

通信のための確率推論の基礎

林 和則

大阪市立大学

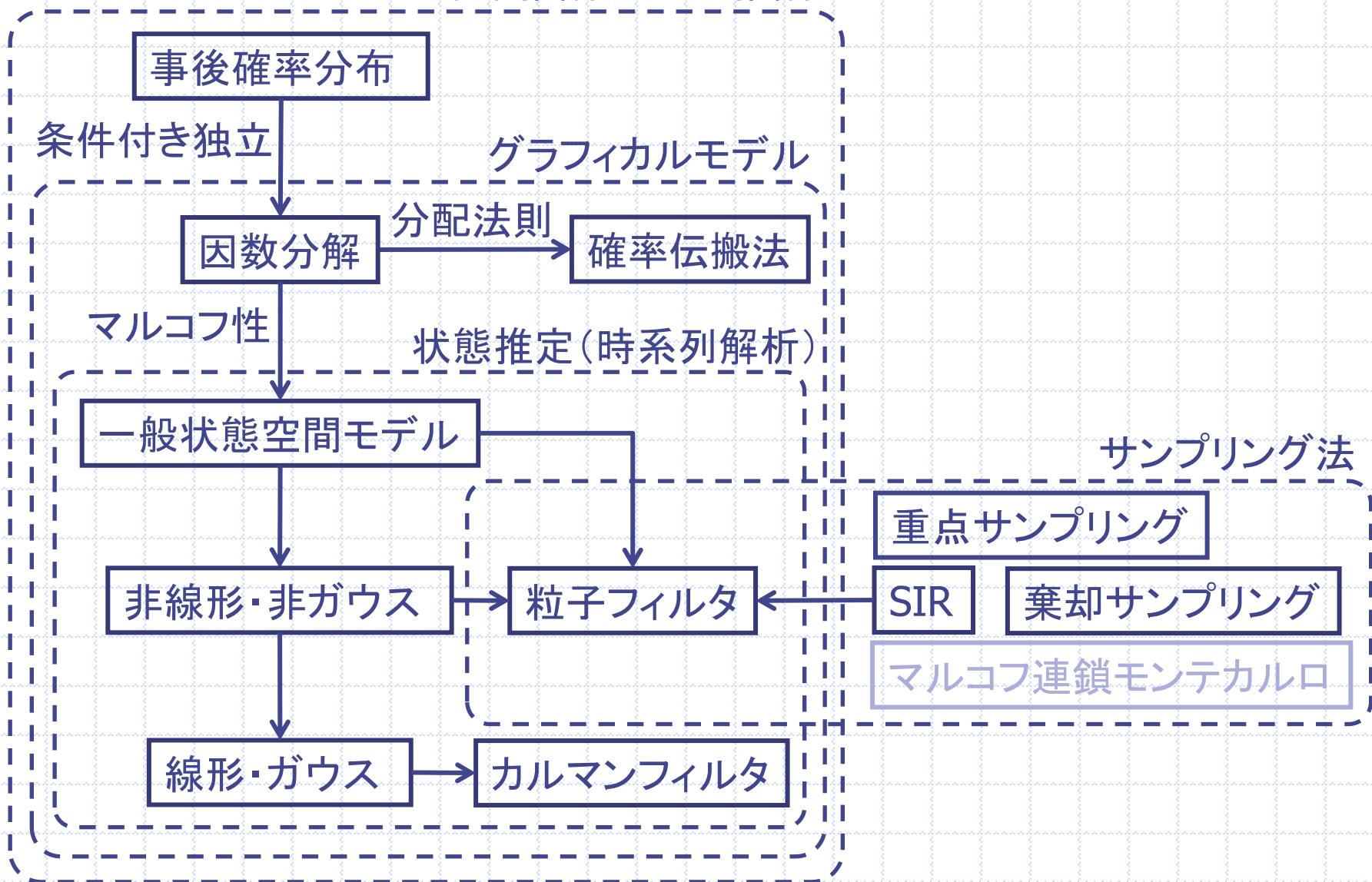
E-mail: kazunori@eng.osaka-cu.ac.jp

目次

- ◆ 基礎事項の確認
- ◆ 確率推論問題
- ◆ 状態推定
- ◆ 確率伝搬法
- ◆ 確率伝搬法の応用例

お話しする内容の関係

確率推論(ベイズ推論)



目次

- ◆ 基礎事項の確認
- ◆ 確率推論問題
- ◆ 状態推定
- ◆ 確率伝搬法
- ◆ 確率伝搬法の応用例

離散確率変数の基本法則

離散確率変数： X, Y (実現値 x_i ($i = 1, \dots, M$), y_j ($j = 1, \dots, L$))

確率分布： $P(x_i) = \Pr(X = x_i)$

同時確率分布： $P(x_i, y_j) = \Pr(X = x_i, Y = y_j)$

条件付確率分布： $P(y_j|x_i) = \Pr(Y = y_j|X = x_i)$

確率変数は大文字
その実現値は小文字
(スカラーとベクトル
を表記上区別しないの
で注意)

確率の基本法則：

加法定理： $P(x) = \sum_y P(x, y)$ 周辺化

乗法定理： $P(x, y) = P(y|x)P(x)$

$P(x, y) = P(y)P(x)$ が成り立つとき X, Y は独立であるという

ベイズ則：

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x|y)P(y)}{\sum_y P(x|y)P(y)}$$

連続確率変数の基本法則

連続確率変数：任意の $c \in \mathbb{R}$ に対して $\Pr(X = c) = 0$ となる確率変数 X

確率密度関数： $p(x)$ $\Pr(a \leq X \leq b) = \int_a^b p(x)dx$

確率密度関数の性質： $p(x) \geq 0, \int_{-\infty}^{\infty} p(x)dx = 1$

累積分布関数： $P(a)$ $\Pr(X \leq a) = P(a) = \int_{-\infty}^a p(x)dx$

確率の基本法則：

加法定理： $p(x) = \int p(x, y)dy$ 周辺化

乗法定理： $p(x, y) = p(y|x)p(x)$

独立性の定義、およびベイズ則も離散確率変数の場合と同様
(確率分布を密度関数に、和を積分に置き換える)

条件付き独立

確率変数 X, Y, Z の全ての実現値 x, y, z について,

$$P(x, y|z) = P(x|z)P(y|z)$$

が成立するとき、確率変数 X, Y は Z を与えた下で条件付き独立
といい $X \perp Y|Z$ と書く

条件付き独立の性質の例：

$X \perp Y|Z$ のとき

$$P(x, y, z) = \frac{P(x, z)P(y, z)}{P(z)}$$

$$P(x|y, z) = P(x|z)$$

ベイジアンネットワーク

確率変数 X_1, X_2, \dots, X_n の同時確率分布 $P(x_1, x_2, \dots, x_n)$ に対応するベイジアンネットワークは次の性質をみたす有向非巡回グラフ（矢印の向きにたどって同じノードに戻ることがない有向グラフ）

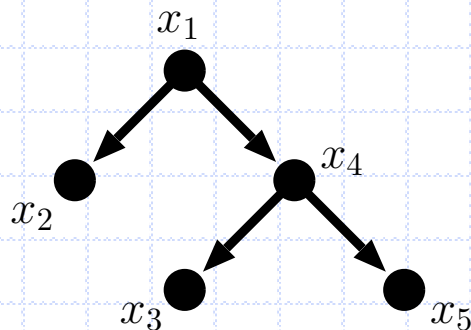
1. 確率変数が各ノードに対応
2. 同時確率分布の因数分解が

$$P(x_1, x_2, \dots, x_n) = P(x_1|S_1)P(x_2|S_2) \cdots P(x_n|S_n)$$

で与えられるとき、 S_i が x_i の親ノードの集合。

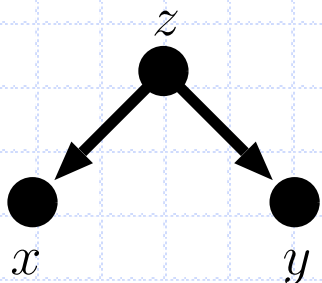
（すなわち、 $x_j \in S_i \rightarrow x_i$ に対応する有向エッジが存在）

具体例： $P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2|x_1)P(x_3|x_4)P(x_4|x_1)P(x_5|x_4)$



条件付き独立性：3ノードの例

(a) tail-to-tail:



$$P(x, y, z) = P(x|z)P(y|z)p(z)$$

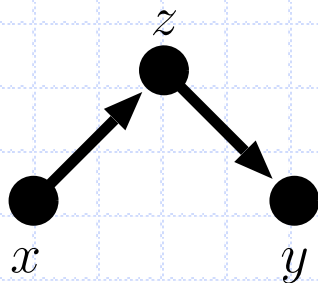
$$P(x, y) = \sum_z \{P(x|z)P(y|z)P(z)\}$$

独立でない

$$\begin{aligned} P(x, y|z) &= \frac{P(x, y, z)}{P(z)} \\ &= \frac{P(x|z)P(y|z)P(z)}{P(z)} \\ &= P(x|z)P(y|z) \end{aligned}$$

条件付き独立

(b) head-to-tail:



$$P(x, y, z) = P(x)P(y|z)P(z|x)$$

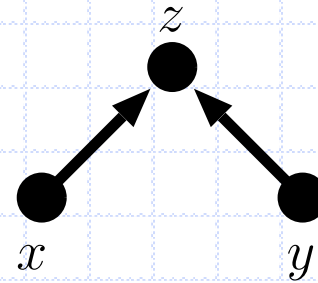
$$P(x, y) = \sum_z \{P(x)P(y|z)P(z|x)\}$$

独立でない

$$\begin{aligned} P(x, y|z) &= \frac{P(x)P(y|z)P(z|x)}{P(z)} \\ &= \frac{P(x, z)P(y|z)}{P(z)} \\ &= P(x|z)P(y|z) \end{aligned}$$

条件付き独立

(c) head-to-head:



$$P(x, y, z) = P(x)P(y)P(z|x, y)$$

$$\begin{aligned} P(x, y) &= \sum_z \{P(x)P(y)P(z|x, y)\} \\ &= P(x)P(y) \end{aligned}$$

独立

$$P(x, y|z) = \frac{P(x)P(y)P(z|x, y)}{P(z)}$$

条件付き独立でない

ファクターグラフ

多変数関数の因子分解を表すグラフ

多変数関数： $f(x_1, x_2, \dots, x_n) = f_1(A_1)f_2(A_2)\cdots f_m(A_m)$
 (A_1, A_2, \dots, A_m は x_1, x_2, \dots, x_n の部分集合)

関数ノード：ローカル関数 $f_i(A_i)$ に対応

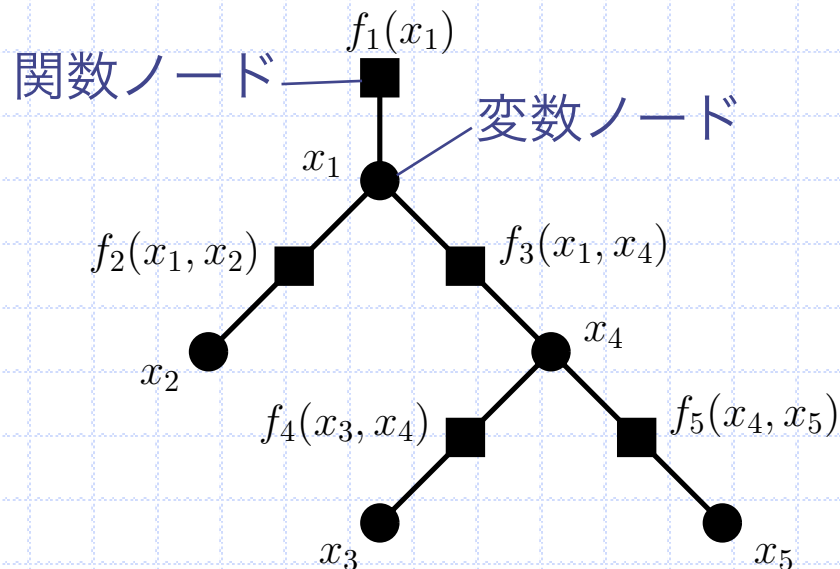
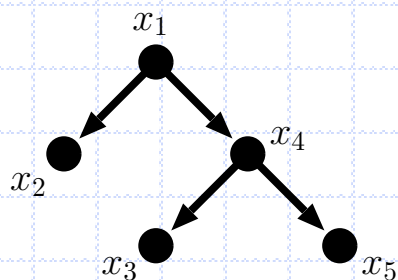
変数ノード：変数 x_j に対応

ノード接続ルール：

$f_i(A_i)$ に対応する関数ノードと A_i に含まれる変数に対応する変数ノードがエッジで接続される

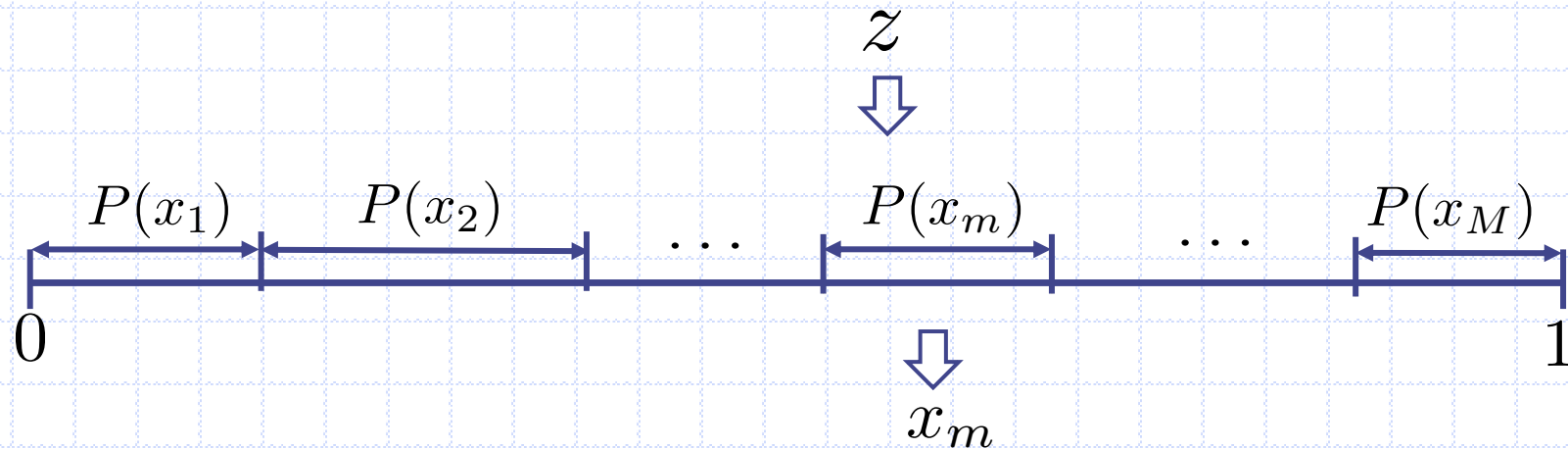
具体例：

$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5) &= P(x_1)P(x_2|x_1)P(x_3|x_4)P(x_4|x_1)P(x_5|x_4) \\ &= f_1(x_1)f_2(x_1, x_2)f_3(x_3, x_4)f_4(x_1, x_4)f_5(x_4, x_5) \end{aligned}$$



離散分布からのサンプル生成

- ・ 区間 $(0, 1)$ で一様に分布する乱数 z の発生源がある
- ・ 離散確率変数 X (実現値 $x_i (i = 1, \dots, M)$, 確率分布 $P(x_i)$) からのサンプルを生成したい
- ・ 区間 $(0, 1)$ を幅 $P(x_i)$ の M 区間に分割し, 発生した z が $P(x_m)$ の区間に落ち込んだら x_m をサンプルとして生成



連続分布からのサンプル生成：逆関数法

- ・ 区間 $(0, 1)$ で一様に分布する乱数 z の発生源がある
- ・ ある関数 $f(\cdot)$ を用いて $y = f(z)$ と変換する
- ・ 変換後の y が所望の分布に従うように $f(z)$ を選ぶ

- ・ y の確率密度関数： $p(y) = p(z) \left| \frac{dz}{dy} \right| \quad p(z) = 1$

$$z = h(y) \equiv \int_{-\infty}^y p(y') dy'$$

$$y = h^{-1}(z)$$

y の累積分布関数 $h(y)$ の逆関数に一様乱数 z を代入すればよい

逆関数法の例：指数分布の場合

• y の確率密度関数： $p(y) = \lambda \exp(-\lambda y), \quad 0 \leq y < \infty$

• y の累積分布関数： $z = h(y) = \int_{-\infty}^y p(y') dy'$

$$= \int_0^y \lambda \exp(-\lambda y') dy'$$
$$= \left[-\exp(-\lambda y') \right]_0^y$$
$$= 1 - \exp(-\lambda y)$$

• 逆関数（変換式）：

$$y = h^{-1}(z)$$
$$= -\frac{\ln(1 - z)}{\lambda}$$

累積分布関数が求まるか？

ガウス分布の場合：

逆関数が求まるか？

Marsaglia, Box-Muller

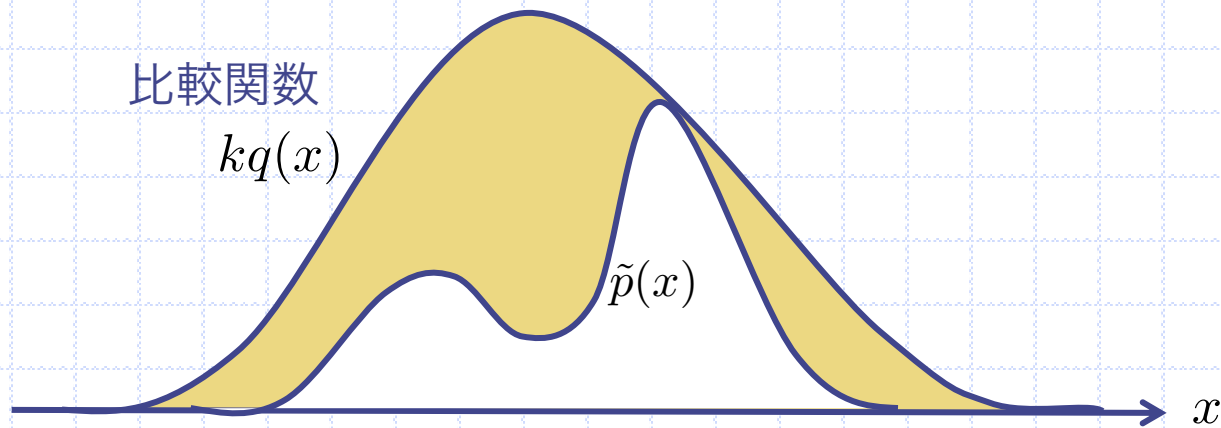
棄却サンプリング：準備

$p(x)$ からサンプリングしたいが、直接は困難

- 任意の x について、 $p(x)$ の値はある正規化定数 Z_p を除いて容易に分かる ($p(x) = \frac{1}{Z_p} \tilde{p}(x)$ で $\tilde{p}(x)$ は分かるけど $p(x)$ は分からない)
- 容易にサンプルを抽出できる分布 $q(x)$ を用意 (提案分布)
- 定数 k を導入し、全ての x に対して $kq(x) \geq \tilde{p}(x)$ となるようにする

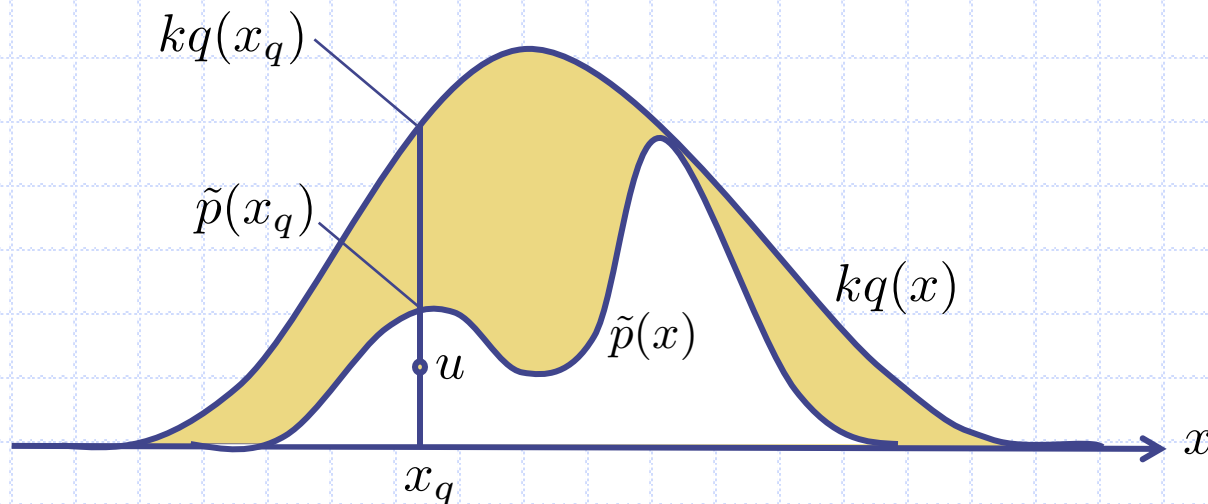
棄却サンプリング：準備

- $p(x)$ からサンプリングしたいが、直接は困難
- 任意の x について、 $p(x)$ の値はある正規化定数 Z_p を除いて容易に分かる ($p(x) = \frac{1}{Z_p} \tilde{p}(x)$ で $\tilde{p}(x)$ は分かるけど $p(x)$ は分からない)
- 容易にサンプルを抽出できる分布 $q(x)$ を用意 (提案分布)
- 定数 k を導入し、全ての x に対して $kq(x) \geq \tilde{p}(x)$ となるようにする



棄却サンプリング：手順

- ・ サンプルを抽出できる分布 $q(x)$ を用意（提案分布）
- ・ $q(x)$ に従う確率変数 X_q のサンプル（実現値） x_q を生成
- ・ 区間 $[0, kq(x_q)]$ の一様分布に従う確率変数 U のサンプル u を生成
- ・ $u > \tilde{p}(x_q)$ ならば x_q を棄却，そうでなければ x_q を保持
- ・ 保持された x_q は確率分布 $p(x)$ に従う確率変数のサンプル



棄却サンプリング：原理

- 保持された X_q の累積分布関数：

$$\begin{aligned}
 \Pr(X_q \leq a | \text{keep}) &= \frac{\Pr(X_q \leq a, \text{keep})}{\Pr(\text{keep})} \\
 &= \frac{\int_{-\infty}^{\infty} \int_0^{kq(x)} I(x \leq a, 0 \leq u \leq \tilde{p}(x)) q(x) \frac{1}{kq(x)} du dx}{\int_{-\infty}^{\infty} \int_0^{kq(x)} I(0 \leq u \leq \tilde{p}(x)) q(x) \frac{1}{kq(x)} du dx} \\
 &= \frac{\int_{-\infty}^a \frac{\tilde{p}(x)}{k} dx}{\int_{-\infty}^{\infty} \frac{\tilde{p}(x)}{k} dx} && I(\cdot) : \text{特性関数} \\
 &= \frac{\int_{-\infty}^a \frac{\tilde{p}(x)}{Z_p} dx}{\int_{-\infty}^{\infty} \frac{\tilde{p}(x)}{Z_p} dx} \\
 &= \int_{-\infty}^a p(x) dx && p(x) \text{ の累積分布関数}
 \end{aligned}$$

重点サンプリング：考え方

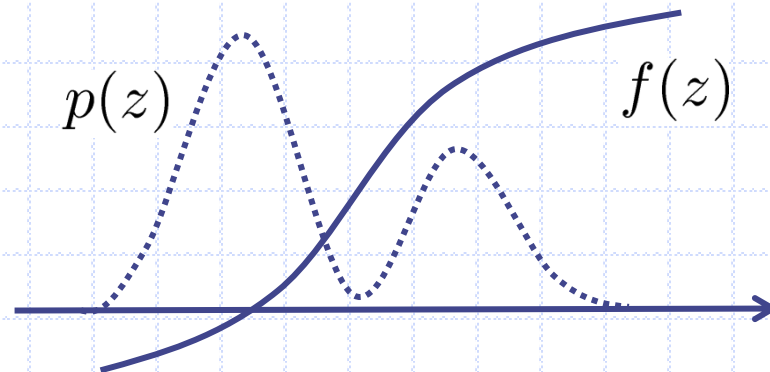
- ・ サンプリングを行う目的は多くの場合、ある関数 $f(z)$ の確率分布 $p(z)$ の下での期待値計算 $\mathbb{E}[f] = \int f(z)p(z)dz$

- ・ 重点サンプリングは期待値を直接近似する手法
($p(z)$ からのサンプル生成ではない)

- ・ z の空間を均一なグリッドで離散化すると $\mathbb{E}[f] \simeq \frac{1}{L} \sum_{l=1}^L f(z^{(l)})p(z^{(l)})$ と近似できるが、一様サンプリングは効率が悪い



$f(z)p(z)$ が大きな領域から
サンプル点を取りたい



重点サンプリング：手順

- 容易にサンプルを抽出できる分布 $q(z)$ を用意（提案分布）
- $q(z)$ からのサンプルを $\{z^{(l)}\}$ とする
- 重点サンプリングによる期待値の近似：

$$\begin{aligned}\mathbb{E}[f] &= \int f(z)p(z)dz \\ &= \int f(z)\frac{p(z)}{q(z)}q(z)dz \\ &\simeq \frac{1}{L} \sum_{l=1}^L f(z^{(l)})\frac{p(z^{(l)})}{q(z^{(l)})}\end{aligned}$$

$$r_l = \frac{p(z^{(l)})}{q(z^{(l)})} : \text{重要度重み}$$

重点サンプリング：手順(続)

・ $p(z)$ の正規化定数が分からない場合 $p(z) = \frac{\tilde{p}(z)}{Z_p}$

$$\begin{aligned}\mathbb{E}[f] &= \int f(z)p(z)dz \\ &= \int f(z)\frac{p(z)}{q(z)}q(z)dz \\ &= \frac{Z_q}{Z_p} \int f(z)\frac{\tilde{p}(z)}{\tilde{q}(z)}q(z)dz \\ &\simeq \frac{Z_q}{Z_p} \frac{1}{L} \sum_{l=1}^L f(z^{(l)})\tilde{r}_l\end{aligned}$$

$$\left[q(z) = \frac{\tilde{q}(z)}{Z_q} \quad \tilde{r}_l = \frac{\tilde{p}(z^{(l)})}{\tilde{q}(z^{(l)})} \right]$$

$$\begin{aligned}\frac{Z_p}{Z_q} &= \frac{1}{Z_q} \int \tilde{p}(z)dz \\ &= \int \frac{\tilde{p}(z)}{\tilde{q}(z)}q(z)dz \\ &\simeq \frac{1}{L} \sum_{l=1}^L \tilde{r}_l\end{aligned}$$

$$\begin{aligned}\mathbb{E}[f] &\simeq \sum_{l=1}^L f(z^{(l)})w_l \\ w_l &= \frac{\tilde{r}_l}{\sum_{m=1}^L \tilde{r}_m}\end{aligned}$$

SIR (sampling/importance-resampling)

棄却サンプリング同様，提案分布 $q(x)$ を用いるが定数 k を決定することなく， $p(x)$ に（近似的に）従うサンプルを生成する手法

SIRの手順：

- サンプルを抽出できる分布 $q(x)$ を用意（提案分布）
- $q(x)$ から L 個のサンプル $x^{(1)}, x^{(2)}, \dots, x^{(L)}$ を生成
- $w_l = \frac{\tilde{p}(x^{(l)})/q(x^{(l)})}{\sum_{n=1}^L \tilde{p}(x^{(n)})/q(x^{(n)})}$ によって重み w_1, w_2, \dots, w_L を決定
- 確率質量が w_1, w_2, \dots, w_L の離散分布 $(x^{(1)}, x^{(2)}, \dots, x^{(L)})$ に従う確率変数 X_r の L 個のサンプルを抽出（再サンプリング）する

SIR：原理

- 再サンプリングされた値の累積分布関数：

$$\begin{aligned}
 \Pr(X_r \leq a) &= \sum_{l:x^{(l)} \leq a} w_l \\
 &= \frac{\sum_{l:x^{(l)} \leq a} \frac{\tilde{p}(x^{(l)})}{q(x^{(l)})}}{\sum_{n=1}^L \frac{\tilde{p}(x^{(n)})}{q(x^{(n)})}} \\
 &= \frac{\frac{1}{L} \sum_{l=1}^L I(x^{(l)} \leq a) \frac{\tilde{p}(x^{(l)})}{q(x^{(l)})}}{\frac{1}{L} \sum_{n=1}^L \frac{\tilde{p}(x^{(n)})}{q(x^{(n)})}}
 \end{aligned}$$

$I(\cdot)$ ：特性関数



$L \rightarrow \infty$

$$\begin{aligned}
 \Pr(X_r \leq a) &\rightarrow \frac{\int I(x \leq a) \frac{\tilde{p}(x)}{q(x)} q(x) dx}{\int \frac{\tilde{p}(x)}{q(x)} q(x) dx} \\
 &= \frac{\int I(x \leq a) \tilde{p}(x) dx}{\int \tilde{p}(x) dx} \\
 &= \frac{\int I(x \leq a) p(x) dx}{p(x) \text{ の累積分布関数}}
 \end{aligned}$$

目次

- ◆ 基礎事項の確認
- ◆ 確率推論問題
- ◆ 状態推定
- ◆ 確率伝搬法
- ◆ 確率伝搬法の応用例

確率推論問題

ベイズの定理： $P(y|x) = \frac{P(x|y)P(y)}{P(x)}$ X ：未知の（興味のある）確率変数
 Y ：観測可能な確率変数

$P(x|y)$ ：事後分布 (posterior distribution)



$P(x)$ ：事前分布 (prior distribution)

$P(y|x)$ ：尤度関数 (likelihood function)

確率推論問題： Y の観測に基づいて、事後分布 $P(x|y)$ を求めること

最大事後確率推定：

$Y = y_o$ を観測したとき、推定値 $X = \hat{x}$ が正しい確率は

$$\Pr(X = \hat{x}, Y = y_o) = P(\hat{x}|y_o)P(y_o)$$

正しい確率が最大（誤り確率が最小）の推定値は

$$\hat{x}_{\text{map}} = \arg \max_{x_i} P(x_i|y_o)$$

最尤推定： $\Pr(X = \hat{x}, Y = y_o) = P(y_o|\hat{x})P(\hat{x})$

$\hat{x}_{\text{ml}} = \arg \max_{x_i} P(y_o|x_i)$ $P(x)$ が一様するとき \hat{x}_{map} と一致

確率推論問題 (続)

未知の確率変数： X_1, \dots, X_m

観測可能な確率変数： $X_{m+1} = a_{m+1}, \dots, X_n = a_n$

事後分布： $P(x_1, \dots, x_m | a_{m+1}, \dots, a_n)$

周辺事後分布： $P(x_i | a_{m+1}, \dots, a_n), \quad i = 1, \dots, m$

最大事後確率推定 (ブロック毎)：

$$\{\hat{x}_1, \dots, \hat{x}_m\} = \arg \max_{x_1, \dots, x_m} P(x_1, \dots, x_m | a_{m+1}, \dots, a_n)$$

最大事後確率推定 (成分毎)：

$$\hat{x}_i = \arg \max_{x_i} P(x_i | a_{m+1}, \dots, a_n)$$

どちらも、素朴に計算すると指数オーダーの計算量。
ただし、周辺事後確率の最大化は同時分布の構造 (因数分解) を利用することで計算量を削減可能な場合がある。

$$\begin{aligned} & \Pr(X_1 = x_1 | X_{m+1} = a_{m+1}, \dots, X_n = a_n) \\ &= \frac{\Pr(X_1 = x_1, X_{m+1} = a_{m+1}, \dots, X_n = a_n)}{\Pr(X_{m+1} = a_{m+1}, \dots, X_n = a_n)} \\ &= \alpha \sum_{x_2, \dots, x_m} P(x_1, x_2, \dots, x_m, a_{m+1}, \dots, a_n) \end{aligned}$$

α ：規格化定数

目次

- ◆ 基礎事項の確認
- ◆ 確率推論問題
- ◆ 状態推定
- ◆ 確率伝搬法
- ◆ 確率伝搬法の応用例

状態空間モデル

興味のある離散時間確率過程（状態）： X_1, \dots, X_T

観測可能な確率変数： Y_1, \dots, Y_T

(一般化された)同時分布の密度関数:

$$x_{1:T} = \{x_1, \dots, x_T\}$$

$$\begin{aligned} & p(x_1, \dots, x_T, y_1, \dots, y_T) \\ &= p(y_T | x_1, \dots, x_T, y_1, \dots, y_{T-1}) p(x_1, \dots, x_T, y_1, \dots, y_{T-1}) \\ &= p(y_T | x_{1:T}, y_{1:T-1}) p(x_{1:T}, y_{1:T-1}) \\ &= p(y_T | x_{1:T}, y_{1:T-1}) p(x_T | x_{1:T-1}, y_{1:T-1}) p(x_{1:T-1}, y_{1:T-1}) \\ &= p(y_T | x_{1:T}, y_{1:T-1}) p(x_T | x_{1:T-1}, y_{1:T-1}) \\ &\quad \cdot p(y_{T-1} | x_{1:T-1}, y_{1:T-2}) p(x_{T-1} | x_{1:T-2}, y_{1:T-2}) p(x_{1:T-2}, y_{1:T-2}) \\ &\quad \vdots \end{aligned}$$

$$= \prod_{t=1}^T p(y_t | x_{1:t}, y_{1:t-1}) p(x_t | x_{1:t-1}, y_{1:t-1})$$

$$\begin{aligned} p(y_1 | x_{1:1}, y_{1:0}) &= p(y_1 | x_1) \\ p(x_1 | x_{1:0}, y_{1:0}) &= p(x_1) \end{aligned}$$

状態空間モデル (続)

$$p(x_{1:T}, y_{1:T}) = \prod_{t=1}^T p(y_t | x_{1:t}, y_{1:t-1}) p(x_t | x_{1:t-1}, y_{1:t-1})$$



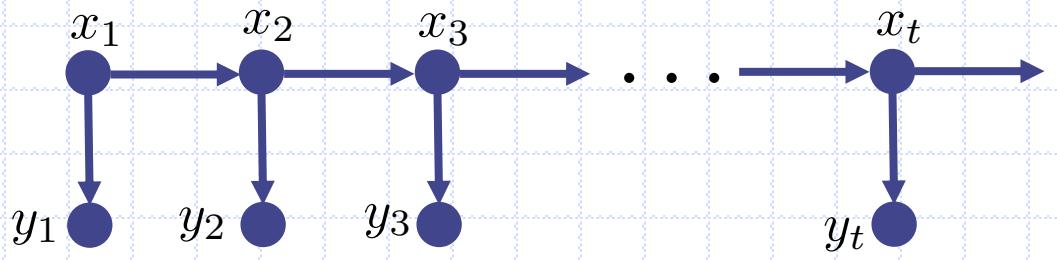
モデルを仮定 (マルコフ性)

$$= \prod_{t=1}^T p(y_t | x_t) p(x_t | x_{t-1})$$

$$p(x_1 | x_0) = p(x_1)$$

$p(x_t | x_{1:t-1}, y_{1:t-1}) = p(x_t | x_{t-1})$: システムモデル
 $p(y_t | x_{1:t}, y_{1:t-1}) = p(y_t | x_t)$: 観測モデル

鎖状グラフィカルモデル:



$$X_{t+1} \perp\!\!\!\perp X_{t-1} | X_t, \quad X_{t+1} \perp\!\!\!\perp Y_t | X_t \quad \text{が成り立つ}$$

状態変数が離散変数の場合

隠れマルコフモデル (HMM)

状態空間モデル (続)

一般状態空間モデル：

$$x_t \sim p(x_t | x_{t-1}) \quad : \text{システムモデル}$$

$$y_t \sim p(y_t | x_t) \quad : \text{観測モデル}$$

非線形・非ガウス型状態空間モデル：

$$x_t = f_t(x_{t-1}, w_t) \quad : \text{システムモデル}$$

$$y_t = h_t(x_t, v_t) \quad : \text{観測モデル}$$

f_t, h_t : 非線形関数

w_t, v_t : 白色雑音

線形・ガウス型状態空間モデル：

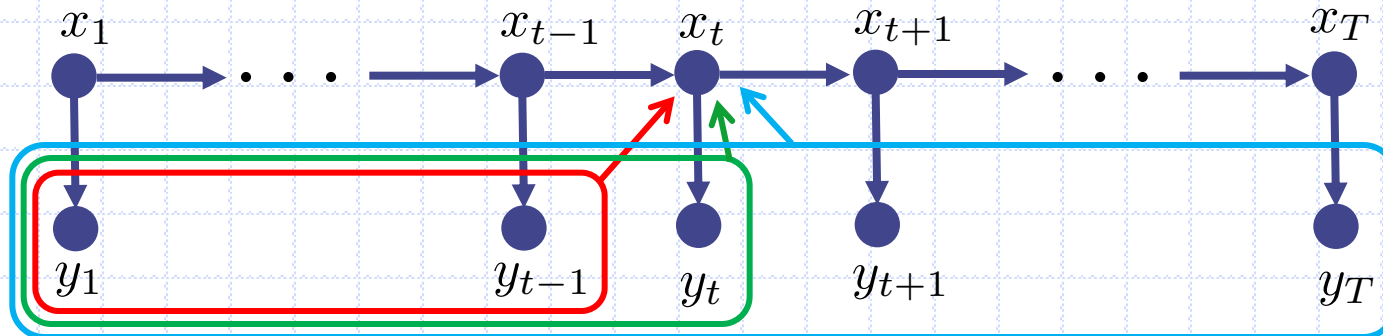
$$x_t = F_t x_{t-1} + G_t w_t \quad : \text{システムモデル}$$

$$y_t = H_t x_t + v_t \quad : \text{観測モデル}$$

F_t, G_t, H_t : 行列

w_t, v_t : 白色ガウス雑音

重要な分布



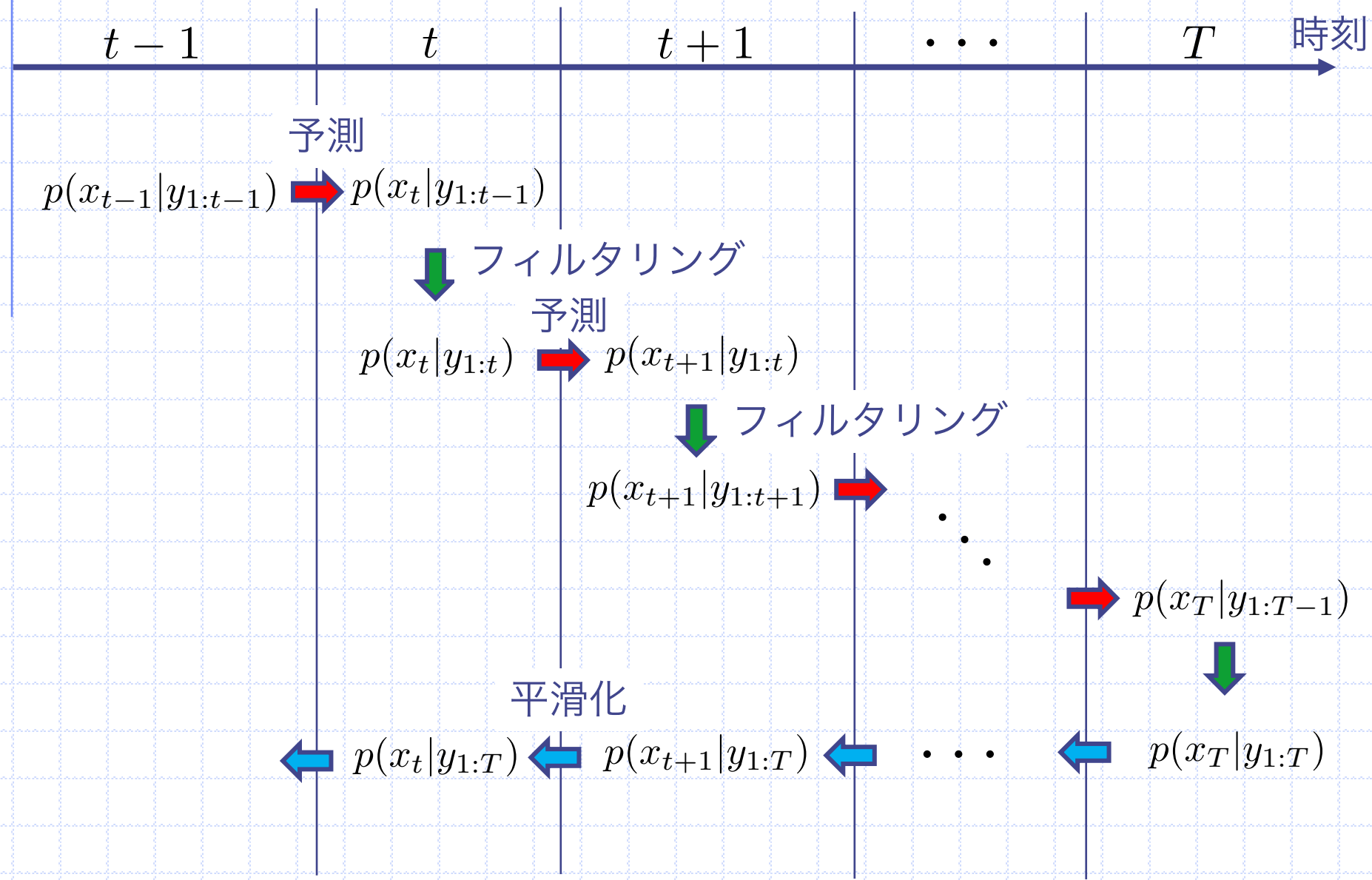
予測分布： $p(x_t | y_{1:t-1})$

フィルタ分布： $p(x_t | y_{1:t})$

平滑化分布： $p(x_t | y_{1:T})$

← 周辺事後分布

状態推定の手順



予測分布

$p(x_{t-1}|y_{1:t-1})$ から $p(x_t|y_{1:t-1})$ を求める

$$p(x_t|y_{1:t-1}) = \int p(x_t, x_{t-1}|y_{1:t-1}) dx_{t-1}$$

$$= \int \boxed{p(x_t|x_{t-1}, y_{1:t-1})} p(x_{t-1}|y_{1:t-1}) dx_{t-1}$$

$$= \int \underbrace{p(x_t|x_{t-1})}_{\text{システムモデル}} \underbrace{p(x_{t-1}|y_{1:t-1})}_{\text{時刻 } t-1 \text{ でのフィルタ分布}} dx_{t-1}$$

: Markov性

フィルタ分布

$p(x_t|y_{1:t-1})$ から $p(x_t|y_{1:t})$ を求める

$$\begin{aligned}
 p(x_t|y_{1:t}) &= p(x_t|y_{1:t-1}, y_t) \\
 &= \frac{p(x_t, y_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \\
 &= \frac{p(x_t, y_t|y_{1:t-1})}{\int p(x_t, y_t|y_{1:t-1}) dx_t} \\
 &= \frac{p(y_t|x_t, y_{1:t-1})p(x_t|y_{1:t-1})}{\int p(y_t|x_t, y_{1:t-1})p(x_t|y_{1:t-1}) dx_t} \\
 &= \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1}) dx_t}
 \end{aligned}$$

: Markov性
(観測モデル)

観測
モデル 時刻 $t-1$ での
予測分布

平滑化分布

$p(x_{t+1}|y_{1:T})$ から $p(x_t|y_{1:T})$ を求める

$$p(x_t|y_{1:T}) = \int p(x_t, x_{t+1}|y_{1:T}) dx_{t+1}$$

$$= \int \underline{p(x_t|x_{t+1}, y_{1:T})} p(x_{t+1}|y_{1:T}) dx_{t+1}$$

↓ 有向分離 (d-separation)

$$= \int \underline{p(x_t|x_{t+1}, y_{1:t})} p(x_{t+1}|y_{1:T}) dx_{t+1}$$

$$= \int \frac{p(x_t, x_{t+1}|y_{1:t})}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T}) dx_{t+1}$$

$$= \int \frac{p(x_{t+1}|x_t, y_{1:t}) p(x_t|y_{1:t})}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T}) dx_{t+1}$$

$$= p(x_t|y_{1:t}) \int \frac{p(x_{t+1}|x_t)}{p(x_{t+1}|y_{1:t})} p(x_{t+1}|y_{1:T}) dx_{t+1}$$

有向分離

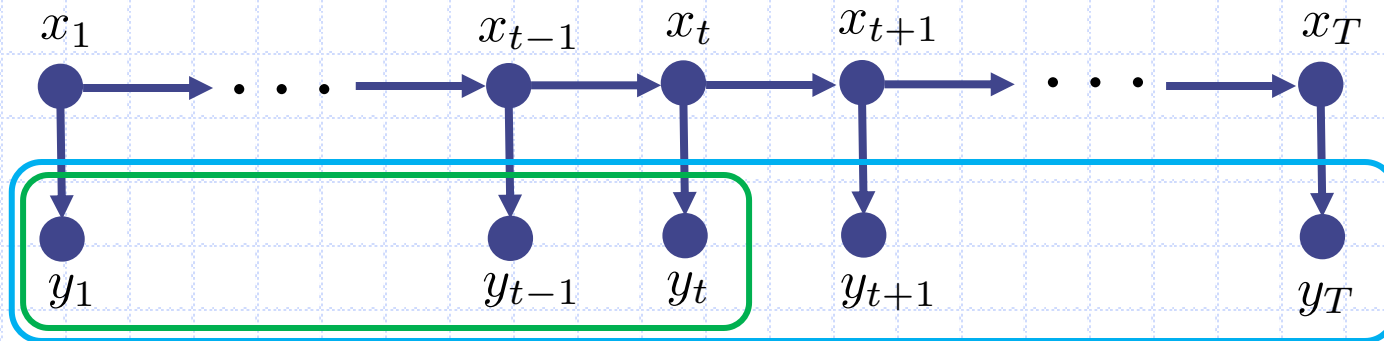
積法則を用いると…

$$p(x_t, y_{1:T} | x_{t+1}) = \underline{p(x_t | x_{t+1}, y_{1:T})} p(y_{1:T} | x_{t+1})$$

条件付き独立性を用いると…

$$\begin{aligned} p(x_t, y_{1:T} | x_{t+1}) &= p(x_t, y_{1:t}, y_{t+1:T} | x_{t+1}) \quad \searrow : \text{条件付き独立} \\ &= p(x_t, y_{1:t} | x_{t+1}) p(y_{t+1:T} | x_{t+1}) \\ &= p(x_t | x_{t+1}, y_{1:t}) p(y_{1:t} | x_{t+1}) p(y_{t+1:T} | x_{t+1}) \\ &= p(x_t | x_{t+1}, y_{1:t}) p(y_{1:t}, y_{t+1:T} | x_{t+1}) \\ &= \underline{p(x_t | x_{t+1}, y_{1:t})} p(y_{1:T} | x_{t+1}) \end{aligned}$$

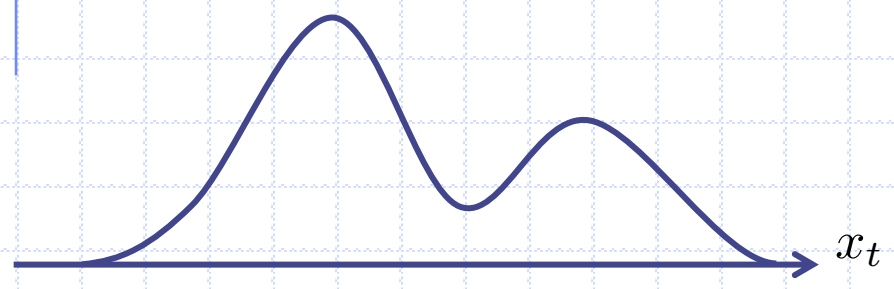
d-separation



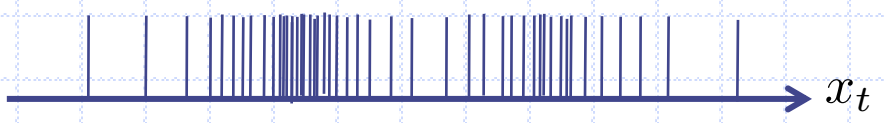
粒子による確率分布の近似

連続分布である $p(x_t|y_{1:t-1})$ や $p(x_t|y_{1:t})$ を複数の「粒子」(サンプル)による離散分布で近似する

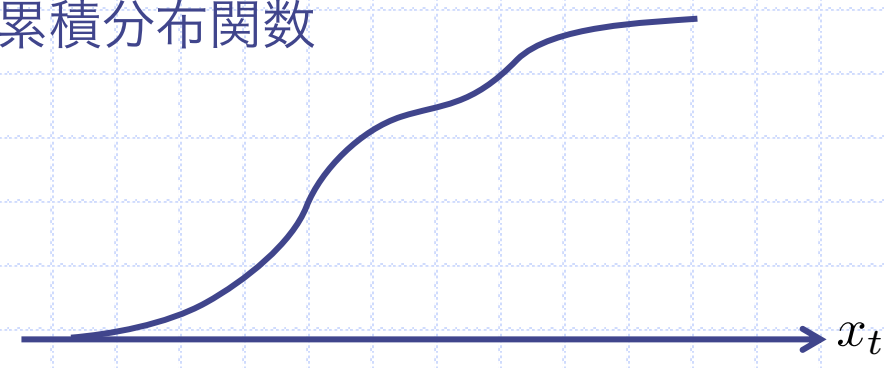
確率密度関数



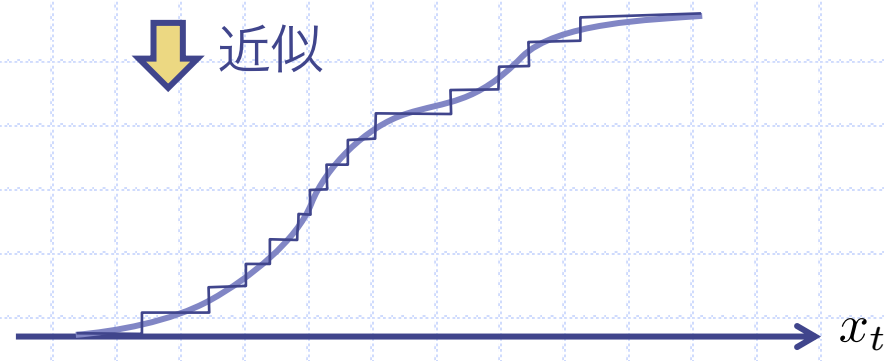
↓ 近似



累積分布関数



↓ 近似



粒子フィルタの手順

$p(x_{t-1}|y_{1:t-1})$ を近似するサンプル： $\{x_{t-1}^{(i)}, i = 1, \dots, L\}$



予測

$p(x_t|y_{1:t-1})$ を近似するサンプル： $\{\hat{x}_t^{(i)}, i = 1, \dots, L\}$



フィルタリング

$p(x_t|y_{1:t})$ を近似するサンプル： $\{x_t^{(i)}, i = 1, \dots, L\}$

粒子フィルタ：予測

非線形・非ガウス型状態空間モデル：

$$x_t = f_t(x_{t-1}, w_t)$$

予測分布：

$$\begin{aligned} p(x_t | y_{1:t-1}) &= \int p(x_t | x_{t-1}) p(x_{t-1} | y_{1:t-1}) dx_{t-1} \\ &= \int \left\{ \int p(x_t, w_t | x_{t-1}) dw_t \right\} p(x_{t-1} | y_{1:t-1}) dx_{t-1} \\ &= \int \left\{ \int p(x_t | x_{t-1}, w_t) p(w_t | x_{t-1}) dw_t \right\} p(x_{t-1} | y_{1:t-1}) dx_{t-1} \\ &= \int \left\{ \int \delta(x_t - f_t(x_{t-1}, w_t)) p(w_t) dw_t \right\} p(x_{t-1} | y_{1:t-1}) dx_{t-1} \\ &\approx \frac{1}{L} \sum_{i=1}^L \delta\left(x_t - f_t\left(x_{t-1}^{(i)}, w_t^{(i)}\right)\right) \end{aligned}$$

粒子フィルタ：予測（続）

粒子フィルタによる予測分布の計算手順：

- ・時刻 $t - 1$ の状態のフィルタ分布 $p(x_{t-1}|y_{1:t-1})$ を近似する粒子 $\{x_{t-1}^{(i)}, i = 1, \dots, L\}$ を用意
- ・システム雑音の分布 $p(w_t)$ からの L 個のサンプル $\{w_t^{(i)}, i = 1, \dots, L\}$ を生成
- ・時刻 t の状態の予測分布 $p(x_t|y_{1:t-1})$ を近似する粒子を次で計算

$$\hat{x}_t^{(i)} = f_t(x_{t-1}^{(i)}, w_t^{(i)}), \quad i = 1, \dots, L$$

粒子フィルタ：フィルタリング

フィルタ分布：

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t}$$

$$\propto \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(x)}$$

$$\frac{p(x)}{q(x)} \quad \text{提案分布 (既にサンプルあり)}$$

⇒ SIRの利用

粒子フィルタによるフィルタ分布の計算手順：

- 時刻 t の状態の予測分布 $p(x_t|y_{1:t-1})$ を近似する粒子

$\{\hat{x}_t^{(i)}, i = 1, \dots, L\}$ を用意

- 確率質量が

$\frac{p(y_t|\hat{x}_t^{(i)})}{\sum_{n=1}^L p(y_t|\hat{x}_t^{(n)})}$ である $\{\hat{x}_t^{(i)}, i = 1, \dots, L\}$ 上の離散分布

から L 回再サンプリングすることで、時刻 t の状態のフィルタ分布

$p(x_t|y_{1:t})$ を近似する粒子 $\{x_t^{(i)}, i = 1, \dots, L\}$ を得る

カルマンフィルタ

線形・ガウス型状態空間モデルの場合、 $p(x_t|y_{1:t-1})$ と $p(x_t|y_{1:t})$ はガウス分布になるので、条件付期待値と共分散行列だけを追いかければよい

条件付き期待値：
$$E[X_t|y_{1:s}] = \bar{x}_{t|s}$$

条件付き共分散行列：
$$E[(X_t - \hat{x}_{t|s})(X_t - \bar{x}_{t|s})^T | y_{1:s}] = P_{t|s}$$

予測分布：
$$p(x_t|y_{1:t-1}) = \mathcal{N}(\bar{x}_{t|t-1}, P_{t|t-1})$$

フィルタ分布：
$$p(x_t|y_{1:t}) = \mathcal{N}(\bar{x}_{t|t}, P_{t|t})$$

カルマンフィルタ：

フィルタリング：
$$K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1}$$
$$\bar{x}_{t|t} = \bar{x}_{t|t-1} + K_t (y_t - H_t \bar{x}_{t|t-1})$$
$$P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}$$

予測：
$$\bar{x}_{t+1|t} = F_t \bar{x}_{t|t}$$
$$P_{t+1|t} = F_t P_{t|t} F_t^T + G_t Q_t G_t^T$$

目次

- ◆ 基礎事項の確認
- ◆ 確率推論問題
- ◆ 状態推定
- ◆ 確率伝搬法
- ◆ 確率伝搬法の応用例

確率伝搬法の原理

同時確率分布の分解:

$$\begin{aligned}
 P(x_1, \dots, x_n) &= P(x_1)P(x_2, \dots, x_n|x_1) \\
 &= P(x_1)P(x_2|x_1)P(x_3, \dots, x_n|x_1, x_2) \\
 &= P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \cdots P(x_n|x_1, x_2, \dots, x_{n-1})
 \end{aligned}$$

必ずしも全ての条件が有効ではない

具体例: $P(x_1, x_2, x_3, x_4, x_5)$

$$\begin{aligned}
 &= P(x_1)P(x_2|x_1)P(x_4|x_1, x_2)P(x_3|x_1, x_2, x_4)P(x_5|x_1, x_2, x_3, x_4) \\
 &= P(x_1)P(x_2|x_1)P(x_4|x_1)P(x_3|x_4)P(x_5|x_4)
 \end{aligned}$$

$$\Pr(X_1 = x_1 | X_3 = a_3) = \frac{\Pr(X_1 = x_1, X_3 = a_3)}{\Pr(X_3 = a_3)}$$

$$= \alpha \sum_{x_2, x_4, x_5} P(x_1, x_2, a_3, x_4, x_5)$$

$$= \alpha \sum_{x_2, x_4, x_5} P(x_1)P(x_2|x_1)P(X_3 = a_3|x_4)P(x_4|x_1)P(x_5|x_4)$$

$$= \alpha P(x_1) \left(\sum_{x_2} P(x_2|x_1) \right) \left(\sum_{x_4} P(X_3 = a_3|x_4)P(x_4|x_1) \right) \left(\sum_{x_5} P(x_5|x_4) \right)$$

分配法則により
ローカルな
周辺化へ

和-積 アルゴリズム

木構造をもつファクターグラフで表現された多変数関数

$$f(x_1, x_2, \dots, x_n) = f_1(A_1)f_2(A_2) \cdots f_m(A_m)$$

の周辺化関数 $g_r(x_r) = \sum_{x_1, \dots, x_n \setminus x_r} f(x_1, \dots, x_n)$

をファクターグラフ上で計算するメッセージ伝搬アルゴリズム

周辺化関数計算の手順：

1. $f(x_1, x_2, \dots, x_n) = f_1(A_1)f_2(A_2) \cdots f_m(A_m)$ からファクターグラフを作る
2. 求めたい周辺化関数を $g_r(x_r)$ とするとき、変数ノード x_r を根とする木としてファクターグラフを描き直す
3. 木の下側（葉側）のノードから上側（根側）のノードの順に以下に説明する処理のルールに従って計算する

和-積 アルゴリズム (続)

変数ノード x_k での処理:

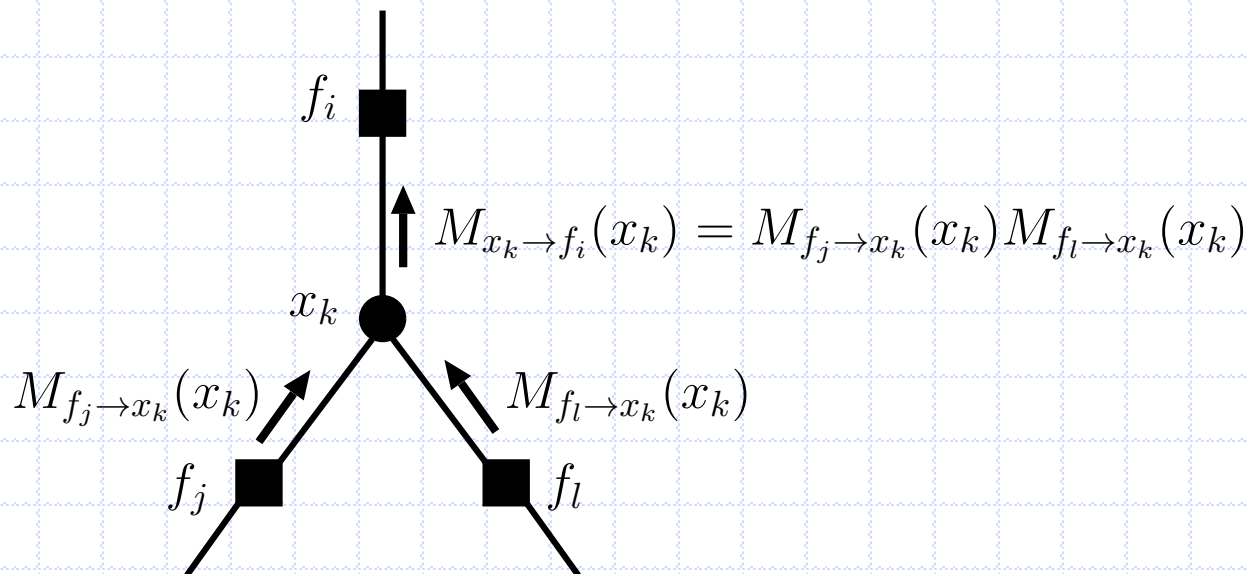
変数ノード x_k から直接つながっている関数ノード f_i へのメッセージ

$$M_{x_k \rightarrow f_i}(x_k) = \prod_{a \in N(x_k) \setminus f_i} M_{a \rightarrow x_k}(x_k)$$

$N(x_k)$ は x_k に接続するノードの集合

$M_{a \rightarrow x_k}(x_k)$ はノード a から x_k に届いたメッセージ

x_k が葉のときは $M_{x_k \rightarrow f_i}(x_k) = 1$



和-積 アルゴリズム (続)

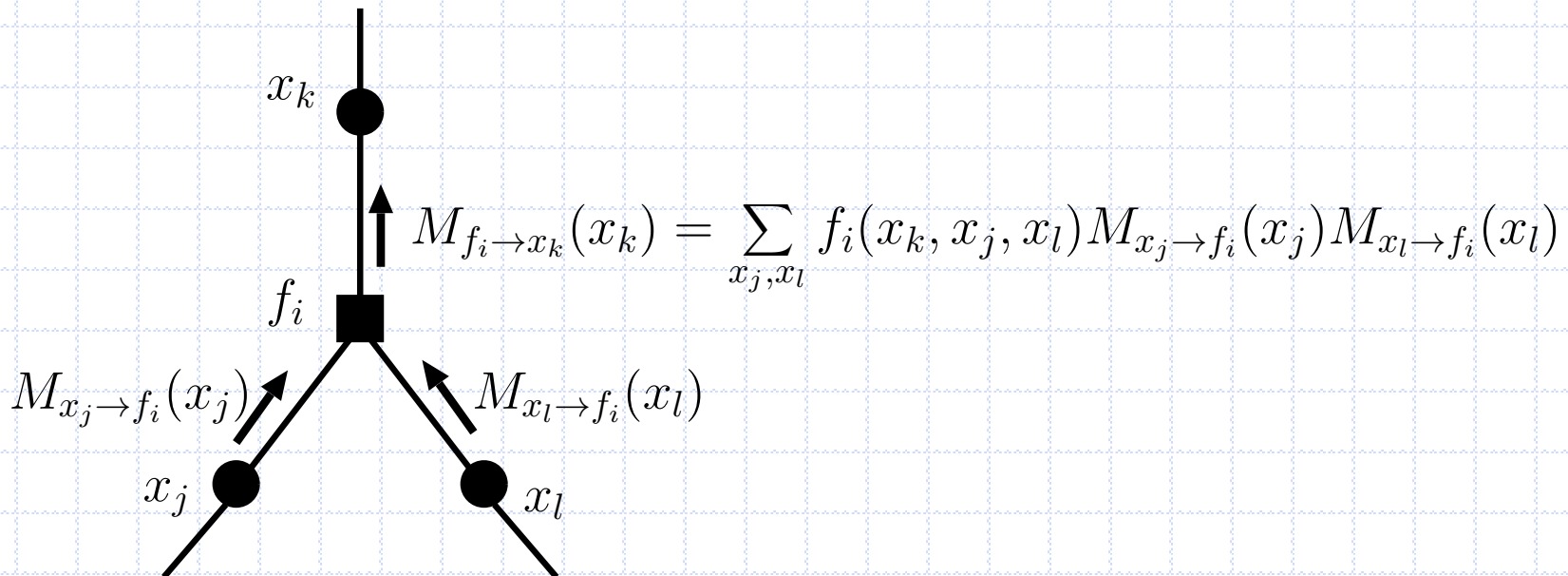
関数ノード f_i での処理:

関数ノード f_i から直接つながっている変数ノード x_k へのメッセージ

$$M_{f_i \rightarrow x_k}(x_k) = \sum_{A_i \setminus x_k} \left(f_i(A_i) \prod_{a \in A_i \setminus x_k} M_{a \rightarrow f_i}(a) \right)$$

$M_{a \rightarrow f_i}(a)$ はノード a から f_i に届いたメッセージ

f_i が葉のときは $M_{f_i \rightarrow x_k}(x_k) = f_i(x_k)$

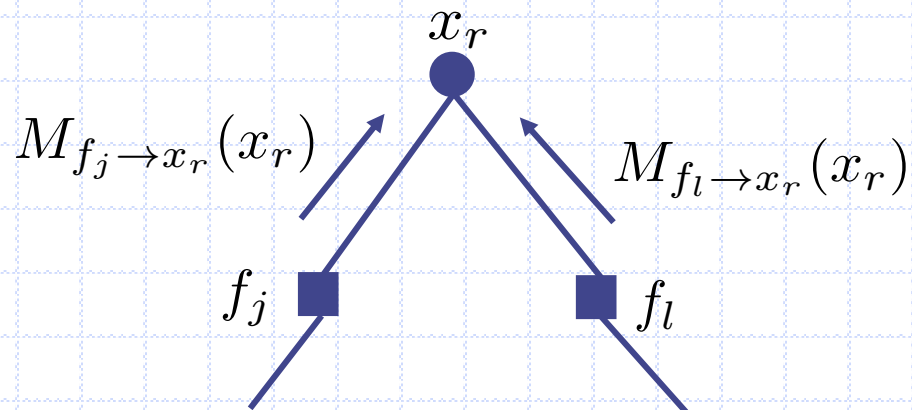


和-積 アルゴリズム (続)

ルートノード x_r での処理:

ルートの変数ノード x_r での処理

$$M_{x_r}(x_r) = \prod_{a \in N(x_r)} M_{a \rightarrow x_r}(x_r) = g_r(x_r)$$



和-積 アルゴリズムのしくみ

ファクターグラフが木構造をもつと次のように因数分解可能

$$f(x_1, \dots, x_n) = \prod_{a \in N(x_r)} F_a(x_r, B_a)$$

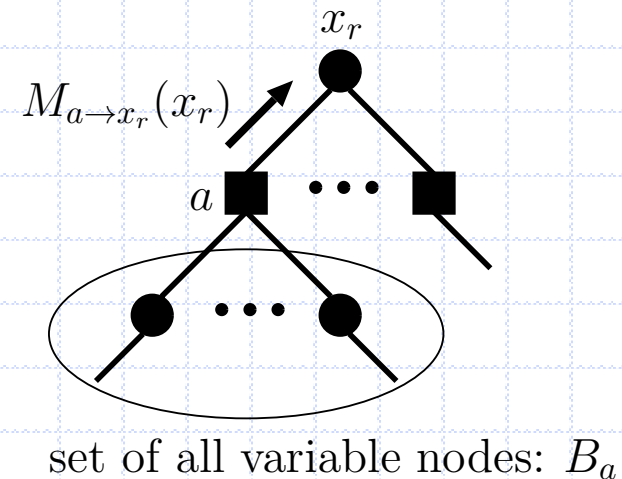
B_a : a を介して x_r に繋がっている変数の集合
 (x_r, B_a) , $a \in N(x_r)$ は x_r のみを共有

$$g_r(x_r) = \sum_{x_1, \dots, x_n \setminus x_r} \prod_{a \in N(x_r)} F_a(x_r, B_a)$$

$$= \prod_{a \in N(x_r)} \left(\sum_{B_a} F_a(x_r, B_a) \right)$$

$$= \prod_{a \in N(x_r)} M_{a \rightarrow x_r}(x_r)$$

ルートノードでの処理



ただし, $M_{a \rightarrow x_r}(x_r) \stackrel{\text{def}}{=} \sum_{B_a} F_a(x_r, B_a)$ ← a 以下の情報で計算可能

補足：和と積の順序の交換について

簡単な構成のグラフの例：

$$N(x_r) = \{f_1, f_2\} \quad B_{f_1} = \{x_1, x_2\}, \quad B_{f_2} = \{x_3, x_4\}$$

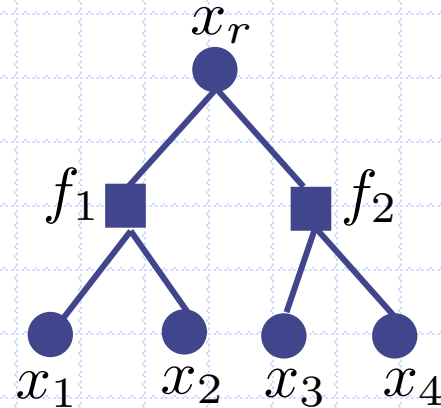
$$\sum_{x_1, x_2, x_3, x_4} \prod_{a \in \{f_1, f_2\}} F_a(x_r, B_a)$$

$$= \sum_{x_1, x_2, x_3, x_4} F_{f_1}(x_r, x_1, x_2) F_{f_2}(x_r, x_3, x_4)$$

$$= \sum_{x_1, x_2} \sum_{x_3, x_4} F_{f_1}(x_r, x_1, x_2) F_{f_2}(x_r, x_3, x_4)$$

$$= \left(\sum_{x_1, x_2} F_{f_1}(x_r, x_1, x_2) \right) \left(\sum_{x_3, x_4} F_{f_2}(x_r, x_3, x_4) \right)$$

$$= \prod_{a \in \{f_1, f_2\}} \sum_{B_a} F_a(x_r, B_a)$$



$(x_r, B_{f_1}), (x_r, B_{f_2})$ は
 x_r のみを共有

和-積 アルゴリズムのしくみ(続)

関数ノード a に直接接続する変数ノードを $x_r, x_{a,1}, \dots, x_{a,M}$ とし、
変数ノード $x_{a,m}$ を介して a につながる全ての変数の集合を $B_{a,m}$ とする

$$F_a(x_r, B_a) = a(x_r, x_{a,1}, \dots, x_{a,M}) G_{a,1}(x_{a,1}, B_{a,1}) \cdots G_{a,M}(x_{a,M}, B_{a,M})$$

という形にかけるので

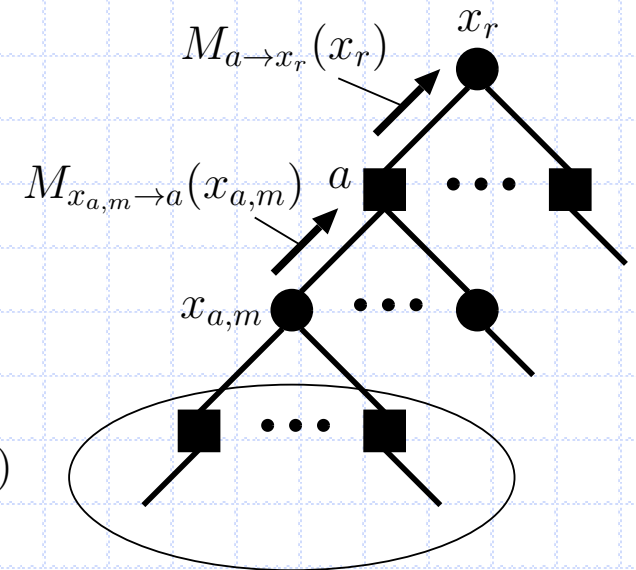
$$M_{a \rightarrow x_r}(x_r)$$

$$= \sum_{B_a} a(x_r, x_{a,1}, \dots, x_{a,M}) \prod_{m=1}^M G_{a,m}(x_{a,m}, B_{a,m})$$

$$= \sum_{x_{a,1}, \dots, x_{a,M}} a(x_r, x_{a,1}, \dots, x_{a,M}) \prod_{m=1}^M \sum_{B_{a,m}} G_{a,m}(x_{a,m}, B_{a,m})$$

$$= \sum_{x_{a,1}, \dots, x_{a,M}} a(x_r, x_{a,1}, \dots, x_{a,M}) \prod_{x_{a,m} \in N(a) \setminus x_r} M_{x_{a,m} \rightarrow a}(x_{a,m})$$

関数ノードでの処理



set of all variable nodes: $B_{a,m}$

ただし, $M_{x_{a,m} \rightarrow a}(x_{a,m}) \stackrel{\text{def}}{=} \sum_{B_{a,m}} G_{a,m}(x_{a,m}, B_{a,m}) \leftarrow x_{a,m}$ 以下の情報で計算可能

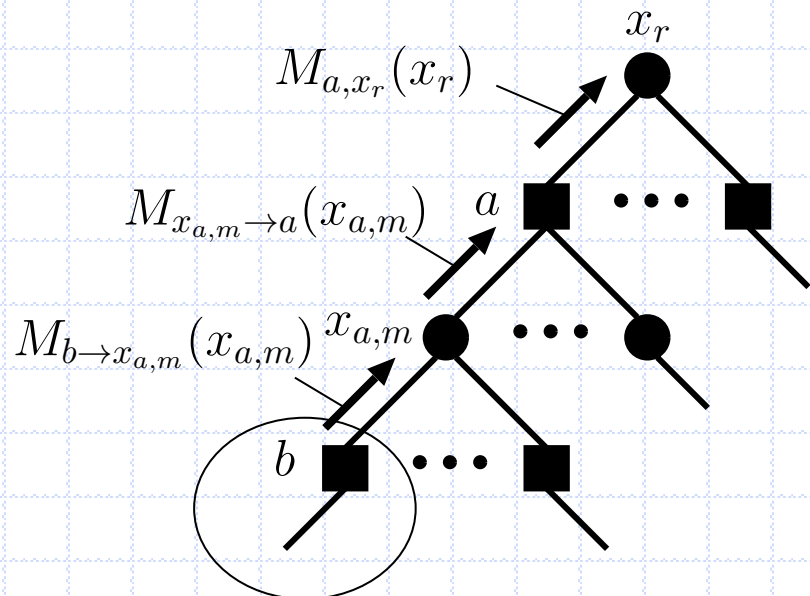
和-積 アルゴリズムのしくみ(続)

変数ノード $x_{a,m}$ にノード $b (\neq a)$ を介してつながる全ての変数の集合： $B_{a,m,b}$

$$G_{a,m}(x_{a,m}, B_{a,m}) = \prod_{b \in N(x_{a,m}) \setminus a} F_{a,b}(x_{a,m}, B_{a,m,b}) \quad \text{とかけるので}$$

$$\begin{aligned} & M_{x_{a,m} \rightarrow a}(x_{a,m}) \\ &= \sum_{B_{a,m}} \prod_{b \in N(x_{a,m}) \setminus a} F_{a,b}(x_{a,m}, B_{a,m,b}) \\ &= \prod_{b \in N(x_{a,m}) \setminus a} \sum_{B_{a,m,b}} F_{a,b}(x_{a,m}, B_{a,m,b}) \\ &= \prod_{b \in N(x_{a,m}) \setminus a} M_{b \rightarrow x_{a,m}}(x_{a,m}) \end{aligned}$$

変数ノードでの処理



set of all variable nodes: $B_{a,m,b}$

和-積アルゴリズムの注意点

確率推論への適用法：

1. $P(x_r | x_i, x_j, x_k, x_l, \dots)$ などを $f(x_r, x_i, x_j, x_k, x_l, \dots)$ のようにおく
2. 周辺事後確率の計算を周辺化関数の計算に帰着させる
3. ファクターグラフを作成
4. 和-積アルゴリズムを実行（観測値がある場合は以下のようにする）

$$x_k \text{ の観測値が } a_k \text{ のとき } M_{x_k \rightarrow f_i}(x_k) = \delta_{x_k, a_k}$$

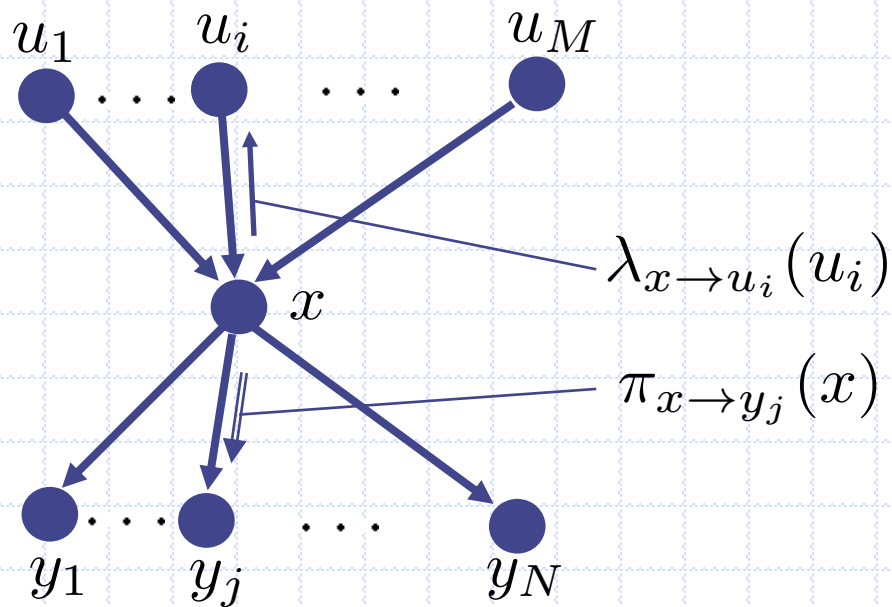
グラフのループについて：

- ・ループがある場合にはどのような解が得られるか理論的な保証はないが、経験的に多くの応用でよい結果が得られる
- ・ループがある場合の理論的な説明付けや収束特性解析，対策法など
 - ✓ 情報幾何に基く解釈
 - ✓ ベーテ自由エネルギーの停留点
 - ✓ 密度発展法
 - ✓ EXITチャート法
 - ✓ ガウス近似法
 - ✓ 一般化確率伝搬法

PearlのBPアルゴリズム

- x が親 u_i に送るメッセージ：

$$\lambda_{x \rightarrow u_i}(u_i) = \sum_{u_1, \dots, u_M \setminus u_i} \left(\sum_x \left(P(x|u_1, \dots, u_M) \prod_{k=1}^N \lambda_{y_k \rightarrow x}(x) \right) \prod_{k=1 \setminus i}^M \pi_{u_k \rightarrow x}(u_k) \right)$$



- x が子 y_j に送るメッセージ：

$$\pi_{x \rightarrow y_j}(x) = \left(\prod_{k=1 \setminus j}^N \lambda_{y_k \rightarrow x}(x) \right) \sum_{u_1, \dots, u_M} \left(P(x|u_1, \dots, u_M) \prod_{k=1}^M \pi_{u_k \rightarrow x}(u_k) \right)$$

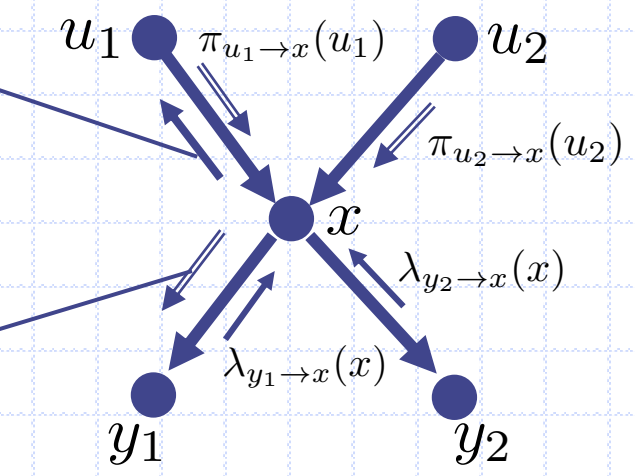
PearlのBPアルゴリズムと和-積アルゴリズムの例

$$P(u_1, u_2, x, y_1, y_2) = P(x|u_1, u_2)P(y_1|x)P(y_2|x)P(u_1)P(u_2)$$

$$\sum_{u_2} \left(\sum_x P(x|u_1, u_2) \lambda_{y_1 \rightarrow x}(x) \lambda_{y_2 \rightarrow x}(x) \right) \pi_{u_2 \rightarrow x}(u_2)$$

$$= \sum_x \lambda_{y_1 \rightarrow x}(x) \lambda_{y_2 \rightarrow x}(x) \sum_{u_2} P(x|u_1, u_2) \pi_{u_2 \rightarrow x}(u_2)$$

$$\lambda_{y_2 \rightarrow x}(x) \sum_{u_1, u_2} P(x|u_1, u_2) \pi_{u_1 \rightarrow x}(u_1) \pi_{u_2 \rightarrow x}(u_2)$$



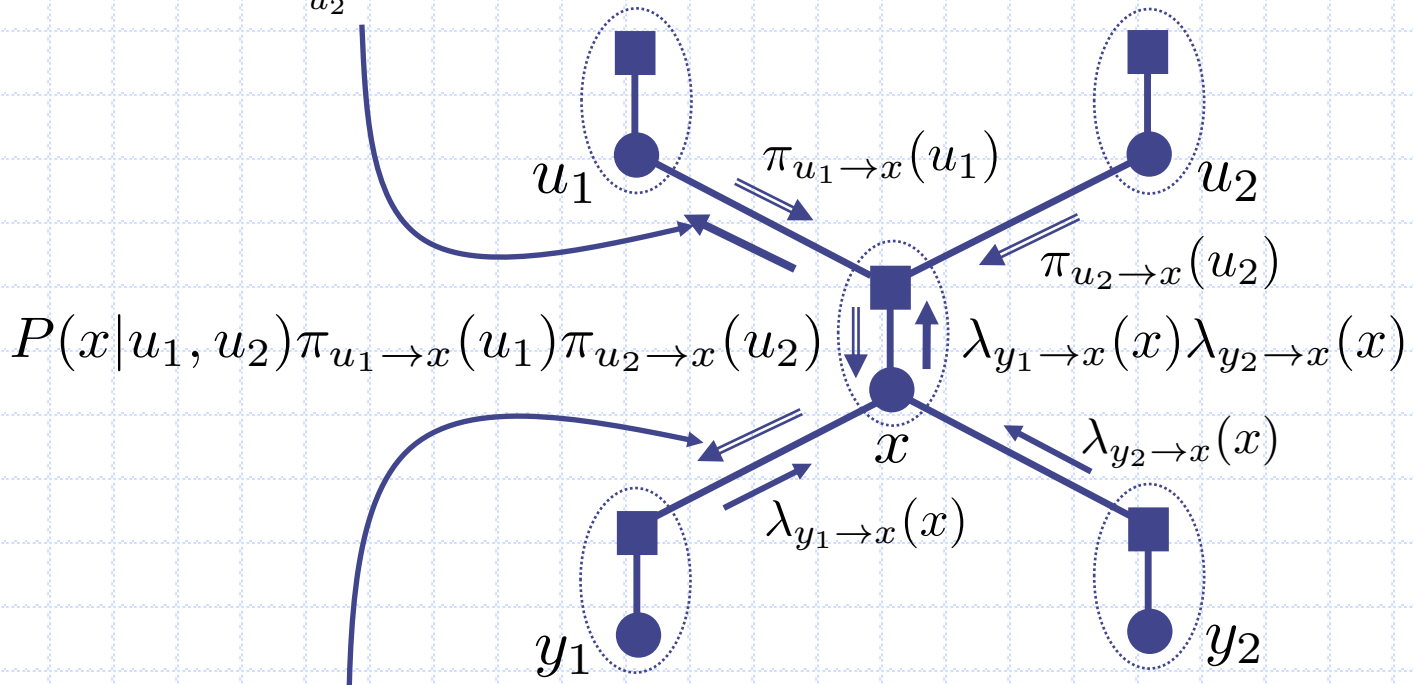
ベイジアンネットワーク表現

PearlのBPアルゴリズムと和-積アルゴリズムの例

$$P(u_1, u_2, x, y_1, y_2) = P(x|u_1, u_2)P(y_1|x)P(y_2|x)P(u_1)P(u_2)$$

$$\sum_{u_2} \left(\sum_x P(x|u_1, u_2) \lambda_{y_1 \rightarrow x}(x) \lambda_{y_2 \rightarrow x}(x) \right) \pi_{u_2 \rightarrow x}(u_2)$$

$$= \sum_x \lambda_{y_1 \rightarrow x}(x) \lambda_{y_2 \rightarrow x}(x) \sum_{u_2} P(x|u_1, u_2) \pi_{u_2 \rightarrow x}(u_2)$$



ファクターグラフ表現

$$\lambda_{y_2 \rightarrow x}(x) \sum_{u_1, u_2} P(x|u_1, u_2) \pi_{u_1 \rightarrow x}(u_1) \pi_{u_2 \rightarrow x}(u_2)$$

目次

- ◆ 基礎事項の確認
- ◆ 確率推論問題
- ◆ 状態推定
- ◆ 確率伝搬法
- ◆ 確率伝搬法の応用例

誤り訂正符号の例：線形符号

情報語（長さ K ）： $\mathbf{u} = [u_1 \dots u_K]^T$



符号化（ N 個の冗長を追加, $K/(K+N)$ を符号化率という）

符号語（長さ $K+N$ ）：

$$\mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ \vdots \\ c_{K+N} \end{bmatrix} = \underbrace{\begin{bmatrix} \text{生成行列} \\ \mathbf{G} \end{bmatrix}}_K \begin{bmatrix} u_1 \\ \vdots \\ u_K \end{bmatrix}_{K+N}$$

検査行列： $\underbrace{\mathbf{H}}_{K+N} \left. \vphantom{\mathbf{H}} \right\} N$ ← 行ベクトルは \mathbf{G} の列ベクトルの直交補空間
 $\mathbf{HG} = \mathbf{0}$

受信語： $\mathbf{c}' = \mathbf{c} + \mathbf{e}$ ← 誤り

$$\mathbf{Hc}' = \mathbf{Hc} + \mathbf{He} = \mathbf{HG}\mathbf{u} + \mathbf{He} = \underline{\mathbf{He}}$$

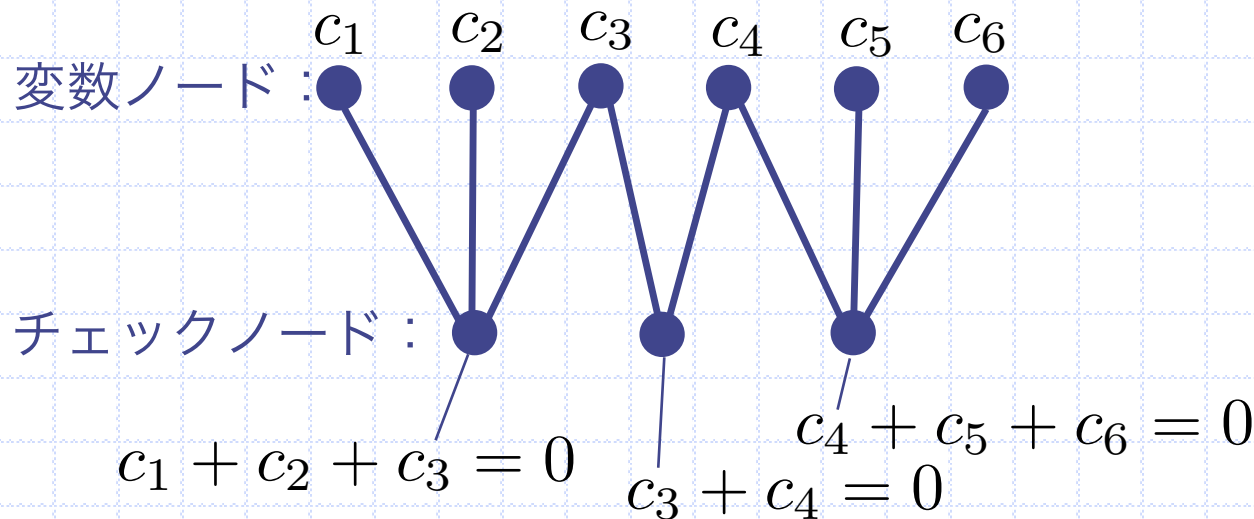
これから誤りの検出や訂正を行う

低密度パリティ検査 (LDPC) 符号

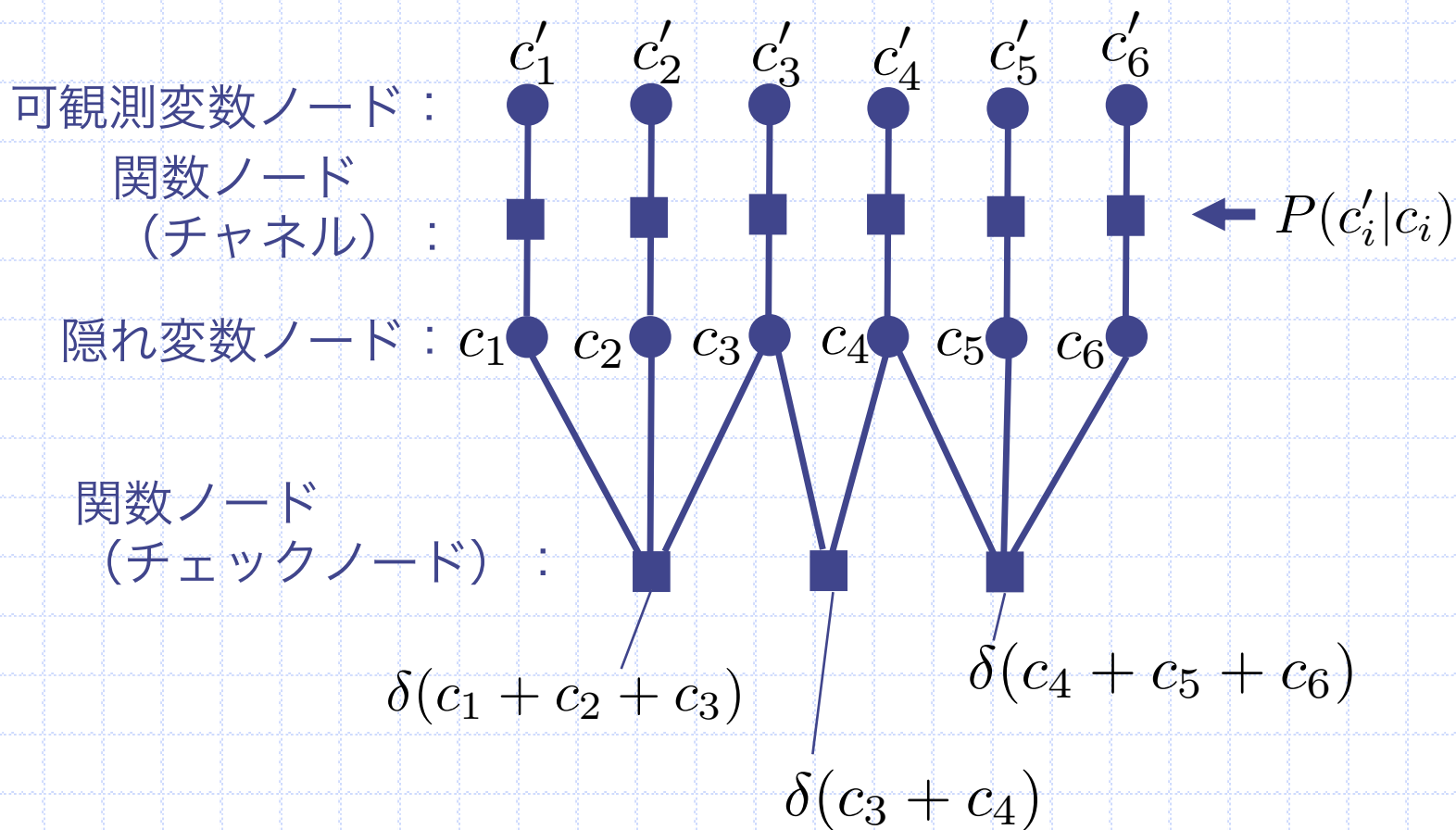
非零要素の割合が非常に小さい「低密度」の検査行列によって定義

$$\mathbf{Hc} = \mathbf{0}$$

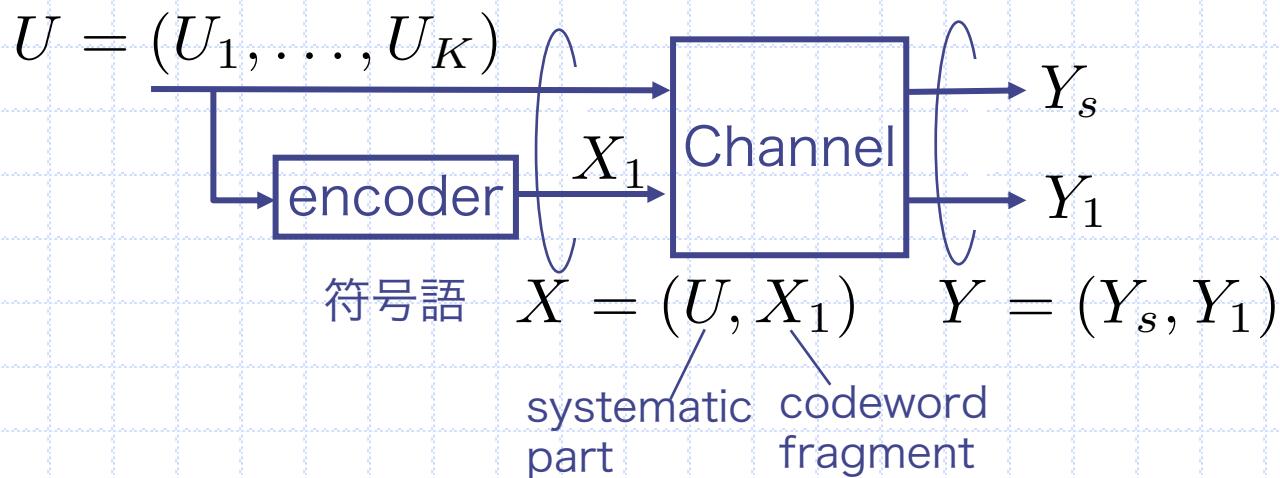
検査行列： $\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$



低密度パリティ検査 (LDPC) 符号： ファクターグラフ表現



組織符号とその最大事後確率復号



通信路の遷移確率: $P(y|x) \stackrel{def}{=} \Pr(Y = y|X = x)$

$$P(y|x) = P(y_s, y_1|u, x_1)$$

$$= P(y_s|u)P(y_1|x_1)$$

$$= \left(\prod_{i=1}^K P(y_{si}|u_i) \right) \cdot P(y_1|x_1)$$

周辺事後確率: $\text{BEL}_i(a) \stackrel{def}{=} \Pr\{U_i = a|Y_s = y_s, Y_1 = y_1\}$

($a \in A$ で, A は情報シンボルのアルファベット)

組織符号とその最大事後確率復号

$$\begin{aligned}
 \text{BEL}_i(a) &= \Pr\{U_i = a | Y = y\} \\
 &= \frac{\Pr\{U_i = a, Y = y\}}{\Pr\{Y = y\}} \\
 &= \alpha \Pr\{U_i = a, Y = y\} \\
 &= \alpha \sum_{u_1, \dots, u_K : u_i = a} P(y, u) \\
 &= \alpha \sum_{u_1, \dots, u_K : u_i = a} P(y|u) P(u) \quad \left(\begin{array}{l} P(u_i) = \pi_{u_i}(u_i) \\ P(y_{sj} | u_j) = \lambda_{y_{sj} \rightarrow u_j}(u_j) \end{array} \right) \\
 &= \alpha \sum_{u_1, \dots, u_K : u_i = a} P(y_1 | x_1) P(y_s | u) \prod_{j=1}^K P(u_j) \\
 &= \alpha \sum_{u_1, \dots, u_K : u_i = a} P(y_1 | x_1) \prod_{j=1}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j) \\
 &= \alpha \lambda_{y_{si} \rightarrow u_i}(a) \pi_{u_i}(a) \sum_{u_1, \dots, u_K : u_i = a} P(y_1 | x_1) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j)
 \end{aligned}$$

組織符号とその最大事後確率復号

$$\text{BEL}_i(a) = \alpha \sum_{u_1, \dots, u_K : u_i = a} P(y_1 | x_1) \prod_{j=1}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j)$$

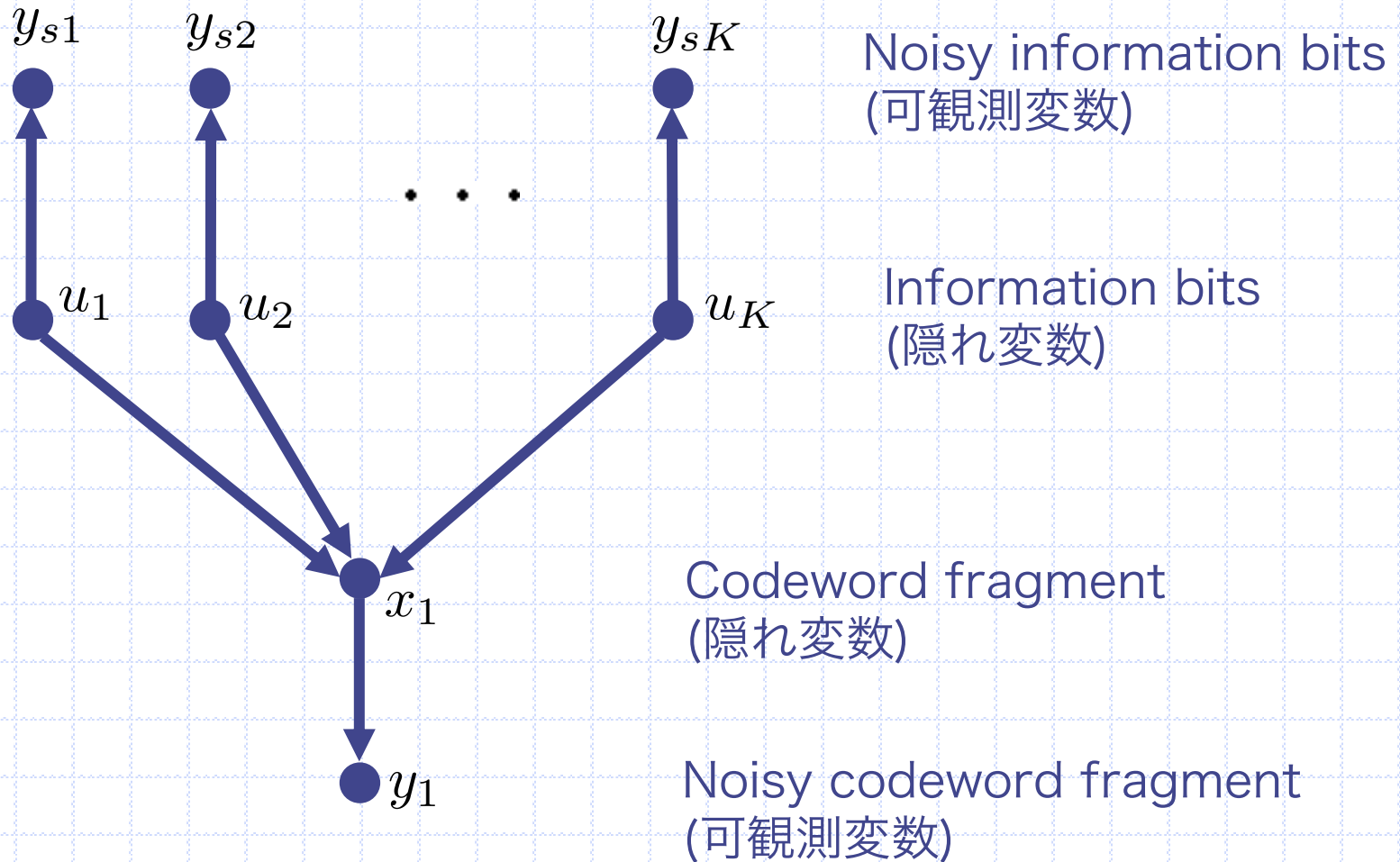
$$= \alpha \lambda_{y_{si} \rightarrow u_i}(a) \pi_{u_i}(a) \sum_{u_1, \dots, u_K : u_i = a} P(y_1 | x_1) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j)$$

systematic
evidence

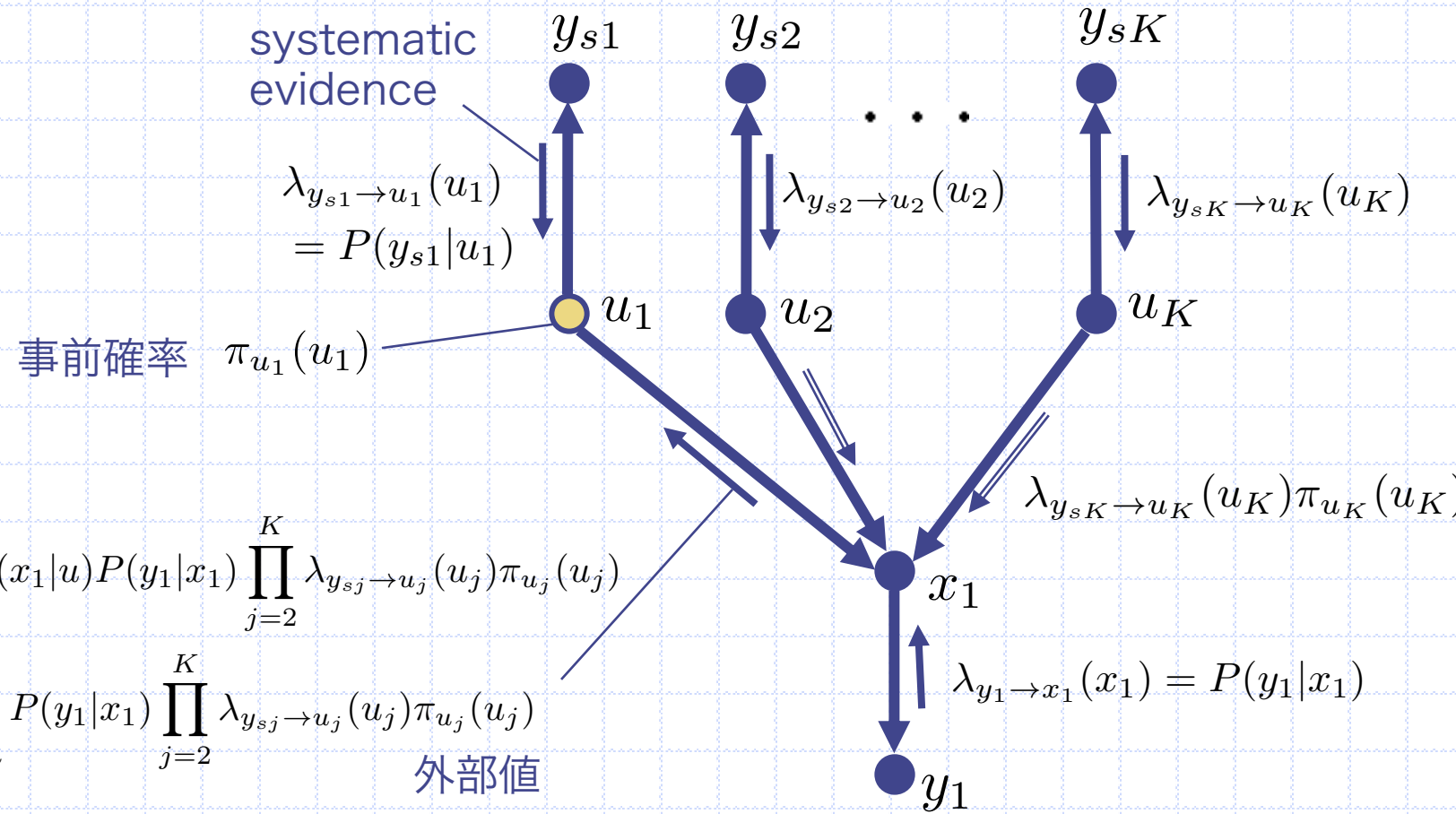
a priori
Information
(事前確率)

extrinsic value
(外部値)

組織符号のベイジアンネットワーク表現



組織符号のベイジアンネットワーク表現にBPを適用

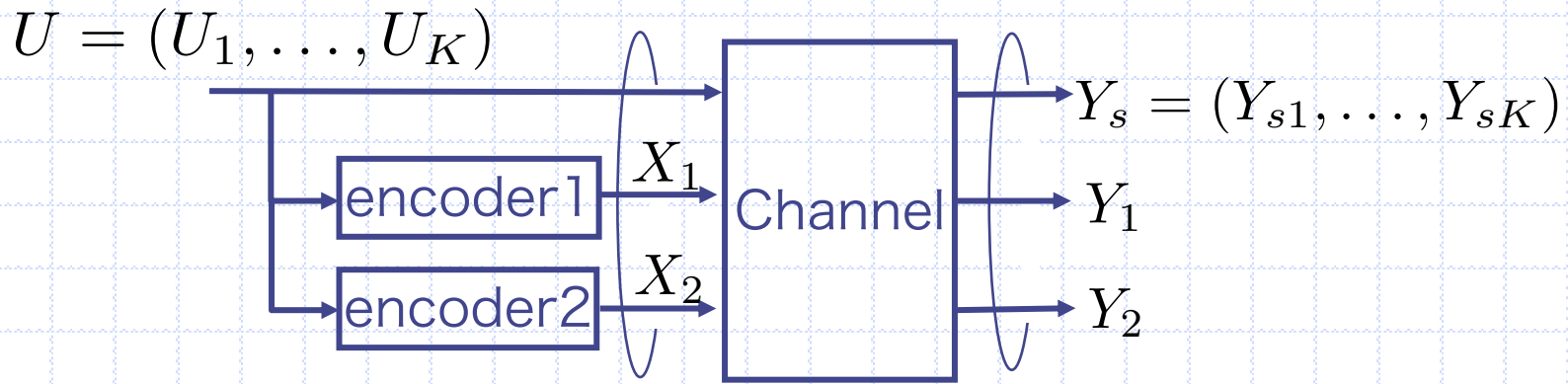


$$\sum_{u_1, \dots, u_K: u_1=a} P(x_1|u) P(y_1|x_1) \prod_{j=2}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j)$$

$$= \sum_{u_1, \dots, u_K: u_1=a} P(y_1|x_1) \prod_{j=2}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j)$$

- \longrightarrow π : 子 \rightarrow 親のメッセージ
- \Longrightarrow λ : 親 \rightarrow 子のメッセージ

並列接続組織符号（ターボ符号）とその最大事後確率復号



$$C : U \longrightarrow X = (U, X_1, X_2) \qquad Y = (Y_s, Y_1, Y_2)$$

通信路の遷移確率:

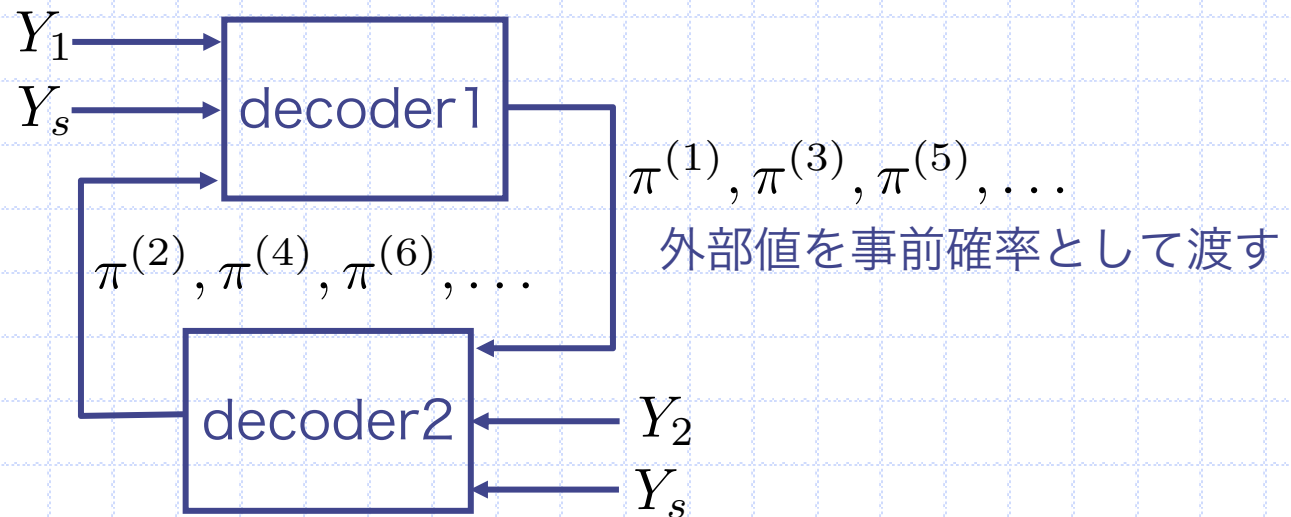
$$\begin{aligned} P(y|x) &= P(y_s, y_1, y_2 | u, x_1, x_2) \\ &= P(y_s | u) P(y_1 | x_1) P(y_2 | x_2) \\ &= \left(\prod_{i=1}^K P(y_{si} | u_i) \right) P(y_1 | x_1) P(y_2 | x_2) \end{aligned}$$

並列接続組織符号（ターボ符号）とその最大事後確率復号(続)

シンボル毎最大事後確率復号をするための事後周辺確率：

$$\begin{aligned} \text{BEL}_i(a) &= \alpha \sum_{u_1, \dots, u_K : u_i = a} P(y_1 | x_1) P(y_2 | x_2) \prod_{j=1}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j) \\ &= \alpha \lambda_{y_{si} \rightarrow u_i}(a) \pi_{u_i}(a) \sum_{u_1, \dots, u_K : u_i = a} P(y_1 | x_1) P(y_2 | x_2) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}(u_j) \end{aligned}$$

ターボ復号法：



並列連接組織符号（ターボ符号）とその最大事後確率復号(続)

• (2m-1)-th Step :

事前確率が $P(u_i) = \pi_{u_i}^{(2m-2)}(u_i)$ で与えられるときの符号語 (U, X_1) のみに対応する周辺事後確率：

$$\Pr\{U_i = a | Y_s = y_s, Y_1 = y_1\} = \alpha \sum_{u_1, \dots, u_K : u_i = a} P(y_1 | x_1) \prod_{j=1}^K \lambda_{y_{s_j} \rightarrow u_j}(u_j) \pi_{u_j}^{(2m-2)}(u_j)$$

$$= \alpha \lambda_i(a) \pi_{u_i}^{(2m-2)}(a) \sum_{u_1, \dots, u_K : u_i = a} P(y_1 | x_1) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{s_j} \rightarrow u_j}(u_j) \pi_{u_j}^{(2m-2)}(u_j)$$

extrinsic value (外部値)

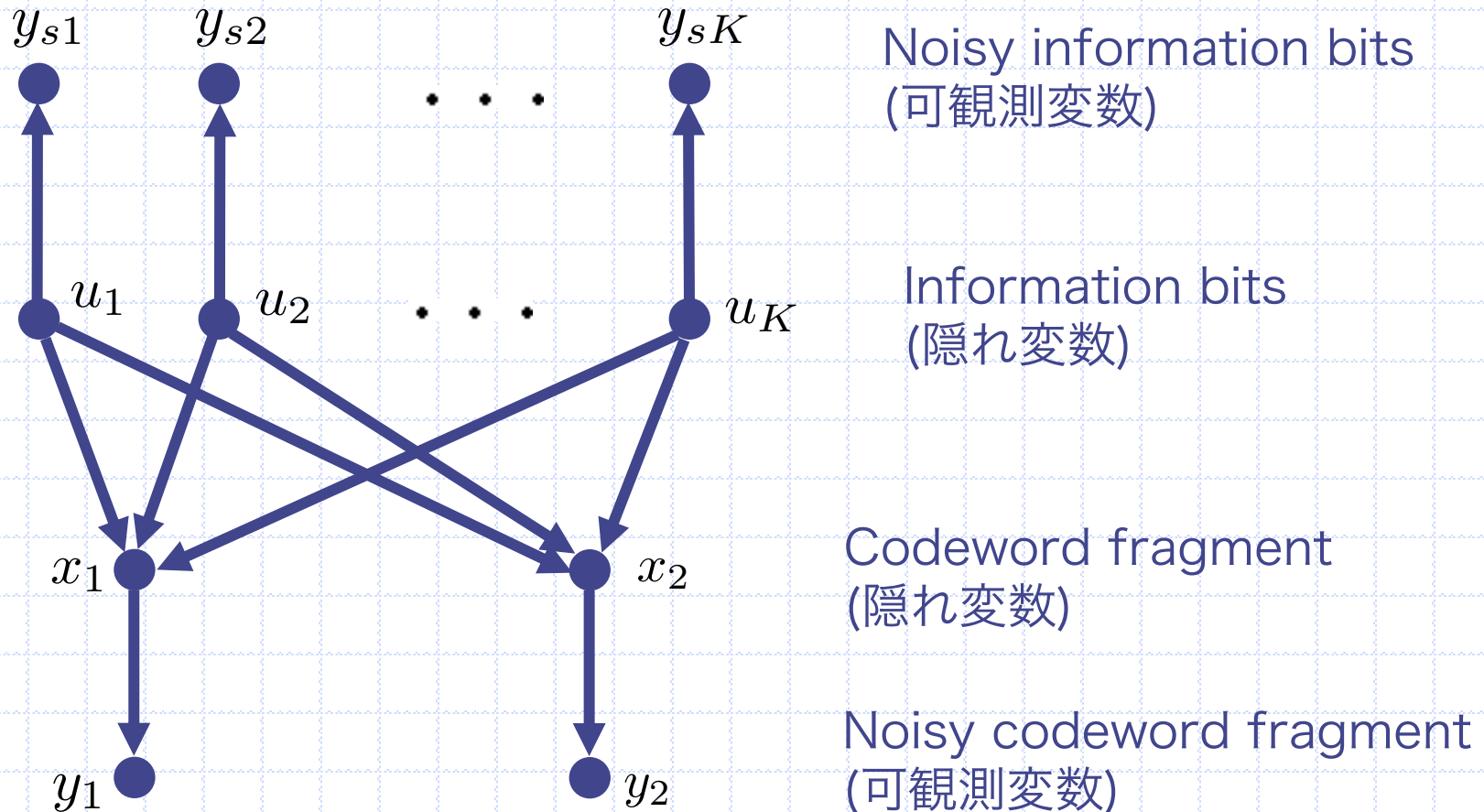
• 2m-th Step :

事前確率が $P(u_i) = \pi_{u_i}^{(2m-1)}(u_i)$ で与えられるときの符号語 (U, X_2) のみに対応する事後周辺分布：

$$\Pr(U_i = a | Y_s = y_s, Y_2 = y_2) = \alpha \lambda_{y_{s_i} \rightarrow u_i}(a) \pi_{u_i}^{(2m-1)}(a) \sum_{u_1, \dots, u_K : u_i = a} P(y_2 | x_2) \prod_{\substack{j=1 \\ j \neq i}}^K \lambda_{y_{s_j} \rightarrow u_j}(u_j) \pi_{u_j}^{(2m-1)}(u_j)$$

(2m+1)-th Step ^

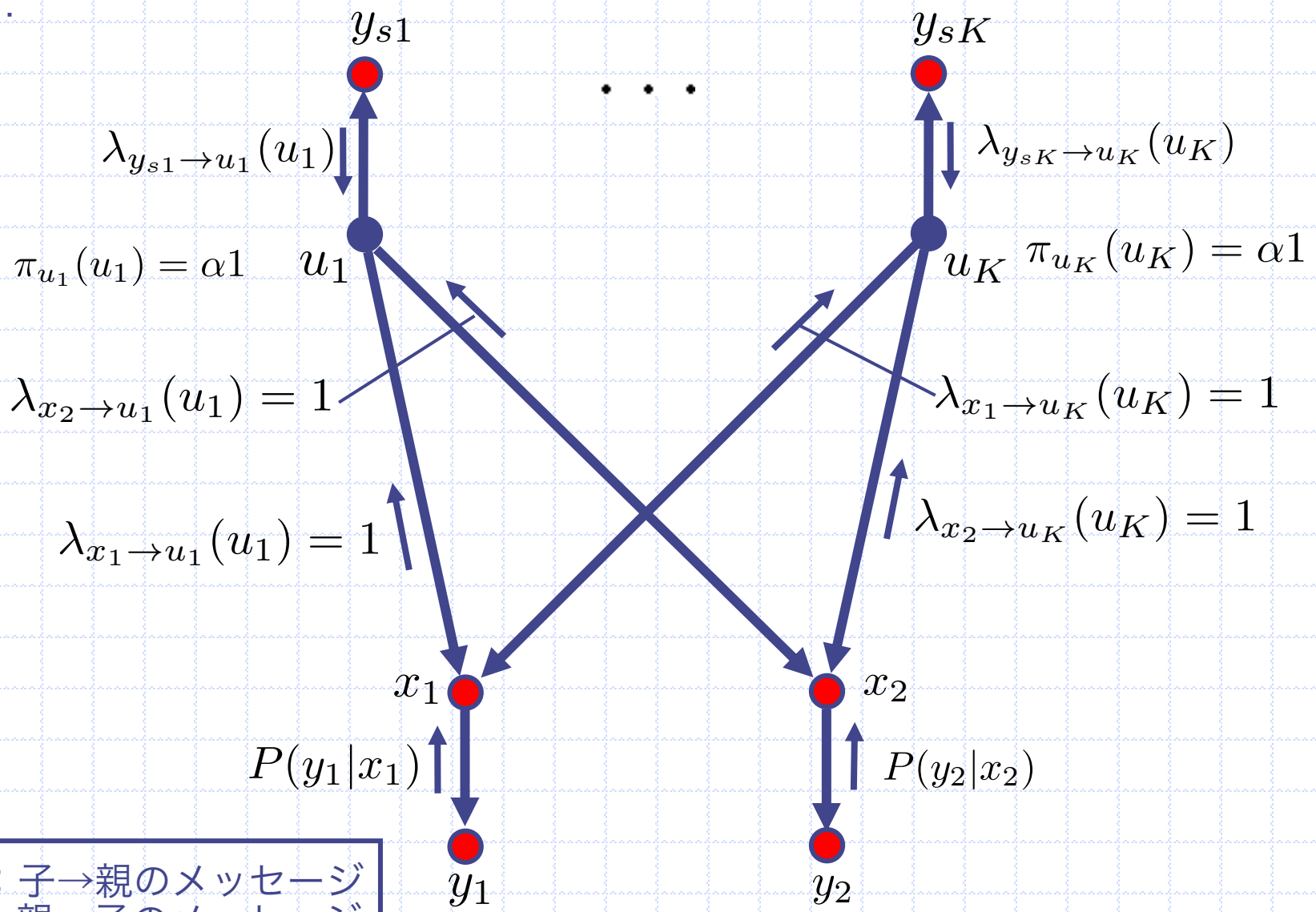
ターボ復号のベイジアンネットワーク表現



ループがあるため、BPを適用しても真の事後周辺分布は得られない
(Loopy BP)

ターボ符号のベイジアンネットワーク表現にBPを適用

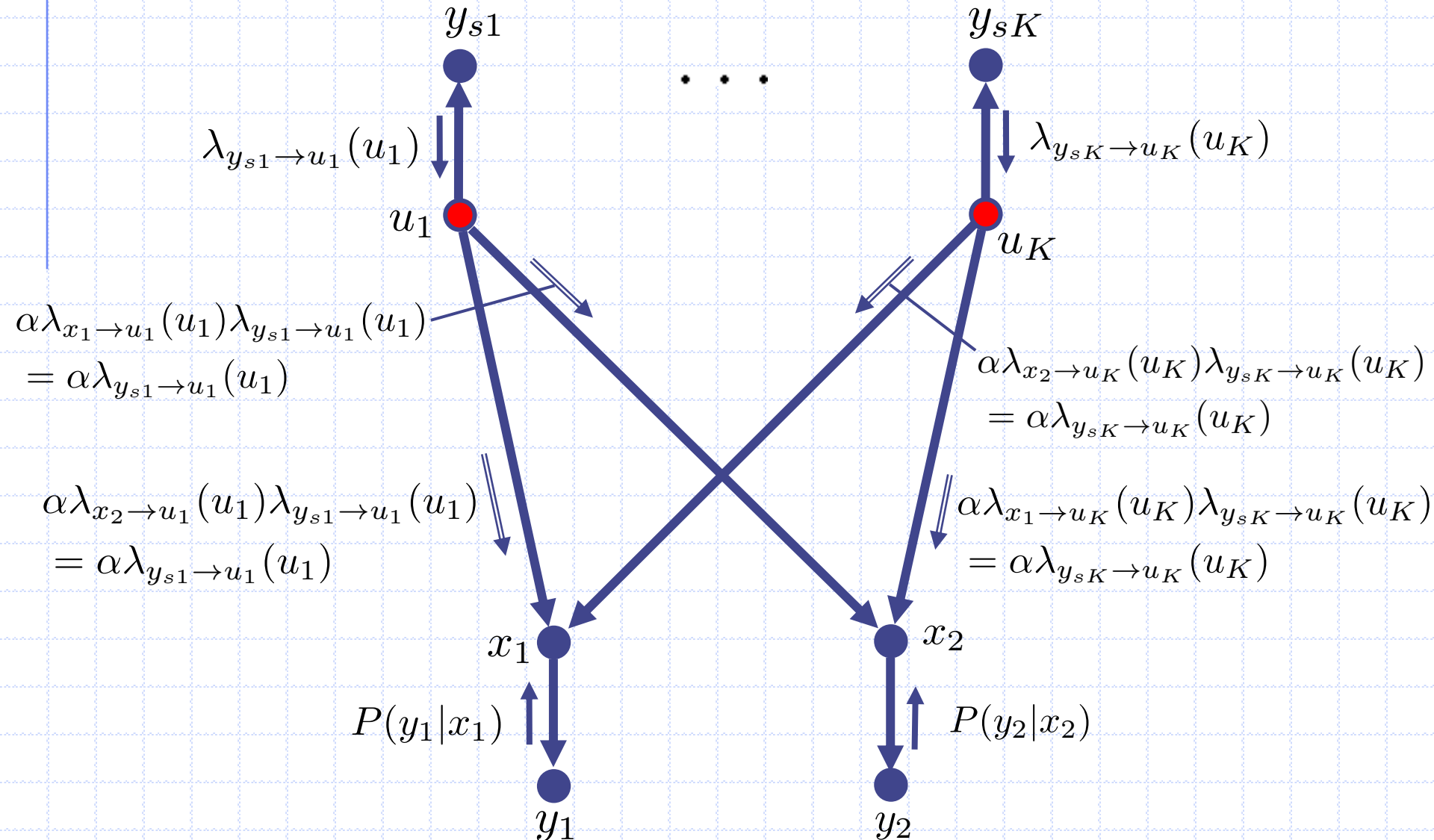
・初期化：



π : 子→親のメッセージ
 λ : 親→子のメッセージ

ターボ符号のベイジアンネットワーク表現にBPを適用 (続)

・ノード u での処理:



ターボ符号のベイジアンネットワーク表現にBPを適用 (続)

・ノード x_1 での処理：

$$\alpha \sum_{u_1, \dots, u_K \setminus u_1} \left(\sum_{x_1} P(x_1|u) P(y_1|x_1) \right) \prod_{j=2}^K \lambda_{y_{sj} \rightarrow u_j}(u_j)$$

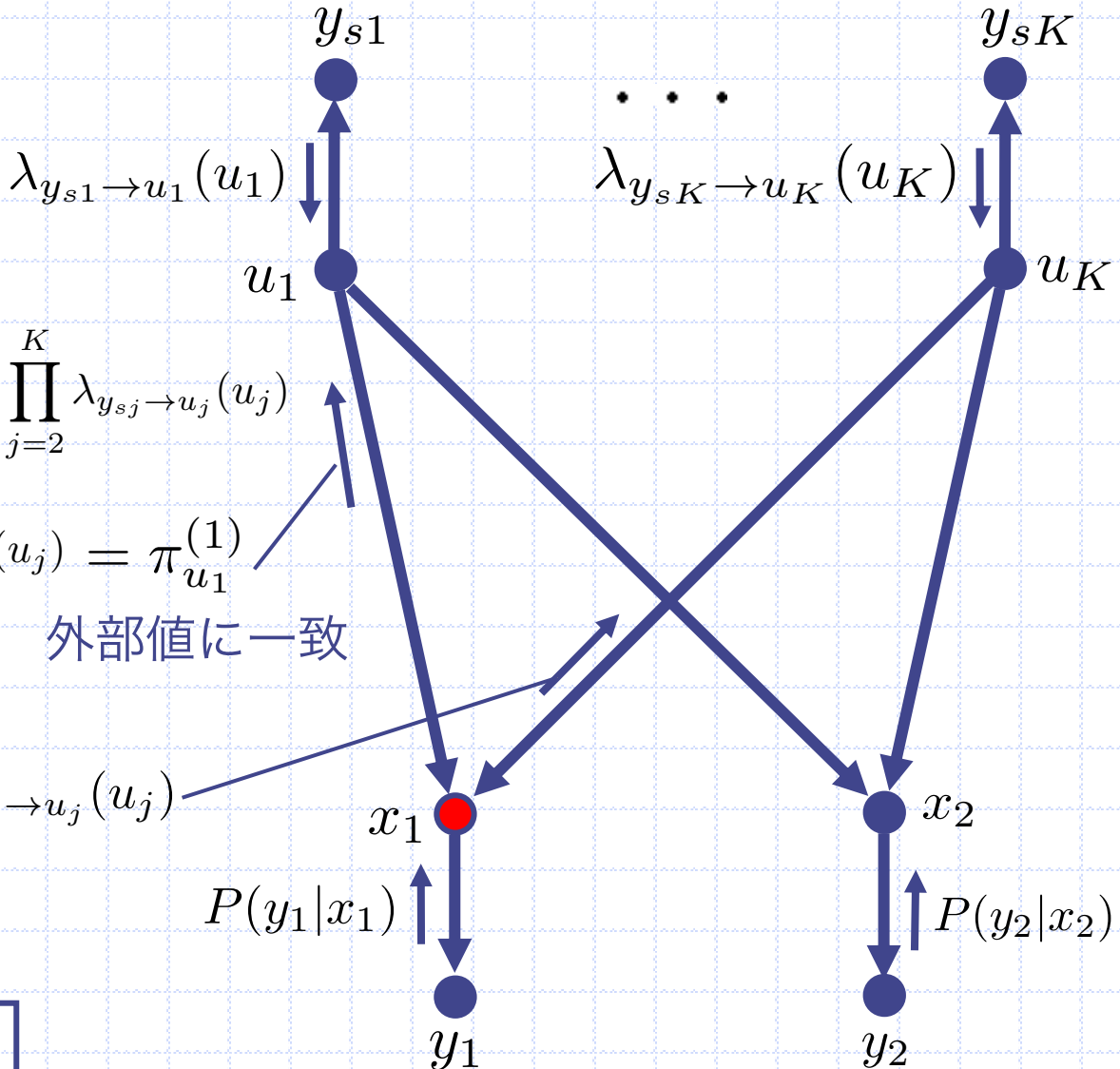
$$= \alpha \sum_{u_1, \dots, u_K \setminus u_1} P(y_1|x_1) \prod_{j=2}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) = \pi_{u_1}^{(1)}$$

外部値に一致

$$\alpha \sum_{u_1, \dots, u_K \setminus u_K} P(y_1|x_1) \prod_{j=1}^{K-1} \lambda_{y_{sj} \rightarrow u_j}(u_j)$$

$$= \pi_{u_K}^{(1)}$$

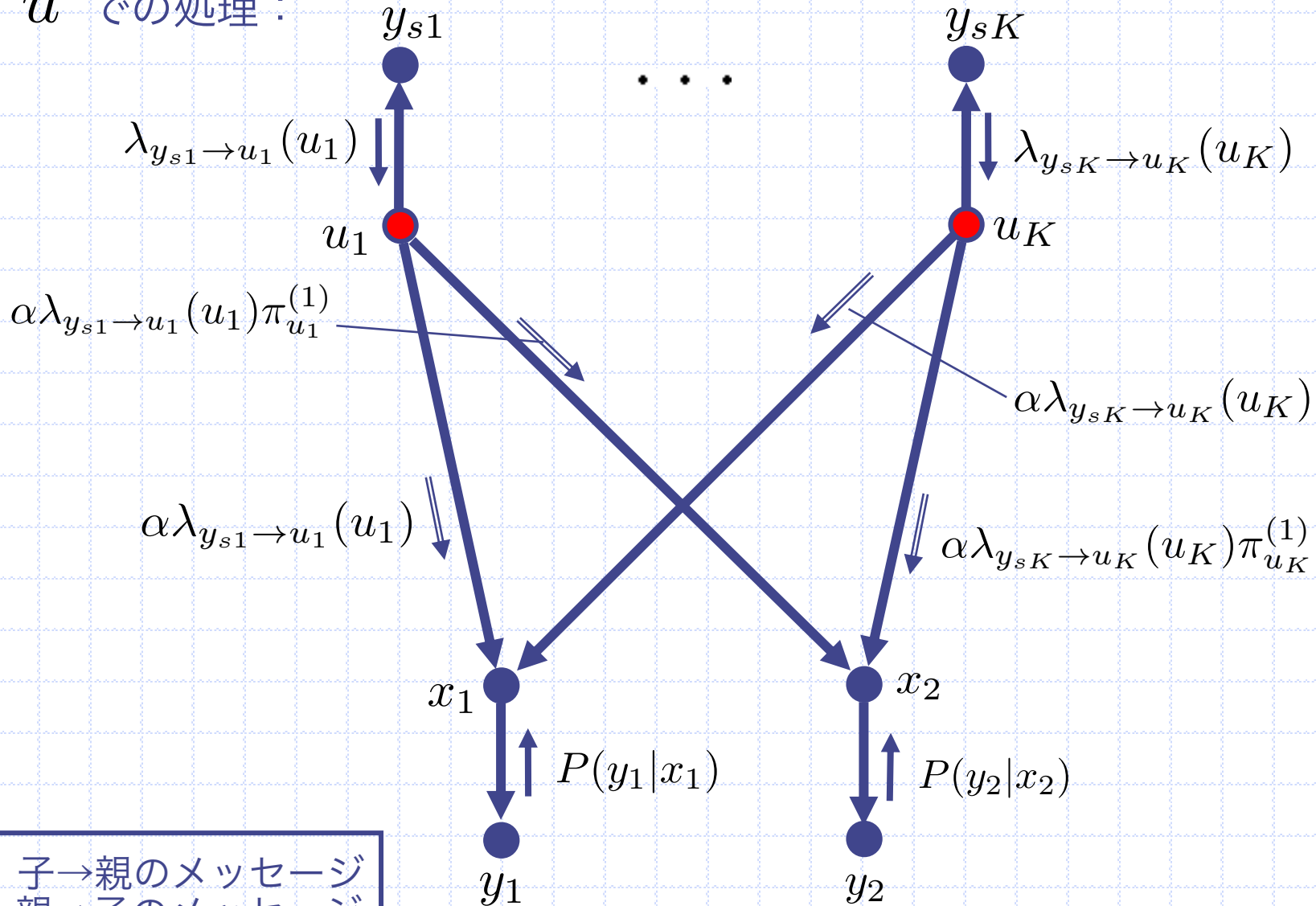
外部値に一致



→ π : 子→親のメッセージ
→ λ : 親→子のメッセージ

ターボ符号のベイジアンネットワーク表現にBPを適用 (続)

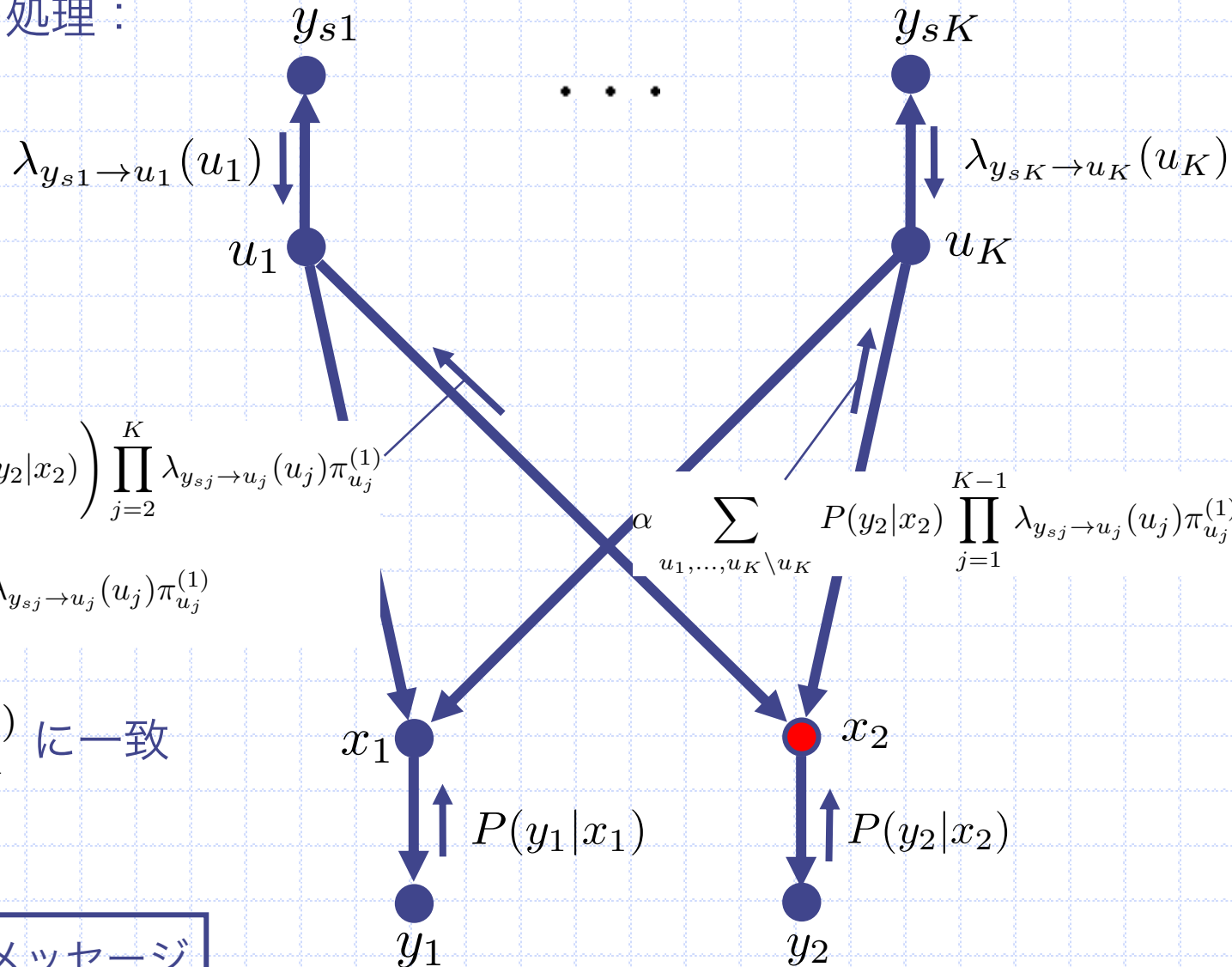
・ノード u での処理：



→ π : 子→親のメッセージ
⇒ λ : 親→子のメッセージ

ターボ符号のベイジアンネットワーク表現にBPを適用 (続)

・ノード x_2 での処理:



$$\alpha \sum_{u_1, \dots, u_K \setminus u_1} \left(\sum_{x_2} P(x_2|u) P(y_2|x_2) \right) \prod_{j=2}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}^{(1)}$$

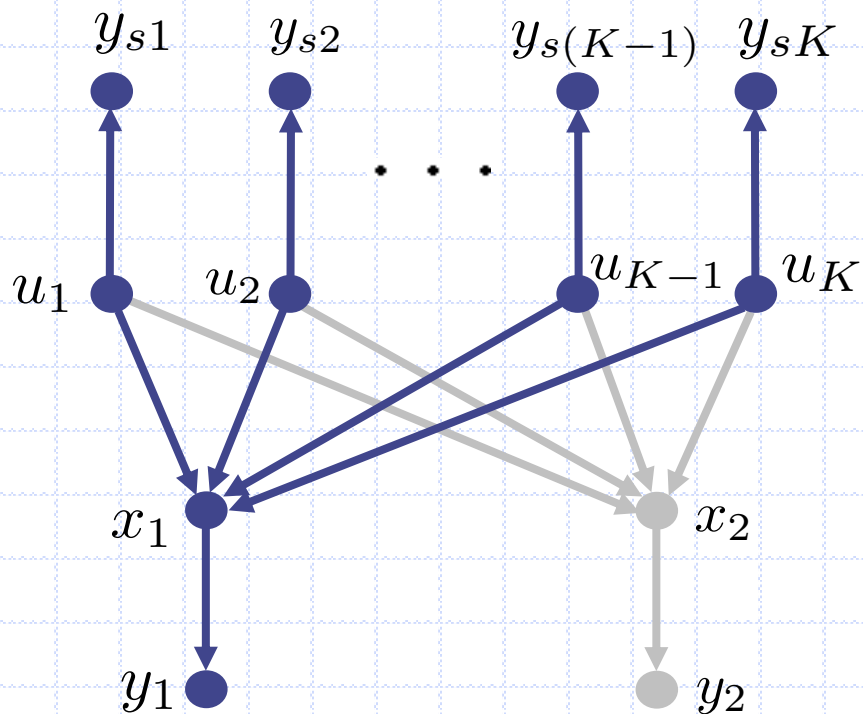
$$= \alpha \sum_{u_1, \dots, u_K \setminus u_1} P(y_2|x_2) \prod_{j=2}^K \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}^{(1)}$$

$$\alpha \sum_{u_1, \dots, u_K \setminus u_K} P(y_2|x_2) \prod_{j=1}^{K-1} \lambda_{y_{sj} \rightarrow u_j}(u_j) \pi_{u_j}^{(1)}$$

これは外部値 $\pi_{u_1}^{(2)}$ に一致

\longrightarrow π : 子 \rightarrow 親のメッセージ
 \longleftarrow λ : 親 \rightarrow 子のメッセージ

ターボ復号で外部値を次段の事前値とする理由



ターボ復号ではそれぞれの要素符号の復号を交互に行うため、例えば符号語 (U, X_1) の復号アルゴリズムからはベイジアンネットワークが上図のように見えている

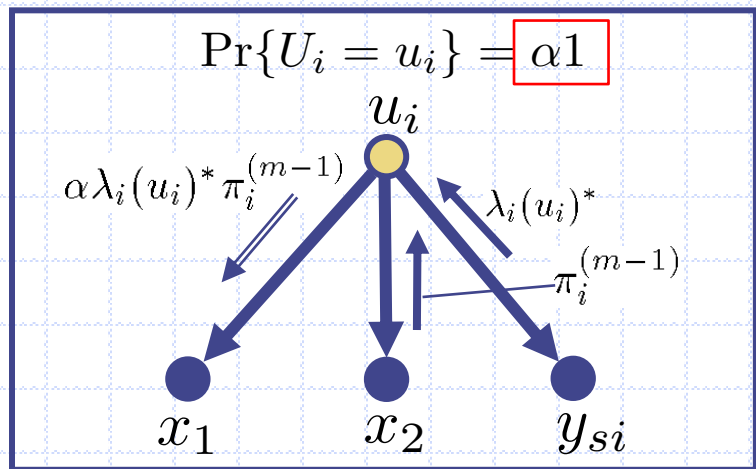
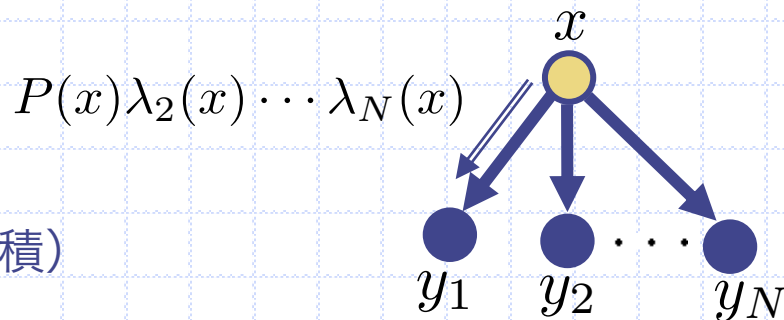
どうすればもう1つの符号語の持つ情報をうまく取り込めるのか？

ターボ復号で外部値を次段の事前値とする理由（続）

もう1つの符号語とはノード u_i を介してつながっている

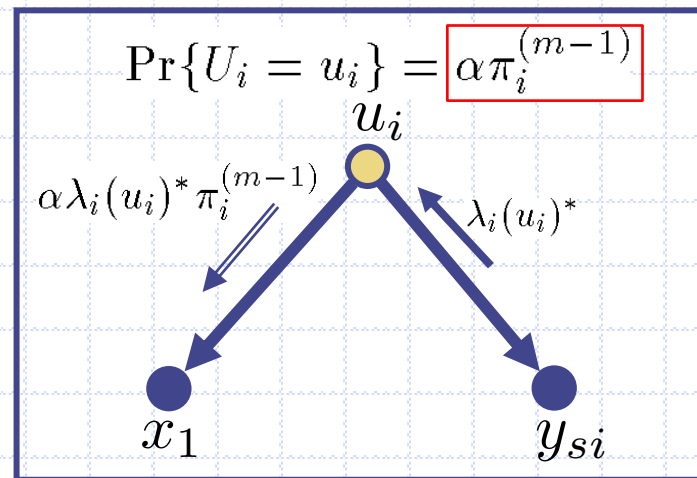
u_i は“親がない，子が複数のノード”

x が y_1 に送るメッセージ
 = (x が持つ事前確率)
 × (y_1 以外から届いたメッセージの積)



x_1 にとっては
 どちらも同じ

(周辺事後確率を計算する際の u_i にとっても同じ)



x_2 からのメッセージを事前確率として用いればよい
 実は外部値に等しい

高速フーリエ変換(FFT)

離散フーリエ変換(DFT):

$$W_k = \sum_{n=0}^{N-1} w(n) e^{-j \frac{2\pi}{N} nk}$$

FFTの原理：

$$W_k = \sum_{n=0}^{\frac{N}{2}-1} w(2n) e^{-j \frac{2\pi}{N} 2nk} + \sum_{n=0}^{\frac{N}{2}-1} w(2n+1) e^{-j \frac{2\pi}{N} (2n+1)k}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} w(2n) e^{-j \frac{2\pi}{N} nk} + e^{-j \frac{2\pi}{N} k} \sum_{n=0}^{\frac{N}{2}-1} w(2n+1) e^{-j \frac{2\pi}{N} nk}$$

N/2ポイントのDFT



2ポイントになるまで繰り返す (バタフライ演算)

高速フーリエ変換(FFT) (続)

N=8の場合：

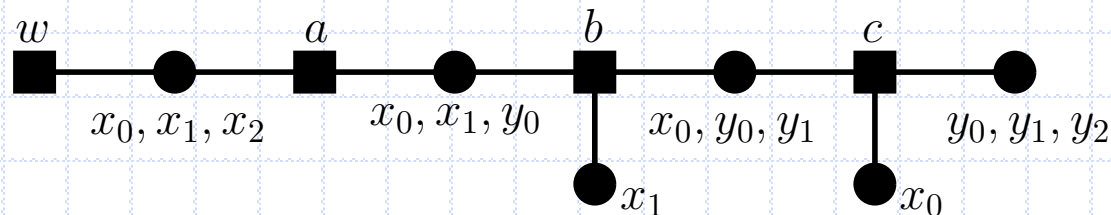
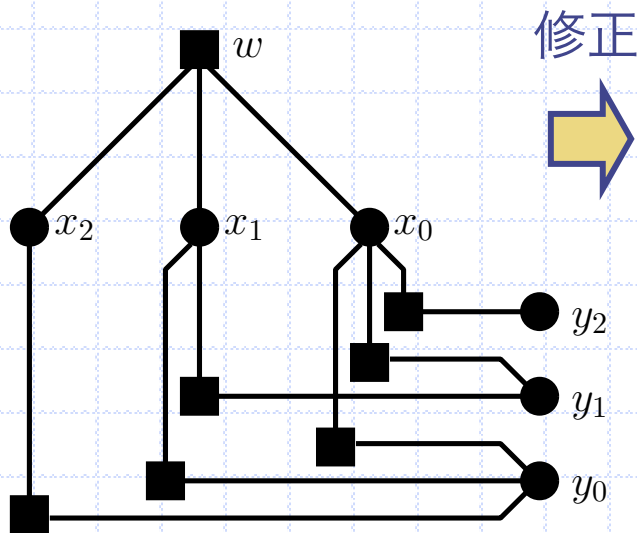
$$\left[\begin{array}{l} \text{2進表現} \\ n = 4x_2 + 2x_1 + x_0 \\ k = 4y_2 + 2y_1 + y_0 \end{array} \quad x_i, y_i \in \{0, 1\} \right]$$

$$W_k = \sum_{n=0}^7 w(n) e^{-j \frac{2\pi}{8} nk}$$

$$= \sum_{x_0, x_1, x_2} w(4x_2 + 2x_1 + x_0) e^{-j \frac{2\pi}{8} (4x_2 + 2x_1 + x_0)(4y_2 + 2y_1 + y_0)}$$

$$= \sum_{x_0, x_1, x_2} w(4x_2 + 2x_1 + x_0) (-1)^{x_2 y_0} (-1)^{x_1 y_1} (-1)^{x_0 y_2} (j)^{-x_0 y_1} (j)^{-x_1 y_0} e^{-j \frac{2\pi}{8} (x_0 y_0)}$$

$$= \sum_{x_0} (-1)^{x_0 y_2} (j)^{-x_0 y_1} e^{-j \frac{2\pi}{8} x_0 y_0} \sum_{x_1} (-1)^{x_1 y_1} (j)^{-x_1 y_0} \sum_{x_2} w(4x_2 + 2x_1 + x_0) (-1)^{x_2 y_0}$$



これに対する和-積sum-productアルゴリズムはFFTのアルゴリズムに等しい

近似メッセージ伝搬法 (AMP) : ℓ_1 最適化

- ・ 近似メッセージ伝搬法 (Approximate Message Passing, AMP)
- ・ ℓ_1 最適化問題：未知のスパースベクトル $\mathbf{b} = [b_1, \dots, b_N]^T \in \mathbb{R}^N$ をその線形観測 $\mathbf{y} = \mathbf{A}\mathbf{b} \in \mathbb{R}^M$ から再構成する手法

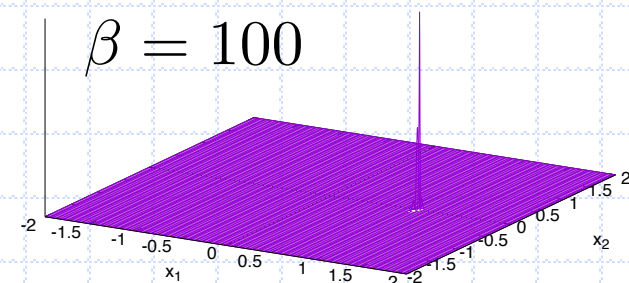
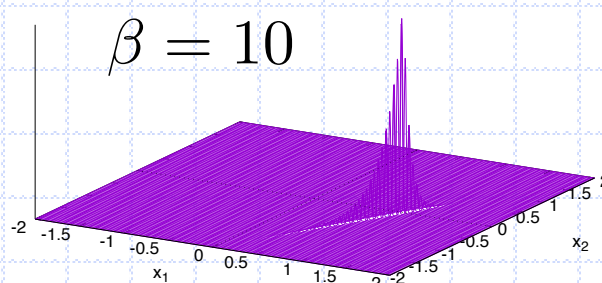
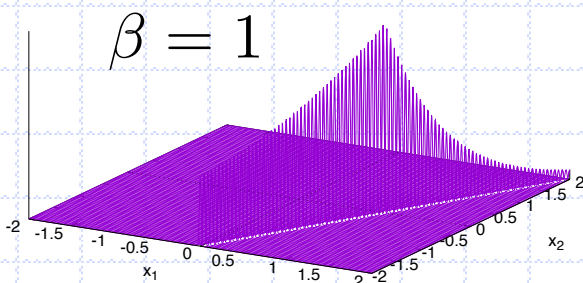
$$\hat{\mathbf{b}} = \arg \min_{\mathbf{s} \in \mathbb{R}^N} \|\mathbf{s}\|_1 \quad \text{subject to } \mathbf{y} = \mathbf{A}\mathbf{s}$$



対応する確率分布

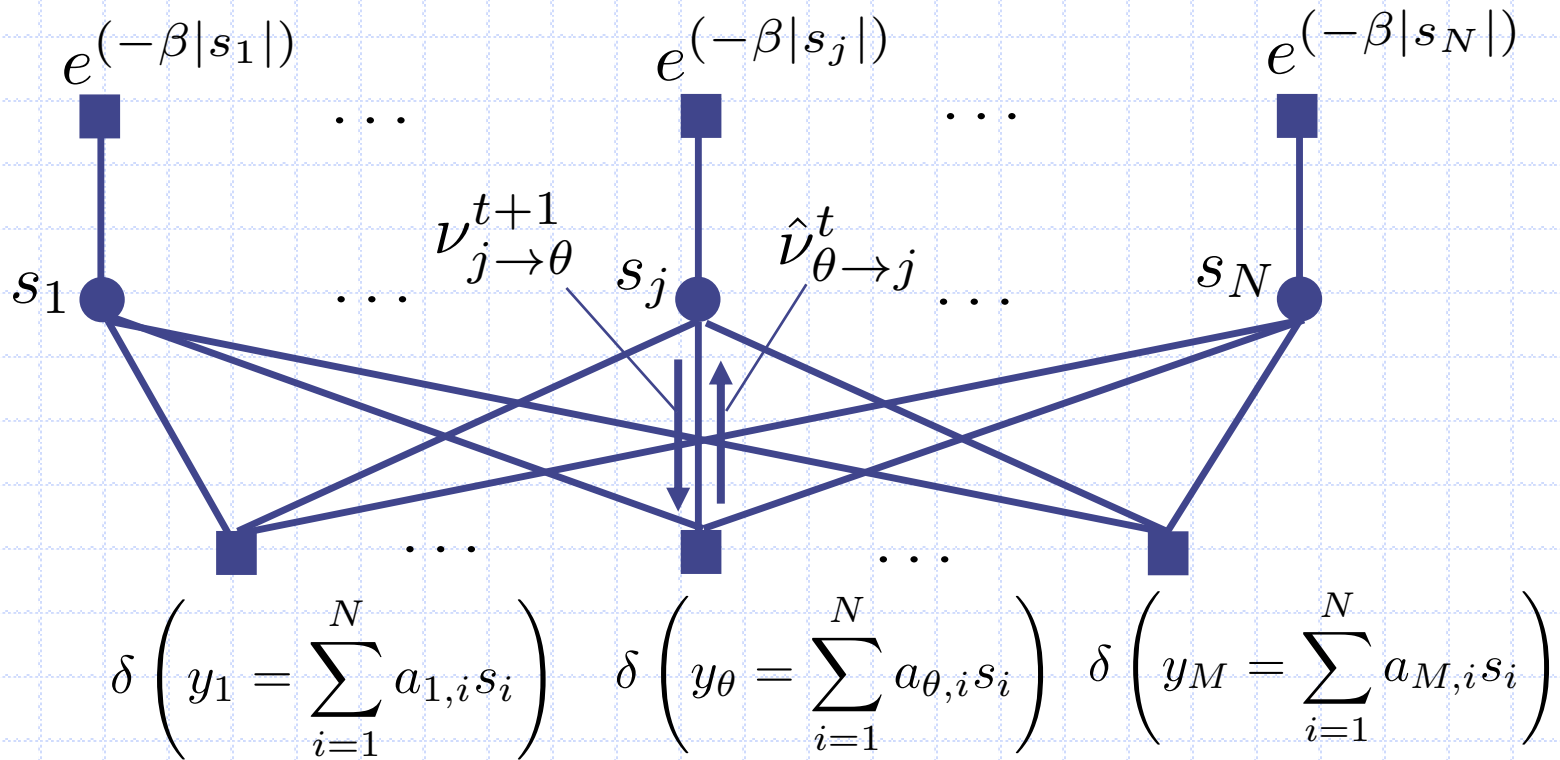
$$\mu(\mathbf{s}) \propto \prod_{j=1}^N \exp(-\beta |s_j|) \prod_{\theta=1}^M \delta \left(y_\theta = \sum_{i=1}^N a_{\theta,i} s_i \right)$$

$\beta \rightarrow +\infty$ を考えると、確率質量が最適解に集中する



近似メッセージ伝搬法 (AMP) : ℓ_1 最適化

- ℓ_1 最適化のファクターグラフ表現：



和-積アルゴリズムの
メッセージ：

$$\nu_{j \rightarrow \theta}^{t+1}(s_j) \propto \exp(-\beta|s_j|) \prod_{\zeta \neq \theta} \hat{\nu}_{\zeta \rightarrow j}^t(s_j)$$

$$\hat{\nu}_{\theta \rightarrow j}^t(s_j) \propto \int \delta\left(y_\theta = \sum_{i=1}^N a_{\theta,i} s_i\right) \prod_{k \neq j} \nu_{k \rightarrow \theta}^t(s_k) ds_{\setminus j}$$

近似メッセージ伝搬法 (AMP) : ℓ_1 最適化

- 大システム極限 ($M, N \rightarrow \infty, M/N = \Delta$) を考える

$\hat{\nu}_{\theta \rightarrow j}^t(s_j) \leftarrow$ 平均 $\frac{z_{\theta \rightarrow j}^t}{a_{\theta, j}}$, 分散 $\frac{\hat{\tau}_{\theta \rightarrow j}^t}{\beta a_{\theta, j}^2}$ のガウス分布で近似

ただし,

$$z_{\theta \rightarrow j}^t = y_{\theta} - \sum_{i \neq j} a_{\theta, i} x_{i \rightarrow \theta}^t$$

$$\hat{\tau}_{\theta \rightarrow j}^t = \sum_{i \neq j} a_{\theta, i}^2 \tau_{i \rightarrow \theta}^t$$

で, $x_{i \rightarrow \theta}^t, \frac{\tau_{i \rightarrow \theta}^t}{\beta}$ は $\nu_{i \rightarrow \theta}^t$ の平均と分散

近似メッセージ伝搬法 (AMP) : ℓ_1 最適化

ガウス近似を利用すると

$$\begin{aligned}\hat{\nu}_{\theta \rightarrow j}^t(s_j) &\approx \hat{\phi}_{\theta \rightarrow j}^t(s_j) \\ &:= \sqrt{\frac{\beta a_{\theta,j}^2}{2\pi \hat{\tau}_{\theta \rightarrow j}^t}} \exp \left\{ -\frac{\beta}{2\hat{\tau}_{\theta \rightarrow j}^t} (a_{\theta,j} s_j - z_{\theta \rightarrow j}^t)^2 \right\}\end{aligned}$$

となり、 $\nu_{i \rightarrow \theta}^t(s_i)$ ($i \neq j$) の平均と分散のみから $\hat{\nu}_{\theta \rightarrow j}^t(s_j)$ の近似値が計算可能

$\nu_{j \rightarrow \theta}^{t+1}(s_j)$ の平均と分散について考える

近似メッセージ伝搬法 (AMP) : ℓ_1 最適化

次の分布に従う確率変数の平均と分散をそれぞれ $F_\beta(u, c)$, $G_\beta(u, c)$ とする

$$f_\beta(s; u, c) \propto \exp \left\{ -\beta|s| - \frac{\beta}{2c}(s-u)^2 \right\},$$

- $\hat{\tau}_{\theta \rightarrow j}^t$ が θ, j に依存しない ($\hat{\tau}_{\theta \rightarrow j}^t = \hat{\tau}^t$ と書ける) とする
- $\hat{\nu}_{\theta \rightarrow j}^t(s_j) = \hat{\phi}_{\theta \rightarrow j}^t(s_j)$ であるとする.



このとき, $\nu_{j \rightarrow \theta}^{t+1}(s_j) = \phi_{j \rightarrow \theta}^{t+1}(s_j) \left\{ 1 + \mathcal{O} \left(\frac{s_j^2}{M} \right) \right\}$ と書ける

$$\text{ただし, } \phi_{j \rightarrow \theta}^{t+1}(s_j) := f_\beta \left(s_j, \sum_{\zeta \neq \theta} a_{\zeta, j} z_{\zeta \rightarrow j}^t, \hat{\tau}^t \right)$$

$\nu_{j \rightarrow \theta}^{t+1}(s_j)$ の平均 :

分散 :

$$x_{j \rightarrow \theta}^{t+1} = F_\beta \left(\sum_{\zeta \neq \theta} a_{\zeta, j} z_{\zeta \rightarrow j}^t, \hat{\tau}^t \right), \quad \tau_{j \rightarrow \theta}^t = \beta G_\beta \left(\sum_{\zeta \neq \theta} a_{\zeta, j} z_{\zeta \rightarrow j}^t, \hat{\tau}^t \right)$$

近似メッセージ伝搬法 (AMP) : ℓ_1 最適化

$\nu_{j \rightarrow \theta}^t(s_j)$ の平均 : $x_{j \rightarrow \theta}^t$

分散 : $\tau_{j \rightarrow \theta}^t / \beta$

$\nu_{j \rightarrow \theta}^{t+1}(s_j)$ の平均 :

$$x_{j \rightarrow \theta}^{t+1} = F_{\beta} \left(\sum_{\zeta \neq \theta} a_{\zeta, j} z_{\zeta \rightarrow j}^t, \hat{\tau}^t \right),$$

分散 :

$$\tau_{j \rightarrow \theta}^t = \beta G_{\beta} \left(\sum_{\zeta \neq \theta} a_{\zeta, j} z_{\zeta \rightarrow j}^t, \hat{\tau}^t \right)$$

なので, 次の更新式が得られる

$$x_{j \rightarrow \theta}^{t+1} = F_{\beta} \left(\sum_{\zeta \neq \theta} a_{\zeta, j} z_{\zeta \rightarrow j}^t, \hat{\tau}^t \right),$$

$$z_{\theta \rightarrow j}^t = y_{\theta} - \sum_{i \neq j} a_{\theta, i} x_{i \rightarrow \theta}^t,$$

θ, j に依存しない形に近似

$$\hat{\tau}_{\theta \rightarrow j}^{t+1} = \frac{\beta}{M} \sum_{i \neq j} G_{\beta} \left(\sum_{\zeta \neq \theta} a_{\zeta, i} z_{\zeta \rightarrow i}^t, \hat{\tau}^t \right).$$

$$\hat{\tau}^{t+1} = \frac{\beta}{M} \sum_{i=1}^N G_{\beta} \left(\sum_{\zeta=1}^M a_{\zeta, i} z_{\zeta \rightarrow i}^t, \hat{\tau}^t \right)$$

近似メッセージ伝搬法 (AMP) : ℓ_1 最適化

$\beta \rightarrow +\infty$ の極限を考える

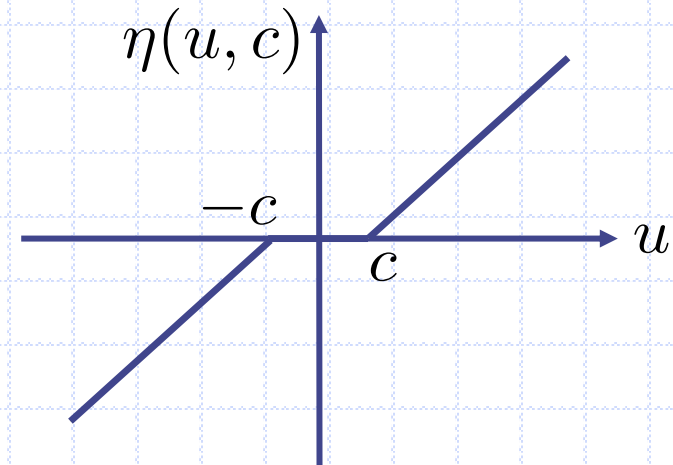
平均値は $f_\beta(s; u, c) \propto \exp \left\{ -\beta|s| - \frac{\beta}{2c}(s-u)^2 \right\}$, の指数部の最大値に支配される

$$F_\beta(u, c) \rightarrow \arg \min_{s \in \mathbb{R}} \left\{ |s| + \frac{1}{2c}(s-u)^2 \right\} = \eta(u, c)$$

ただし,

$$\eta(u, c) = \begin{cases} u + c & (u < -c) \\ 0 & (-c \leq u \leq c) \\ u - c & (c < u) \end{cases}$$

弱しきい値関数



近似メッセージ伝搬法 (AMP) : ℓ_1 最適化

$\beta \rightarrow +\infty$ の極限を考える

分散 $G_\beta(u, c)$ は平均 $\eta(u, c)$ の近傍での $f_\beta(s; u, c)$ の値を調べることで得る

1) $-c \leq u \leq c$ のとき, $\eta(u, c) = 0$ なので $s = 0$ の近傍を考えると

$$f_\beta(s; u, c) \propto \exp \left\{ -\beta|s| - \frac{\beta}{2c}(s - u)^2 \right\} \approx \exp(-\beta|s|)$$

: ラプラス分布

と近似されるので, 分散は $2/\beta^2$

2) $c < u$ のとき, $\eta(u, c) = u - c > 0$ なので

$$f_\beta(s; u, c) \approx \exp \left\{ -\frac{\beta}{2c}(s - u + c)^2 \right\} : \text{ガウス分布}$$

と近似されるので, 分散は c/β

近似メッセージ伝搬法 (AMP) : ℓ_1 最適化

したがって,

$$\lim_{\beta \rightarrow \infty} F_{\beta}(u, c) = \begin{cases} u + c & (u < -c) \\ 0 & (-c \leq u \leq c), \\ u - c & (c < u) \end{cases}$$

$$\lim_{\beta \rightarrow \infty} \beta \mathbf{G}_{\beta}(u, c) = c \eta'(u, c) = \begin{cases} c & (u < -c) \\ 0 & (-c \leq u \leq c) \\ c & (c < u) \end{cases}$$

となる.

更新式 :

$$x_{j \rightarrow \theta}^{t+1} = \eta \left(\sum_{\zeta \neq \theta} a_{\zeta i} z_{\zeta \rightarrow i}^t, \hat{\tau}^t \right),$$

$$z_{\theta \rightarrow j}^t = y_{\theta} - \sum_{i \neq j} a_{\theta, i} x_{i \rightarrow \theta}^t,$$

$$\hat{\tau}^{t+1} = \frac{\hat{\tau}^t}{M} \sum_{i=1}^N \eta' \left(\sum_{\zeta=1}^M a_{\zeta, i} z_{\zeta \rightarrow i}^t, \hat{\tau}^t \right).$$

近似メッセージ伝搬法 (AMP) : ℓ_1 最適化

メッセージが以下のように近似されるとする

$$x_{j \rightarrow \theta}^t = x_j^t + \partial x_{j \rightarrow \theta}^t + \mathcal{O}\left(\frac{1}{N}\right)$$

ただし,

$$\partial x_{j \rightarrow \theta}^t, \partial z_{\theta \rightarrow j}^t = \mathcal{O}(1/\sqrt{N})$$

$$z_{\theta \rightarrow j}^t = z_{\theta}^t + \partial z_{\theta \rightarrow j}^t + \mathcal{O}\left(\frac{1}{N}\right)$$

このとき, 次が導かれる

$$x_j^{t+1} = \eta \left(\sum_{\zeta=1}^M a_{j,\zeta} z_{\zeta}^t + x_j^t, \hat{\tau}^t \right) + o_N(1)$$

$N \rightarrow \infty$ で, $\rightarrow 0$ となる項

$$z_{\theta}^t = y_{\theta} - \sum_{i=1}^N a_{\theta,i} x_i^t + \frac{1}{M} z_{\theta}^{t-1} \sum_{k=1}^N \eta' \left(\sum_{\zeta} a_{\zeta,k} z_{\zeta}^{t-1} + x_k^{t-1}, \hat{\tau}^{t-1} \right) + o_N(1)$$

近似メッセージ伝搬法 (AMP) : ℓ_1 最適化

↓ ベクトル表記, $o_N(1)$ を無視

$$\mathbf{x}^{t+1} = \eta(\mathbf{A}^T \mathbf{z}^t + \mathbf{x}^t, \hat{\tau}^t)$$

$$\mathbf{z}^t = \mathbf{y} - \mathbf{A}\mathbf{x}^t + \frac{1}{\Delta} \mathbf{z}^{t-1} \langle \eta'(\mathbf{A}^T \mathbf{z}^{t-1} + \mathbf{x}^{t-1}, \hat{\tau}^{t-1}) \rangle$$

同様に

$$\hat{\tau}^t = \frac{\hat{\tau}^{t-1}}{\Delta} \langle \eta'(\mathbf{A}^T \mathbf{z}^{t-1} + \mathbf{x}^{t-1}, \hat{\tau}^{t-1}) \rangle$$

$\langle \cdot \rangle$: ベクトルの平均

$\hat{\tau}$ をパラメータとみなす場合 :

$$\mathbf{x}^{t+1} = \eta(\mathbf{A}^T \mathbf{z}^t + \mathbf{x}^t, \lambda \sigma_t)$$

$$\mathbf{z}^t = \mathbf{y} - \mathbf{A}\mathbf{x}^t + \frac{1}{\Delta} \mathbf{z}^{t-1} \langle \eta'(\mathbf{A}^T \mathbf{z}^{t-1} + \mathbf{x}^{t-1}, \lambda \sigma_{t-1}) \rangle$$

ただし, $\lambda > 0$, $\sigma_t^2 := \|\mathbf{x}^t - \mathbf{b}\|_2^2 / N$

まとめ

- ・ 必要な基礎事項の復習
 - ・ 確率変数と確率の基本法則
 - ・ ベイズ則
 - ・ 条件付き独立とグラフィカルモデル
 - ・ サンプルング法
- ・ ベイズ則に基づく確率推論
- ・ マルコフ性を用いた状態空間モデルの導出と、予測とフィルタリングによる逐次的な状態推定および平滑化による周辺事後分布の導出
- ・ 確率伝搬法としてファクターグラフ上の和-積アルゴリズムとPearlのBPアルゴリズムを説明
- ・ 確率伝搬法の応用例として、LDPC符号、ターボ符号の復号アルゴリズム、FFT、近似メッセージ伝搬法を紹介