **Could we acqure images using less/efficient measurements?**

This slide is copy from https://www.raeng.org.uk/publications/other/candes-presentation-frontiers-of-engineering

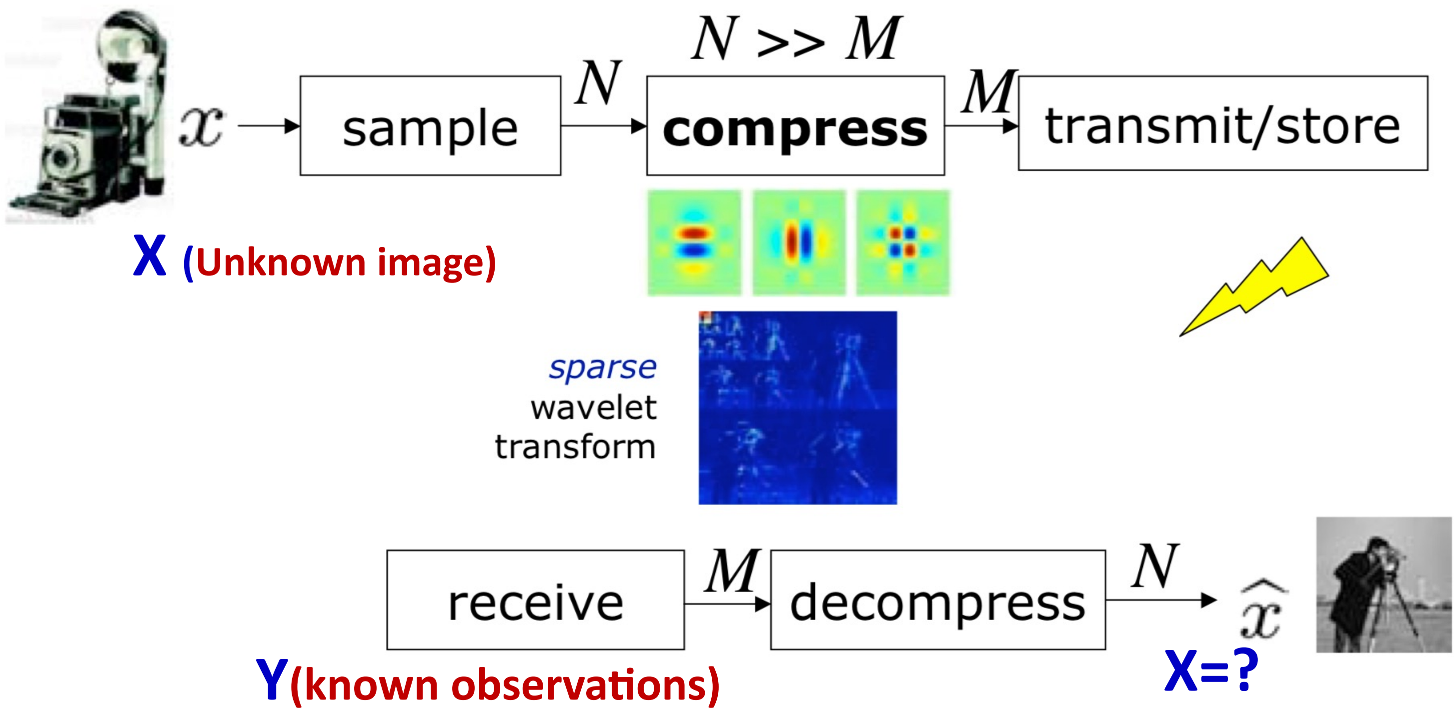
Background

◻ Compressed Sensing

Raw: 15MB   JPEG: 150KB

- **Massive data acuisition**
- **Most of the data is reduntant**
- **Wasteful meaurements**

- **Could we acqure images using less/efficient measurements?**



Background

- **Bayesian Perspective**

$\mathbf{x} \sim p(\mathbf{x}) \longrightarrow$ **transformations** $\longrightarrow \mathbf{y}$

Unknown Signal

Known Observations

what is x?

Background

☐ Bayesian Perspective

Prior distribution

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

Unknown Signal

likelihood distribution

$p(\mathbf{y}|\mathbf{x})$

$\mathbf{y}$

Known Observations

Posterior distribution

Thomas Bayes (1702-1761)

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

Bayes' rule

$$\mathbf{Z} \triangleq p(\mathbf{y}) = \sum_{\mathbf{x}} p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$$

evidence (partition function)

Background

Bayesian Perspective

Prior distribution

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

Unknown Signal

likelihood distribution

$$p(\mathbf{y}|\mathbf{x})$$

$\mathbf{y}$

Known Observations

Posterior distribution

Thomas Bayes (1702-1761)

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

Bayes' rule

$$\mathbf{Z} \triangleq p(\mathbf{y}) = \sum_{\mathbf{x}} p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$$

evidence (partition function)

Goal

marginal distribution $p(x_i|\mathbf{y}) = \sum_{x_j \neq x_i} p(\mathbf{x}|\mathbf{y})$

i = 1,..., N

posterior mean $\hat{x}_i = E\left(x_i|\mathbf{y}\right) = \sum_{x_i} x_i p(x_i|\mathbf{y})$

Background

- Bayesian Perspective

likelihood distribution

Prior distribution

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

Unknown Signal

$$p(\mathbf{y}|\mathbf{x})$$

$\mathbf{y}$

Known Observations

Posterior distribution

Thomas Bayes (1702-1761)

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

Bayes' rule

$$\mathbf{Z} \triangleq p(\mathbf{y}) = \sum_{\mathbf{x}} p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$$

evidence (partition function)

- Goal

marginal distribution

$$p(x_i|\mathbf{y}) = \sum_{x_j \neq x_i} p(\mathbf{x}|\mathbf{y})$$

posterior mean

$$\hat{x}_i = E\left(x_i|\mathbf{y}\right) = \sum_{x_i} x_i p(x_i|\mathbf{y})$$

Curse of Dimentionality!

e.g., N spins, O(2^N)

We have to resort to approximate Bayesin Inference methods

Outline

- Background
- **Variational Inference**
- Expectation Propagation
- A Unified EP Perspective on AMP and its extensions
- Conclusion

Variational Inference

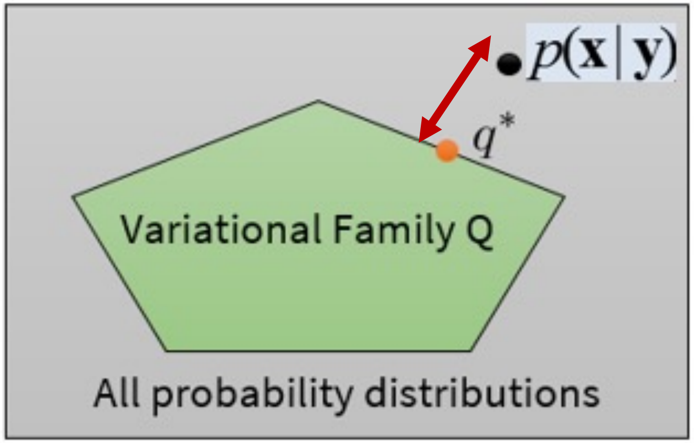
□ Two Common Approaches of Approximate Inference

Variational Inference **vs.** **Sampling Method**

Variational Inference	Sampling Method
• **Deterministic**	• **Stochastic**
• **Biased**	• **Unbiased**
• **Scalable**	• **Non-scalable**

Variational Inference

◻ Basic Principle

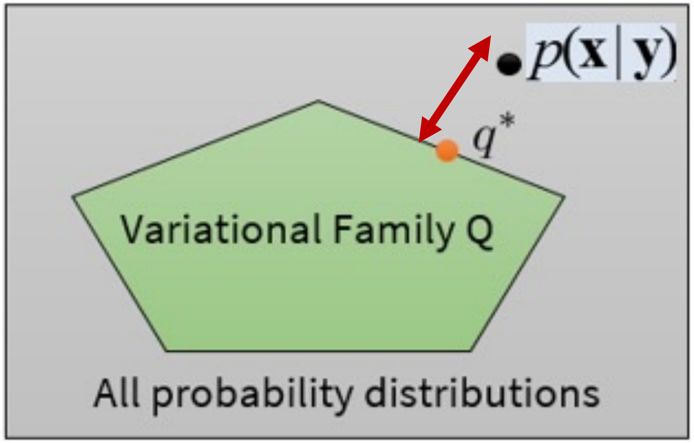


To approxmate complicated target distribtuion p with a simple distribution q as close to p as possible

$$q \approx p$$

Variational Inference

☐ Basic Principle



To approxmate complicated target distribtuion p with a simple distribution q as close to p as possible

$$q \approx p$$

Optimization problem $q^* = \arg\min_{q \in Q} KL(q(\mathbf{x}) || p(\mathbf{x}|\mathbf{y}))$

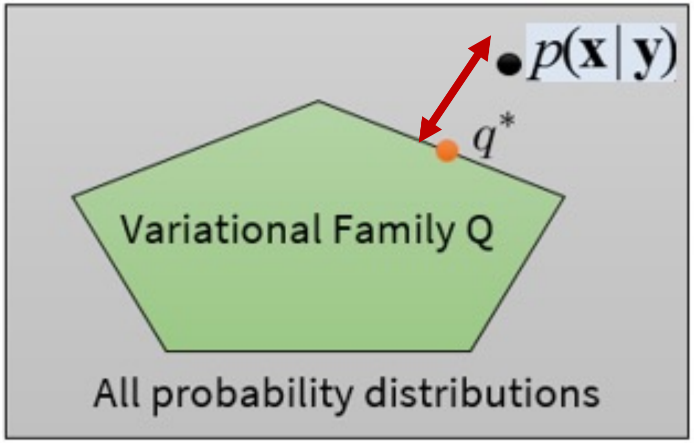
KL divergence

"distance"

$$KL(q||p) = \sum_{\mathrm{x}} q(\mathrm{x}) \log \frac{q(\mathrm{x})}{p(\mathrm{x})}$$

Variational Inference

☐ Basic Principle



$$\min_{q \in Q} KL(q(\mathbf{x})||p(\mathbf{x}|\mathbf{y}))$$

To approxmate complicated target distribtuion p with a simple distribution q as close to p as possible

$$q(\mathrm{x}) \log \frac{q(\mathrm{x})}{p(\mathrm{x})}$$

q ≈ p

Optimization problem

KL divergence

"distance"

- **Non-negativity of KL**
  KL(p||q) >=0 and KL(p||q) =0 if and only if p = q

"Gibbs inequality"

- **Non-symmetry of KL**
  KL(p||q) is not equal to KL(q||p)

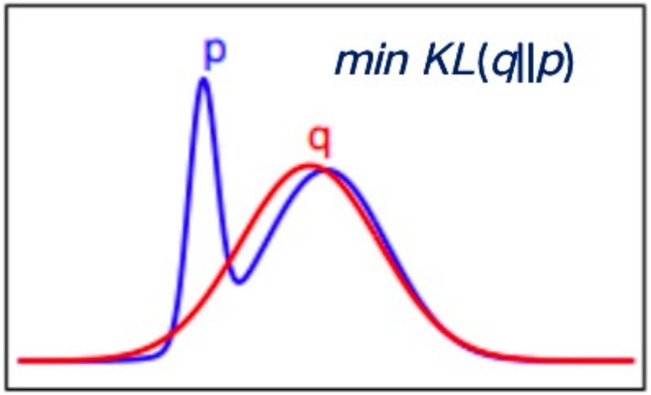
qseudo distance

Variational Inference

Basic Principle

- **KL divergence**

$$KL(q||p) = \sum q(x) \log \frac{q(x)}{p(x)}$$



$\neq$

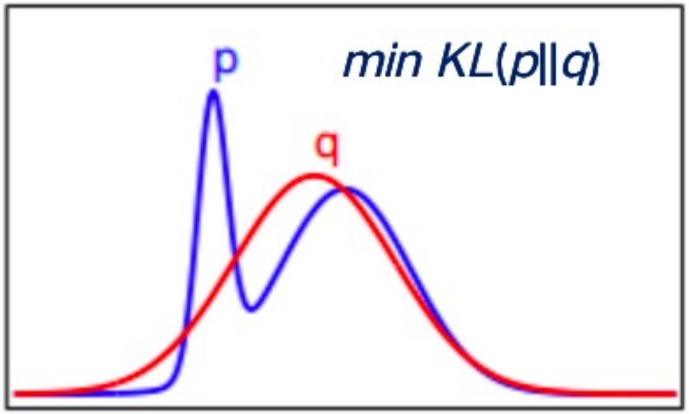


figure copied from [Bishop06]

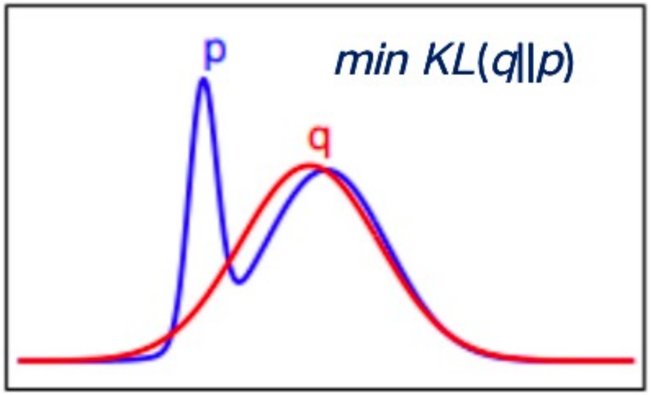
Variational Inference

☐ Basic Principle

- **KL divergence**

$$KL(q||p) = \sum q(x) \log \frac{q(x)}{p(x)}$$

goal



$\neq$

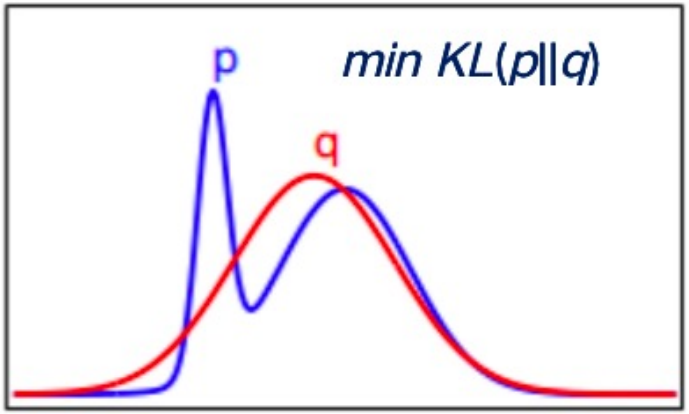


figure c

Remember that VI uses *KL*(*q*||*p*)

dilemma!

To calculate the KL divergence, **we must know the target distribution in adance**, which is our primary goal!

Variational Inference

- **ELBO bound**

$$KL(q(\mathbf{x})||p(\mathbf{x}|\mathbf{y})) = \sum_{\mathrm{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})} = \sum_{\mathrm{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})p(\mathbf{y})}{p(\mathbf{x},\mathbf{y})}$$

Bayes' Rule

$$= \sum_{\mathrm{x}} q(\mathbf{x}) \log q(\mathbf{x}) - \sum_{\mathrm{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) + \log p(\mathbf{y})$$

Expansion

≥0 **"Gibbs inequality"**

Variational Inference

- ELBO bound

$$KL(q(\mathbf{x})||p(\mathbf{x}|\mathbf{y})) = \sum_{\mathrm{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})} = \sum_{\mathrm{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})p(\mathbf{y})}{p(\mathbf{x},\mathbf{y})}$$ **Bayes' Rule**

$$= \sum_{\mathrm{x}} q(\mathbf{x}) \log q(\mathbf{x}) - \sum_{\mathrm{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) + \log p(\mathbf{y})$$ **Expansion**

≥0 "Gibbs inequality"

$$\underbrace{\log p(\mathbf{y})}_{\text{Log Partition function}} \geq \underbrace{\sum_{\mathrm{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) - \sum_{\mathrm{x}} q(\mathbf{x}) \log q(\mathbf{x})}_{\text{Evidence Lower Bound (ELBO)}}$$

Variational Inference

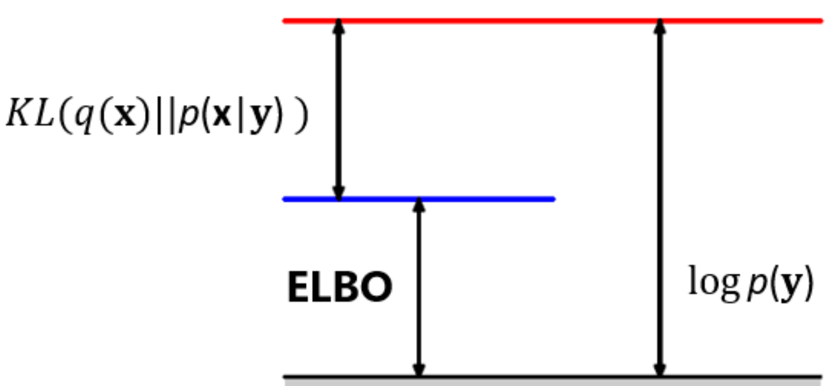
- **ELBO bound**

$$KL(q(\mathbf{x})||p(\mathbf{x}|\mathbf{y})) = \sum_{\mathrm{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})} = \sum_{\mathrm{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})p(\mathbf{y})}{p(\mathbf{x},\mathbf{y})}$$ **Bayes' Rule**

$$= \sum\nolimits_{\mathrm{x}} q(\mathbf{x}) \log q(\mathbf{x}) - \sum\nolimits_{\mathrm{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) + \log p(\mathbf{y})$$ **Expansion**

≥0 **"Gibbs inequality"**

$$\underbrace{\log p(\mathbf{y})}_{\text{Log Partition function}} \geq \underbrace{\sum_{\mathrm{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) - \sum_{\mathrm{x}} q(\mathbf{x}) \log q(\mathbf{x})}_{\text{Evidence Lower Bound (ELBO)}}$$

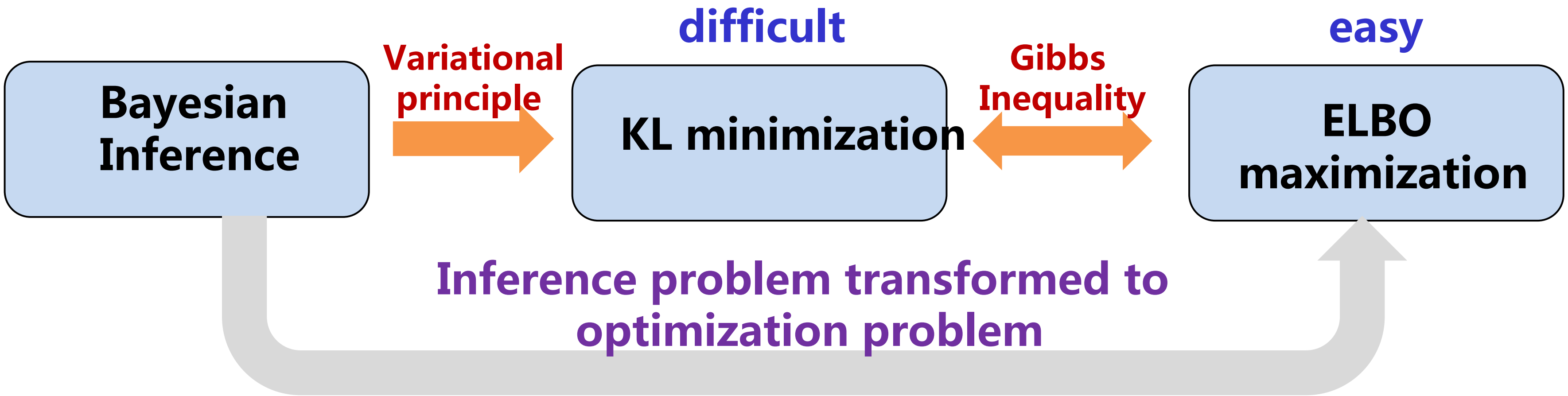


minimize KL = maximize ELBO

Variational Inference

- **ELBO bound**

Big Picture of VI



Variational Inference

☐ Analogy between different planets

Computer Science Planet

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

evidence lower bound

$$\mathrm{ELBO} \triangleq \langle \log p(\mathbf{x}, \mathbf{y}) \rangle_q + H(q(\mathbf{x})) \leq \log p(\mathbf{y})$$

Statisticl physics Planet

$$p(\mathbf{s}|\beta, J) = \frac{e^{-\beta E(\mathbf{s}, J)}}{Z(\beta, J)}$$

free energy

variational free energy

$$\beta F_q(\mathbf{J}) \triangleq \beta \langle E(\mathbf{s}, J) \rangle_q - H(q(\mathbf{s})) \geq -\log Z(\beta, J)$$

Statistical Physics	Computer Science/Information Theory
Spins/dgrees of freedom $\mathbf{s}$	Hidden variables/signal of interest $\mathbf{x}$
Couplings/quenched disorder $\mathbf{J}$	Data observations $\mathbf{y}$
Boltzmann factor $e^{-\beta E(\mathbf{s}, J)}$	Joint distribution $p(\mathbf{x}, \mathbf{y})$
Partition function $Z(\beta, J)$	Evidence $p(\mathbf{y})$
Energy $\beta E(\mathbf{s}, J)$	Negative log-joint distribution $-\log p(\mathbf{x}, \mathbf{y})$
Free Energy $-\log Z(\beta, J)$	Negative log evidence $-\log p(\mathbf{y})$
Variational distribution $q(\mathbf{s})$	Variational distribution $q(\mathbf{x}|\mathbf{y})$
Variational free energy $\beta F_q(\mathbf{J})$	Negative ELBO -ELBO

Table modified from Table I in [Mehta et al 19]

Variational Inference

◻ Why transfroming inference to optimization?

$$\max \text{ELBO} = \sum_{\mathbf{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) - \sum_{\mathbf{x}} q(\mathbf{x}) \log q(\mathbf{x})$$

There are a bunch of optimization methods we could leverage!

- **different choice of q**
  - ✓ **structure:** mean-field, Bethe, etc.
  - ✓ **parametric:** Gaussian, nueral network, etc.

- **different optimization methods**
  - ✓ coordient descent
  - ✓ gradient descent
  - ✓ stochastic gradient descent
  - ✓ natural gradient descent
  - ✓ ......

Different combinations lead to different inference algorithms

Mean-filed Approximation

Mean Field Approximation

$$\max \textbf{ELBO} = \sum_{\mathrm{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) - \sum_{\mathrm{x}} q(\mathbf{x}) \log q(\mathbf{x})$$

Mean Field structure

$$q(\mathbf{x}) = \prod_i q(x_i)$$

different variables are independent

ELBO

$$= \sum_{\mathrm{x}} \prod_i q(x_i) \log p(\mathbf{x},\mathbf{y}) - \sum_i q(x_i) \log q(x_i)$$

Mean-filed Approximation

◻ Mean Field Approximation

$$\max \textbf{ELBO} = \sum_{\mathrm{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) - \sum_{\mathrm{x}} q(\mathbf{x}) \log q(\mathbf{x})$$

Mean Field structure

$$q(\mathbf{x}) = \prod_i q(x_i)$$

different variables are independent

Using coordinate descent optimization, we obtain the variational message passing (VMP) algorithm:

ELBO

$$= \sum_{\mathrm{x}} \prod_i q(x_i) \log p(\mathbf{x},\mathbf{y}) - \sum_i q(x_i) \log q(x_i)$$

Input: A model $p(\mathbf{x},\mathbf{y})$, a dataset

Output: $q(\mathbf{x}) = \prod_i q(x_i)$

1: Initialize variational factors $q(\mathbf{x})$
2: **while** the ELBO has not converged **do**
3: **for** $i \in 1, 2, \cdots, d$ **do**
4: $q(x_i) \propto \exp\left\{\mathbb{E}_{\prod_{j \neq i} q(x_j)}[\log p(\mathbf{x},\mathbf{y})]\right\}$
5: **end for**
6: Compute ELBO
7: **end while**

Bethe Approximation

- Bethe approximation/Kikuchi Approximation

$$\max \text{ELBO} = \sum_{\mathrm{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) - \sum_{\mathrm{x}} q(\mathbf{x}) \log q(\mathbf{x})$$

Bethe Approximation: $q(\mathbf{x}) = \dfrac{\prod_a q(\mathbf{x}_a)}{\prod_i [q(x_i)]^{d_i-1}}$ **pair-wise correlations**

ELBO with Bethe Approximation

$$\text{ELBO} = \sum_a \sum_{\mathrm{x}_a} q(\mathbf{x}_a) \log f_a(\mathbf{x}_a) - \sum_i q(x_i) \log q(x_i)$$

$$- \sum_a \sum_{\mathrm{x}_a} q(\mathbf{x}_a) \log q(\mathbf{x}_a) + \sum_i (d_i - 1) \sum_{x_i} q(x_i) \log q(x_i)$$

s.t.

Bethe Approximation

- Bethe approximation/Kikuchi Approximation

$$\max \mathbf{ELBO} = \textstyle\sum_{\mathrm{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) - \sum_{\mathrm{x}} q(\mathbf{x}) \log q(\mathbf{x})$$

Bethe Approximation: $q(\mathbf{x}) = \dfrac{\prod_a q(\mathbf{x}_a)}{\prod_i [q(x_i)]^{d_i - 1}}$ **pair-wise correlations**

ELBO with Bethe Approximation

$$\mathbf{ELBO} = \sum_a \sum_{\mathrm{x_a}} q(\mathbf{x}_a) \log f_a(\mathbf{x}_a) + \sum_i q(x_i) \log q(x_i)$$

$$- \sum_a \sum_{\mathrm{x_a}} q(\mathbf{x}_a) \log q(\mathbf{x}_a) + \sum_i (d_i - 1) \sum_{x_i} q(x_i) \log q(x_i)$$

s.t.

$$m_{a\to i}(x_i) = \sum_{\mathrm{x_a} \backslash x_i} f_a(\mathbf{x}_a) \prod_j m_{j\to a}(x_i)$$

Langrange Multiplier

Belief Propagation(BP)

$$m_{i\to a}(x_i) = \prod_{b\neq a} m_{b\to i}(x_i)$$

Message passing on the factor graph

[Yedidia et al 02,05]

Belief Propagation

A Toy Example

There are 4 random discrete variables, each taking 10 possible values randomly.
The joint distribution is

$$p\left(x_1, x_2, x_3, x_4\right) = p\left(x_1\right) p\left(x_2 | x_1\right) p\left(x_3 | x_2\right) p\left(x_4 | x_2\right)$$

Question: How to compute the maginal distribution $p\left(x_2\right)$?

Belief Propagation

A Toy Example

There are 4 random discrete variables, each taking 10 possible values randomly.
The joint distribution is

$$p(x_1, x_2, x_3, x_4) = p(x_1)\, p(x_2|x_1)\, p(x_3|x_2)\, p(x_4|x_2)$$

Question: How to compute the maginal distribution $p(x_2)$

Direct Answer： marginalize out all other varibles of the joint distribution

$$p(x_2) = \sum_{x_1, x_3 x_4} p(x_1)\, p(x_2|x_1)\, p(x_3|x_2)\, p(x_4|x_2)$$

Number of values of	x_2	10
For each combination	x_1, x_3, x_4	3 multiplications
Number of combinations	x_1, x_3, x_4	10*10*10 = 10^3

$$\mathcal{O}\left(10^4\right)$$

Total Number of Multiplications :	10*3*10^3 =3*10^4
Total Number of Additions :	10*(10^3 -1) $\approx 10^4$

Belief Propagation

◻ A Toy Example

There are 4 random discrete variables, each taking 10 possible values randomly.
The joint distribution is

$$p(x_1, x_2, x_3, x_4) = p(x_1)\, p(x_2|x_1)\, p(x_3|x_2)\, p(x_4|x_2)$$

Question: How to compute the maginal distribution $p(x_2)$

Direct Answer： marginalize out all other varibles of the joint distribution

$$p(x_2) = \sum_{x_1, x_3 x_4} p(x_1)\, p(x_2|x_1)\, p(x_3|x_2)\, p(x_4|x_2)$$

Number of values of	x_2	**10**
For each combination	x_1, x_3, x_4	**3 multiplications**
Number of combinations	x_1, x_3, x_4	**10*10*10 = 10^3**

$$\mathcal{O}\left(10^4\right)$$

Total Number of Multiplications :	**10*3*10^3 =3*10^4**
Total Number of Additions :	**10*(10^3 -1) $\approx 10^4$**

The structure of the joint distribution is totally ignored!

Belief Propagation

A Toy Example

The distributive law

$$ab + ac = a\left(b + c\right)$$

Belief Propagation

◻ A Toy Example

The distributive law

$$ab + ac = a\left(b + c\right)$$

Alternative Answer:

$$p\left(x_2\right) = \sum_{x_1, x_3 x_4} p\left(x_1\right) p\left(x_2|x_1\right) p\left(x_3|x_2\right) p\left(x_4|x_2\right)$$

$$= \underbrace{\sum_{x_1} p\left(x_1\right) p\left(x_2|x_1\right)}_{m_A(x_2)} \underbrace{\sum_{x_3} p\left(x_3|x_2\right)}_{m_B(x_2)} \underbrace{\sum_{x_4} p\left(x_4|x_2\right)}_{m_C(x_2)}$$

The distributive law

Belief Propagation

◻ A Toy Example

The distributive law

$$ab + ac = a\left(b + c\right)$$

Alternative Answer：

$$p\left(x_2\right) = \sum_{x_1, x_3 x_4} p\left(x_1\right) p\left(x_2 | x_1\right) p\left(x_3 | x_2\right) p\left(x_4 | x_2\right)$$

$$= \underbrace{\sum_{x_1} p\left(x_1\right) p\left(x_2 | x_1\right)}_{m_A(x_2)} \underbrace{\sum_{x_3} p\left(x_3 | x_2\right)}_{m_B(x_2)} \underbrace{\sum_{x_4} p\left(x_4 | x_2\right)}_{m_C(x_2)}$$

The distributive law

The original problem is divided into small sub-problems

Total Number of Multiplications : 10*(10+2) = 120

Total Number of Additions : 10*(9+9+9) = 270

$$\mathcal{O}\left(10^2\right)$$

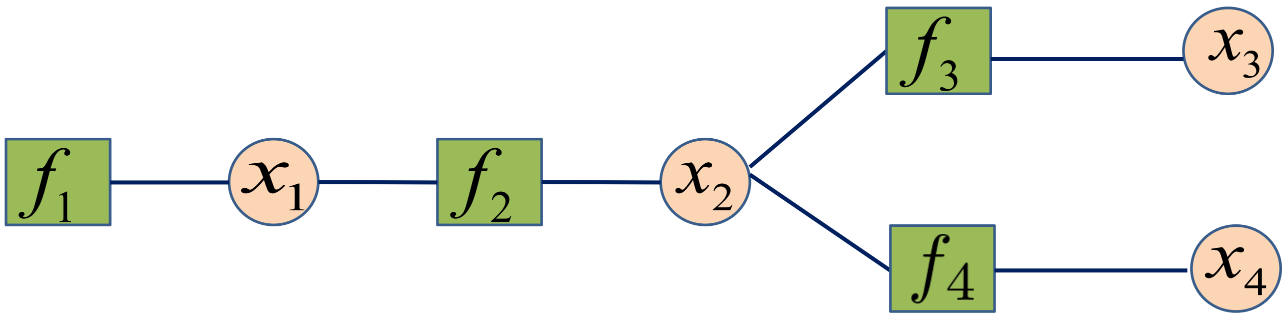
Belief Propagation

◻ A Toy Example

$$p\left(x_1, x_2, x_3, x_4\right) = \underbrace{p\left(x_1\right)}_{f_1} \underbrace{p\left(x_2 | x_1\right)}_{f_2} \underbrace{p\left(x_3 | x_2\right)}_{f_3} \underbrace{p\left(x_4 | x_2\right)}_{f_4}$$

Factor graph

- **circle nodes represent random variables**
- **square nodes represent factorizing functions**
- **function node *f* connects varible node *x* if and only if *x* is one of argument of *f***



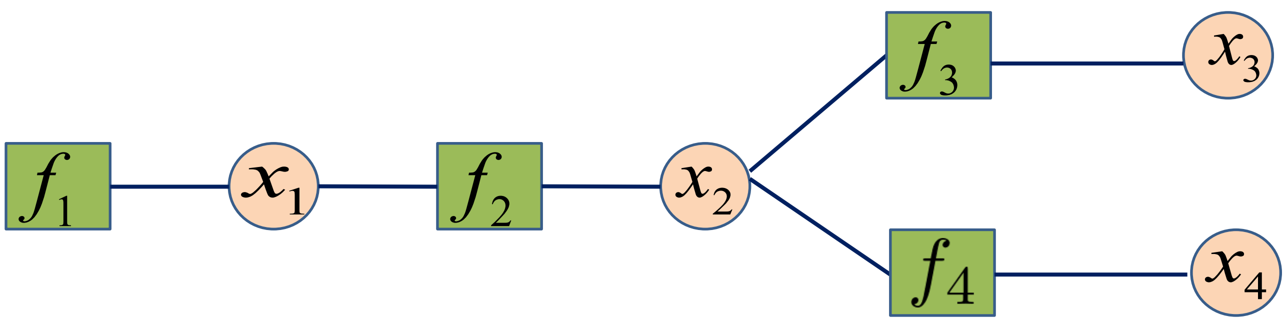
Belief Propagation

A Toy Example

$$p(x_1, x_2, x_3, x_4) = \underbrace{p(x_1)}_{f_1}\,\underbrace{p(x_2|x_1)}_{f_2}\,\underbrace{p(x_3|x_2)}_{f_3}\,\underbrace{p(x_4|x_2)}_{f_4}$$

Factor graph

- circle nodes represent random variables
- square nodes represent factorizing functions
- function node ***f*** connects varible node ***x*** if and only if ***x*** is one of argument of ***f***



$$p(x_2) = \sum_{x_1} p(x_1)\, p(x_2|x_1) \sum_{x_3} p(x_3|x_2) \sum_{x_4} p(x_4|x_2)$$

Inference process ⟷ **Message Passing on graph**

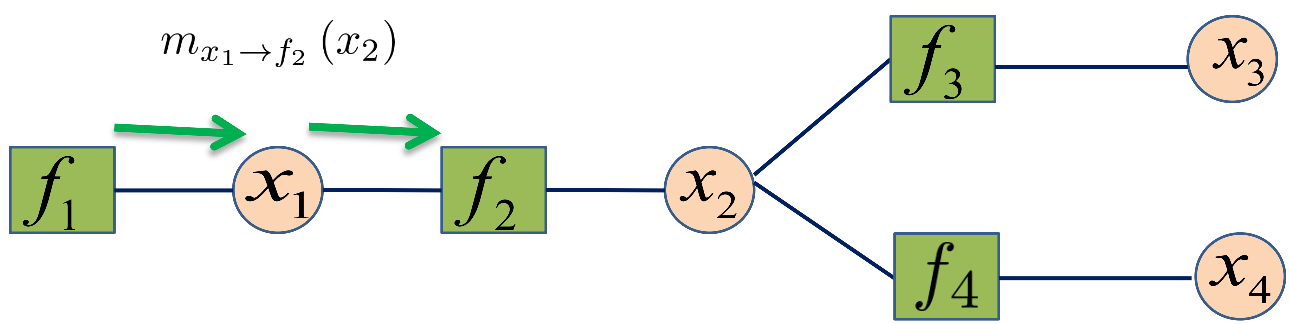
Belief Propagation

□ A Toy Example

$$p(x_1, x_2, x_3, x_4) = \underbrace{p(x_1)}_{f_1}\,\underbrace{p(x_2|x_1)}_{f_2}\,\underbrace{p(x_3|x_2)}_{f_3}\,\underbrace{p(x_4|x_2)}_{f_4}$$

Factor graph

- **circle nodes represent random variables**
- **square nodes represent factorizing functions**
- **function node *f* connects varible node *x* if and only if *x* is one of argument of *f***



$$p(x_2) = \sum_{x_1} \overbrace{p(x_1)}^{m_{x_1 \to f_2}(x_2)}\, p(x_2|x_1) \sum_{x_3} p(x_3|x_2) \sum_{x_4} p(x_4|x_2)$$

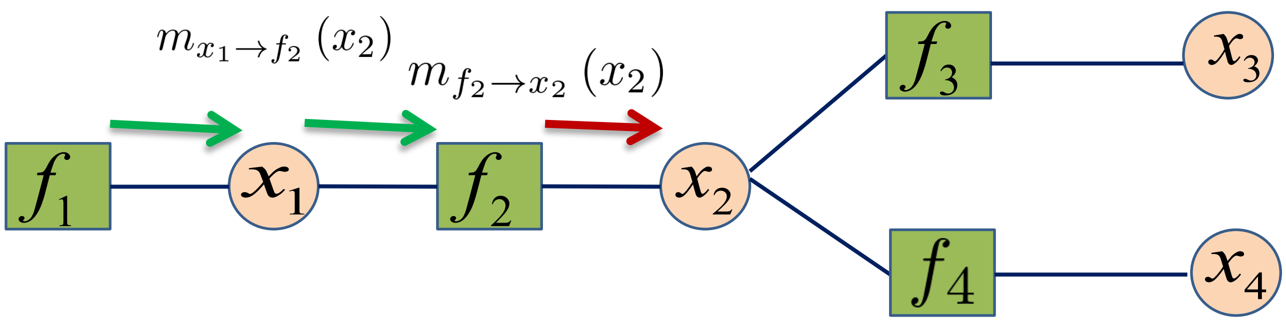
Belief Propagation

◻ A Toy Example

$$p\left(x_1, x_2, x_3, x_4\right) = \underbrace{p\left(x_1\right)}_{f_1} \underbrace{p\left(x_2|x_1\right)}_{f_2} \underbrace{p\left(x_3|x_2\right)}_{f_3} \underbrace{p\left(x_4|x_2\right)}_{f_4}$$

Factor graph

- circle nodes represent random variables
- square nodes represent factorizing functions
- function node ***f*** connects varible node ***x*** if and only if ***x*** is one of argument of ***f***



$$p(x_2) = \underbrace{\sum_{x_1} \overbrace{p\left(x_1\right)}^{m_{x_1 \to f_2}\left(x_2\right)} p\left(x_2|x_1\right)}_{m_{f_2 \to x_2}\left(x_2\right)} \sum_{x_3} p\left(x_3|x_2\right) \sum_{x_4} p\left(x_4|x_2\right)$$

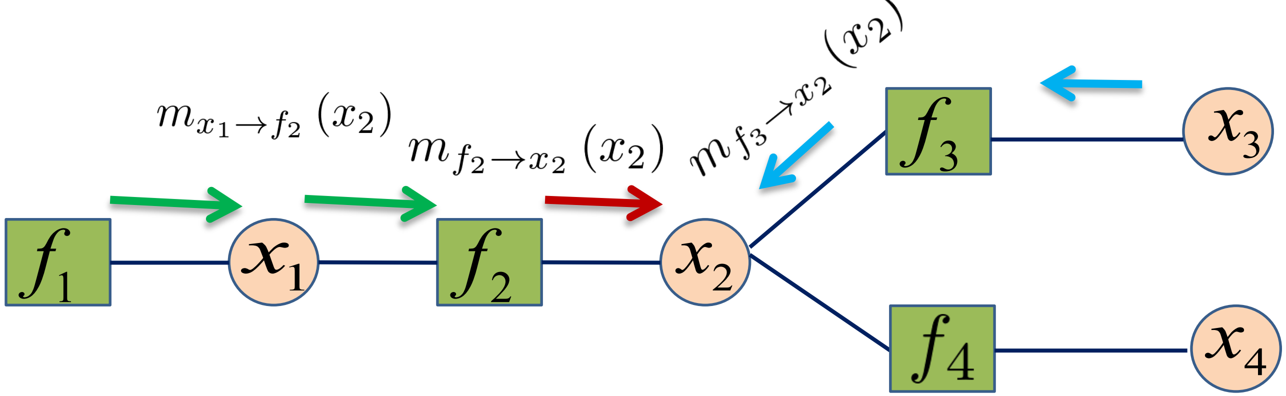
Belief Propagation

◻ A Toy Example

$$p\left(x_1, x_2, x_3, x_4\right) = \underbrace{p\left(x_1\right)}_{f_1} \underbrace{p\left(x_2 | x_1\right)}_{f_2} \underbrace{p\left(x_3 | x_2\right)}_{f_3} \underbrace{p\left(x_4 | x_2\right)}_{f_4}$$

Factor graph

- **circle nodes represent random variables**
- **square nodes represent factorizing functions**
- **function node *f* connects varible node *x* if and only if *x* is one of argument of *f***



$$p(x_2) = \underbrace{\sum_{x_1} \overbrace{p\left(x_1\right)}^{m_{x_1 \to f_2}(x_2)} p\left(x_2 | x_1\right)}_{m_{f_2 \to x_2}(x_2)} \underbrace{\sum_{x_3} p\left(x_3 | x_2\right)}_{m_{f_3 \to x_2}(x_2)} \sum_{x_4} p\left(x_4 | x_2\right)$$

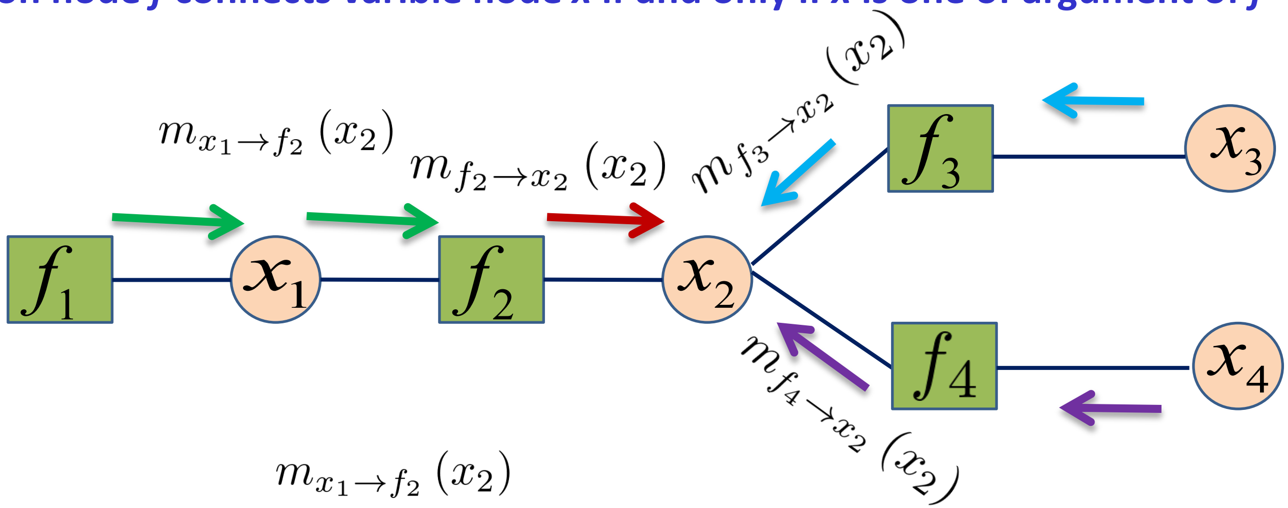
Belief Propagation

◻ A Toy Example

$$p(x_1, x_2, x_3, x_4) = \underbrace{p(x_1)}_{f_1}\,\underbrace{p(x_2|x_1)}_{f_2}\,\underbrace{p(x_3|x_2)}_{f_3}\,\underbrace{p(x_4|x_2)}_{f_4}$$

Factor graph

- circle nodes represent random variables
- square nodes represent factorizing functions
- function node ***f*** connects varible node ***x*** if and only if ***x*** is one of argument of ***f***



$$p(x_2) = \underbrace{\sum_{x_1} \overbrace{p(x_1)}^{m_{x_1 \to f_2}(x_2)} p(x_2|x_1)}_{m_{f_2 \to x_2}(x_2)} \underbrace{\sum_{x_3} p(x_3|x_2)}_{m_{f_3 \to x_2}(x_2)} \underbrace{\sum_{x_4} p(x_4|x_2)}_{m_{f_4 \to x_2}(x_2)}$$

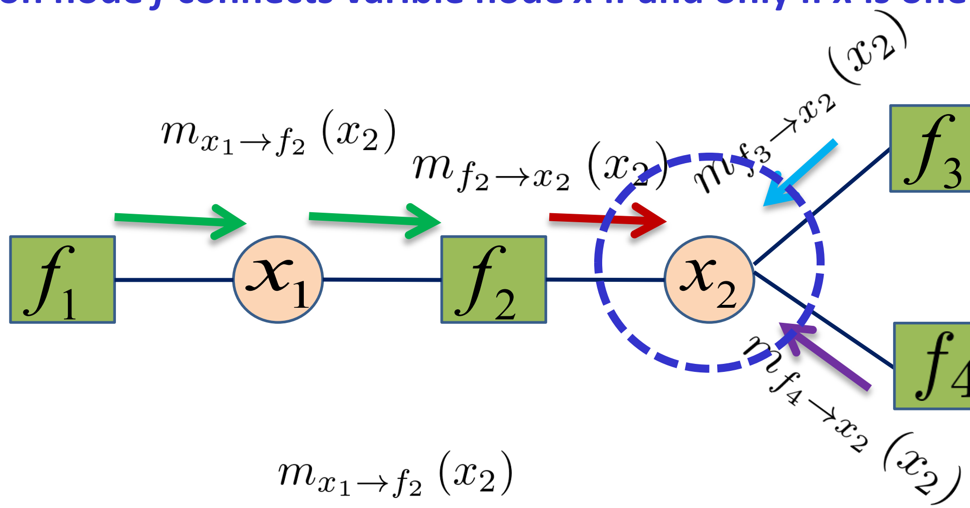
Belief Propagation

- **A Toy Example**

$$p(x_1, x_2, x_3, x_4) = \underset{f_1}{p(x_1)}\, \underset{f_2}{p(x_2|x_1)}\, \underset{f_3}{p(x_3|x_2)}\, \underset{f_4}{p(x_4|x_2)}$$

Factor graph

- **circle nodes represent random variables**
- **square nodes represent factorizing functions**
- **function node *f* connects varible node *x* if and only if *x* is one of argument of *f***



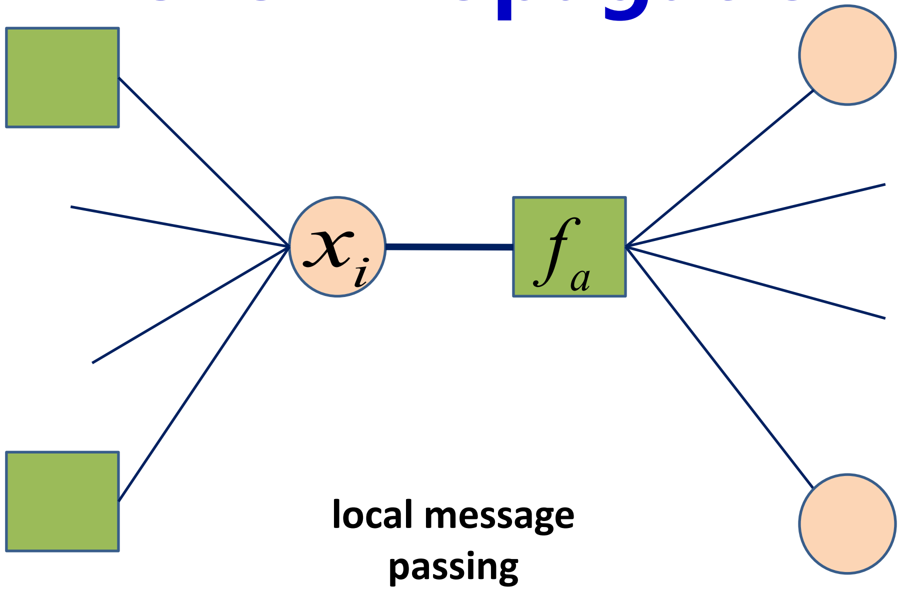
$$p(x_2) = \underbrace{\sum_{x_1} \overbrace{p(x_1)}^{m_{x_1\to f_2}(x_2)}\, p(x_2|x_1)}_{m_{f_2\to x_2}(x_2)}\, \underbrace{\sum_{x_3} p(x_3|x_2)}_{m_{f_3\to x_2}(x_2)}\, \underbrace{\sum_{x_4} p(x_4|x_2)}_{m_{f_4\to x_2}(x_2)}$$

product of all incoming messages

Belief Propagation

- Factor Graph

BP on general graph



local message passing

Loopy Blief Propagation (LBP)

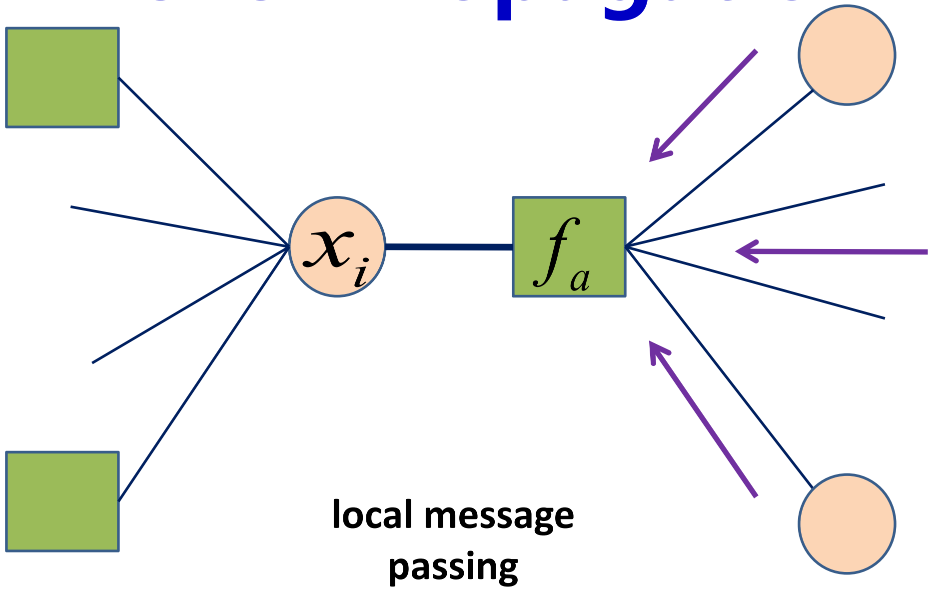
Factor to variable

Variable to factor

Belief Propagation

- Factor Graph

BP on
general graph



Loopy Blief Propagation (LBP)

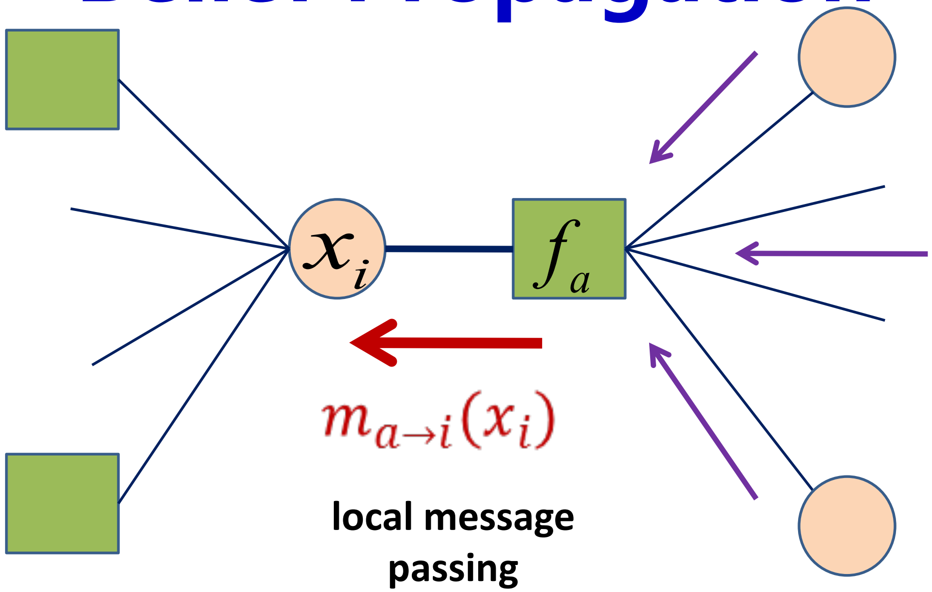
Factor to variable

Variable to factor

Belief Propagation

- Factor Graph

BP on
general graph



Loopy Blief Propagation (LBP)

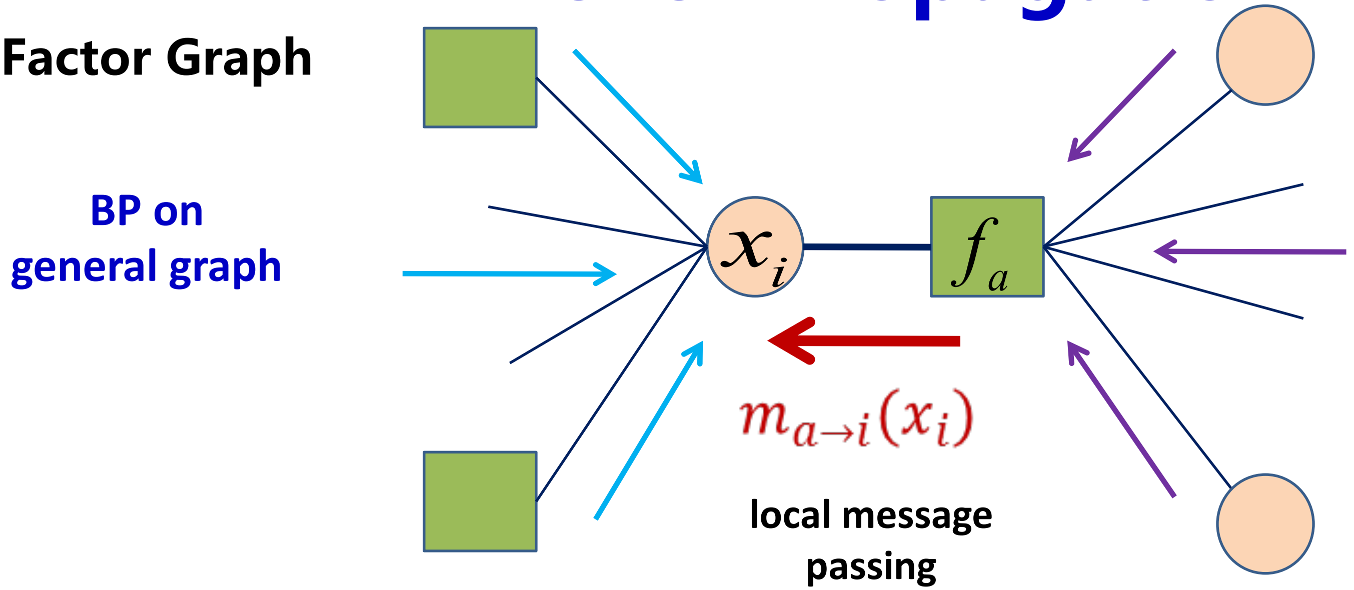
Factor to variable

$$m_{a\to i}(x_i) = \sum_{x_j, j\neq i} f_a(\mathbf{x}_a) \prod_{j\neq i} m_{j\to a}(x_j)$$

Variable to factor

Belief Propagation

- Factor Graph



Loopy Blief Propagation (LBP)

Factor to variable

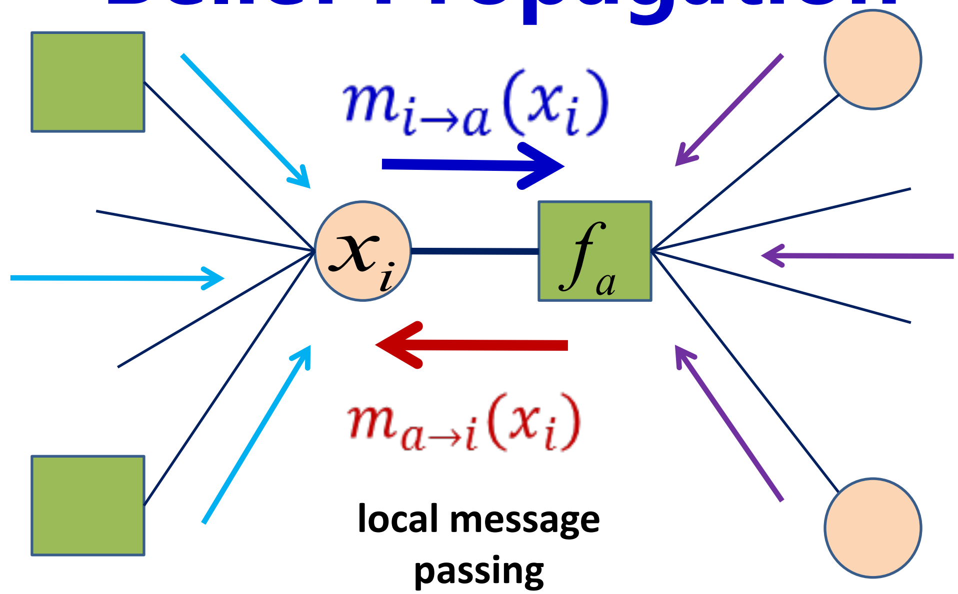
$$m_{a\to i}(x_i) = \sum_{x_j, j\neq i} f_a(\mathbf{x}_a) \prod_{j\neq i} m_{j\to a}(x_j)$$

Variable to factor

Belief Propagation

- Factor Graph

BP on general graph



local message passing

Loopy Blief Propagation (LBP)

Factor to variable
$$m_{a\to i}(x_i) = \sum_{x_j, j\neq i} f_a(\mathbf{x}_a) \prod_{j\neq i} m_{j\to a}(x_j)$$

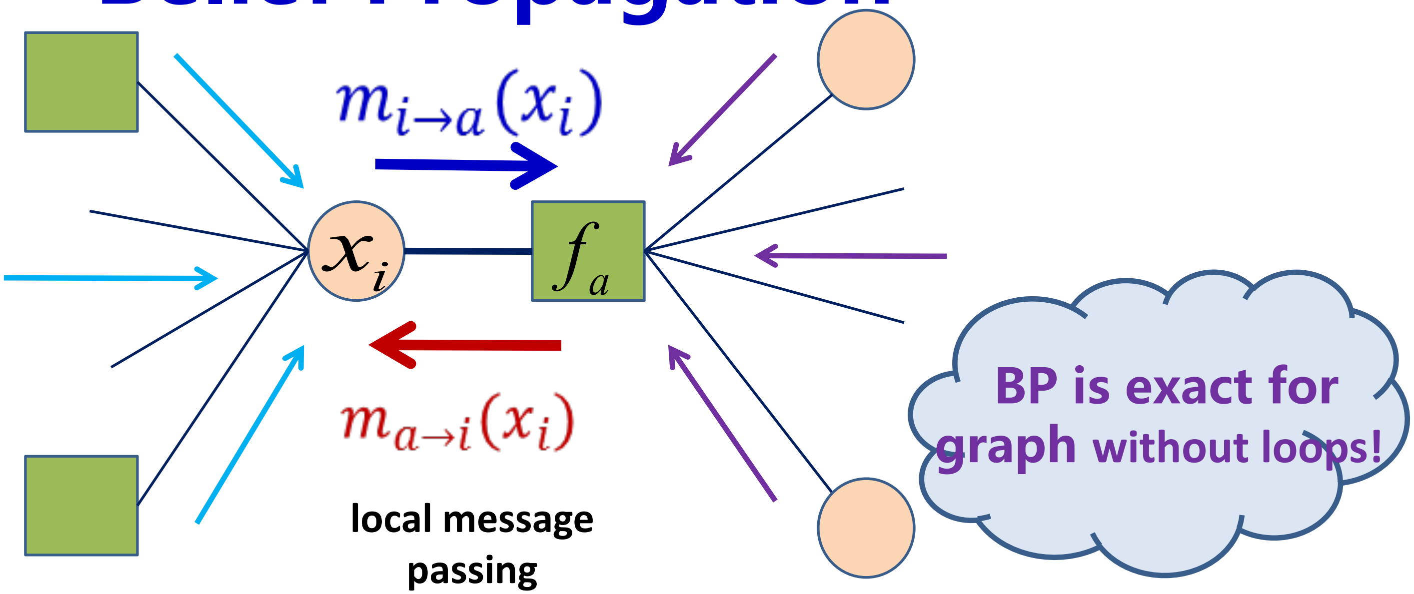
Variable to factor
$$m_{i\to a}(x_i) = \prod_{b\neq a} m_{b\to i}(x_i)$$

Excluding incomming message itself

Belief Propagation

- Factor Graph

BP on general graph



Loopy Blief Propagation (LBP)

Factor to variable

$$m_{a\to i}(x_i) = \sum_{x_j, j\neq i} f_a(\mathbf{x}_a) \prod_{j\neq i} m_{j\to a}(x_j)$$

Iterations

(graph with loops)

Variable to factor

$$m_{i\to a}(x_i) = \prod_{b\neq a} m_{b\to i}(x_i)$$

Excluding incomming message itself

Belief Propagation

figure copied from http://computerrobotvision.org/2009/tutorial_day/crv09_belief_propagation_v2.pdf

Message passing is a beautiful algorithmic framework to tackle difficult problems using divide and conquer by local compuation and information sharing

Parametric Approximation

Parameterization

$$\max \textbf{ELBO} = \sum_{\mathrm{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) - \sum_{\mathrm{x}} q(\mathbf{x}) \log q(\mathbf{x})$$

Parameterization $q(\mathbf{x}) \equiv q(\mathbf{x}; \boldsymbol{\phi})$

- **Exponential family** $q(\mathbf{x}; \boldsymbol{\phi}) = \mathbf{exp}\{\langle \boldsymbol{\phi}, \boldsymbol{\eta}(\mathbf{x})\rangle - A(\boldsymbol{\phi})\}$ **E.g., Gaussian, Bernoulli, exponential…**
- **Deep nueral network, such as that used in vatiational auto-encoder (VAE)**

Parametric Approximation

☐ Parameterization

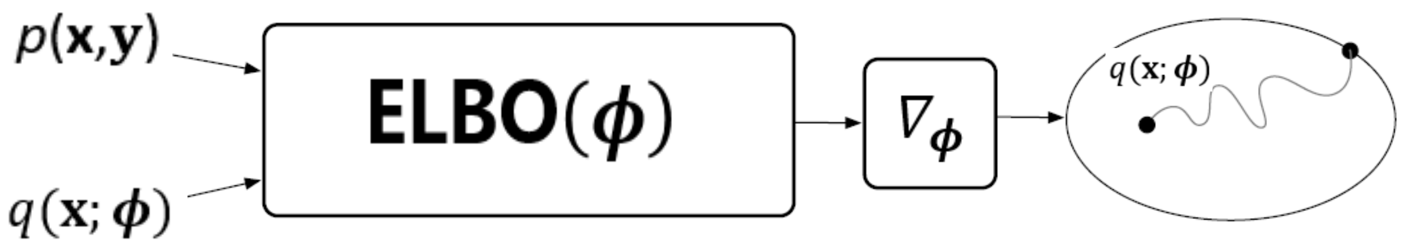
$$\max \mathbf{ELBO} = \sum_{\mathrm{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) - \sum_{\mathrm{x}} q(\mathbf{x}) \log q(\mathbf{x})$$

Parameterization $q(\mathbf{x}) \equiv q(\mathbf{x}; \boldsymbol{\phi})$

- **Exponential family** $q(\mathbf{x}; \boldsymbol{\phi}) = \exp\{\langle \boldsymbol{\phi}, \boldsymbol{\eta}(\mathbf{x}) \rangle - A(\boldsymbol{\phi})\}$ **E.g., Gaussian, Bernoulli, exponential…**
- **Deep nueral network, such as that used in vatiational auto-encoder (VAE)**

New Optimization Obejective

$$\max_{\boldsymbol{\phi}} \mathbf{ELBO}(\boldsymbol{\phi}) = \sum_{\mathrm{x}} q(\mathbf{x}; \boldsymbol{\phi}) \log p(\mathbf{x},\mathbf{y}) - \sum_{\mathrm{x}} q(\mathbf{x}; \boldsymbol{\phi}) \log q(\mathbf{x}; \boldsymbol{\phi})$$



Stochastic variational inference framework

The variational parameters are optimzied usign SGD

[Hoffman et al 2013]

Parametric Approximation

□ Parameterization

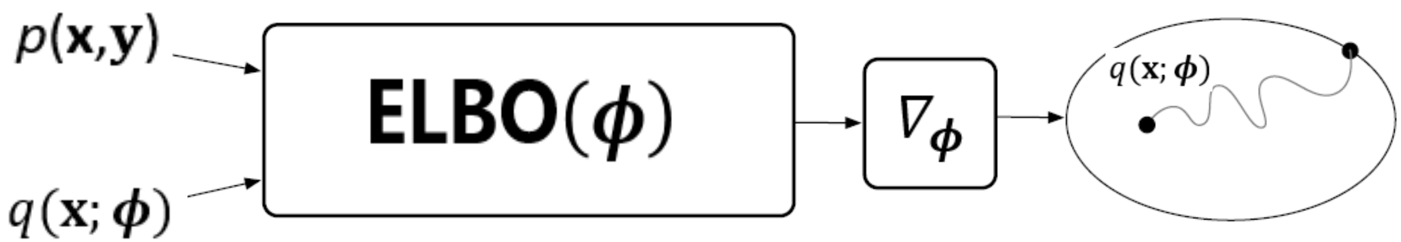
$$\max \textbf{ELBO} = \sum_{\mathrm{x}} q(\mathbf{x}) \log p(\mathbf{x},\mathbf{y}) - \sum_{\mathrm{x}} q(\mathbf{x}) \log q(\mathbf{x})$$

Parameterization $q(\mathbf{x}) \equiv q(\mathbf{x};\boldsymbol{\phi})$

- **Exponential family** $q(\mathbf{x};\boldsymbol{\phi}) = \mathbf{exp}\{\langle \boldsymbol{\phi}, \boldsymbol{\eta}(\mathbf{x})\rangle - A(\boldsymbol{\phi})\}$ **E.g., Gaussian, Bernoulli, exponential...**
- **Deep nueral network, such as that used in vatiational auto-encoder (VAE)**

New Optimization Obejective

$$\max_{\boldsymbol{\phi}} \textbf{ELBO}(\boldsymbol{\phi}) = \sum_{\mathrm{x}} q(\mathbf{x};\boldsymbol{\phi}) \log p(\mathbf{x},\mathbf{y}) - \sum_{\mathrm{x}} q(\mathbf{x};\boldsymbol{\phi}) \log q(\mathbf{x};\boldsymbol{\phi})$$



Stochastic variational inference framework

The variational parameters are optimzied usign SGD

The original integration problem boilds down to derivative problem

Outline

- Background
- Variational Inference
- **Expectation Propagation**
- A Unified EP Perspective on AMP and its extensions
- Conclusion

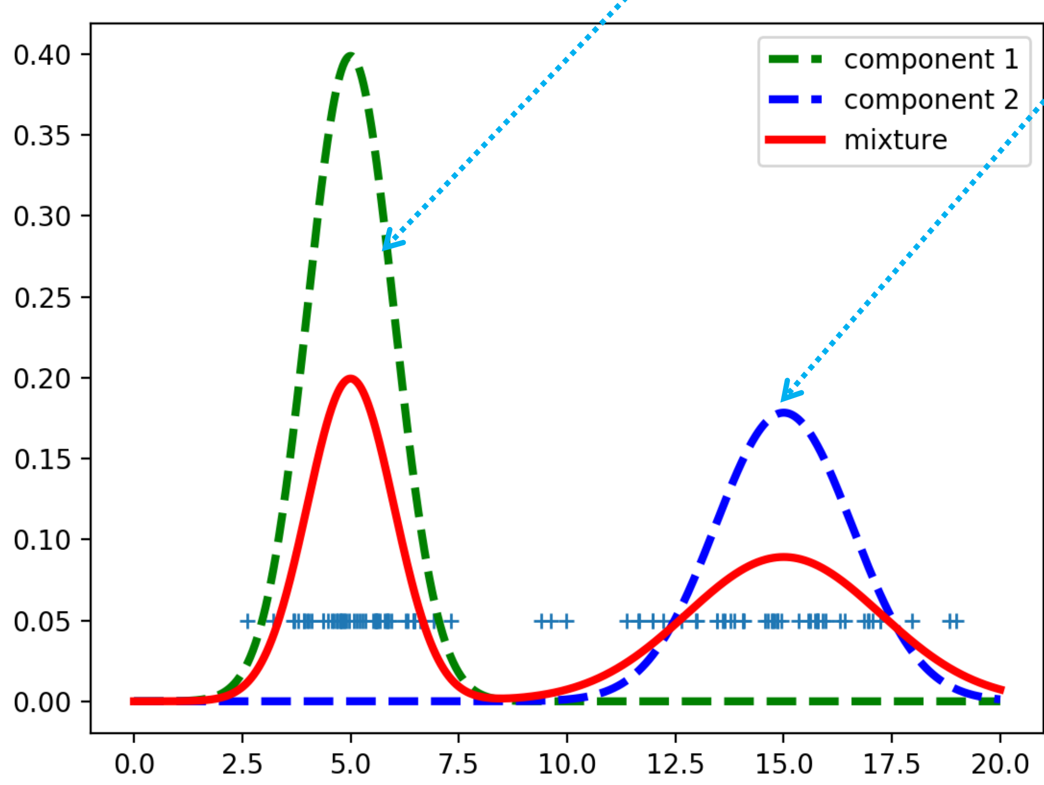
A Toy Problem

- **Problem Statement**

  we obtain a sequence of data points $y_i, i = 1 \ldots N$

$$p(y_i|x) = \underbrace{0.5\mathcal{N}(y_i; x, 1)}_{\text{component one}} + \underbrace{0.5\mathcal{N}(y_i; x+10, 5)}_{\text{component two}}$$

PDF



What is the value of x?

This example is modified from example in [Minka01b]

A Toy Problem

Probabilistic Modeling

- **prior distribution** $p(x) = \mathcal{N}(x; 0, 100)$ **Guassian prior**
- **likelihood distribution** $p(y_i|x) = 0.5\mathcal{N}(y_i; x, 1) + 0.5\mathcal{N}(y_i; x+10, 5)$

After obtaining *N* observations, the joint distribution could be written as

$$p(x, \mathbf{y}) = p(x) \prod_{i=1}^{N} p(y_i|x)$$

- **posterior distribution**

$$p(x|\mathbf{y}) = \frac{p(x) \prod_{i=1}^{N} p(y_i|x)}{p(\mathbf{y})}$$

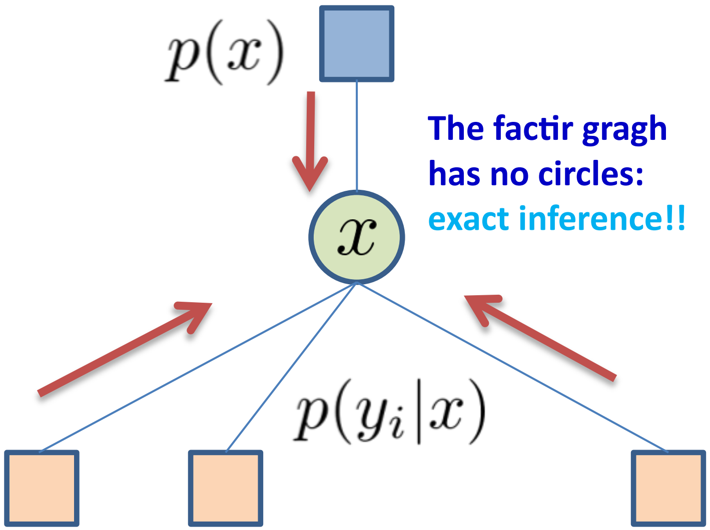
We could perform Bayesian inference to compute the posterior distribution

All the codes for this toy example are available:
https://github.com/mengxiangming/ep-demo

A Toy Problem

- Factor Graph and Belief Propagation

$$p(x, \mathbf{y}) = p(x) \prod_{i=1}^{N} p(y_i|x)$$



The factir gragh has no circles: exact inference!!

Belief Propagation

factor to vaiable: $m_{i \to x}(x) = p(y_i|x)$

vaiable update: $q(x) = p(x) \prod_{i=1}^{N} m_{i \to x}(x)$

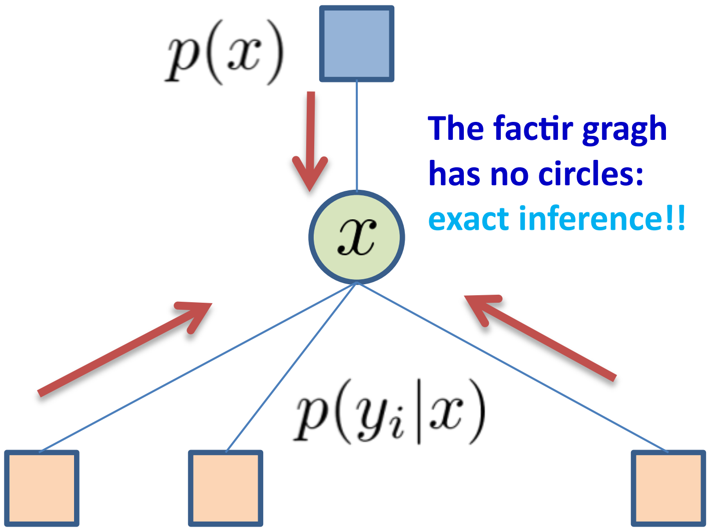
Already Done?

$$p(y_i|x) = 0.5\mathcal{N}(y_i; x, 1) + 0.5\mathcal{N}(y_i; x + 10, 5) \qquad p(x) = \mathcal{N}(x; 0, 100)$$

A Toy Problem

- **Factor Graph and Belief Propagation**

$$p(x, \mathbf{y}) = p(x) \prod_{i=1}^{N} p(y_i|x)$$



The factir gragh has no circles: exact inference!!

Belief Propagation

factor to vaiable: $m_{i \to x}(x) = p(y_i|x)$

vaiable update: $q(x) = p(x) \prod_{i=1}^{N} m_{i \to x}(x)$

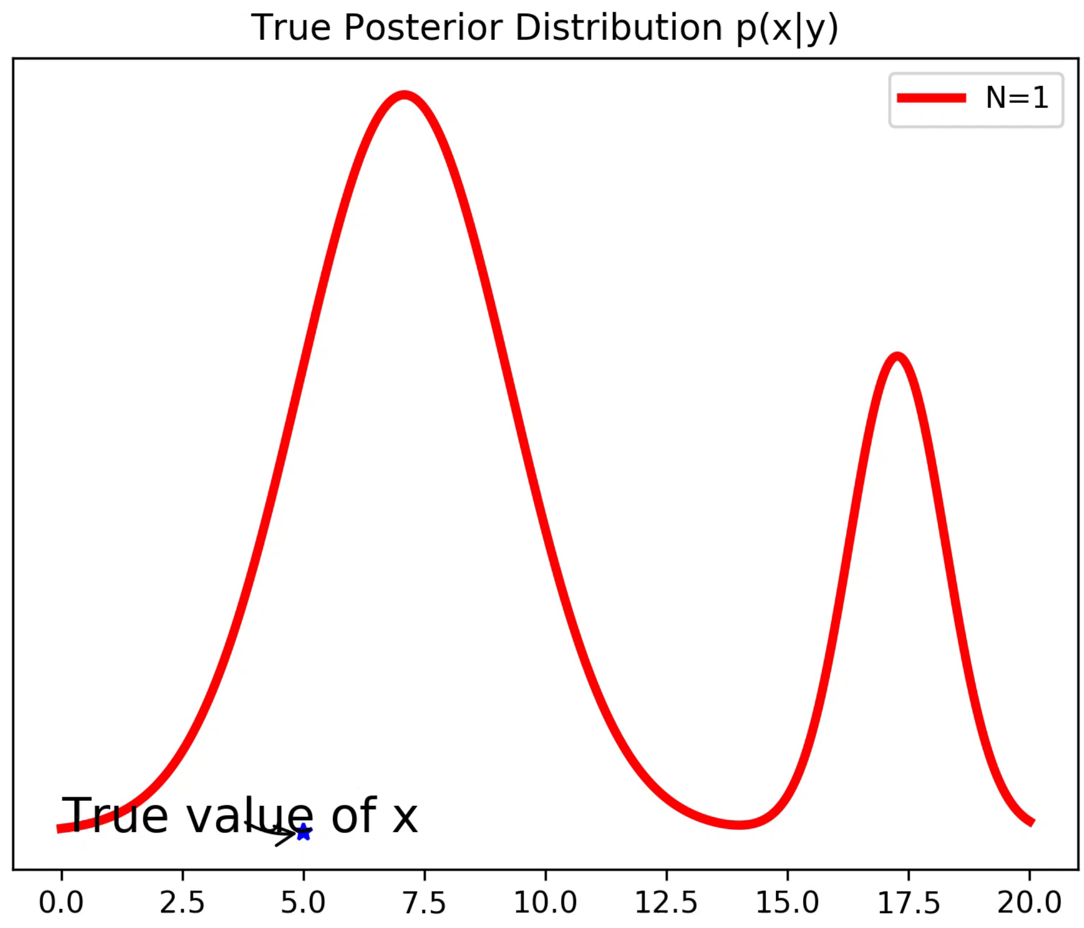
Already Done?

$$p(y_i|x) = 0.5\mathcal{N}(y_i; x, 1) + 0.5\mathcal{N}(y_i; x + 10, 5) \qquad p(x) = \mathcal{N}(x; 0, 100)$$

- **The posterior distribution is a mixture of 2^N Gaussians.**
- **The computational complexity is exponential with *N***

A Toy Problem

- The True Posterior



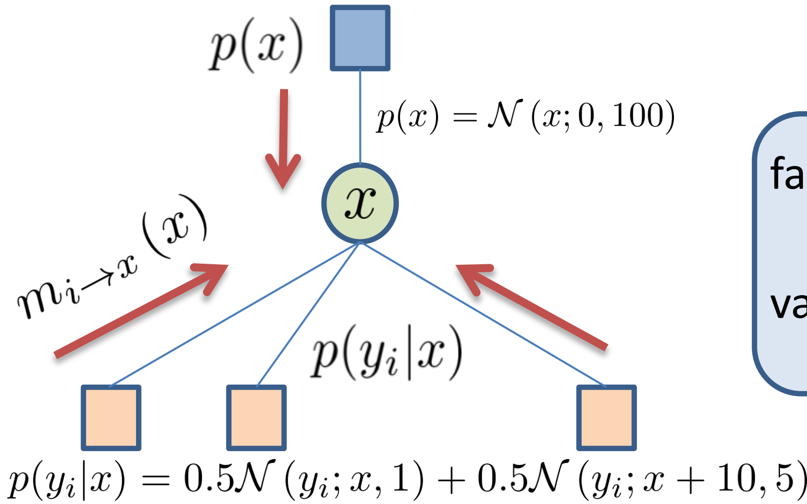
A Toy Problem

- The True Posterior

Approximating the posterior as one Gaussian distribution

A Toy Problem

A Naive Approximation



Approximating each BP message itself as Gaussian distribution independently

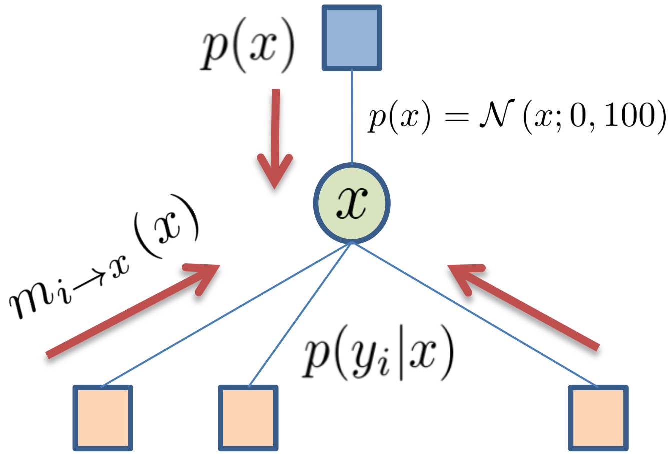
factor to vaiable: $m_{i \to x}(x) \approx \mathcal{N}(x; m_i, v_i)$

vaiable update: $q(x) \approx p(x) \prod_{i=1}^{N} \mathcal{N}(x; m_i, v_i)$

Naive Gaussian Message Approximation

A Toy Problem

A Naive Approximation



$p(y_i|x) = 0.5\mathcal{N}(y_i; x, 1) + 0.5\mathcal{N}(y_i; x+10, 5)$

Approximating each BP message itself as Gaussian distribution independently

factor to vaiable: $m_{i \to x}(x) \approx \mathcal{N}(x; m_i, v_i)$

vaiable update: $q(x) \approx p(x) \prod_{i=1}^{N} \mathcal{N}(x; m_i, v_i)$

Naive Gaussian Message Approximation

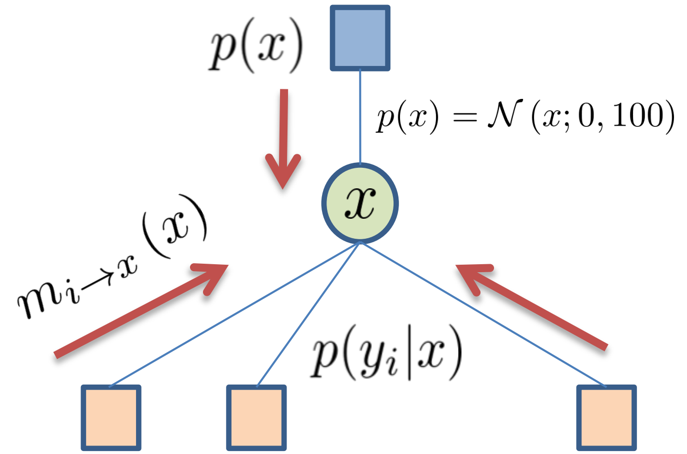
True Posterior $$p(x|\mathbf{y}) \propto p(x) \prod_{i=1}^{N} p(y_i|x)$$

Approximate $$p(x|\mathbf{y}) \propto p(x) \prod_{i=1}^{N} \mathcal{N}(x; m_i, v_i)$$

Each non-Gaussian likelihood is approximated as a Gaussian factor

A Toy Problem

☐ A Naive Approximation



$p(y_i|x) = 0.5\mathcal{N}(y_i; x, 1) + 0.5\mathcal{N}(y_i; x+10, 5)$

Approximating each BP message itself as Gaussian distribution independently

factor to vaiable: $m_{i\to x}(x) \approx \mathcal{N}(x; m_i, v_i)$

vaiable update: $q(x) \approx p(x) \prod_{i=1}^{N} \mathcal{N}(x; m_i, v_i)$

Naive Gaussian Message Approximation

True Posterior $$p(x|\mathbf{y}) \propto p(x) \prod_{i=1}^{N} p(y_i|x)$$

Approximate $$p(x|\mathbf{y}) \propto p(x) \prod_{i=1}^{N} \mathcal{N}(x; m_i, v_i)$$

Each non-Gaussian likelihood is approximated as a Gaussian factor

The posterior will be also Gaussian due to the product rule of Gaussian

$$\mathcal{N}(x; m, v) \propto \mathcal{N}(x; m_1, v_1)\,\mathcal{N}(x; m_2, v_2)$$

$$\frac{1}{v} = \frac{1}{v_1} + \frac{1}{v_2} \qquad \frac{m}{v} = \frac{m_1}{v_1} + \frac{m_2}{v_2}$$

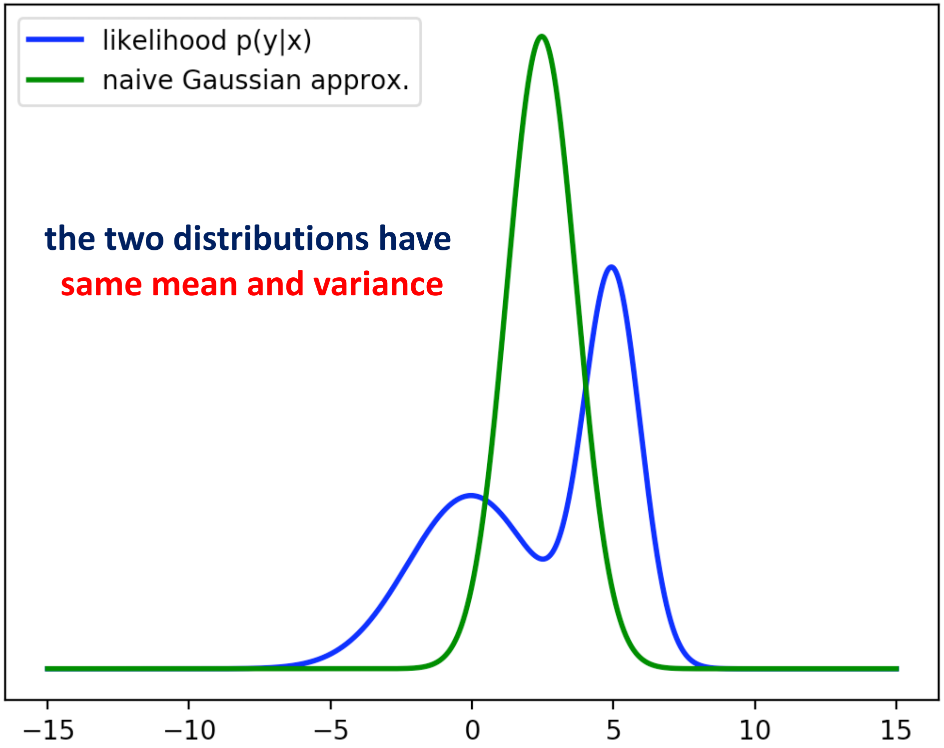
A Toy Problem

A Naive Approximation

For each non-Gaussian message

$$p(y_i|x) = 0.5\mathcal{N}(y_i; x, 1) + 0.5\mathcal{N}(y_i; x + 10, 5)$$

Gaussian Approximation $$\tilde{t}_i(x) \triangleq \mathrm{Proj}\left[p(y_i|x)\right]$$ Gassian Projection Operator



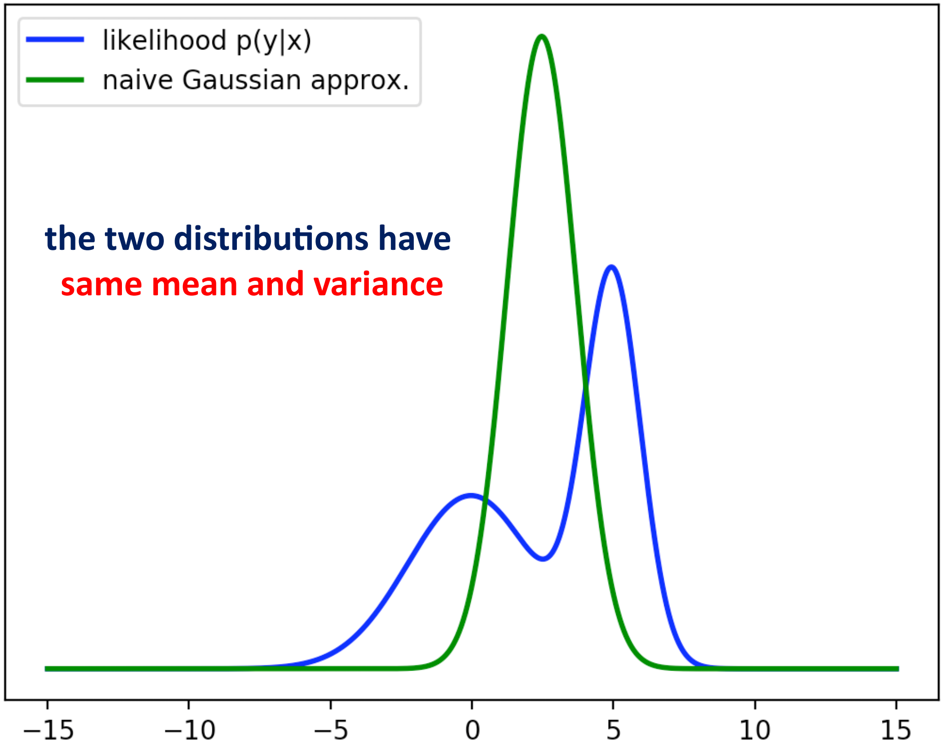
A Toy Problem

☐ A Naive Approximation

For each non-Gaussian message

$$p(y_i|x) = 0.5\mathcal{N}(y_i; x, 1) + 0.5\mathcal{N}(y_i; x+10, 5)$$

Gaussian Approximation $\tilde{t}_i(x) \triangleq \mathrm{Proj}\left[p(y_i|x)\right]$ Gassian Projection Operator



$$q(x) = \underset{q \in \mathrm{Gaussian}}{\arg\max}\ KL(p(x)\,||\,q(x))$$

equivalent

$$q(x) \triangleq \mathrm{Proj}\left[p(x)\right] = \mathcal{N}(x; m, v) \qquad m = \mathtt{E}_{p(x)}(x) \qquad v = \mathtt{Var}_{p(x)}(x)$$

Moment Matching

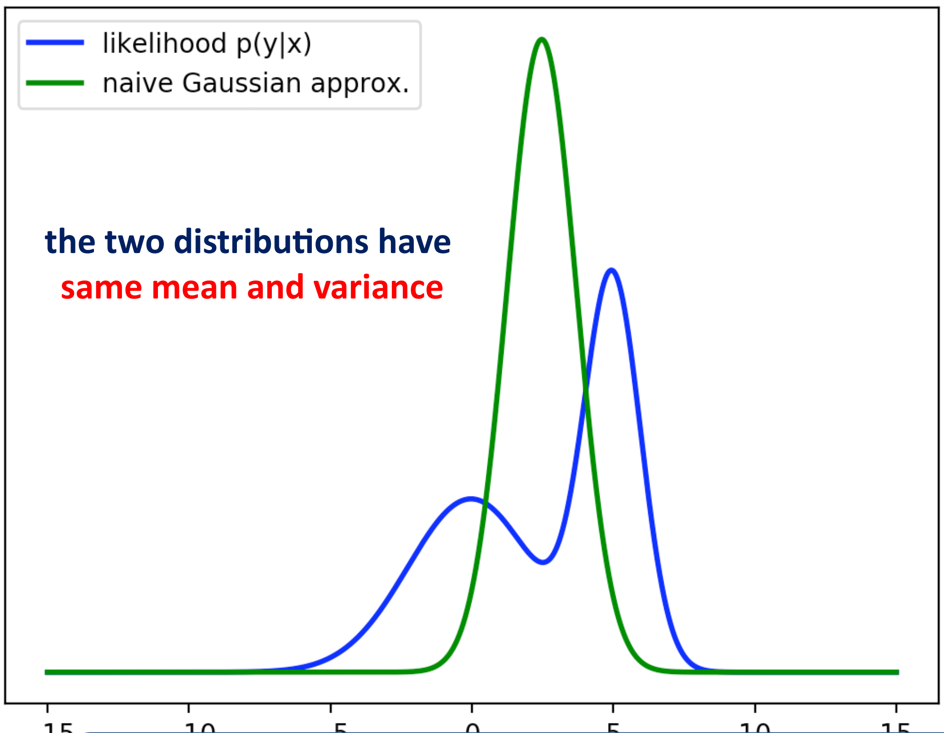
A Toy Problem

A Naive Approximation

For each non-Gaussian message

$$p(y_i|x) = 0.5\mathcal{N}\left(y_i; x, 1\right) + 0.5\mathcal{N}\left(y_i; x+10, 5\right)$$

Gaussian Approximation $\tilde{t}_i\left(x\right) \triangleq \mathrm{Proj}\left[p\left(y_i|x\right)\right]$ Gassian Projection Operator



$$q\left(x\right) = \underset{q \in \mathrm{Gaussian}}{\arg\max}\ KL(p\left(x\right)||q\left(x\right))$$

equivalent

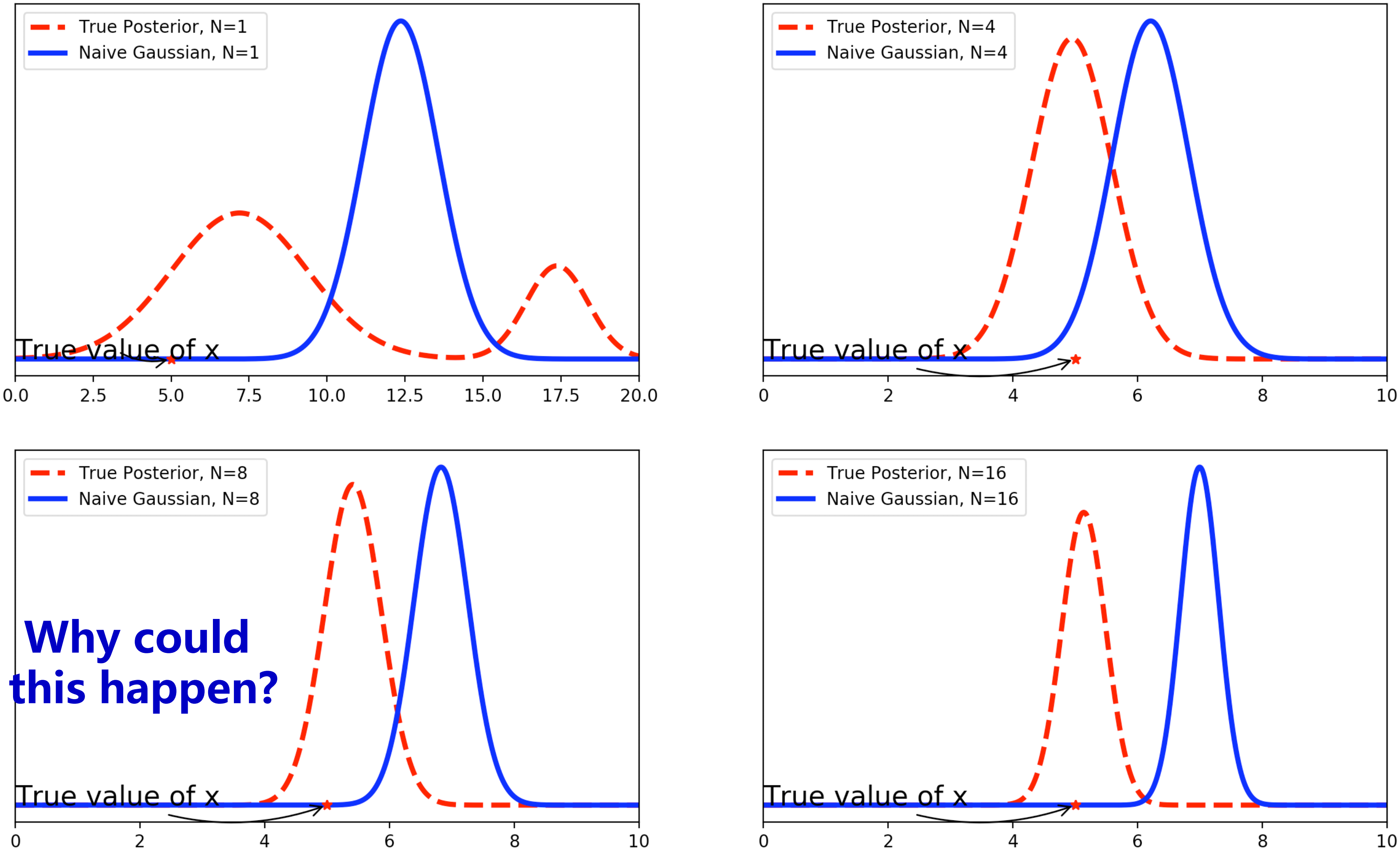
$$q\left(x\right) \triangleq \mathrm{Proj}\left[p\left(x\right)\right] \qquad m = \mathtt{E}_{p(x)}\left(x\right)$$
$$= \mathcal{N}\left(x; m, v\right) \qquad v = \mathtt{Var}_{p(x)}\left(x\right)$$

Moment Matching

Then, how will the posterior be like?

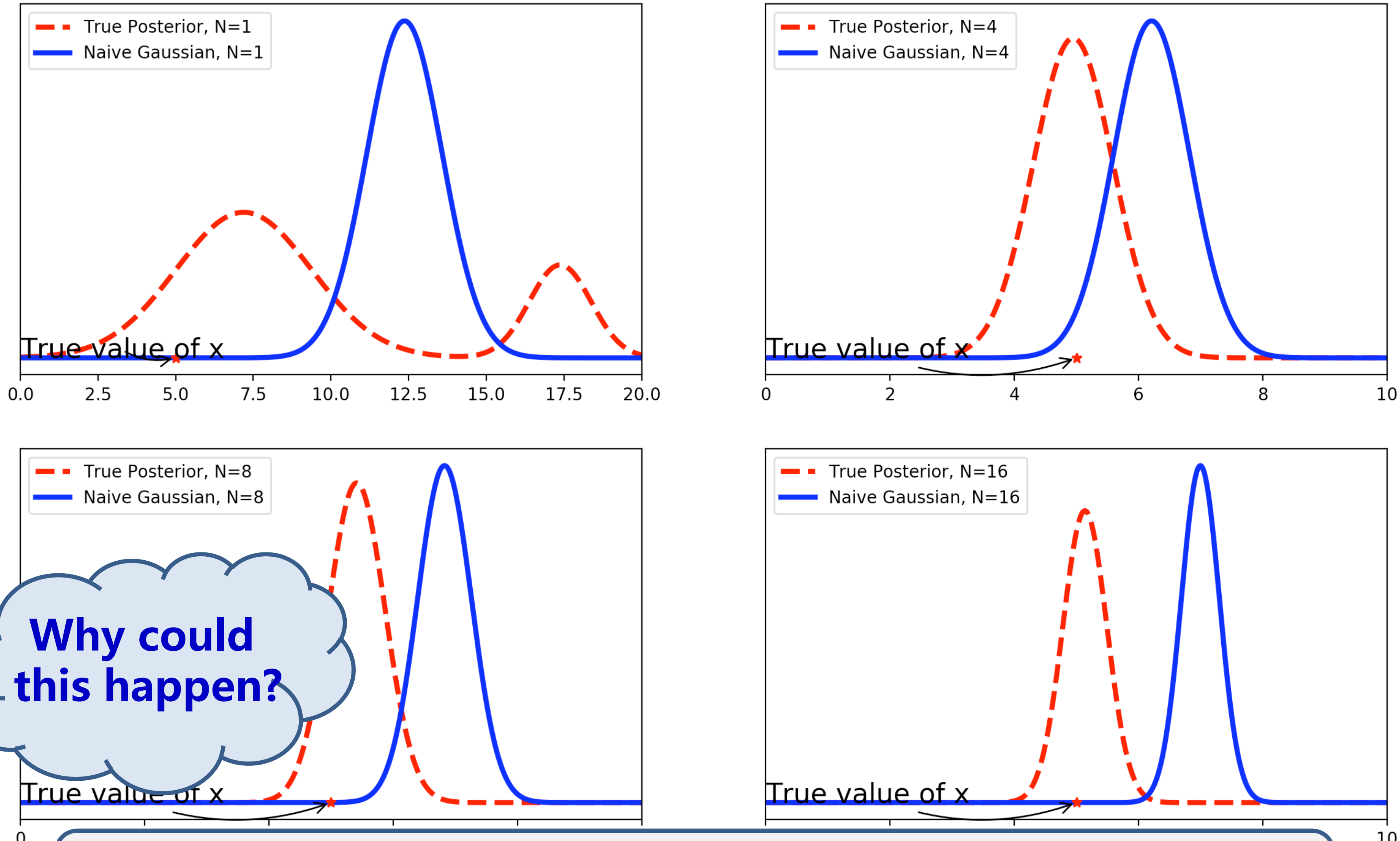
A Toy Problem

□ A Naive Approximation



A Toy Problem

☐ A Naive Approximation



Why could this happen?

There is still a big discrepancy between the true posterior and naive Gaussian approximation, even when the true postrior approaches Gaussian!

A Toy Problem

- A Naive Approximation

Because it is ~~naive~~ selfish

A Toy Problem

- A Naive Approximation

Because it is ~~naive~~ selfish

Each factor (message) only cares about itself when making approximations
while forgetting the ultimate goal is to make a good approximation to the global posterior

A Toy Problem

An Alternative Gaussian Approximation

Consider the simple case of N = 1 (only one observation)

True posterior $$p(x|\mathbf{y}) \propto \underbrace{p(x)}_{\textbf{Gauss}}\underbrace{p(y_1|x)}_{\textbf{Non-Gauss}}$$

- **Step 1: Approximating the product** $p(x)p(y_1|x)$ **as Gaussian** $\mathrm{Proj}\left[p(x)p(y_1|x)\right]$

A Toy Problem

An Alternative Gaussian Approximation

Consider the simple case of N = 1 (only one observation)

True posterior $$p(x|\mathbf{y}) \propto \underbrace{p(x)}_{\text{Gauss}}\underbrace{p(y_1|x)}_{\text{Non-Gauss}}$$

- **Step 1: Approximating the product** $p(x)p(y_1|x)$ **as Gaussian** $\mathrm{Proj}\left[p(x)p(y_1|x)\right]$
- **Step 2: Divide the Gaussian** $\mathrm{Proj}\left[p(x)p(y_1|x)\right]$ **by** $p(x)$ **to obtain a Gaussian**

$$\tilde{t}_1(x) = \frac{\mathrm{Proj}\left[p(x)\,p(y_1|x)\right]}{p(x)}$$

taking care of $p(x)$ when approximating $p(y_1|x)$

A Toy Problem

An Alternative Gaussian Approximation

Consider the simple case of N = 1 (only one observation)

True posterior $p(x|\mathbf{y}) \propto \underbrace{p(x)}_{\text{Gauss}}\underbrace{p(y_1|x)}_{\text{Non-Gauss}}$

- **Step 1: Approximating the product** $p(x)p(y_1|x)$ **as Gaussian** $\mathrm{Proj}\left[p(x)p(y_1|x)\right]$
- **Step 2: Divide the Gaussian** $\mathrm{Proj}\left[p(x)p(y_1|x)\right]$ **by** $p(x)$ **to obtain a Gaussian**

$$\tilde{t}_1(x) = \frac{\mathrm{Proj}\left[p(x)\,p(y_1|x)\right]}{p(x)}$$

taking care of $p(x)$ when approximating $p(y_1|x)$

$$q(x|y_1) \propto p(x)\,\tilde{t}_1(x)$$

Posterior Gauss approximation

A Toy Problem

◻ Assumed Density Filtering (ADF)

Consider general case of N obersvations

True posterior $p(x|\mathbf{y}) \propto p(x)p(y_1|x)p\left(y_2|x\right)p\left(y_3|x\right)\cdots p\left(y_N|x\right)$

Approximate posterior $q\left(x|\mathbf{y}\right) \propto p\left(x\right)\tilde{t}_1\left(x\right)\tilde{t}_2\left(x\right)\tilde{t}_3\left(x\right)\cdots\tilde{t}_N\left(x\right)$

A Toy Problem

☐ Assumed Density Filtering (ADF)

Consider general case of N obersvations

True posterior

$$p(x|\mathbf{y}) \propto p(x)p(y_1|x)p\left(y_2|x\right)p\left(y_3|x\right)\cdots p\left(y_N|x\right)$$

Approximate posterior

$$q\left(x|\mathbf{y}\right) \propto p\left(x\right)\tilde{t}_1\left(x\right)\tilde{t}_2\left(x\right)\tilde{t}_3\left(x\right)\cdots\tilde{t}_N\left(x\right)$$

ADF Algorithm

- Initialize $q^0\left(x\right) = p\left(x\right)$
- For each new observation y_i

Inclusion $$\hat{p}\left(x\right) = \frac{q^{i-1}\left(x\right)p\left(y_i|x\right)}{\int q^{i-1}\left(x\right)p\left(y_i|x\right)dx}$$

Projection $$q^i\left(x\right) = \mathrm{Proj}\left[\hat{p}\left(x\right)\right]$$

$$\tilde{t}_i\left(x\right) \propto \frac{q^i\left(x\right)}{q^{i-1}\left(x\right)}$$ only implicitly made

A Toy Problem

☐ Assumed Density Filtering (ADF)

Consider general case of N obersvations

True posterior $$p(x|\mathbf{y}) \propto p(x)p(y_1|x)p(y_2|x)\,p(y_3|x)\cdots p(y_N|x)$$

Approximate posterior $$q(x|\mathbf{y}) \propto p(x)\,\tilde{t}_1(x)\tilde{t}_2(x)\tilde{t}_3(x)\cdots\tilde{t}_N(x)$$

ADF Algorithm

- Initialize $q^0(x) = p(x)$
- For each new observation y_i

Inclusion $$\hat{p}(x) = \frac{q^{i-1}(x)\,p(y_i|x)}{\int q^{i-1}(x)\,p(y_i|x)\,dx}$$

Projection $$q^i(x) = \mathrm{Proj}\left[\hat{p}(x)\right]$$

$$\tilde{t}_i(x) \propto \frac{q^i(x)}{q^{i-1}(x)}$$ only implicitly made

ADF is one kind of sequential Gaussian Projector [Minka01b]

$$\overrightarrow{\quad q^{i-1}(x) \quad}$$

$$\tilde{t}_1(x) \quad \tilde{t}_2(x) \quad p(y_3|x) \quad p(y_4|x)\,p(y_5|x)$$

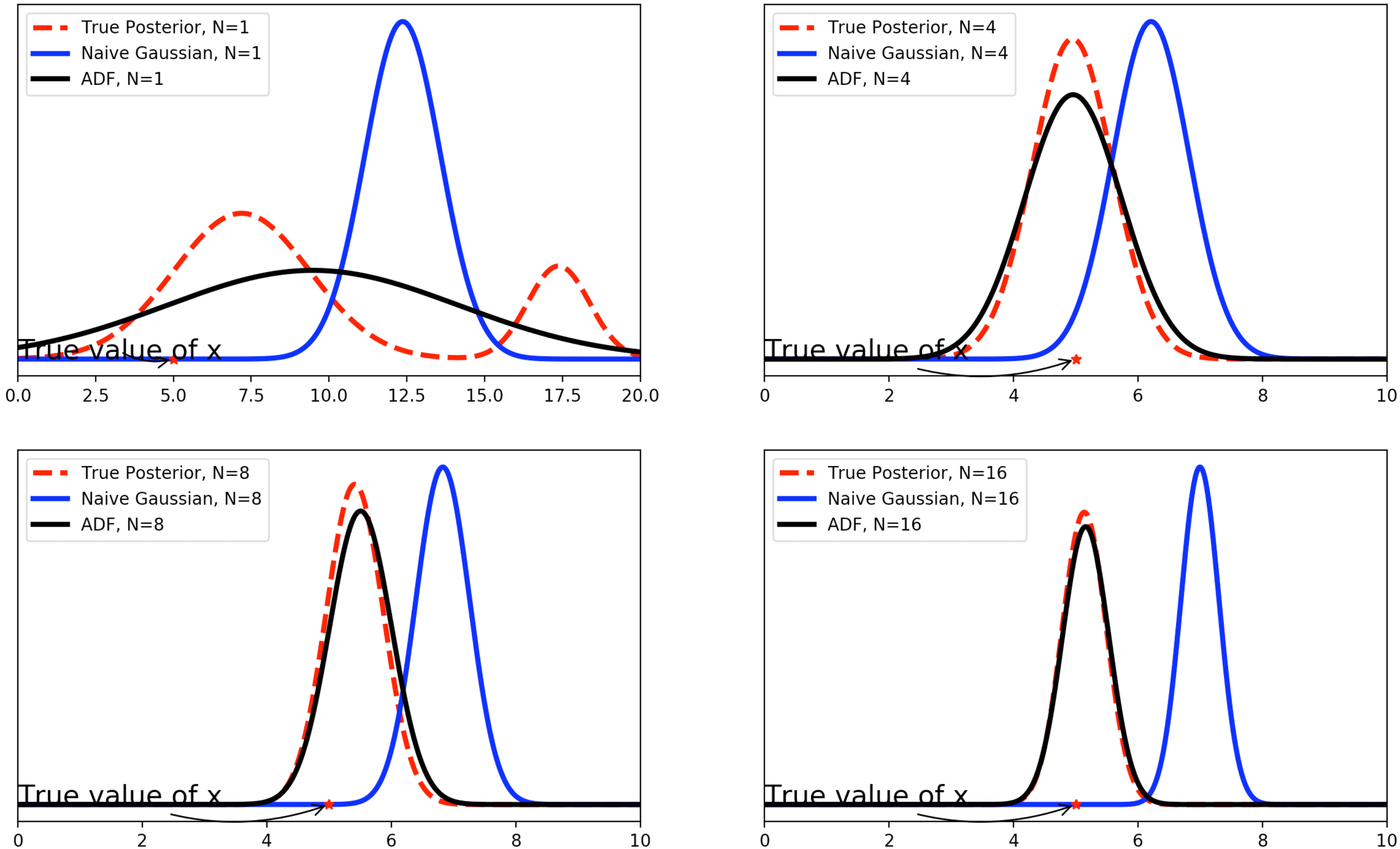
$$\downarrow$$

$$\tilde{t}_1(x) \quad \tilde{t}_2(x) \quad \tilde{t}_3(x) \quad p(y_4|x)\,p(y_5|x)$$

$$\overrightarrow{\quad q^{i}(x) \quad}$$

A Toy Problem

- **Assumed Density Filtering (ADF)**

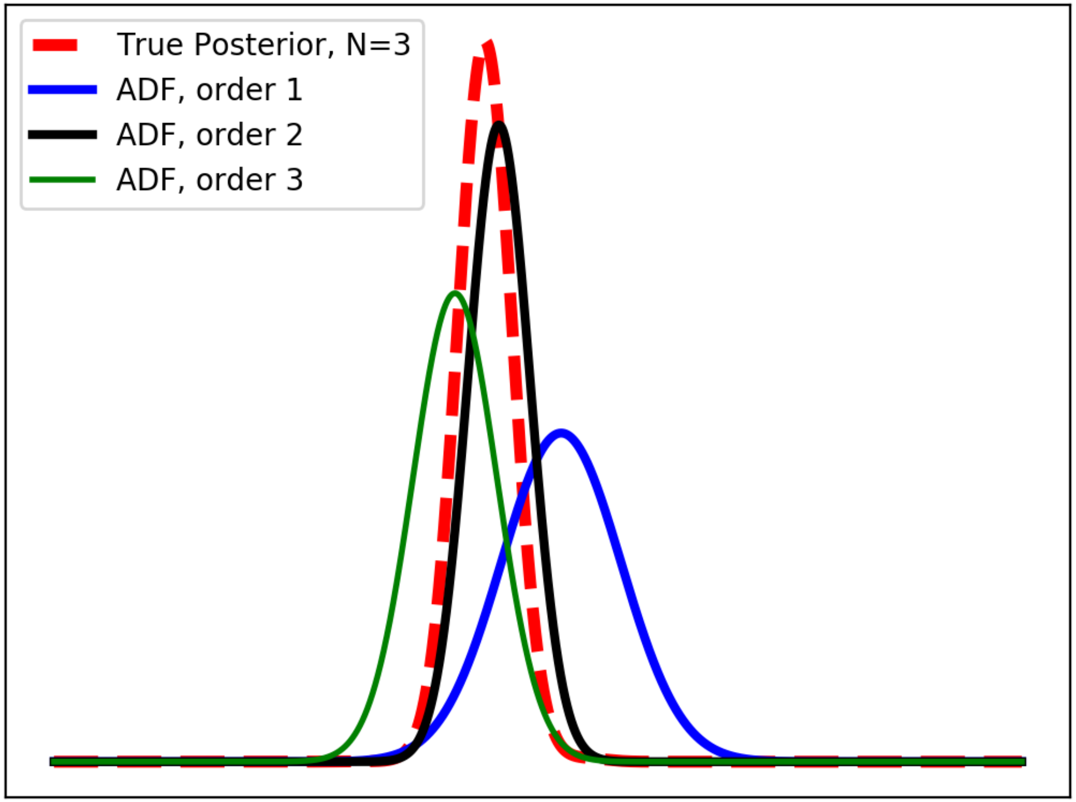


ADF is much better than naive Gauss approximation

A Toy Problem

- **Assumed Density Filtering (ADF)**

However, ADF is sensitive to the order of approximations !



How to avoid the effect of different ordering

A Toy Problem

- **Expectation Propagation**

  Expectation Propagation = ADF + Iteratively Refine

A Toy Problem

Expectation Propagation

Expectation Propagation = ADF + Iteratively Refine

True posterior $$p(x|\mathbf{y}) \propto p(x)p(y_1|x)p(y_2|x)\,p(y_3|x)\cdots p(y_N|x)$$

Approximate posterior $$q(x|\mathbf{y}) \propto p(x)\,\tilde{t}_1(x)\tilde{t}_2(x)\tilde{t}_3(x)\cdots\tilde{t}_N(x)$$

EP Algorithm

- Initialize $\tilde{t}_i(x), i = 1\ldots N, q(x) = p(x)\prod_i \tilde{t}_i(x)$
- For iter = 1… Num_iter

  For *i* in 1…*N*

division $$q^{\setminus i}(x) \propto \frac{q(x)}{\tilde{t}_i(x)} = p(x)\prod_{j\neq i}\tilde{t}_j(x)$$

inclusion $$\hat{p}(x) = \frac{q^{\setminus i}(x)\,p(y_i|x)}{\int q^{\setminus i}(x)\,p(y_i|x)dx}$$

projection $$q(x) = \mathrm{Proj}\left[\hat{p}(x)\right]$$

refinement $$\tilde{t}_i(x) \propto \frac{q(x)}{q^{\setminus i}(x)}$$

A Toy Problem

Expectation Propagation

Expectation Propagation = ADF + Iteratively Refine

True posterior $p(x|\mathbf{y}) \propto p(x)p(y_1|x)p\left(y_2|x\right)p\left(y_3|x\right)\cdots p\left(y_N|x\right)$

Approximate posterior $q\left(x|\mathbf{y}\right) \propto p\left(x\right)\tilde{t}_1\left(x\right)\tilde{t}_2\left(x\right)\tilde{t}_3\left(x\right)\cdots\tilde{t}_N\left(x\right)$

EP Algorithm

- Initialize $\tilde{t}_i\left(x\right), i = 1 \ldots N, q\left(x\right) = p\left(x\right)\prod_i \tilde{t}_i\left(x\right)$
- For iter = 1… Num_iter

  For *i* in 1…*N*

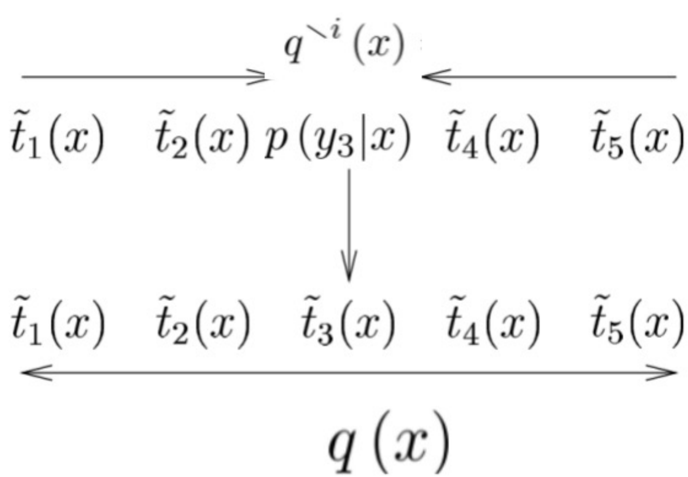
division $q^{\setminus i}\left(x\right) \propto \dfrac{q\left(x\right)}{\tilde{t}_i\left(x\right)} = p\left(x\right)\prod_{j \neq i}\tilde{t}_j\left(x\right)$

inclusion $\hat{p}\left(x\right) = \dfrac{q^{\setminus i}\left(x\right)p\left(y_i|x\right)}{\int q^{\setminus i}\left(x\right)p\left(y_i|x\right)dx}$

projection $q\left(x\right) = \mathrm{Proj}\left[\hat{p}\left(x\right)\right]$

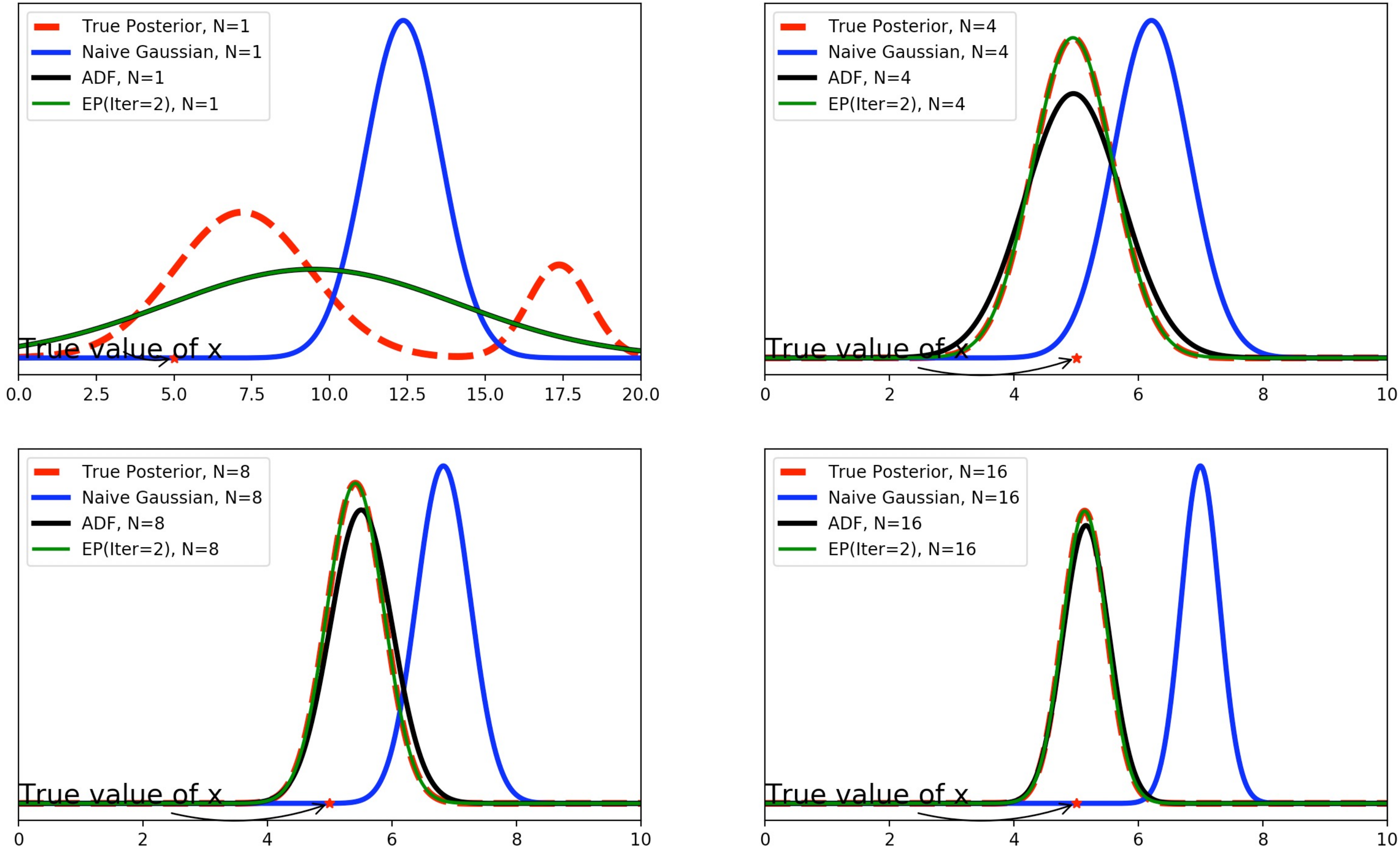
refinement $\tilde{t}_i\left(x\right) \propto \dfrac{q\left(x\right)}{q^{\setminus i}\left(x\right)}$

EP is an iterative refinement of ADF and is not affected by order[Minka01b]



A Toy Problem

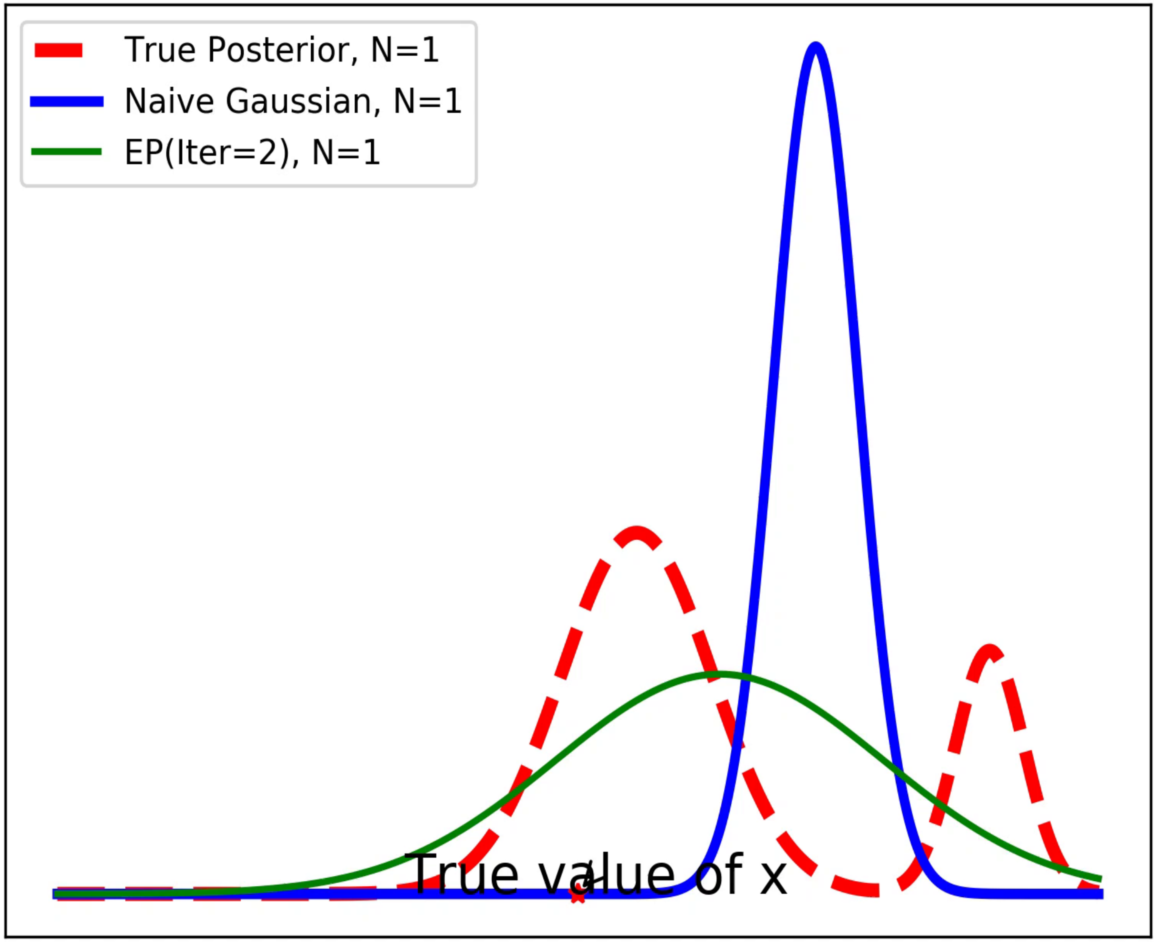
- **Expectation Propagation**



EP approximation is close to the true posterior !

A Toy Problem

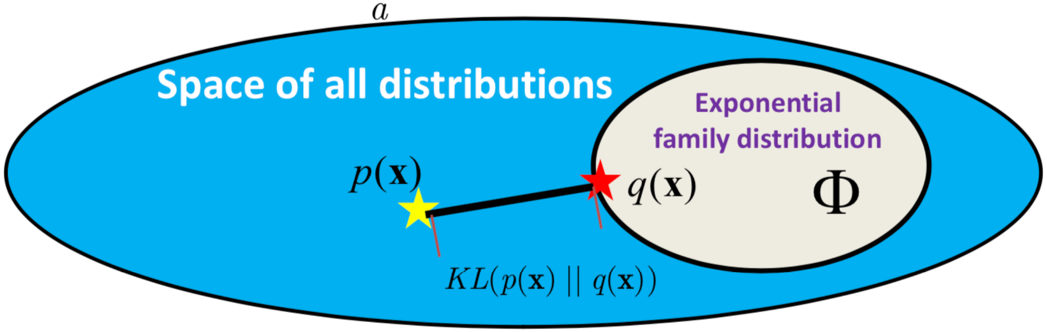
- **Expectation Propagation**



EP as Optimization

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

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



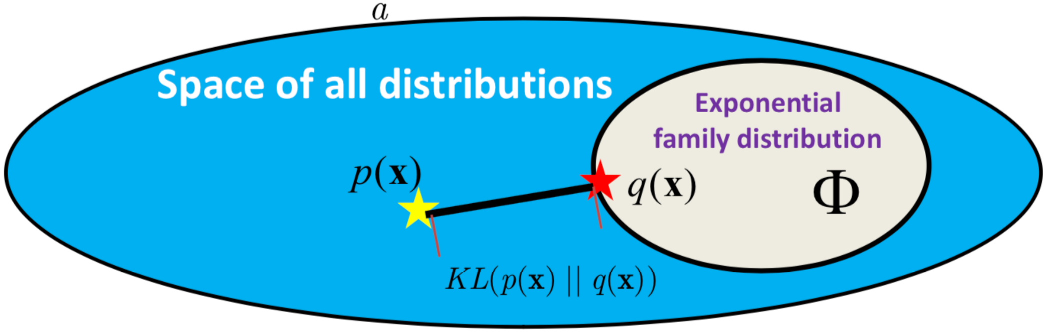
$$q(\boldsymbol{x}) = h(\boldsymbol{x}) \exp\left\{\boldsymbol{\theta}^T \phi(\boldsymbol{x}) + g(\boldsymbol{\theta})\right\}$$

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

EP as Optimization

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

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



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

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

Iterative local optimization

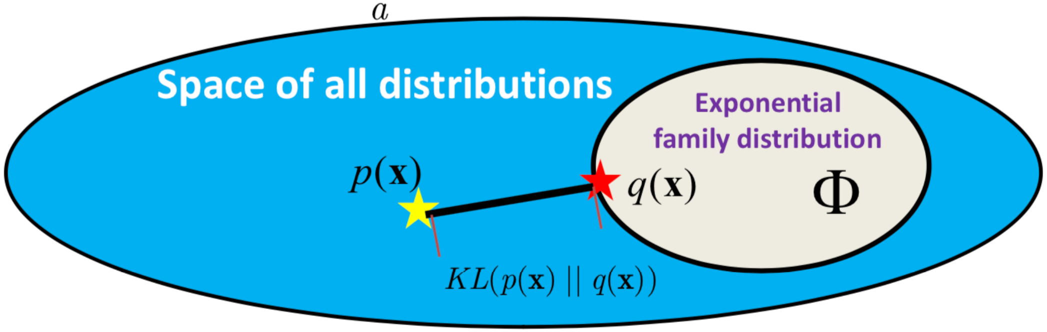
Iteratively refine each factor

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

EP as Optimization

Expectation Propagation (EP) [Minka01] [Opper05]

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



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

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

Iterative local optimization

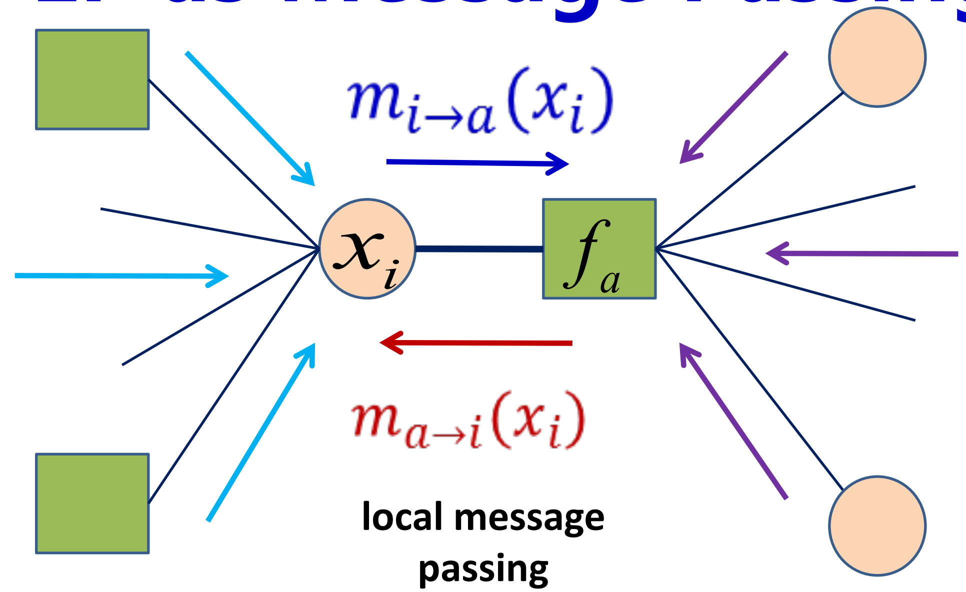
Iteratively refine each factor

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

- **BP minimizes *KL(q||p)* while EP minimizes *KL(p||q)***
- **EP can be also implemented as message passing on factor graph**

EP as Message Passing

- **Factor Graph**



Expectation Propagation

Factor to variable

$$m_{a\to i}(x_i) = \frac{\mathrm{Proj}\left[m_{i\to a}(x_i)\sum_{x_j, j\neq i} f_a(\mathbf{x}_a)\prod_{j\neq i} m_{j\to a}(x_j)\right]}{m_{i\to a}(x_i)}$$

Iterations

Variable to factor

$$m_{i\to a}(x_i) = \prod_{b\neq a} m_{b\to i}(x_i)$$

Excluding incomming message itself

EP vs. BP

Expectation Propagation **vs.** **Belief Propagation**

- **minimize KL (p||q)**
- **A generalization of BP**
- **Discrete & continuous variable**
- Might iterative without loop

- **minimize KL (q||p)**
- **EP with fully factorization**
- **Dicrete variable**
- **Non-iterative without loop**

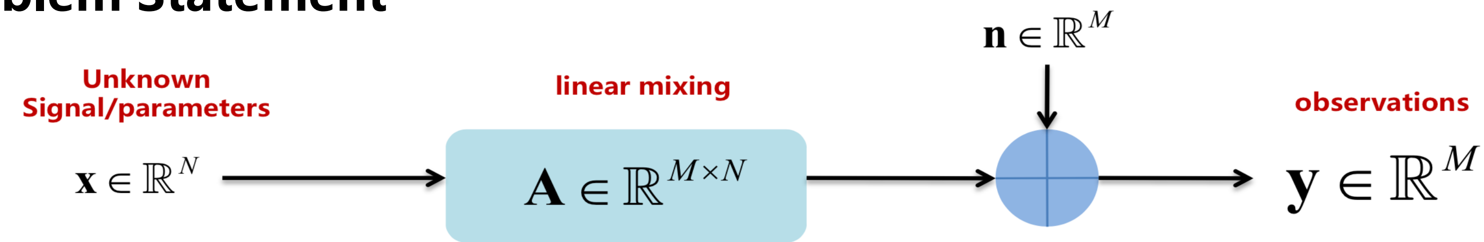
- EP is related to the cavity method in physics [M´ezard et al 87] [Opper&Saad 01]

Outline

- Background
- Variational Inference
- Expectation Propagation
- **A Unified EP Perspective on AMP and its extensions**
- Conclusion

Linear Observations

◻ Problem Statement



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

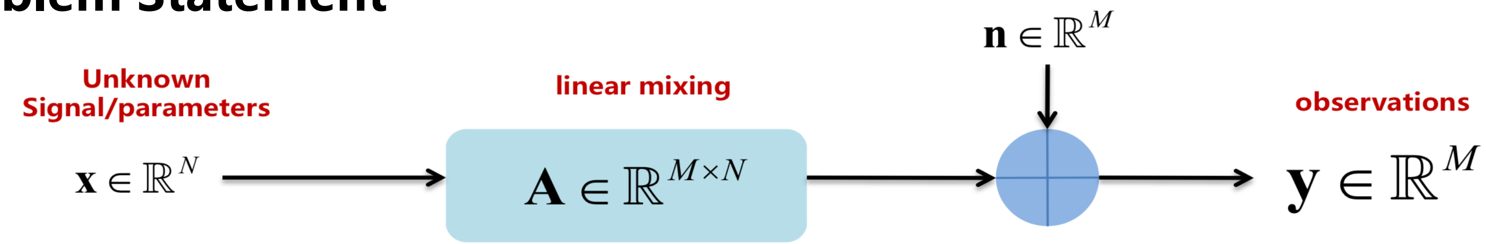
$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$$

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

- **The goal is to recover signal x given the observations y.**
- **A fundamental problem in communication, compressed sensing, statistics**

Linear Observations

Problem Statement



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

- **The goal is to recover signal x given the observations y.**
- **A fundamental problem in communication, compressed sensing, statistics**

First, we write the the joint distribution can be written as follows

$$p(\mathbf{x}, \mathbf{y}) = p_0(\mathbf{x}) p(\mathbf{y}|\mathbf{x})$$

$$= p_0(\mathbf{x}) \frac{1}{(2\pi\sigma^2)^{\frac{M}{2}}} e^{-\frac{(\mathbf{y}-\mathbf{A}\mathbf{x})^T(\mathbf{y}-\mathbf{A}\mathbf{x})}{2\sigma^2}}$$

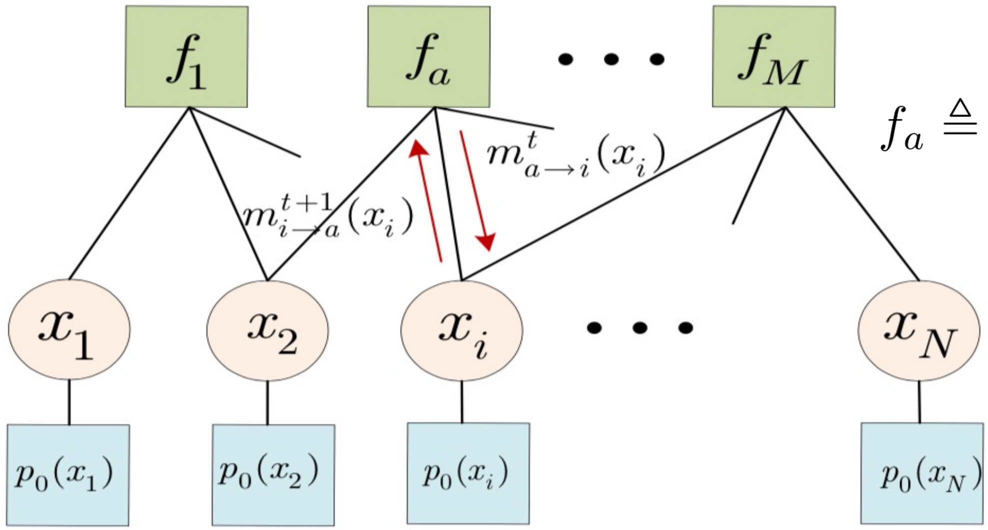
Vector-Form

$$= \prod_{i=1}^{N} p_0(x_i) \prod_{a=1}^{M} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_a - \sum A_{ai} x_i)^2}{2\sigma^2}}$$

Fully-Factorized

Linear Observations

□ Fully-Factorized Factor Graph



$$f_a \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_a - \sum A_{ai} x_i)^2}{2\sigma^2}}$$

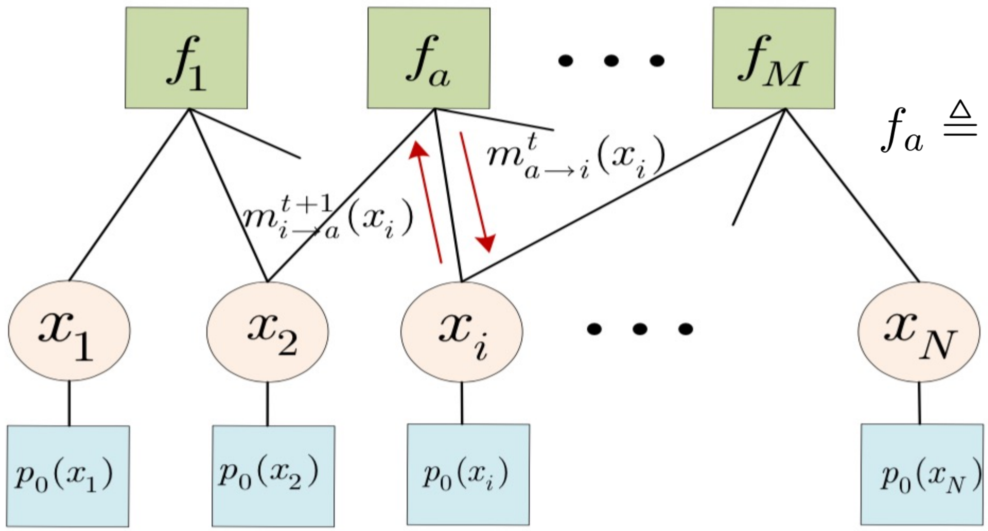
Expectation Propagation (EP)

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

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

Linear Observations

- **Fully-Factorized** **Factor Graph**



$$f_a \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_a - \sum A_{ai} x_i)^2}{2\sigma^2}}$$

Expectation Propagation (EP)

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

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

It seems quite easy ?

Linear Observations

□ 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

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

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

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

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

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

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

Linear Observations

□ 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

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

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

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

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

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

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

Still Too Complicated!

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

Linear Observations

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

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

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

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

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

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

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

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, **neglect the high-order terms** in large system limit.

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

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

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

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

Linear Observations

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

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

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

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

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

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

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

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, **neglect the high-order terms** in large system limit.

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

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

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

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

- After some algebra, the number of messages is reduced to ***O*(*M*+*N*)** and we obtain AMP

AMP Algorithm

Initialization

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

Factor node update

$$V_a^t = \sum_i A_{ai}^2 \nu_i^t$$

$$Z_a^t = \sum_i A_{ai}\hat{x}_i^t - \frac{(y_a - Z_a^{t-1})}{\sigma^2 + V_a^{t-1}} V_a^t$$

Onsager term

Variable node update

$$\Sigma_i^t = 1 \Big/ \sum_a \frac{A_{ai}^2}{\sigma^2 + V_a^t}$$

$$R_i^t = \hat{x}_i^t + \Sigma_i^t \sum_a \frac{A_{ai}(y_a - Z_a^t)}{\sigma^2 + V_a^t}$$

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

End

Linear Complexity O(MN)

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.

Relation to AMP

◻ 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

- **Comments**
  - ✓ The first AMP-like method was derived by Kabashima for CDMA detection [Kabashima 03] and later derived by Donoho et. al for compressed sensing [DMM09].
  - ✓ 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]

Relation to AMP

☐ 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

- **Comments**
  - ✓ The first AMP-like method was derived by Kabashima for CDMA detection [Kabashima 03] and later derived by Donoho et. al for compressed sensing [DMM09].
  - ✓ 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]

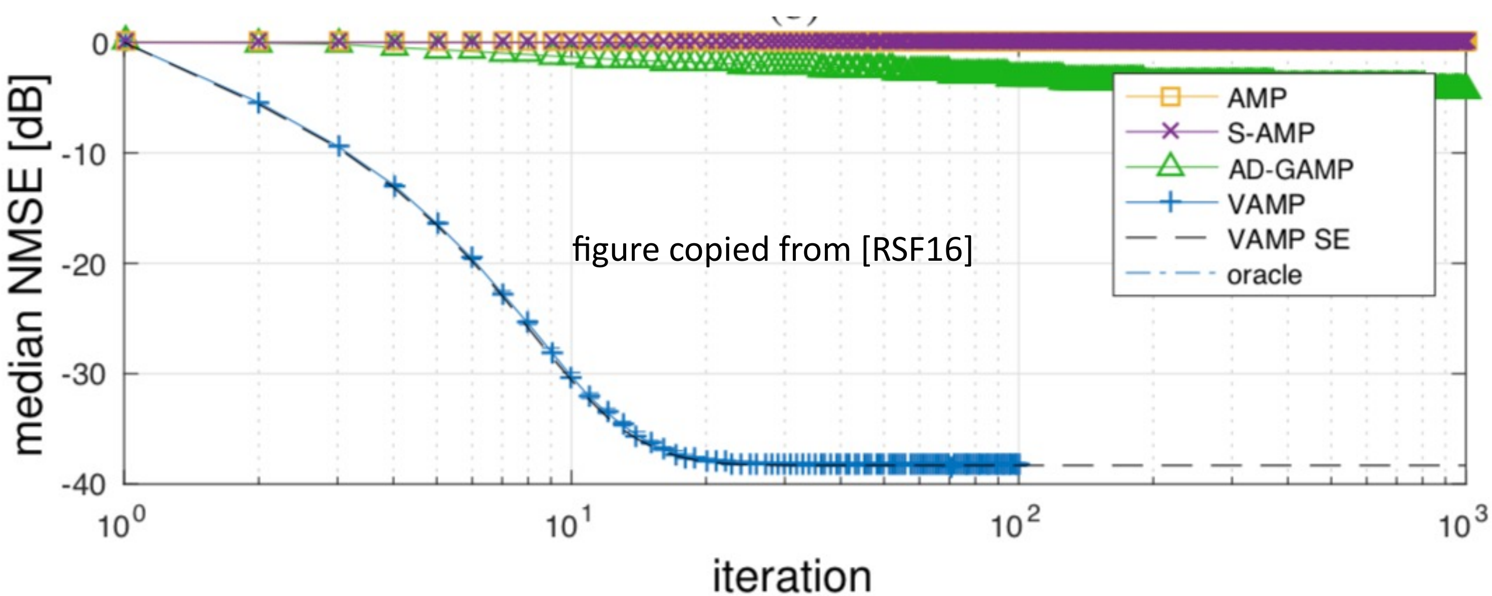
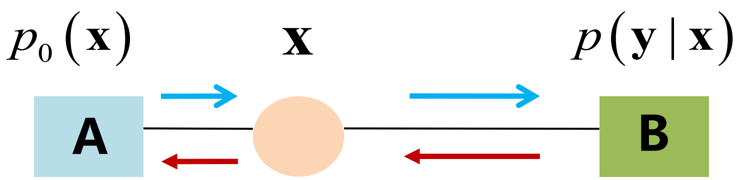


figure copied from [RSF16]

EP Perspective on VAMP

□ Vector-form Factor Graph

$$p(\mathbf{x},\mathbf{y}) = \prod_{i=1}^{N} p_0(x_i) \prod_{a=1}^{M} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_a - \sum A_{ai} x_i)^2}{2\sigma^2}}$$



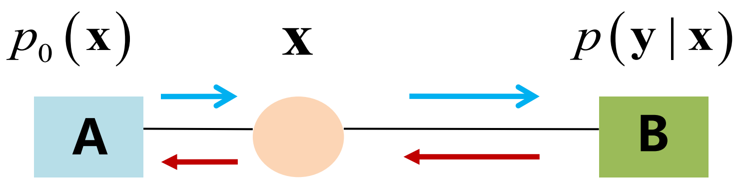
vector-form factor graph

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{\frac{M}{2}}} e^{-\frac{(\mathbf{y}-\mathbf{A}\mathbf{x})^T(\mathbf{y}-\mathbf{A}\mathbf{x})}{2\sigma^2}}$$

EP Perspective on VAMP

☐ Vector-form Factor Graph

$$p(\mathbf{x},\mathbf{y}) = \prod_{i=1}^{N} p_0(x_i) \prod_{a=1}^{M} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_a - \sum A_{ai}x_i)^2}{2\sigma^2}}$$



vector-form factor graph

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{\frac{M}{2}}} e^{-\frac{(\mathbf{y}-\mathbf{A}\mathbf{x})^T(\mathbf{y}-\mathbf{A}\mathbf{x})}{2\sigma^2}}$$

$$m_{A\to x}(x_i) = \frac{\mathrm{Proj}\left[p_0(x_i)\, m_{x\to A}(x_i)\right]}{m_{x\to A}(x_i)} = \mathcal{N}\left(x_i; m_{i\to A}, v_{i\to A}\right)$$

$$m_{x\to B}(x_i) = m_{A\to x}(x_i)$$

$$m_{B\to x}(x_i) = \int \mathcal{N}\left(\mathbf{y}; \mathbf{A}\mathbf{x}, \sigma^2\mathbf{I}\right) \prod m_{x\to B}(x_i)\, dx_{j\neq i} = \mathcal{N}\left(x_i; m_{B\to i}, v_{B\to i}\right)$$

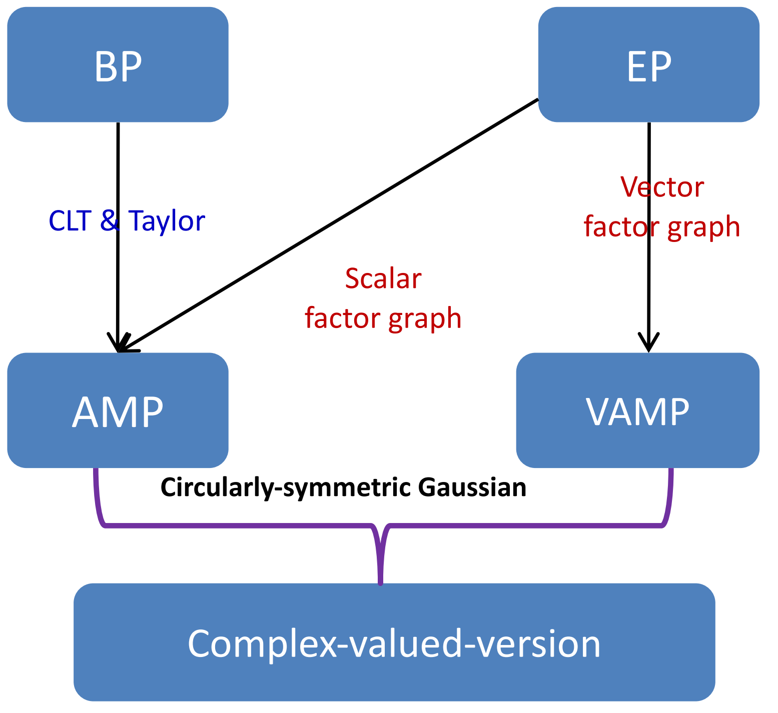
$$m_{x\to A}(x_i) = m_{B\to x}(x_i)$$

This is exactly the MMSE form of VAMP

[RSF16]

A Unified Perspective

◻ An EP Perspective on AMP



- **The EP perspective of AMP and VAMP:**
  - ✓ Explicitly establishing the relationship between AMP
  - ✓ 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)

NonLinear Observations

☐ Background

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

- **The measurements are often obtained in a nonlinear way**
  - **one-bit (quantized) compressed sensing**
  - **phase retrival**
  - **logistic regression**
  - **....**

Inference on Generalized linear model (GLM)

NonLinear Observations

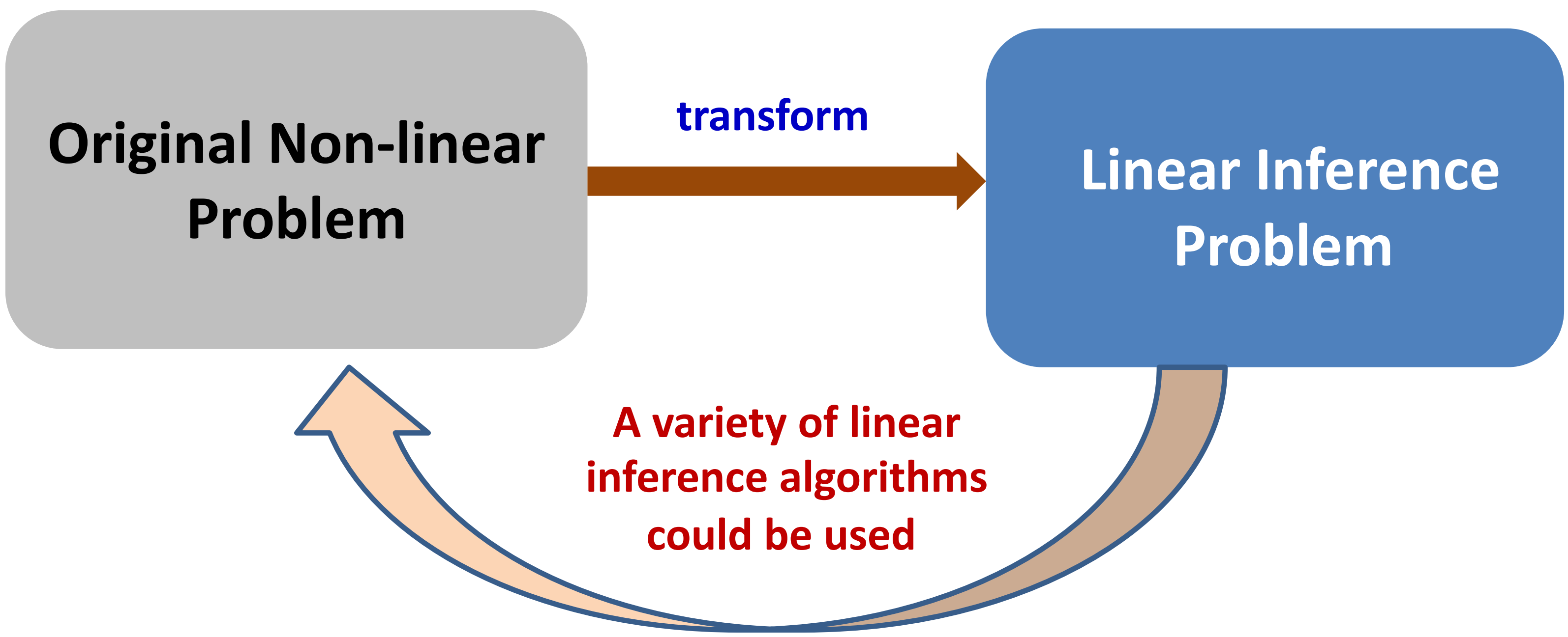
Basic Idea:

Is it possible to transform the nonlinear inference problem to linear inference problems?

NonLinear Observations

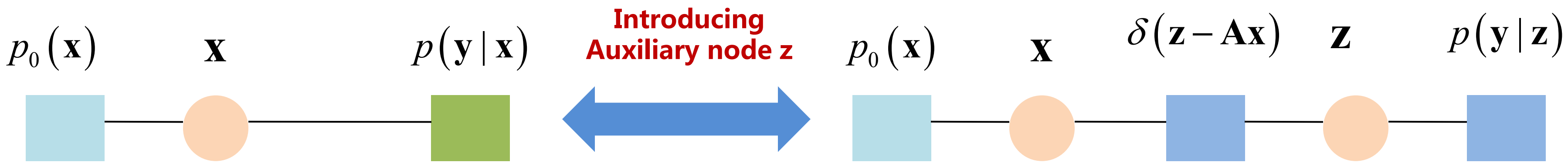
Basic Idea:

Is it possible to transform the nonlinear inference problem to linear inference problems?



A Unified Inference Framework for GLM

□ Two Equivalent Factor Graphs of GLM

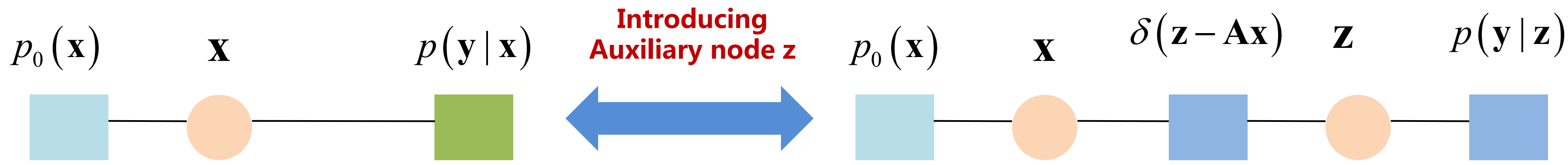


(a) factor graph of GLM

(b) Equivalent factor graph of GLM

A Unified Inference Framework for GLM

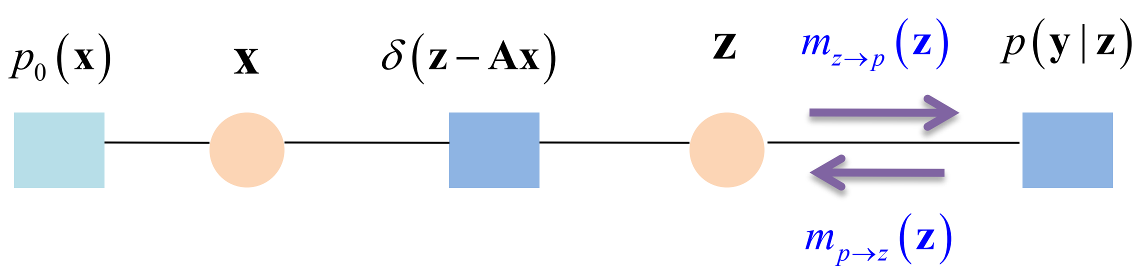
☐ Two Equivalent Factor Graphs of GLM



(a) factor graph of GLM

(b) Equivalent factor graph of GLM

☐ Decoupling GLM into SLM via EP



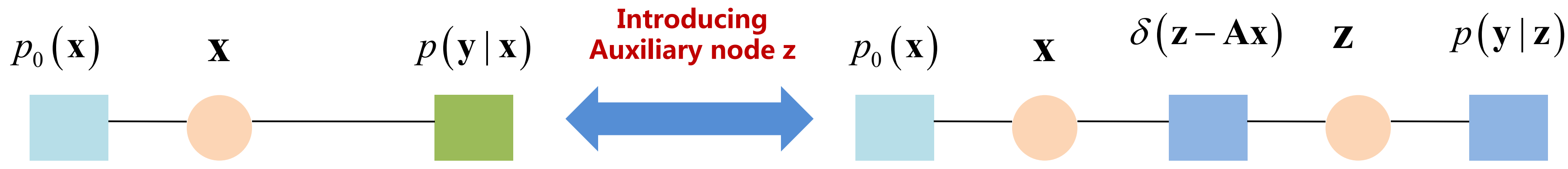
$$m_{z\to p}^{t-1}(\mathbf{z}) \propto N\left(\mathbf{z}; z_A^{ext}(t-1), v_A^{ext}(t-1)I\right)$$

EP message passing
(t-th iteration)

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

A Unified Inference Framework for GLM

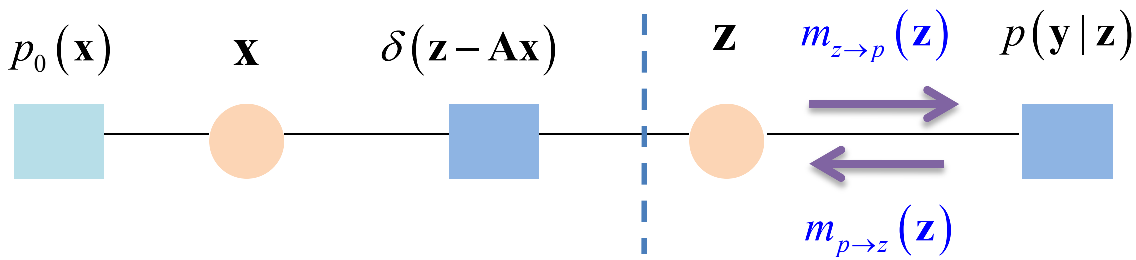
☐ Two Equivalent Factor Graphs of GLM



(a) factor graph of GLM

(b) Equivalent factor graph of GLM

☐ Decoupling GLM into SLM via EP



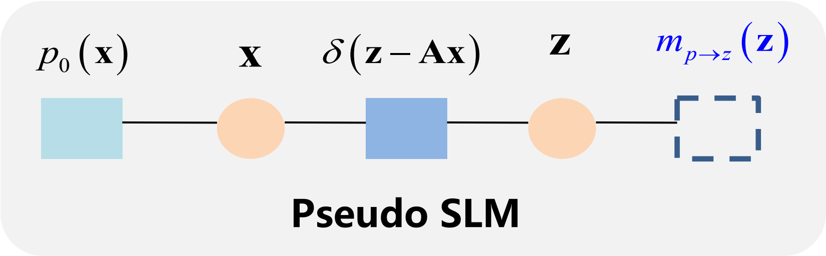
$$m_{z\to p}^{t-1}(\mathbf{z}) \propto N\left(\mathbf{z}; z_A^{ext}(t-1), v_A^{ext}(t-1)I\right)$$

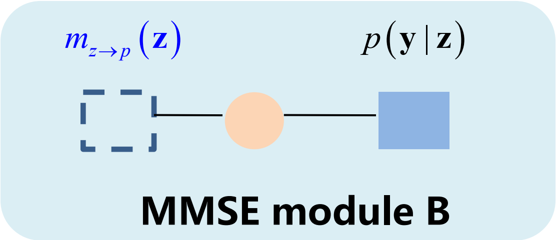
EP message passing
(t-th iteration)

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

A Unified Inference Framework for GLM

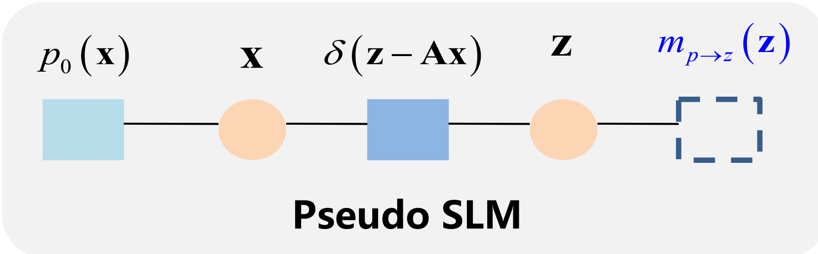
◻ Decoupling GLM into SLM via EP

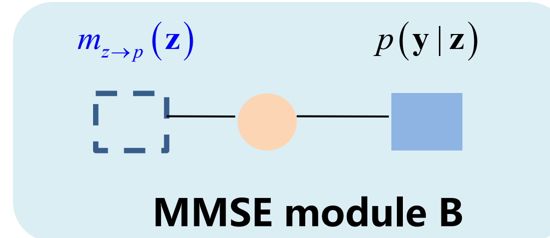




A Unified Inference Framework for GLM

◻ Decoupling GLM into SLM via EP





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

Pseudo Linear Models

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

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

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

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

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

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

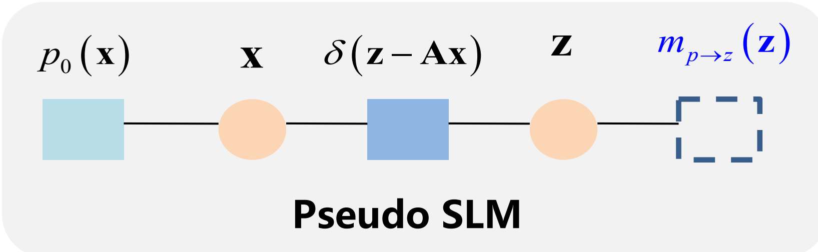
$$\frac{1}{v_B^{ext}(t)} = \frac{1}{v_B^{post}(t)} - \frac{1}{v_A^{ext}(t-1)}$$

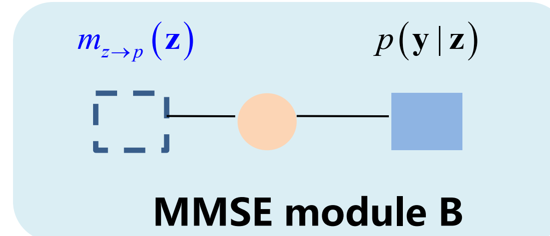
$$\frac{\mathbf{z}_B^{ext}(t)}{v_B^{ext}(t)} = \frac{\mathbf{z}_B^{post}(t)}{v_B^{post}(t)} - \frac{\mathbf{z}_A^{ext}(t-1)}{v_A^{ext}(t-1)}$$

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

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

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

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

Module A

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

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

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

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

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

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

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

Universal Algorithm Design [MWZ18]

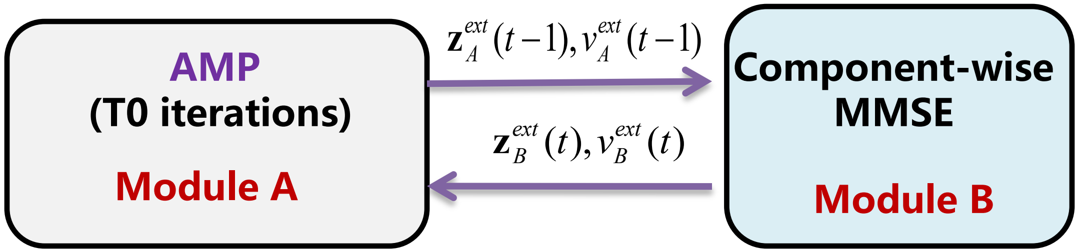
Unified Inference Framework for GLM
- Initialization $\mathbf{z}_A^{ext}(0), v_A^{ext}(0)$
- For t = 1: T, Do
  1. Perform component-wise MMSE
  2. Update $\mathbf{z}_B^{ext}(t), v_B^{ext}(t)$
  3. Perform **SLM inference** one or more iterations
  4. Compute $\mathbf{z}_A^{post}(t), v_A^{post}(t)$ and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

[MWZ18] 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

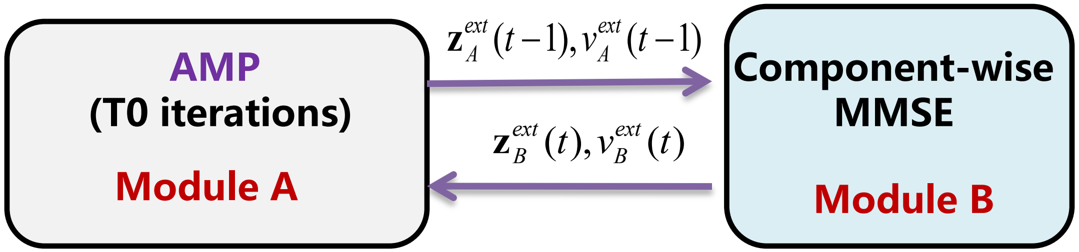


The Gr-AMP Algorithm

- 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



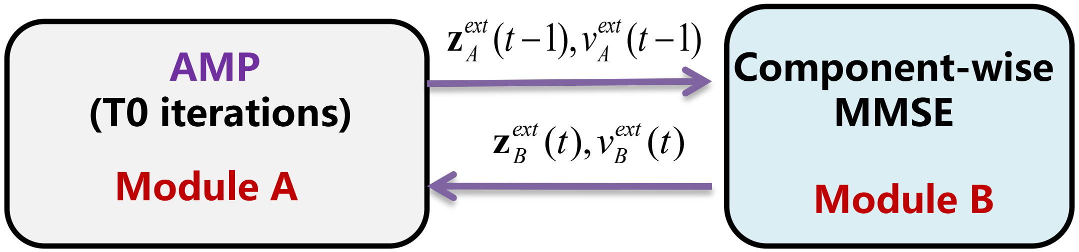
The Gr-AMP Algorithm

- 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 GAMP when T0 = 1 and thus **provides an EP perspective on GAMP** [MWZ18]
    In essance, GAMP first transforms nonlinear model to linear model using EP and then directly apply AMP on the linear model in each iteration.
  - ✓ This perspective provides **a concise derivation of GAMP using EP as in** [MWZ18]
  - ✓ A more flexible message passing schedule: double-loop implementation.

A Unified Inference Framework for GLM

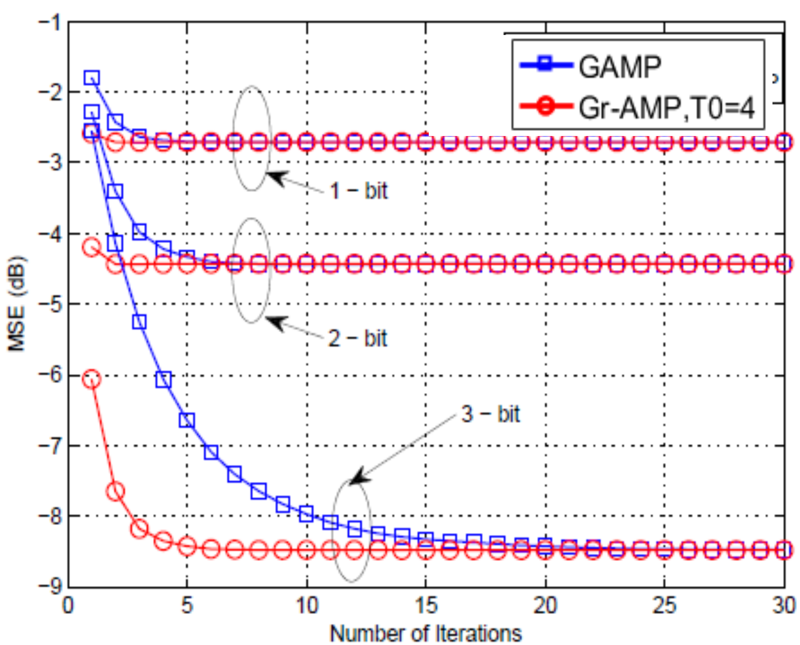
☐ From AMP to Gr-AMP



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

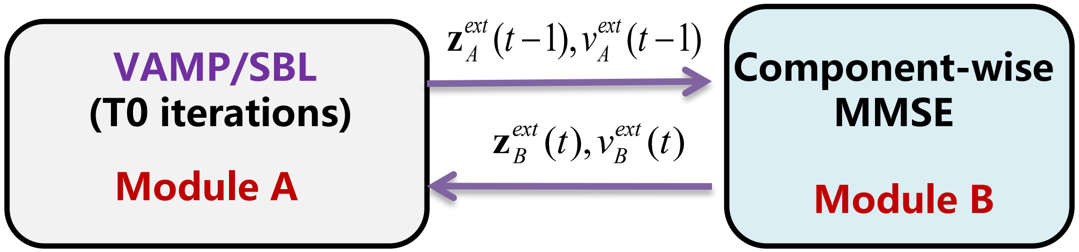
- **Relation of Gr-AMP to GAMP**
  - ✓ Gr-AMP is precisely GAMP when T0 = 1 and thus **provides an EP perspective on GAMP** [MWZ18]
    In essance, GAMP first transforms nonlinear model to linear model using EP and then directly apply AMP on the linear model in each iteration.
  - ✓ This perspective provides **a concise derivation of GAMP using EP as in** [MWZ18]
  - ✓ 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.

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

VAMP/SBL
(T0 iterations)
Module A

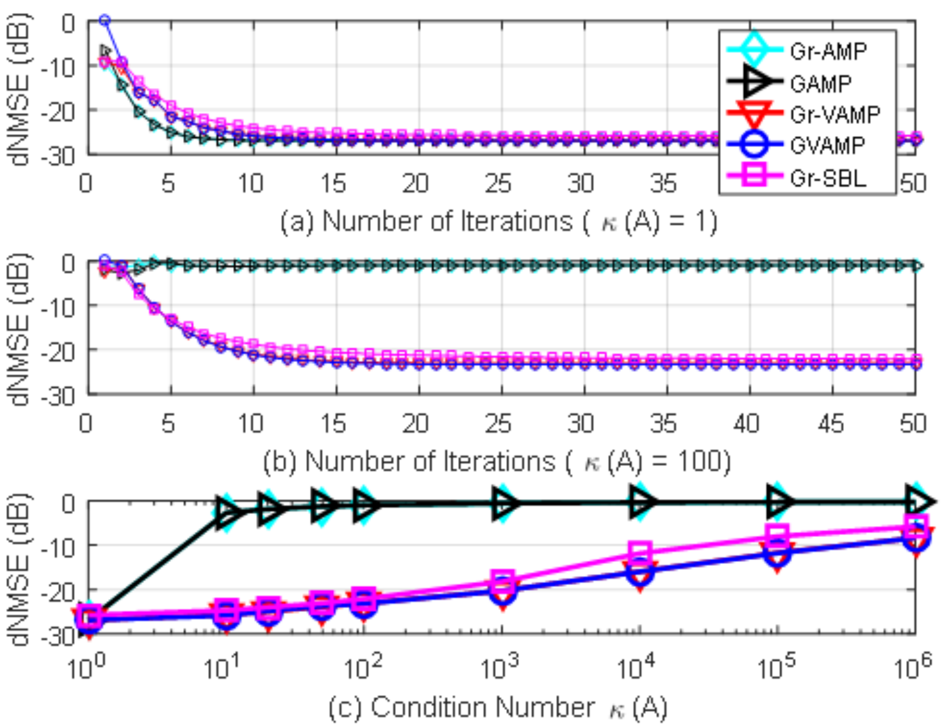
$\mathbf{z}_A^{ext}(t-1), v_A^{ext}(t-1)$ →

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

Component-wise MMSE
Module B

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.

Code available: https://github.com/mengxiangming/glmcode

Conclusions

- **A high-bias low-variance introduction to approximate Bayesian inference**
- **An overview of variational inference framewrok**
- **A tutorial introducition of expection propagation**
- **A unified EP perspective on AMP and its extensions.**

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
- [Minka01b]Minka, Thomas P. “A family of algorithms for approximate bayesian inference”, (thesis) 2001.
- [Opper05]] 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.

References

- [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.
- [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.
- [Yedidia et al 05] Yedidia J S, Freeman W T, Weiss Y. Constructing free-energy approximations and generalized belief propagation algorithms[J]. IEEE Transactions on information theory, 2005, 51(7): 2282-2312.
- [Yedidia et al 02] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. Technical Report TR2001-22, Mitsubishi Electric Research Labs, January 2002.
- [Opper&Saad 01] M. Opper and D. Saad. Adaptive TAP equations. In M. Opper and D. Saad, editors, Advanced mean field methods: Theory and practice, pages 85–98. MIT Press, 2001.
- [Hoffman et al 2013] Hoffman, Matthew D., et al. "Stochastic variational inference." The Journal of Machine Learning Research 14.1 (2013): 1303-1347.
- [M´ezard et al 87] M. M´ezard, G. Parisi and M. A. Virasoro, Spin Glass Theory and Beyond (World Scientific, Singapore, 1987).
- [Bishop06] Bishop C M. Pattern recognition and machine learning[M]. springer, 2006.
- [Mehta et al 19] Mehta P, Bukov M, Wang C H, et al. A high-bias, low-variance introduction to machine learning for physicists[J]. Physics reports, 2019.

Thank You

ありがとうございます

Q&A