Generalization of Factor Graphs and Belief Propagation for Quantum Information Science

End-of-First-Year Oral Exam

Michael X. CAO

Department of Information Engineering, CUHK

September 25, 2015
$g(x) = \prod_a f_a(x_a)$

where $f_a : \mathcal{X}_a \rightarrow \mathbb{R}_{\geq 0}$ is a real nonnegative valued function.
\[ g(x) = \prod_a f_a(x_a) \]

where \( f_a : \mathcal{X}_a \rightarrow \mathbb{R}_{\geq 0} \) is a real nonnegative valued function.

\[ g(x) = \bigotimes_a f_a \]

where \( f_a \in \mathcal{L}_H^+(\mathcal{X}_a) \) is a PSD operator on \( \mathcal{X}_a \), i.e., \([f_a]_{x_{\partial a}, x'_{\partial a}}\) is a PSD matrix.
$$g(x) = \prod_a f_a(x_a)$$

where $f_a : \mathcal{X}_a \rightarrow \mathbb{R}_{\geq 0}$ is a real nonnegative valued function.

$$g(x) = \bigotimes_a f_a$$

where $f_a \in \mathcal{L}_{H}^+(\mathcal{X}_a)$ is a PSD operator on $\mathcal{X}_a$, i.e., $[f_a]_{x_{\partial a}, x'_{\partial a}}$ is a PSD matrix.
$g(x) = \prod_a f_a(x_a)$

where $f_a : \mathcal{X}_a \to \mathbb{R}_{\geq 0}$ is a real nonnegative valued function.

$g(x) = \bigotimes_a f_a$

where $f_a \in \mathcal{L}_{H}^+(\mathcal{X}_a)$ is a PSD operator on $\mathcal{X}_a$, i.e.,

$[f_a]_{x_{\delta a}, x'_{\delta a}}$ is a PSD matrix.

How about calculating the partition sum?
Outline

1. Classical Factor Graphs
   - Modeling
     - “Closing-the-box” Operation

2. Quantum Factor Graphs
   - A Motivating Example
   - Quantum Factor Graph
   - Construction of a QNFG
   - Several Examples

3. Problem of Calculating the Partition Sum
   - Sum-Product / Belief Propagation Algorithm
   - Exploration on Variational Approach

4. End Matters
Factor Graph representing Factorization

Classically, a factor graph describes a factorization of a global function, which is often related to a probability model.

An LDPC Code applied on a binary independent channel
Classically, a factor graph describes a factorization of a global function, which is often related to a probability model.

An LDPC Code applied on a binary independent channel

\[ x_i, y_i \in \mathbb{F}_2 \quad \forall i \in \{1, \cdots, n\} ; \]

\[ f_+ (x) \triangleq 1 \left\{ \sum_{i \text{ incoming}} x_i = 0 \right\}. \]
Classically, a factor graph describes a factorization of a global function, which is often related to a probability model.

An LDPC Code applied on a binary independent channel

\[ x_i, y_i \in \mathbb{F}_2 \quad \forall i \in \{1, \cdots, n\}; \]

\[ f_+ (x) \triangleq 1 \left\{ \sum_{i \text{ incoming}} x_i = 0 \right\}. \]

A problem of interest:
Calculate the marginal distribution of \( x_i \) given fixed \( \{y_i\}_{i=1}^n \).
Classically, a factor graph describes a factorization of a global function, which is often related to a probability model.

A problem of interest: Calculate the marginal distribution of \( x_i \) given fixed \( \{ y_i \}_{i=1}^n \).

\[
g = \prod_i p_{Y_i | x_i} \cdot \prod_k f_k \propto p_{Y | x} \\
\sum_{x_j, j \neq i} g(x, y) \propto p_{Y=y | x_i}
\]
Factor Graph representing Factorization

Classically, a factor graph describes a factorization of a global function, which is often related to a probability model.

\begin{align*}
\prod_{\{x_i, x_j\} \text{ adjacent}} f(x_i, x_j) \cdot \prod_{i, j} h_{ij}(x_i, x_j)
\end{align*}

A simplified $n \times n$ Ising Model
Classically, a factor graph describes a factorization of a global function, which is often related to a probability model.

A simplified $n \times n$ Ising Model

$x_{i,j} \in \{-1, 1\} \ \forall i, j$;

$h_{ij}(x_{i,j}) = \exp \left( \frac{x_i \cdot \tilde{h}_{ij}}{T} \right) \ \forall i, j$;

$f(x_1, x_2) \triangleq \exp \left( \frac{x_1 \cdot x_2}{T} \right)$. 

...
Classically, a factor graph describes a factorization of a global function, which is often related to a probability model.

\[
\begin{align*}
    f(x_1, x_2) & \triangleq \exp \left( \frac{x_1 \cdot x_2}{T} \right) \\
    h_{ij}(x_{i,j}) & = \exp \left( \frac{x_i \cdot \tilde{h}_{ij}}{T} \right) \quad \forall i, j;
\end{align*}
\]

A problem of interest: Calculate the partition function

\[
Z(T) \triangleq \sum_{x} \prod_{a,b \text{ adjacent}} f(x_a, x_b) \cdot \prod_{i,j} h_{ij}(x_{i,j}).
\]
In general, a factor graph for factorization

\[ g(x) = \prod_{a \in F} f_a(x_{\partial a}) \]
Factor Graph representing Factorization

In general, a factor graph for factorization

\[ g(x) = \prod_{a \in \mathcal{F}} f_a(x_{\partial a}) \]

is a bipartite graph \( G = (\mathcal{F}, \mathcal{V}, \mathcal{E}) \) between \( \mathcal{F} \) and \( \mathcal{V} \) with edge set

\[ \mathcal{E} = \{(i, a) \in \mathcal{F} \times \mathcal{V} : i \in \partial a\}. \]
Factor Graph representing Factorization

In general, a factor graph for factorization

\[ g(x) = \prod_{a \in \mathcal{F}} f_a(x_{\partial a}) \]

is a bipartite graph \( \mathcal{G} = (\mathcal{F}, \mathcal{V}, \mathcal{E}) \) between \( \mathcal{F} \) and \( \mathcal{V} \) with edge set

\[ \mathcal{E} = \{ (i, a) \in \mathcal{F} \times \mathcal{V} : i \in \partial a \} \].
Factor Graph representing Factorization

In general, a factor graph for factorization

\[ g(\mathbf{x}) = \prod_{a \in \mathcal{F}} f_a(x_{\partial a}) \]

is a bipartite graph \( \mathcal{G} = (\mathcal{F}, \mathcal{V}, \mathcal{E}) \) between \( \mathcal{F} \) and \( \mathcal{V} \) with edge set

\[ \mathcal{E} = \{(i, a) \in \mathcal{F} \times \mathcal{V} : i \in \partial a\}. \]

\[ \begin{array}{cccccc}
 p & u_A & u_B & b_B & b_A & q_B \\
 x & z & y & z' \\
\end{array} \]

A standard factor graph
Factor Graph representing Factorization

In general, a factor graph for factorization

\[ g(x) = \prod_{a \in F} f_a(x_{\partial a}) \]

is a bipartite graph \( G = (F, V, E) \) between \( F \) and \( V \) with edge set

\[ E = \{(i, a) \in F \times V : i \in \partial a\}. \]

A standard factor graph
Classical Factor Graphs

Modeling

**Factor Graph representing Factorization**

In general, a factor graph for factorization

\[ g(x) = \prod_{a \in \mathcal{F}} f_a(x_{\partial a}) \]

is a bipartite graph \( G = (\mathcal{F}, \mathcal{V}, \mathcal{E}) \) between \( \mathcal{F} \) and \( \mathcal{V} \) with edge set

\[ \mathcal{E} = \{(i, a) \in \mathcal{F} \times \mathcal{V} : i \in \partial a\}. \]
Factor Graph representing Factorization

In general, a factor graph for factorization

\[ g(x) = \prod_{a \in F} f_a(x_{\partial a}) \]

is a bipartite graph \( G = (F, V, E) \) between \( F \) and \( V \) with edge set

\[ E = \{(i, a) \in F \times V : i \in \partial a\}. \]
Factor Graph representing Factorization

In general, a factor graph for factorization

\[ g(x) = \prod_{a \in F} f_a(x_{\partial a}) \]

is a bipartite graph \( G = (F, V, E) \) between \( F \) and \( V \) with edge set

\[ E = \{(i, a) \in F \times V : i \in \partial a\} \].

A standard factor graph

A normal factor graph
Outline

1 Classical Factor Graphs
   - Modeling
   - “Closing-the-box” Operation

2 Quantum Factor Graphs
   - A Motivating Example
   - Quantum Factor Graph
   - Construction of a QNFG
   - Several Examples

3 Problem of Calculating the Partition Sum
   - Sum-Product / Belief Propagation Algorithm
   - Exploration on Variational Approach

4 End Matters
Normal Factor Graph and “Closing-the-box” Operation

Traditionally, we assume all the factors to be nonnegative. Thus, any marginal function is still a *measure.*

\[
p_0 \quad x_0 \quad p_1 \quad x_1 \quad p_2 \quad x_2 \quad p_3 \quad x_3
\]

*Normal Factor Graph for a hidden Markov model of length 3:*

\[
p (y_1, \ldots, y_3, x_0, \ldots, x_3) = p_0 (x_0) \prod_{k=1}^{3} p_k (y_k, x_k | x_{k-1})
\]
Traditionally, we assume all the factors to be nonnegative. Thus, any marginal function is still a *measure*.

\[
p_0(x_0) \prod_{k=1}^{3} p_k(y_k, x_k | x_{k-1})
\]

**Normal Factor Graph for a hidden Markov model of length 3:**

\[
p(y_1, \ldots, y_3, x_0, \ldots, x_3) = p_0(x_0) \prod_{k=1}^{3} p_k(y_k, x_k | x_{k-1})
\]

**Exterior Function of above dashed box:**

\[
p_{y_1, y_2, y_3 | x_0}(y_1, y_2, y_3 | x_0) = \\
\sum_{x_1, x_2, x_3} p_1(y_1, x_1 | x_0) p_2(y_2, x_2 | x_1) p_3(y_3, x_3 | x_2).
\]
Normal Factor Graph and “Closing-the-box” Operation

Traditionally, we assume all the factors to be nonnegative. Thus, any marginal function is still a measure.

\[ p_0 \]

\[ p_{y_1, y_2, y_3 | x_0} (y_1, y_2, y_3 | x_0) \]

\[ y_1 \]
\[ y_2 \]
\[ y_3 \]

Normal Factor Graph for a hidden Markov model of length 3:
\[ p (y_1, \ldots, y_3, x_0, \ldots, x_3) = p_0 (x_0) \prod_{k=1}^{3} p_k (y_k, x_k | x_{k-1}) \]

Exterior Function of above dashed box:
\[ p_{y_1, y_2, y_3 | x_0} (y_1, y_2, y_3 | x_0) = \sum_{x_1, x_2, x_3} p_1 (y_1, x_1 | x_0) p_2 (y_2, x_2 | x_1) p_3 (y_3, x_3 | x_2). \]

“Closing-the-box” Operation: Replacing the box with a factor corresponding to its exterior function
Outline

1. Classical Factor Graphs
   - Modeling
   - “Closing-the-box” Operation

2. Quantum Factor Graphs
   - A Motivating Example
   - Quantum Factor Graph
   - Construction of a QNFG
   - Several Examples

3. Problem of Calculating the Partition Sum
   - Sum-Product / Belief Propagation Algorithm
   - Exploration on Variational Approach

4. End Matters
Factor graphs can be used to represent quantum probabilities if more general factors are allowed [Loeliger and Vontobel, 2015, Loeliger and Vontobel, 2012].

The global function:

\[
g(x, y, \tilde{x}, \tilde{x}') \triangleq p(x) U(\tilde{x}, x) U^H(x, \tilde{x}') B^H(y, \tilde{x}) B(\tilde{x}', y)
\]

\[
= p(x) U(\tilde{x}, x) B(\tilde{x}', y) U(\tilde{x}', x) B(\tilde{x}, y).
\]
Factor graphs can be used to represent quantum probabilities if more general factors are allowed [Loeliger and Vontobel, 2015, Loeliger and Vontobel, 2012].

\[
p_X(x) = U^H U = X Y
\]

Factor graph for an elementary quantum system

The exterior function of the dashed box:

\[
p_{Y|X}(y|x) = \sum_{\tilde{x}, \tilde{x}'} U(\tilde{x}, x) B(\tilde{x}', y) \overline{U(\tilde{x}', x) B(\tilde{x}, y)}
\]

\[= \left| \sum_{\tilde{x}} U(\tilde{x}, x) B(\tilde{x}, y) \right|^2.
\]
Factor graphs can be used to represent quantum probabilities if more general factors are allowed [Loeliger and Vontobel, 2015, Loeliger and Vontobel, 2012].

Factor graph for an elementary quantum system

Normal Factor Graph for a hidden Markov model of length 3:

\[ p(y_1, \ldots, y_3, x_0, \ldots, x_3) = p_0(x_0) \prod_{k=1}^{3} p_k(y_k, x_k | x_{k-1}) \]
Factor Graph representing Quantum Probabilities

Factor graphs can be used to represent quantum probabilities if more general factors are allowed [Loeliger and Vontobel, 2015, Loeliger and Vontobel, 2012].

\[
p(x) = U^H B^H B H U^H
\]

**Factor graph for an elementary quantum system**

\[
p(x) = \hat{U} B^H B H \hat{U}
\]

*Redraw of above*
Factor graphs can be used to represent quantum probabilities if more general factors are allowed [Loeliger and Vontobel, 2015, Loeliger and Vontobel, 2012].

\[
p(x) = U U^H B^H B^{-1} = X Y
\]

Factor graph for an elementary quantum system

Redraw of above
Outline

1. Classical Factor Graphs
   - Modeling
   - “Closing-the-box” Operation

2. Quantum Factor Graphs
   - A Motivating Example
   - Quantum Factor Graph
   - Construction of a QNFG
   - Several Examples

3. Problem of Calculating the Partition Sum
   - Sum-Product / Belief Propagation Algorithm
   - Exploration on Variational Approach

4. End Matters
Quantum Normal Factor Graph (QNFG) as a simplified model

In this case, we have

\[
\begin{align*}
\text{Global function:} & \quad g(x, x', \tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}') = p(x) U(\tilde{x}, x) B(\tilde{x}', y) U(\tilde{x}', x) B(\tilde{x}, y) \cdot 1 \{x = x'\} 1 \{\tilde{y} = \tilde{y}' = y\} \\
\text{Exterior function:} & \quad f_1((x, \tilde{y}), (x', \tilde{y}')) = \hat{U}, \hat{B} \rangle L H(\tilde{X}) \\
\text{Exterior function:} & \quad f_2(y) = \hat{P} \otimes I_y y, f_1 \rangle L H(X \otimes \tilde{Y})
\end{align*}
\]
Quantum Normal Factor Graph (QNFG) as a simplified model

\[ \text{diag}(p(x)) \]

In this case, we have

Global function:

\[ g(x, x', \tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}', y) = p(x)U(\tilde{x}, x)B(\tilde{x}', y)U(\tilde{x}', x)B(\tilde{x}, y) \]
\[ \cdot 1 \{ x = x' \} 1 \{ \tilde{y} = \tilde{y}' = y \} \]
Quantum Normal Factor Graph (QNFG) as a simplified model

In this case, we have

Global function: $g(x, x', \tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}', y) = p(x)U(\tilde{x}, x)B(\tilde{x}', y)\overline{U(\tilde{x}', x)B(\tilde{x}, y)} \cdot 1 \{x = x'\} 1 \{\tilde{y} = \tilde{y}' = y\}$

Exterior function: $f_1((x, \tilde{y}), (x', \tilde{y}')) = \sum_{\tilde{x}, \tilde{x}'}\hat{U}(\tilde{x}, x; \tilde{x}', x') \cdot \hat{B}(\tilde{x}, \tilde{y}; \tilde{x}', \tilde{y}')$
Quantum Normal Factor Graph (QNFG) as a simplified model

In this case, we have

**Global function:** \( g \left( x, x', \tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}', y \right) = p(x)U(\tilde{x}, x)B(\tilde{x}', y)\overline{U(\tilde{x}', x)B(\tilde{x}, y)} \cdot 1 \{ x = x' \} 1 \{ \tilde{y} = \tilde{y}' = y \} \)

**Exterior function:** \( f_1 \left( (x, \tilde{y}), (x', \tilde{y}') \right) = \left\langle \hat{U}, \hat{B} \right\rangle_{\mathcal{L}_H}(\tilde{x}) \)
Quantum Normal Factor Graph (QNFG) as a simplified model

In this case, we have

Global function: \[ g (x, x', \tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}', y) = p(x) U (\tilde{x}, x) B (\tilde{x}', y) \overline{U (\tilde{x}', x) B (\tilde{x}, y)} \cdot 1 \{x = x'\} 1 \{\tilde{y} = \tilde{y}' = y\} \]

Exterior function: \[ f_1 ((x, \tilde{y}), (x', \tilde{y}')) = \left\langle \hat{U}, \hat{B} \right\rangle \mathcal{L}_H (\tilde{x}) \]

Exterior function: \[ f_2 (y) = \sum_x p(x) f_1 ((x, y), (x, y)) \]
Quantum Normal Factor Graph (QNFG) as a simplified model

In this case, we have

Global function: $g(x, x', \tilde{x}, \tilde{x}', \tilde{y}, \tilde{y}', y) = p(x)U(\tilde{x}, x)B(\tilde{x}', y)\overline{U(\tilde{x}', x)B(\tilde{x}, y)}$\cdot \mathbb{1}\{x = x'\}\mathbb{1}\{\tilde{y} = \tilde{y}' = y\}$

Exterior function: $f_1((x, \tilde{y}), (x', \tilde{y}')) = \langle \hat{U}, \hat{B} \rangle L_H(\tilde{x})$

Exterior function: $f_2(y) = \langle \hat{P} \otimes I_y, f_1 \rangle L_H(x \otimes \tilde{y})$
Quantum Normal Factor Graph (QNFG) as a simplified model

Definition 1 (Quantum Normal Factor Graph)

A quantum normal factor graph or QNFG is a normal factor graph where each variable edge may stands for one or a pair of variables. For each factor (indexed by $a \in \mathcal{F}$)

$$f_a (x_{\partial a}, x'_{\partial a}, y_{\delta a})$$

is a PSD operator over $x_{\partial a}$, given $y_{\delta a}$ fixed arbitrarily.
Quantum Normal Factor Graph (QNFG) as a simplified model

Definition 1 (Quantum Normal Factor Graph)

A quantum normal factor graph or QNFG is a normal factor graph where each variable edge may stands for one or a pair of variables. For each factor (indexed by $a \in \mathcal{F}$)

$$f_a (x_{\partial a}, x'_{\partial a}; y_{\delta a})$$

is a PSD operator over $\mathcal{X}_{\partial a}$, given $y_{\delta a}$ fixed arbitrarily.
Quantum Normal Factor Graph (QNFG) as a simplified model

Definition 1 (Quantum Normal Factor Graph)

A quantum normal factor graph or QNFG is a normal factor graph where each variable edge may stand for one or a pair of variables. For each factor (indexed by $a \in \mathcal{F}$)

$$f_a (x_{\partial a}, x'_{\partial a}; y_{\delta a})$$

is a PSD operator over $\mathcal{X}_{\partial a}$, given $y_{\delta a}$ fixed arbitrarily.
Quantum Normal Factor Graph (QNFG) as a simplified model

 Definition 1 (Quantum Normal Factor Graph)

A quantum normal factor graph or QNFG is a normal factor graph where each variable edge may stands for one or a pair of variables. For each factor (indexed by $a \in \mathcal{F}$)

$$f_a (x_{\partial a}, x'_{\partial a}; y_{\delta a})$$

is a PSD operator over $\mathcal{X}_{\partial a}$, given $y_{\delta a}$ fixed arbitrarily.
Quantum Normal Factor Graph (QNFG) as a simplified model

**Definition 1 (Quantum Normal Factor Graph)**

A *quantum normal factor graph* or QNFG is a normal factor graph where each variable edge may stand for one or a pair of variables. For each factor (indexed by $a \in \mathcal{F}$)

$$f_a (x_{\partial a}, x'_{\partial a}; y_{\delta a})$$

is a PSD operator over $\mathcal{X}_{\partial a}$, given $y_{\delta a}$ fixed arbitrarily. We can define *quantum factor graph* (QFG) similarly, allowing some variable nodes to have degree higher than 2.
Outline

1. Classical Factor Graphs
   - Modeling
   - "Closing-the-box" Operation

2. Quantum Factor Graphs
   - A Motivating Example
   - Quantum Factor Graph
   - Construction of a QNFG
   - Several Examples

3. Problem of Calculating the Partition Sum
   - Sum-Product / Belief Propagation Algorithm
   - Exploration on Variational Approach

4. End Matters
Conversion into QNFG: Squeeze

\[ \hat{U}((\tilde{x}, x), (\tilde{x}', x')) \triangleq U(\tilde{x}, x) \cdot \overline{U(\tilde{x}', x')} \]
\[ = \text{vec}(U) \text{vec}(U)^H \]
Conversion into QNFG: Equality

Classical factor graph

```
X
```

Quantum Normal Factor Graph

```
I
X
X'
```

$I$ is the identity matrix, i.e., $I(x, x') = \delta(x, x')$. 
Conversion into QNFG: Merge

Classical factor graph

$$p(x) \quad = \quad X \quad X'$$

Quantum Normal Factor Graph

$$\text{diag}(p(x)) \quad X \quad X'$$

$$\text{diag}(p(x))(x, x') = \begin{cases} 
p(x) & \text{if } x = x' \\
0 & \text{otherwise} \end{cases}$$
Conversion into QNFG: Parametrize

Classical factor graph

Quantum Normal Factor Graph

\[ I_y(\tilde{y}, \tilde{y}') = \begin{cases} 
1 & \text{if } \tilde{y} = \tilde{y}' = y \\
0 & \text{otherwise} 
\end{cases} \]
Outline

1. Classical Factor Graphs
   - Modeling
   - “Closing-the-box” Operation

2. Quantum Factor Graphs
   - A Motivating Example
   - Quantum Factor Graph
   - Construction of a QNFG
   - Several Examples

3. Problem of Calculating the Partition Sum
   - Sum-Product / Belief Propagation Algorithm
   - Exploration on Variational Approach

4. End Matters
**Example 1**

Unitary Evolution over time in $n$ steps followed by a single projective measure.
Example 1

Unitary Evolution over time in $n$ steps followed by a single projective measure

$$
\rho_1 (x_1, x'_1) = \sum_{x, x'} \hat{U}_1 \left( (x_1, x), (x'_1, x') \right) p(x, x') \delta(x, x')
$$
Example 1

\[ \rho_1 (x_1, x'_1) = \sum_{x, x'} \hat{U}_1 \left( (x_1, x), (x'_1, x') \right) p(x, x') \delta(x, x') = \left\langle \hat{U}_1, \text{diag}(p) \right\rangle_{\mathcal{L}_H(x)} \]

Unitary Evolution over time in \( n \) steps followed by a single projective measure
Example 1

\[
\begin{align*}
\rho_1(x_1, x'_1) &= \sum_{x, x'} \hat{U}_1 \left( (x_1, x), (x'_1, x') \right) p(x, x') \delta(x, x') = \langle \hat{U}_1, \text{diag}(p) \rangle_{\mathcal{L}_H(x)} \\
\rho_2(x_2, x'_2) &= \sum_{x_1, x'_1} \hat{U}_2 \left( (x_2, x_1), (x'_2, x'_1) \right) \rho(x_1, x'_1)
\end{align*}
\]
Example 1

Unitary Evolution over time in $n$ steps followed by a single projective measure

$$\rho_1(x_1, x'_1) = \sum_{x, x'} \hat{U}_1 \left( (x_1, x), (x'_1, x') \right) p(x, x') \delta(x, x') = \left\langle \hat{U}_1, \text{diag}(p) \right\rangle_{\mathcal{L}_H(x)}$$

$$\rho_2(x_2, x'_2) = \sum_{x_1, x'_1} \hat{U}_2 \left( (x_2, x_1), (x'_2, x'_1) \right) \rho(x_1, x'_1) = \left\langle \hat{U}_2, \rho_1 \right\rangle_{\mathcal{L}_H(x_1)}$$
Example 1

Unitary Evolution over time in n steps followed by a single projective measure

\[ \rho_1 (x_1, x'_1) = \sum_{x, x'} \hat{U}_1 \left( (x_1, x), (x'_1, x') \right) p(x, x') \delta(x, x') = \langle \hat{U}_1, \text{diag}(p) \rangle_{\mathcal{L}_H(x)} \]

\[ \rho_2 (x_2, x'_2) = \sum_{x_1, x'_1} \hat{U}_2 \left( (x_2, x_1), (x'_2, x'_1) \right) \rho(x_1, x'_1) = \langle \hat{U}_2, \rho_1 \rangle_{\mathcal{L}_H(x_1)} \]

\[ \rho_n (x_n, x'_n) = \sum_{x_{n-1}, x'_{n-1}} \hat{U}_{n-1} \left( (x_n, x_{n-1}), (x'_n, x'_{n-1}) \right) \rho(x_{n-1}, x'_{n-1}) \]
Example 1

Unitary Evolution over time in $n$ steps followed by a single projective measure

$$
\rho_1 (x_1, x_1') = \sum_{x, x'} \hat{U}_1 \left( (x_1, x), (x_1', x') \right) p(x, x') \delta(x, x') = \left\langle \hat{U}_1, \text{diag}(p) \right\rangle_{L_H(x)}
$$

$$
\rho_2 (x_2, x_2') = \sum_{x_1, x_1'} \hat{U}_2 \left( (x_2, x_1), (x_2', x_1') \right) \rho(x_1, x_1') = \left\langle \hat{U}_2, \rho_1 \right\rangle_{L_H(x_1)}
$$

$$
\rho_n (x_n, x_n') = \sum_{x_{n-1}, x_{n-1}'} \hat{U}_{n-1} \left( (x_n, x_{n-1}), (x_n', x_{n-1}') \right) \rho(x_{n-1}, x_{n-1}')
$$

$$
= \left\langle \hat{U}_n, \rho_{n-1} \right\rangle_{L_H(x_{n-1})}
$$
Example 1

Unitary Evolution over time in $n$ steps followed by a single projective measure

$$\rho_1 (x_1, x'_1) = \sum_{x,x'} \hat{U}_1 \left( (x_1, x), (x'_1, x') \right) p(x, x') \delta(x, x') = \left\langle \hat{U}_1, \text{diag}(p) \right\rangle_{\mathcal{L}_H(x)}$$

$$\rho_2 (x_2, x'_2) = \sum_{x_1,x'_1} \hat{U}_2 \left( (x_2, x_1), (x'_2, x'_1) \right) \rho_1 (x_1, x'_1) = \left\langle \hat{U}_2, \rho_1 \right\rangle_{\mathcal{L}_H(x_1)}$$

$$\rho_n (x_n, x'_n) = \sum_{x_{n-1},x'_{n-1}} \hat{U}_{n-1} \left( (x_n, x_{n-1}), (x'_n, x'_{n-1}) \right) \rho (x_{n-1}, x'_{n-1})$$

$$= \left\langle \hat{U}_n, \rho_{n-1} \right\rangle_{\mathcal{L}_H(x_{n-1})}$$

Schrödinger representation.
Example 1

Unitary Evolution over time in $n$ steps followed by a single projective measure

$$\varphi_n(x_n, x'_n) = \sum_y \hat{B}\left((x_n, y), (x'_n, y)\right)$$
Example 1

Unitary Evolution over time in $n$ steps followed by a single projective measure

$$\varphi_n(x_n, x'_n) = \sum_y \hat{B} \left( (x_n, y), (x'_n, y) \right) = \left\langle \hat{B}, I_y \right\rangle_{L_H(\tilde{y})}$$
Example 1

Unitary Evolution over time in $n$ steps followed by a single projective measure

$$\varphi_n(x_n, x'_n) = \sum_y \hat{B}\left( (x_n, y), (x'_n, y) \right) = \left\langle \hat{B}, I_y \right\rangle_{\mathcal{L}_H(\tilde{Y})}$$

$$\varphi_{n-1}(x_{n-1}, x'_{n-1}) = \sum_{x_n, x'_n} \hat{U}_{n-1}\left( (x_n, x_{n-1}), (x'_n, x'_{n-1}) \right) \varphi(x_n, x'_n)$$
Example 1

Unitary Evolution over time in $n$ steps followed by a single projective measure

$$\varphi_n(x_n, x'_n) = \sum_y \hat{B} \left( (x_n, y), (x'_n, y) \right) = \left\langle \hat{B}, I_y \right\rangle_{L_H(\tilde{y})}$$

$$\varphi_{n-1}(x_{n-1}, x'_{n-1}) = \sum_{x_n, x'_n} \hat{U}_{n-1} \left( (x_n, x_{n-1}), (x'_n, x'_{n-1}) \right) \varphi(x_n, x'_n) = \left\langle \hat{U}_n, \varphi_n \right\rangle_{L_H(\tilde{x}_n)}$$
Example 1

Unitary Evolution over time in $n$ steps followed by a single projective measure

$$\varphi_n (x_n, x'_n) = \sum_y \hat{B} \left( (x_n, y), (x'_n, y) \right) = \left< \hat{B}, l_y \right>_{L_H(\tilde{y})}$$

$$\varphi_{n-1} (x_{n-1}, x'_{n-1}) = \sum_{x_n, x'_n} \hat{U}_{n-1} \left( (x_n, x_{n-1}), (x'_n, x'_{n-1}) \right) \varphi (x_n, x'_n) = \left< \hat{U}_n, \varphi_n \right>_{L_H(\tilde{x}_n)}$$

$$\varphi_1 = \sum_{x_2, x'_2} \hat{U}_2 \left( (x_2, x_1), (x'_2, x'_1) \right) \varphi_2 (x_2, x'_2)$$
Example 1

Unitary Evolution over time in n steps followed by a single projective measure

\[ \varphi_n(x_n, x'_n) = \sum_y \hat{B} \left( (x_n, y), (x'_n, y) \right) = \left\langle \hat{B}, I_y \right\rangle_{LH}(\tilde{y}) \]

\[ \varphi_{n-1}(x_{n-1}, x'_{n-1}) = \sum_{x_n, x'_n} \hat{U}_{n-1} \left( (x_n, x_{n-1}), (x'_n, x'_{n-1}) \right) \varphi(x_n, x'_n) = \left\langle \hat{U}_n, \varphi_n \right\rangle_{LH}(\tilde{x}_n) \]

\[ \varphi_1 = \sum_{x_2, x'_2} \hat{U}_2 \left( (x_2, x_1), (x'_2, x'_1) \right) \varphi_2(x_2, x'_2) = \left\langle \hat{U}_2, \varphi_2 \right\rangle_{LH}(\tilde{x}_2) \]
Example 1

Unitary Evolution over time in $n$ steps followed by a single projective measure

$$
\varphi_n(x_n, x'_n) = \sum_y \hat{B}\left((x_n, y), (x'_n, y)\right) = \left\langle \hat{B}, I_y \right\rangle_{L_H}(\tilde{y})
$$

$$
\varphi_{n-1}(x_{n-1}, x'_{n-1}) = \sum_{x_n, x'_n} \hat{U}_{n-1}\left((x_n, x_{n-1}), (x'_n, x'_{n-1})\right) \varphi(x_n, x'_n) = \left\langle \hat{U}_n, \varphi_n \right\rangle_{L_H}(\tilde{x}_n)
$$

$$
\varphi_1 = \sum_{x_2, x'_2} \hat{U}_2\left((x_2, x_1), (x'_2, x'_1)\right) \varphi_2(x_2, x'_2) = \left\langle \hat{U}_2, \varphi_2 \right\rangle_{L_H}(\tilde{x}_2)
$$

$$
\varphi_0 = \sum_{x_1, x'_1} \hat{U}_1\left((x_1, x), (x'_1, x')\right) \varphi_1(x_1, x'_1)
$$
Example 1

Unitary Evolution over time in $n$ steps followed by a single projective measure

$$\varphi_n (x_n, x'_n) = \sum_y \hat{B} \left( (x_n, y), (x'_n, y) \right) = \langle \hat{B}, I_y \rangle_{L_H(\tilde{y})}$$

$$\varphi_{n-1} (x_{n-1}, x'_{n-1}) = \sum_{x_n, x'_n} \hat{U}_{n-1} \left( (x_n, x_{n-1}), (x'_n, x'_{n-1}) \right) \varphi (x_n, x'_n) = \langle \hat{U}_{n}, \varphi_n \rangle_{L_H(\tilde{x}_n)}$$

$$\varphi_1 = \sum_{x_2, x'_2} \hat{U}_2 \left( (x_2, x_1), (x'_2, x'_1) \right) \varphi_2 (x_2, x'_2) = \langle \hat{U}_2, \varphi_2 \rangle_{L_H(\tilde{x}_2)}$$

$$\varphi_0 = \sum_{x_1, x'_1} \hat{U}_1 \left( (x_1, x), (x'_1, x') \right) \varphi_1 (x_1, x'_1) = \langle \hat{U}_1, \varphi_1 \rangle_{L_H(\tilde{x}_1)}$$
Example 1

Unitary Evolution over time in \( n \) steps followed by a single projective measure

\[
\varphi_n (x_n, x'_n) = \sum_y \hat{B} \left( (x_n, y), (x'_n, y) \right) = \left< \hat{B}, I_y \right>_{\mathcal{L}_H(\tilde{Y})}
\]

\[
\varphi_{n-1} (x_{n-1}, x'_{n-1}) = \sum_{x_n, x'_n} \hat{U}_{n-1} \left( (x_n, x_{n-1}), (x'_n, x'_{n-1}) \right) \varphi (x_n, x'_n) = \left< \hat{U}_n, \varphi_n \right>_{\mathcal{L}_H(\tilde{x}_n)}
\]

\[
\varphi_1 = \sum_{x_2, x'_2} \hat{U}_2 \left( (x_2, x_1), (x'_2, x'_1) \right) \varphi_2 (x_2, x'_2) = \left< \hat{U}_2, \varphi_2 \right>_{\mathcal{L}_H(\tilde{x}_2)}
\]

\[
\varphi_0 = \sum_{x_1, x'_1} \hat{U}_1 \left( (x_1, x), (x'_1, x') \right) \varphi_1 (x_1, x'_1) = \left< \hat{U}_1, \varphi_1 \right>_{\mathcal{L}_H(\tilde{x}_1)}
\]

Heisenberg representation.
Example 2

A Two-Measurement Quantum System

Here, we assume

$$\sum_{y_k} \sum_{x_k} \hat{A}_k \left( (\tilde{x}_k, x_k), (\tilde{x}_k, x'_k) \right) = \delta (x_k, x'_k)$$
Example 2

Here, we assume

$$\sum_{y_k} \sum_{x_k} \hat{A}_k \left( (\tilde{x}_k, x_k), (\tilde{x}_k, x'_k) \right) = \delta (x_k, x'_k)$$

or, equivalently

$$\sum_{y_k} \left\langle \hat{A}_k^{y_k}, \delta \tilde{x}_k, \tilde{x}'_k \right\rangle_{\mathcal{L}_H} = \delta x_k, x'_k$$
Example 2

A Two-Measurement Quantum System

Here, we assume

$$\sum_{y_k} \sum_{x_k} \hat{A}_k \left( (\tilde{x}_k, x_k), (\tilde{x}_k, x'_k) \right) = \delta(x_k, x'_k)$$

or, equivalently

$$\sum_{y_k} \langle \hat{A}^y_k, \delta\hat{x}_k, \hat{x}_k' \rangle_{\mathcal{L}_H(\hat{x})} = \delta x_k, x'_k$$

A special example:

Projective Measurement with 1-dim Eigenspaces
Example 3

A Quantum System with partial measurement

\[ \text{diag}(\rho(x_0)) \]

\[ \hat{U}_0 \rightarrow X_0' \quad X_0 \]

\[ \hat{A}_1 \rightarrow X_1' \quad \tilde{X}_1' \quad X_1 \]

\[ \hat{U}_1 \rightarrow W_1' \quad W_1 \quad \tilde{W}_1 \]

\[ \hat{A}_2 \rightarrow X_2' \quad \tilde{X}_2' \quad X_2 \]

\[ \hat{U}_2 \rightarrow W_2' \quad W_2 \quad \tilde{W}_2 \]

\[ Y_1 \rightarrow \tilde{X}_1 \]

\[ Y_2 \rightarrow \tilde{X}_2 \]

\[ \hat{U} \rightarrow X_3' \quad X_3 \]

\[ X_0' = X_1 \times W_1 \]

This QFG contains cycles.
Example 3

A Quantum System with partial measurement

\[ x_0 = x_1 \times \mathcal{W}_1 \]
Example 3

A Quantum System with partial measurement

\[ x_0 = x_1 \times \mathcal{W}_1 \]

This QFG contains cycles.
Outline

1. Classical Factor Graphs
   - Modeling
   - “Closing-the-box” Operation

2. Quantum Factor Graphs
   - A Motivating Example
   - Quantum Factor Graph
   - Construction of a QNFG
   - Several Examples

3. Problem of Calculating the Partition Sum
   - Sum-Product / Belief Propagation Algorithm
   - Exploration on Variational Approach

4. End Matters
Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate \( Z(\mathcal{G}) \triangleq \sum_x g(x) \)

\( Z = \text{abcdef} = \sum_{x_5} \text{abd} \cdot \text{cef} \)

Sum-Product Algorithm on a normal factor graph with no cycles
Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate $Z(G) \triangleq \sum_x g(x)$

$\text{Sum-Product Algorithm on a normal factor graph with no cycles}$

\[
f_{bd}(x_4) = \sum_{x_1} f_b(x_1, x_4) f_d(x_1)
\]
Problem of Calculating the Partition Sum

Sum-Product / Belief Propagation Algorithm

Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate \( Z(G) \triangleq \sum_x g(x) \)

\[ f_{bd}(x_4) = \sum_{x_1} f_b(x_1, x_4) f_d(x_1) \]

*Sum-Product Algorithm on a normal factor graph with no cycles*
Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate $Z(\mathcal{G}) \triangleq \sum_x g(x)$

**Sum-Product Algorithm on a normal factor graph with no cycles**

$f_{bd}(x_4) = \sum_{x_1} f_b(x_1, x_4) f_d(x_1)$

$f_{ce}(x_3, x_5) = \sum_{x_2} f_c(x_2, x_3, x_5) f_e(x_2)$
Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate $Z(G) \triangleq \sum_x g(x)$

*Sum-Product Algorithm on a normal factor graph with no cycles*

$$f_{bd}(x_4) = \sum_{x_1} f_b(x_1, x_4) f_d(x_1)$$

$$f_{ce}(x_3, x_5) = \sum_{x_2} f_c(x_2, x_3, x_5) f_e(x_2)$$
Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate $Z (G) \triangleq \sum_{x} g (x)$

Sum-Product Algorithm on a normal factor graph with no cycles

$$f_{bd} (x_4) = \sum_{x_1} f_b (x_1, x_4) f_d (x_1)$$

$$f_{ce} (x_3, x_5) = \sum_{x_2} f_c (x_2, x_3, x_5) f_e (x_2)$$

$$f_{cef} (x_5) = \sum_{x_3} f_{ce} (x_3, x_5) f_f (x_3)$$
Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate $Z(\mathcal{G}) \triangleq \sum_x g(x)$

Sum-Product Algorithm on a normal factor graph with no cycles

\[
f_{bd}(x_4) = \sum_{x_1} f_b(x_1, x_4) f_d(x_1)
\]

\[
f_{ce}(x_3, x_5) = \sum_{x_2} f_c(x_2, x_3, x_5) f_e(x_2)
\]

\[
f_{cef}(x_5) = \sum_{x_3} f_{ce}(x_3, x_5) f_f(x_3)
\]
Problem of Calculating the Partition Sum

Sum-Product / Belief Propagation Algorithm

Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate \( Z(G) \triangleq \sum_x g(x) \)

Sum-Product Algorithm on a normal factor graph with no cycles

\[
\begin{align*}
 f_{bd}(x_4) &= \sum_{x_1} f_b(x_1, x_4) f_d(x_1) \\
 f_{ce}(x_3, x_5) &= \sum_{x_2} f_c(x_2, x_3, x_5) f_e(x_2) \\
 f_{cef}(x_5) &= \sum_{x_3} f_{ce}(x_3, x_5) f_f(x_3) \\
 f_{abd}(x_5) &= \sum_{x_4} f_a(x_4, x_5) f_{bd}(x_4)
\end{align*}
\]

Michael X. CAO (IE@CUHK)
Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate $Z(\mathcal{G}) \triangleq \sum_x g(x)$

Sum-Product Algorithm on a normal factor graph with no cycles

$$f_{bd}(x_4) = \sum_{x_1} f_b(x_1, x_4) f_d(x_1)$$

$$f_{ce}(x_3, x_5) = \sum_{x_2} f_c(x_2, x_3, x_5) f_e(x_2)$$

$$f_{cef}(x_5) = \sum_{x_3} f_{ce}(x_3, x_5) f_f(x_3)$$

$$f_{abd}(x_5) = \sum_{x_4} f_a(x_4, x_5) f_{bd}(x_4)$$
Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate $Z (G) \triangleq \sum_{x} g (x)$

```
f_{bd} (x_4) = \sum_{x_1} f_b (x_1, x_4) f_d (x_1)

f_{ce} (x_3, x_5) = \sum_{x_2} f_c (x_2, x_3, x_5) f_e (x_2)

f_{cef} (x_5) = \sum_{x_3} f_{ce} (x_3, x_5) f_f (x_3)

f_{abd} (x_5) = \sum_{x_4} f_a (x_4, x_5) f_{bd} (x_4)

Z = f_{abcdef} = \sum_{x_5} f_{abd} (x_5) f_{cef} (x_5)
```
Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate $Z(G) \triangleq \sum x g(x)$

$\begin{align*}
\text{abcdef} \\
\text{f}_{bd}(x_4) &= \sum_{x_1} f_b(x_1, x_4) f_d(x_1) \\
\text{f}_{ce}(x_3, x_5) &= \sum_{x_2} f_c(x_2, x_3, x_5) f_e(x_2) \\
\text{f}_{cef}(x_5) &= \sum_{x_3} f_{ce}(x_3, x_5) f_f(x_3) \\
\text{f}_{abd}(x_5) &= \sum_{x_4} f_a(x_4, x_5) f_{bd}(x_4) \\
Z &= f_{abcdef} = \sum_{x_5} f_{abd}(x_5) f_{cef}(x_5)
\end{align*}$

Sum-Product Algorithm on a normal factor graph with no cycles
Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate $Z(G) \triangleq \sum_x g(x)$

Sum-Product Algorithm on a normal factor graph with no cycles

$f_{bd} = \langle f_b, f_d \rangle_{x_1}$

$f_{ce} = \langle f_c, f_e \rangle_{x_2}$

$f_{cef} = \langle f_{ce}, f_f \rangle_{x_3}$

$f_{abd} = \langle f_a, f_{bd} \rangle_{x_4}$

$Z = f_{abcdef} = \langle f_{abd}, f_{cef} \rangle_{x_5}$
Sum-Product Algorithm for Acyclic Factor Graphs

Target: Calculate $Z(G) \triangleq \sum_{x,x'} g(x,x')$

$Z = f_{abcdef} = \langle f_{abd}, f_{cef} \rangle \mathcal{L}_H(x_5)$

$f_{bd} = \langle f_b, f_d \rangle \mathcal{L}_H(x_1)$

$f_{ce} = \langle f_c, f_e \rangle \mathcal{L}_H(x_2)$

$f_{cef} = \langle f_{ce}, f_f \rangle \mathcal{L}_H(x_3)$

$f_{abd} = \langle f_a, f_{bd} \rangle \mathcal{L}_H(x_4)$
More generally, we are applying following two rules:

\[
\begin{align*}
    m_{i \rightarrow a} &\leftarrow \bigotimes_{b \in \partial i \setminus \{a\}} m_{b \rightarrow i} (x_i, x_i') \\
    i &\rightarrow a
\end{align*}
\]
More generally, we are applying following two rules:

\[
m_i \rightarrow a \leftarrow \bigotimes_{b \in \partial i \setminus \{a\}} m_b \rightarrow i (x_i, x'_i)
\]

\[
m_a \rightarrow i \leftarrow \left\langle \bigotimes_{j \in \partial a \setminus \{i\}} m_j \rightarrow a, f_a \right\rangle_{\partial a \setminus \{i\}}
\]
More generally, we are applying following two rules:

\[ m_{i \rightarrow a} \leftarrow \bigotimes_{b \in \partial i \setminus \{a\}} m_{b \rightarrow i}(x_i, x_i') \]

\[ m_{a \rightarrow i} \leftarrow \bigotimes_{j \in \partial a \setminus \{i\}} m_{j \rightarrow a}, f_a \bigg|_{\partial a \setminus \{i\}} \]

with initialization at the leaf factors

\[ m_{a \rightarrow i}(x_i, x_i') = f_a(x_i, x_i') \quad \text{where } \{i\} = \partial a \]
For general QFGs with cycles

Definition 2 (General Sum-Product / Belief Propagation Algorithm for QFG)
For a general QFG $G = \{F, V, E\}$ with global function 

$$g (x, x') = \prod_{a \in F} f_a (x_{\partial a}, x'_{\partial a}) \prod_{i \in V} h_i (x_i, x'_i).$$

(1)

Update rules for belief propagation (BP) algorithm:

$$m_{a \rightarrow i}^{(t+1)} \propto \langle \bigotimes_{j \in \partial a \setminus \{i\}} m_j^{(t)} \cdot f_a \rangle$$

$$m_{i \rightarrow a}^{(t+1)} \propto h_i \cdot \bigotimes_{b \in \partial i \setminus \{a\}} m_{b \rightarrow i}^{(t)}$$

(2) (3)

The messages are said to be fixed-point messages when above equations holds without time-stamp superscripts.
For general QFGs with cycles

Definition 2 (General Sum-Product / Belief Propagation Algorithm for QFG)

For a general QFG $G = \{F, V, E\}$ with global function

$$g(x, x') = \prod_{a \in F} f_a(x_{\partial a}, x'_{\partial a}) \prod_{i \in V} h_i(x_i, x'_i).$$

(1)

Update rules for belief propagation (BP) algorithm:

$$m_{a \rightarrow i} \propto \left\langle \bigotimes_{j \in \partial a \setminus \{i\}} m_{j \rightarrow a}, f_a \right\rangle L_h(x_{\partial a \setminus \{i\}})$$

(2)

$$m_{i \rightarrow a} \propto h_i \cdot \bigotimes_{b \in \partial i \setminus \{a\}} m_{b \rightarrow i}$$

(3)

The messages are said to be fixed-point messages when above equations holds without time-stamp superscripts.
Loop Calculus for SP / BP Algorithms

Theorem 3 (Loop Calculus [Chertkov and Chernyak, 2006, Mori, 2015a, Mori, 2015b])

At BP-fixed point, we have

\[ Z \triangleq \left\langle \bigotimes_{a \in \mathcal{F}} f_a, \bigotimes_{i \in \mathcal{V}} h_i \right\rangle_{\mathcal{L}(\mathcal{X}_\mathcal{V})} = Z_{\text{Bethe}} \left( 1 + \sum_{E \in \mathcal{E}'} \mathcal{K}(E) \right) \]

where the extended loop set is defined as

\[ \mathcal{E}' \triangleq \{ E \subset \mathcal{E} \setminus \{ \phi \} : d_i(E) \neq 1 \forall i \in \mathcal{V}, d_a(E) \neq 1 \forall i \in \mathcal{F} \} \]

where \( \mathcal{K}(E) \) is some function depending on \( E \), and \( \mathcal{K}(\phi) = 1 \), and

\[ Z_{\text{Bethe}} \triangleq \frac{\prod_{a \in \mathcal{F}} Z_a \prod_{i \in \mathcal{V}} Z_i}{\prod_{(i,a) \in \mathcal{E}} Z_{i,a}} = \prod_{a \in \mathcal{F}} \left\langle \bigotimes_{i \in \partial a} m^{(t)}_{i \rightarrow a}, f_a \right\rangle \prod_{i \in \mathcal{V}} \left\langle h_i, \bigotimes_{a \in \partial i} m^{(t)}_{a \rightarrow i} \right\rangle \prod_{(i,a) \in \mathcal{E}} \left\langle m_{a \rightarrow i}, m_{i \rightarrow a} \right\rangle. \]
Loop Calculus for BP Algorithms

Interpretation

Bethe Approximation is exact for acyclic QFG;
Loop Calculus for BP Algorithms

Interpretation

Bethe Approximation is exact for acyclic QFG;
Bethe Approximation is close to the exact value for QFGs with small number of cycles.
Outline

1. Classical Factor Graphs
   - Modeling
   - “Closing-the-box” Operation

2. Quantum Factor Graphs
   - A Motivating Example
   - Quantum Factor Graph
   - Construction of a QNFG
   - Several Examples

3. Problem of Calculating the Partition Sum
   - Sum-Product / Belief Propagation Algorithm
   - Exploration on Variational Approach

4. End Matters
Variational Approach for Classic Factor Graphs

Target: Calculate $Z(G) \triangleq \sum_{x} g(x)$, where $g(x) = \prod_{a \in F} f_a(x_a) \prod_{i \in V} h_i(x_i)$

I: Calculate $F_H \triangleq -\ln Z(G)$;
Variational Approach for Classic Factor Graphs

Target: Calculate \( Z(\mathcal{G}) \triangleq \sum_x g(x) \), where \( g(x) = \prod_{a \in \mathcal{F}} f_a(x_a) \prod_{i \in \mathcal{V}} h_i(x_i) \)

I: Calculate \( F_H \triangleq -\ln Z(\mathcal{G}); \)
II: Minimize \( F_{\text{Gibbs}}(b) \) over all possible global probability function \( b(x); \)

\[
\min_{b \text{ is a probability function}} F_{\text{Gibbs}}(b) \triangleq - \sum_{a \in \mathcal{F}} \sum_x b(x) \ln f_a(x_{\partial a}) \\
- \sum_{i \in \mathcal{V}} \sum_x b(x) \ln h_i(x_i) \\
+ \sum_x b(x) \ln b(x) \\
= F_H + \mathcal{D}(b \parallel p) \geq F_H.
\]
Variational Approach for Classic Factor Graphs

Target: Calculate $Z(G) \triangleq \sum_x g(x)$, where $g(x) = \prod_{a \in F} f_a(x_a) \prod_{i \in V} h_i(x_i)$

I: Calculate $F_H \triangleq -\ln Z(G)$;

II: Minimize $F_{\text{Gibbs}}(b)$ over all possible global probability function $b(x)$;

III: Minimize $F_{\text{Bethe}}(\{b_a\}_{a \in F}, \{b_i\}_{i \in V})$ over all valid marginal probability functions $\{b_a\}_{a \in F}, \{b_i\}_{i \in V}$;

$$F_{\text{Bethe}}((b_a)_{a \in F}, (b_i)_{i \in V}) \triangleq -\sum_{a \in F} \sum_{x_{\partial a}} b_a(x_{\partial a}) \ln f_a(x_{\partial a}) - \sum_{i \in V} \sum_{x_i} b_i(x_i) \ln h_i(x_i)$$

$$+ \sum_{a \in F} \sum_{x_{\partial a}} b_a(x_{\partial a}) \ln b_a(x_{\partial a})$$

$$- \sum_{i \in V} (d_i - 1) \sum_{x_i} b_i(x_i) \ln b_i(x_i)$$

$$= F_{\text{Gibbs}} \text{ for acyclic factor graphs.}$$
Variational Approach for Classic Factor Graphs

Target: Calculate $Z(G) \triangleq \sum_{x} g(x)$, where $g(x) = \prod_{a \in F} f_a(x_a) \prod_{i \in V} h_i(x_i)$

I: Calculate $F_{H} \triangleq - \ln Z(G)$;
II: Minimize $F_{\text{Gibbs}}(b)$ over all possible global probability function $b(x)$;
III: Minimize $F_{\text{Bethe}} \left( \{b_a\}_{a \in F}, \{b_i\}_{i \in V} \right)$ over all valid marginal probability functions $\{b_a\}_{a \in F}, \{b_i\}_{i \in V}$;
IV: Study the Stationary Condition of above optimization problem, which turned out to be equivalent to

$$\min_{b_a, b_i \text{ probability functions}} F_{\text{Bethe}} \left( \{b_a\}_{a \in F}, \{b_i\}_{i \in V} \right)$$

s.t. $\sum_{x \partial a \setminus \{i\}} b_a(x_a) = b_i(x_i) \quad \forall (i, a) \in \mathcal{E}, \forall x_i \in \mathcal{X}_i$
Variational Approach for Classic Factor Graphs

Target: Calculate $Z(G) \triangleq \sum_x g(x)$, where $g(x) = \prod_{a \in \mathcal{F}} f_a(x_a) \prod_{i \in \mathcal{V}} h_i(x_i)$

I: Calculate $F_H \triangleq -\ln Z(G)$;

II: Minimize $F_{\text{Gibbs}}(b)$ over all possible global probability function $b(x)$;

III: Minimize $F_{\text{Bethe}}(\{b_a\}_{a \in \mathcal{F}}, \{b_i\}_{i \in \mathcal{V}})$ over all valid marginal probability functions $\{b_a\}_{a \in \mathcal{F}}, \{b_i\}_{i \in \mathcal{V}}$;

IV: Study the Stationary Condition of above optimization problem, which turned out to be equivalent to

$$b_a \propto f_a \cdot \prod_{i \in \partial a} m_{i \rightarrow a}$$

$$b_i \propto h_1 \cdot \prod_{a \in \partial i} m_{a \rightarrow i}$$

$$\sum_{x_{\partial a \setminus \{i\}}} b_a(x_a) = b_i(x_i) \quad \forall (i, a) \in \mathcal{E}, \forall x_i \in \mathcal{X}_i$$
Variational Approach for Classic Factor Graphs

Target: Calculate $Z(\mathcal{G}) \triangleq \sum_x g(x)$, where $g(x) = \prod_{a \in \mathcal{F}} f_a(x_a) \prod_{i \in \mathcal{V}} h_i(x_i)$

I: Calculate $F_H \triangleq -\ln Z(\mathcal{G})$;
II: Minimize $F_{\text{Gibbs}}(b)$ over all possible global probability function $b(x)$;
III: Minimize $F_{\text{Bethe}}(\{b_a\}_{a \in \mathcal{F}}, \{b_i\}_{i \in \mathcal{V}})$ over all valid marginal probability functions $\{b_a\}_{a \in \mathcal{F}}, \{b_i\}_{i \in \mathcal{V}}$;
IV: Study the Stationary Condition of above optimization problem, which turned out to be equivalent to

$$b_a \propto f_a \cdot \prod_{i \in \partial a} m_{i \rightarrow a}$$
$$b_i \propto h_1 \cdot \prod_{a \in \partial i} m_{a \rightarrow i}$$

$$m_{i \rightarrow a} \leftarrow \bigotimes_{b \in \partial i \setminus \{a\}} m_{b \rightarrow i}(x_i, x'_i); \quad m_{a \rightarrow i} \leftarrow \bigg\langle \bigotimes_{j \in \partial a \setminus \{i\}} m_{j \rightarrow a}, f_a \bigg\rangle_{\partial a \setminus \{i\}}$$
Exploration of Variational Approach for QFGs

Definition 4 (Helmholtz free energy and Gibbs free energy for QFGs)

\[
F_H \triangleq - \ln Z(G)
\]

\[
F_{\text{Gibbs}}(b(x, x')) \triangleq - \sum_{a \in \mathcal{F}} \langle \tilde{b}_a, \ln(f_a) \rangle - \sum_{i \in \mathcal{V}} \langle \tilde{b}_i, \ln(h_i) \rangle + \langle b, \ln(b) \rangle
\]
Exploration of Variational Approach for QFGs

Definition 4 (Helmholtz free energy and Gibbs free energy for QFGs)

\[ F_{\text{H}} \triangleq -\ln Z(G) \]

\[ F_{\text{Gibbs}}(b(x, x')) \triangleq -\sum_{a \in F} \langle \tilde{b}_a, \ln(f_a) \rangle - \sum_{i \in V} \langle \tilde{b}_i, \ln(h_i) \rangle + \langle b, \ln(b) \rangle \]

where the induced \( \{ \tilde{b}_a \}_{a \in F} \) and \( \{ \tilde{b}_i \}_{a \in V} \) are defined as

\[ \tilde{b}_a \triangleq \langle 1, b \rangle _{L_h(x_{V \setminus \{\partial a\}})} , \quad \tilde{b}_i \triangleq \langle 1, b \rangle _{L_h(x_{V \setminus \{i\}})} . \quad (4) \]
Exploration of Variational Approach for QFGs

Definition 4 (Helmholtz free energy and Gibbs free energy for QFGs)

\[ F_H \triangleq -\ln Z (G) \]
\[ F_{\text{Gibbs}} (b(x, x')) \triangleq - \sum_{a \in \mathcal{F}} \langle \tilde{b}_a, \ln (f_a) \rangle - \sum_{i \in \mathcal{V}} \langle \tilde{b}_i, \ln (h_i) \rangle + \langle b, \ln (b) \rangle \]

where the induced \( \{ \tilde{b}_a \}_{a \in \mathcal{F}} \) and \( \{ \tilde{b}_i \}_{a \in \mathcal{V}} \) are defined as

\[ \tilde{b}_a \triangleq \langle 1, b \rangle \mathcal{L}_h (x_{\mathcal{V} \setminus \partial a}) , \quad \tilde{b}_i \triangleq \langle 1, b \rangle \mathcal{L}_h (x_{\mathcal{V} \setminus \{i\}}) . \] (4)

Here, \( \ln (\cdot) \) is performed on matrix level, i.e.,

\[ \ln \left( \mathbf{U} \Lambda \mathbf{U}^H \right) \triangleq \mathbf{U} \text{diag} \left( \{ \ln \Lambda_{k,k} \}_{k} \right) \mathbf{U}^H \]

for any PSD matrix \( \mathbf{U} \Lambda \mathbf{U}^H \), where \( \mathbf{U} \) is unitary and \( \Lambda \) is a non-negative diagonal matrix.
Definition 4 (Helmholtz free energy and Gibbs free energy for QFGs)

\[
F_H \triangleq -\ln \mathbb{Z}(\mathcal{G})
\]

\[
F_{\text{Gibbs}}(b(x, x')) \triangleq -\sum_{a \in \mathcal{F}} \langle \tilde{b}_a, \ln(f_a) \rangle - \sum_{i \in \mathcal{V}} \langle \tilde{b}_i, \ln(h_i) \rangle + \langle b, \ln(b) \rangle
\]

where the induced \( \{\tilde{b}_a\}_{a \in \mathcal{F}} \) and \( \{\tilde{b}_i\}_{i \in \mathcal{V}} \) are defined as

\[
\tilde{b}_a \triangleq \langle 1, b \rangle_{\mathcal{L}_h(\mathcal{X}_{\mathcal{V}\backslash \partial a})}, \quad \tilde{b}_i \triangleq \langle 1, b \rangle_{\mathcal{L}_h(\mathcal{X}_{\mathcal{V}\backslash \{i\}})}.
\] (4)

Here, \( \ln(\cdot) \) is performed on matrix level, i.e.,

\[
\ln\left(\mathbf{U}\Lambda\mathbf{U}^H\right) \triangleq \mathbf{U}\text{diag}\left(\{\ln\Lambda_k,k\}_k\right)\mathbf{U}^H
\]

for any PSD matrix \( \mathbf{U}\Lambda\mathbf{U}^H \), where \( \mathbf{U} \) is unitary and \( \Lambda \) is a non-negative diagonal matrix.
Theorem 5

*We have following relationship between Gibbs free energy and Helmholtz free energy*

\[
F_{Gibbs}(b(x, x')) = F_H + D(b \parallel p)
\]

(5)

*where \( D(b \parallel p) \) of two normalized PSD operator is defined as*

\[
D(b \parallel p) \triangleq \langle b, \ln(b) \rangle_{L_h} - \langle b, \ln(p) \rangle_{L_h}
\]

(6)

*and the quantum probability \( p \) is the normalized global function, i.e.,*

\[
p(x, x') = \frac{1}{Z(G)} g(x, x').
\]
Exploration on Variational Approach for QFGs

Theorem 5

We have following relationship between Gibbs free energy and Helmholtz free energy

\[ F_{\text{Gibbs}}(b(\mathbf{x}, \mathbf{x}')) = F_H + D(b \parallel p) \]  \hspace{1cm} (5)

where \( D(b \parallel p) \) of two normalized PSD operator is defined as

\[ D(b \parallel p) \triangleq \langle b, \ln(b) \rangle_{\mathcal{L}_h} - \langle b, \ln(p) \rangle_{\mathcal{L}_h} \]  \hspace{1cm} (6)

and the quantum probability \( p \) is the normalized global function, i.e., \( p(\mathbf{x}, \mathbf{x}') = \frac{1}{Z(G)} g(\mathbf{x}, \mathbf{x}') \).

Lemma 6 (Non-negativity of Von Neumann Divergence)

\[ D(b \parallel p) \geq 0 \quad "=" \quad b = q \]  \hspace{1cm} (7)
Definition 7 (Approximate $F_{\text{Gibbs}}$)

For above QGF, the *Bethe free energy* w.r.t. marginal beliefs $(b_a)_{a \in \mathcal{F}}, (b_i)_{i \in \mathcal{V}}$ is defined as

$$F_{\text{Bethe}}((b_a)_{a \in \mathcal{F}}, (b_i)_{i \in \mathcal{V}}) \triangleq -\sum_{a \in \mathcal{F}} \langle b_a, \ln(f_a) \rangle - \sum_{i \in \mathcal{V}} \langle b_i, \ln(h_i) \rangle + \sum_{a \in \mathcal{F}} \langle b_a, \ln(b_a) \rangle - \sum_{i \in \mathcal{V}} (d_i - 1) \langle b_i, \ln(b_i) \rangle.$$

Here, $\{b_a\}_{a \in \mathcal{F}}$ and $\{b_i\}_{i \in \mathcal{V}}$ are some given normalized PSD operators on $\{\mathcal{X}_a\}_{a \in \mathcal{F}}$ and $\{\mathcal{X}_i\}_{i \in \mathcal{V}}$, respectively.
Definition 7 (Approximate $F_{\text{Gibbs}}$)

For above QGF, the \textit{Bethe free energy} w.r.t. marginal beliefs $(b_a)_{a \in \mathcal{F}}, (b_i)_{i \in \mathcal{V}}$ is defined as

$$F_{\text{Bethe}} \left( (b_a)_{a \in \mathcal{F}}, (b_i)_{i \in \mathcal{V}} \right) \triangleq - \sum_{a \in \mathcal{F}} \langle b_a, \ln(f_a) \rangle - \sum_{i \in \mathcal{V}} \langle b_i, \ln(h_i) \rangle$$

$$+ \sum_{a \in \mathcal{F}} \langle b_a, \ln(b_a) \rangle - \sum_{i \in \mathcal{V}} (d_i - 1) \langle b_i, \ln(b_i) \rangle.$$ 

Here, $\{b_a\}_{a \in \mathcal{F}}$ and $\{b_i\}_{i \in \mathcal{V}}$ are some given normalized PSD operators on $\{x_{\partial a}\}_{a \in \mathcal{F}}$ and $\{x_i\}_{i \in \mathcal{V}}$, respectively.

Question: Does this “approximation” make sense?
Definition 7 (Approximate $F_{\text{Gibbs}}$)

For above QGF, the Bethe free energy w.r.t. marginal beliefs $(b_a)_{a \in \mathcal{F}}, (b_i)_{i \in \mathcal{V}}$ is defined as

\[
F_{\text{Bethe}} \left( (b_a)_{a \in \mathcal{F}}, (b_i)_{i \in \mathcal{V}} \right) \triangleq - \sum_{a \in \mathcal{F}} \langle b_a, \ln (f_a) \rangle - \sum_{i \in \mathcal{V}} \langle b_i, \ln (h_i) \rangle + \sum_{a \in \mathcal{F}} \langle b_a, \ln (b_a) \rangle - \sum_{i \in \mathcal{V}} \left( d_i - 1 \right) \langle b_i, \ln (b_i) \rangle.
\]

Here, \( \{b_a\}_{a \in \mathcal{F}} \) and \( \{b_i\}_{i \in \mathcal{V}} \) are some given normalized PSD operators on \( \{X_{\partial a}\}_{a \in \mathcal{F}} \) and \( \{X_i\}_{i \in \mathcal{V}} \), respectively.

Question: Does this “approximation” make sense?

Given arbitrary compatible $(b_a)_{a \in \mathcal{F}}, (b_i)_{i \in \mathcal{V}}$, can we always find a corresponding global quantum belief matrix $b$?
Lemma 8

Given an acyclic QFG, for any marginal quantum belief \( \{ b_a \}_{a \in \mathcal{F}} \) and \( \{ b_i \}_{i \in \mathcal{V}} \) with compatibility constrain

\[
b_i (x_i, x'_i) = \sum_{x_{\partial a \setminus \{i\}}, x'_{\partial a \setminus \{i\}}} b_a (x_{\partial a}, x'_{\partial a}) \quad \forall (i, a) \in \mathcal{E}
\]  

\[ (8) \]

\[ \exists b \in \mathcal{L}_H (\mathcal{X}), \text{ normalized with} \]

\[
b_a \triangleq \langle 1, b \rangle_{\mathcal{L}_h (\mathcal{X}_\mathcal{V} \setminus \partial a)} , \quad b_i \triangleq \langle 1, b \rangle_{\mathcal{L}_h (\mathcal{X}_\mathcal{V} \setminus \{i\})} .
\]
Lemma 8

Given an acyclic QFG, for any marginal quantum belief \( \{ b_a \}_{a \in F} \) and \( \{ b_i \}_{i \in V} \) with compatibility constrain

\[
b_i(\mathbf{x}_i, \mathbf{x'}_i) = \sum_{\mathbf{x}_{\partial a \setminus \{i\}}, \mathbf{x'_{\partial a \setminus \{i\}}} } b_a(\mathbf{x}_{\partial a}, \mathbf{x'}_{\partial a}) \quad \forall (i, a) \in \mathcal{E} \tag{8}
\]

\( \exists b \in L_{H}(\mathcal{X}{'}), \) normalized with

\[
b_a \triangleq \langle \mathbf{1}, b \rangle_{L_{h}(\mathcal{X}_V \setminus \{a\})} , \quad b_i \triangleq \langle \mathbf{1}, b \rangle_{L_{h}(\mathcal{X}_V \setminus \{i\})} .
\]

Conjecture 9

The operator \( b \) above is positive semi-definite.
Exploration on Variational Approach for QFGs

Lemma 8

Given an acyclic QFG, for any marginal quantum belief \( \{ b_a \}_{a \in F} \) and \( \{ b_i \}_{i \in V} \) with compatibility constrain

\[
b_i(x_i, x'_i) = \sum_{x_{\partial a \setminus \{i\}}, x'_{\partial a \setminus \{i\}}} b_a(x_{\partial a}, x'_{\partial a}) \quad \forall (i, a) \in E
\]  

\( 8 \)

\( \exists b \in \mathcal{L}_H(\mathcal{X}) \), normalized with

\[
b_a \triangleq \langle 1, b \rangle_{\mathcal{L}_h(\mathcal{X}_{V \setminus \{a\}})}, \quad b_i \triangleq \langle 1, b \rangle_{\mathcal{L}_h(\mathcal{X}_{V \setminus \{i\}})}.
\]

Conjecture 9

The operator \( b \) above is positive semi-definite. In such case,

\[
\min_b F_{\text{Gibbs}}(b) =
\]
Exploration on Variational Approach for QFGs

Lemma 8

Given an acyclic QFG, for any marginal quantum belief \( \{ b_a \}_{a \in \mathcal{F}} \) and \( \{ b_i \}_{i \in \mathcal{V}} \) with compatibility constrain

\[
\forall (i, a) \in \mathcal{E} \quad b_i (x_i, x_i') = \sum_{x_{\partial a} \setminus \{i\}, x_{\partial a} \setminus \{i\}} b_a (x_{\partial a}, x_{\partial a}')
\]  

\( \exists b \in \mathcal{L}_H (\mathcal{X}) \), normalized with

\[
b_a \triangleq \langle 1, b \rangle_{\mathcal{L}_h (\mathcal{X} \setminus \{\partial a\})}, \quad b_i \triangleq \langle 1, b \rangle_{\mathcal{L}_h (\mathcal{X} \setminus \{i\})}.
\]

Conjecture 9

The operator \( b \) above is positive semi-definite. In such case,

\[
\min_b F_{\text{Gibbs}} (b) = \min_{\{b_a\}_{a \in \mathcal{F}}, \{b_i\}_{i \in \mathcal{V}}} F_{\text{Bethe}} \left( \{b_a\}_{a \in \mathcal{F}}, \{b_i\}_{i \in \mathcal{V}} \right)
\]

s.t. Equation (8) holds.
Conjecture 10

The stationary condition of optimization problem

\[
\min \left\{ b \right\}_{a \in F}, \left\{ b \right\}_{i \in V} F_{\text{Bethe}} \left( (b_a)_{a \in F}, (b_i)_{i \in V} \right)
\]

s.t. \( b_i (x_i, x'_i) = \sum_{a \in \partial_i \setminus \{i\}} b_a (x_{\partial a}, x'_{\partial a}) \) \( \forall (i, a) \in E \)
Conjecture 10

The stationary condition of optimization problem

\[
\min_{\{b_a\}_{a \in \mathcal{F}}, \{b_i\}_{i \in \mathcal{V}}} F_{\text{Bethe}} ((b_a)_{a \in \mathcal{F}}, (b_i)_{i \in \mathcal{V}})
\]

\[
\text{s.t. } b_i (x_i, x_i') = \sum_{x_{\partial a}\setminus \{i\}, x_{\partial a}'\setminus \{i\}} b_a (x_{\partial a}, x_{\partial a}') \quad \forall (i, a) \in \mathcal{E}
\]

is equivalent to

\[
b_a (x_{\partial a}, x_{\partial a}') \triangleq \frac{1}{Z_a} f_a (x_{\partial a}, x_{\partial a}') \prod_{i \in \partial a} m_{i \rightarrow a} (x_i, x_i'),
\]

\[
b_i (x_i, x_i') \triangleq \frac{1}{Z_i} h_i (x_i, x_i') \prod_{a \in \partial i} m_{a \rightarrow i} (x_i, x_i').
\]

for some fixed-point messages.
Outlooks

Continue exploration on Variational Approach;
Outlooks

Continue exploration on Variational Approach;

Details in Loop Calculus: Any bound for $\mathcal{K}(E)$?
Outlooks

Continue exploration on Variational Approach;

Details in Loop Calculus: Any bound for $\mathcal{K}(E)$?

Looking for practical class of factorizations where this model can be applied.
References


1 Classical Factor Graphs
   • Modeling
   • “Closing-the-box” Operation

2 Quantum Factor Graphs
   • A Motivating Example
   • Quantum Factor Graph
   • Construction of a QNFG
   • Several Examples

3 Problem of Calculating the Partition Sum
   • Sum-Product / Belief Propagation Algorithm
   • Exploration on Variational Approach

4 End Matters