

# 第13章 系列データ

*Pattern  
Recognition  
and  
Machine  
Learning*

2015/9/20

夏合宿PRML輪読ゼミ

B4 三木 真理子

**BehaviorInNetworks**

都市生活学・ネットワーク行動学研究室

- 1. 系列データと状態空間モデル
- 2. 隠れマルコフモデル
  - 2.1 定式化とその性質
  - 2.2 最尤推定法
  - 2.3 潜在変数の系列を知るには...
- 3. 線形動的システム

この章の目標：

系列データを扱う際に有効な状態空間モデルのうち、代表的な2例である隠れマルコフモデルと線形動的システムの性質を知り、パラメータ推定、推論を行えるようになる！

・・・線形動的システム追いつけませんでした、すみません...

# 1.1. 系列データとは

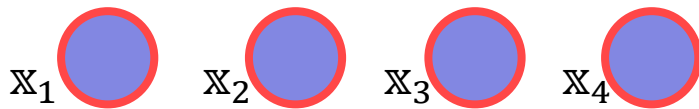
- 系列データ(sequential data)って・・・？  
同質のデータを直列に並べたデータ（朱鷺の杜Wiki参照）  
Ex. 文字列、DNA系列、時系列の測定を通して得られる様々なデータ
- 系列データの特徴  
データ点とデータ点が互いに相関する  
(Ex. 今日雨が降る事象と明日雨が降る事象は独立ではない)  
⇒各々のデータ点が独立同分布に従うという仮定が使えない！
- データ点の相関構造  
(直感的には)時間的に近い観測値同士ほど、強い相関があるだろう  
過去の全ての観測値に依存しているというのは現実的ではない  
⇒未来の予測値が直近の観測値以外の過去の観測値に対して独立だと仮定した**マルコフモデル**について考える

# 1.2. マルコフモデル

ここからは、確率的グラフィカルモデル(8章)の枠組みを用いながら考察していく  
まず、確率の積の規則を用いて、系列データの同時確率は以下のようにかける

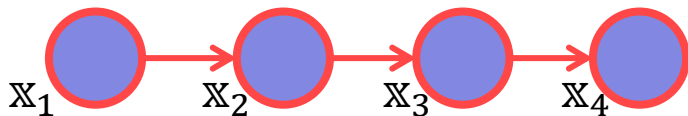
$$p(\mathbb{x}_1, \dots, \mathbb{x}_N) = p(\mathbb{x}_1) \prod_{n=2}^N p(\mathbb{x}_n | \mathbb{x}_1, \dots, \mathbb{x}_{n-1})$$

## ■ 独立同分布として扱う



$$p(\mathbb{x}_1, \dots, \mathbb{x}_N) = \prod_{n=1}^N p(\mathbb{x}_n)$$

## ■ 一次マルコフモデルで考える



$$p(\mathbb{x}_1, \dots, \mathbb{x}_N) = p(\mathbb{x}_1) \prod_{n=2}^N p(\mathbb{x}_n | \mathbb{x}_{n-1})$$

※ $p(\mathbb{x}_n | \mathbb{x}_{n-1})$ がnに限らず一定 (定常時系列の仮定)  $\Rightarrow$  均一マルコフ連鎖

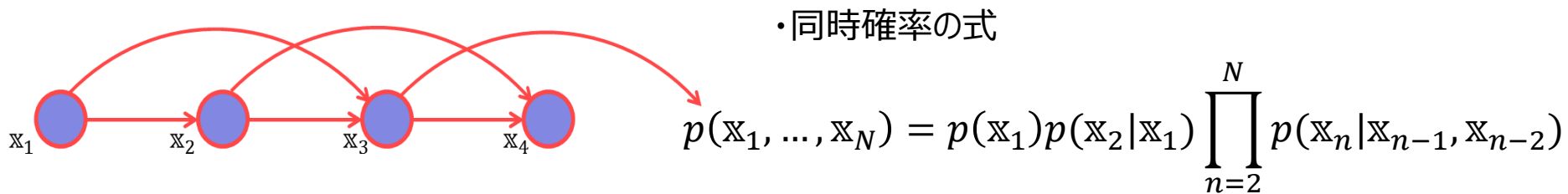
# 1.2. マルコフモデル

一次の均一マルコフ連鎖は、まだかなり制限の強いモデル

⇒データがより過去のデータからも影響を受けるようにモデルを変更

⇒高次マルコフ連鎖

## ■ 二次マルコフ連鎖



まったく同様にM次マルコフ連鎖に拡張することができる。しかし・・・

・仮に $\mathbb{x}_k$ をK個の状態をもつ離散変数とする

⇒  $p(\mathbb{x}_k|\mathbb{x}_{k-1})$ はのパラメータ数は $K(K-1)$ 個！

※ $\mathbb{x}_{k-1}$ のK個の状態の各々に対して $(K-1)$ 個のパラメータをもつ

・このとき、M次マルコフ連鎖によって表されるモデルのパラメータは $K^M(K-1)$ 個！

⇒Mに対して指数的に増大するので、Mが大きいと実用的ではない

# 1.3. 状態空間モデル

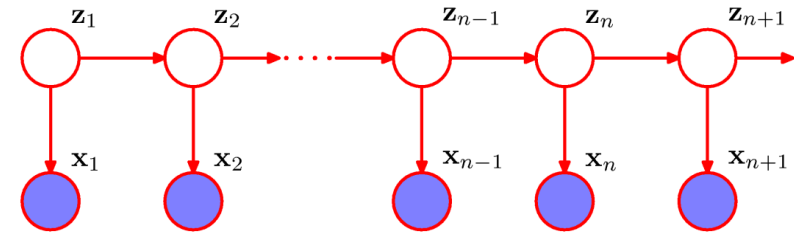
- ① 次数のあるマルコフ性の仮定(M次以前のデータからは独立)に制限されない
- ② 自由パラメータの数がある程度制限される

この2つの性質を満たすモデルを作りたい！

⇒ **状態空間モデル(state space model)**

○状態空間モデル○

- ・観測値 $x_n$ に対し、対応する潜在変数 $z_n$ を導入
  - ※  $x_n$ と $z_n$ の型・次元は異なっていても構わない
- ・潜在変数でマルコフ連鎖を構成する



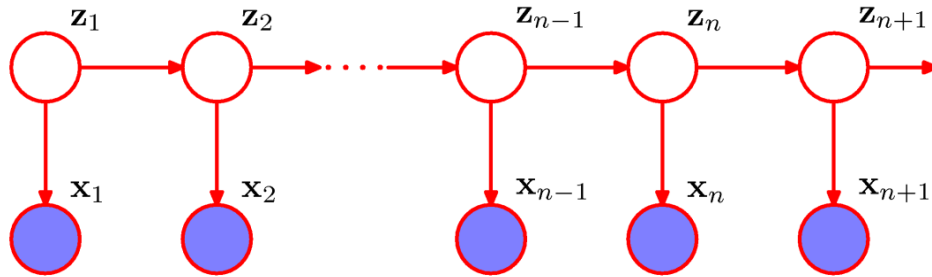
グラフィカルモデルの形から、  
 $z_n$ を観測したとき $z_{n-1}$ と $z_{n+1}$ は独立

$$z_{n+1} \perp\!\!\!\perp z_{n-1} \mid z_n.$$

同時分布を表す式

$$p(x_1, \dots, x_N, z_1, \dots, z_N) = p(z_1) \left[ \prod_{n=2}^N p(z_n | z_{n-1}) \right] \prod_{n=1}^N p(x_n | z_n)$$

# 1.3. 状態空間モデル



有向分離の基準を用いると、

2つの観測変数 $x_n$ と $x_m$ をつなぐ経路は常に存在するが、経由するのは潜在変数のみであり、決して遮断されない。

よって、過去の観測値すべてをあたえたときの観測 $x_{n+1}$ の予測分布  
 $p(x_{n+1} | x_1, \dots, x_n)$ は条件付き独立の性質を何も持たない

⇒  $x_{n+1}$ の予測はすべての過去の観測値に依存するが、どの次数のマルコフ性も満たさない（次数をもつマルコフ性の仮定に制限されない）

潜在変数が離散変数であるような状態空間モデル⇒隠れマルコフモデル(HMM)

潜在変数と観測変数の両方がガウス分布に従う場合⇒線形動的システム

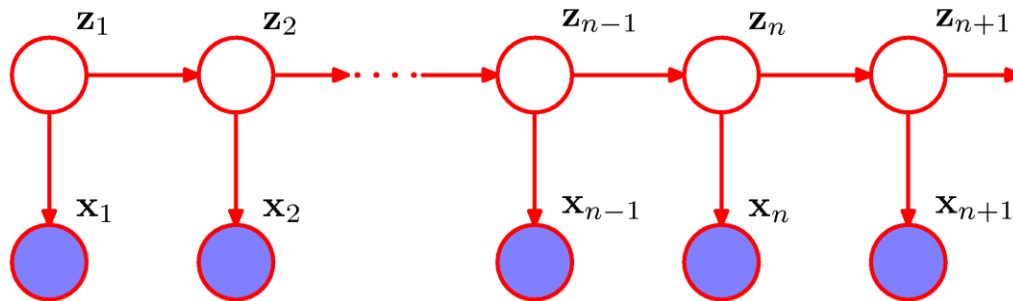
## 2.1 隠れマルコフモデルの記述

隠れマルコフモデル(HMM; hidden Markov model)は、

潜在変数が離散変数である状態空間モデル

※観測変数は離散変数でも連続変数でもよい

- ある1つの時刻に注目すると、  
成分密度 $p(\mathbf{x}_n|z_n)$ で与えられる混合分布に対応する  
⇒潜在変数は離散的な多項変数 $z_n$ であり、  
どの混合成分が対応する観測 $\mathbf{x}_n$ を生成するかを記述する(目標①)





# 2.1 隠れマルコフモデルの記述

目標①: 離散的な多項変数 $z_n$ のどの混合成分が対応する観測 $x_n$ を生成するかを記述する

一対K符号化法を用いる

$z_n$ の確率分布は、 $p(z_n|z_{n-1})$ を通して $z_{n-1}$ の状態に依存する。 $z_n$ はK次元の二値変数なので、 $p(z_n|z_{n-1})$ は遷移確率を要素に持つ数表 $A$ に対応する。

遷移確率は  $A_{jk} \equiv p(z_{nk} = 1 | z_{n-1,j} = 1)$  で定義される。

確率なので、 $0 \leq A_{jk} \leq 1$ と  $\sum_k A_{jk} = 1$  を満たす。

したがって行列 $A$ は $K(K-1)$ 個の独立したパラメータをもつ。

このとき、条件付き分布は、以下のようにかける。

$$p(z_n | z_{n-1}, A) = \prod_{k=1}^K \prod_{j=1}^K A_{jk}^{z_{n-1,j} z_{nk}}$$

最初の潜在ノード $z_1$ は、要素  $\pi_k \equiv p(z_{1k} = 1)$  をもつ確率のベクトル $\pi$ で表される周辺分布 $p(z_1)$ をもつ。

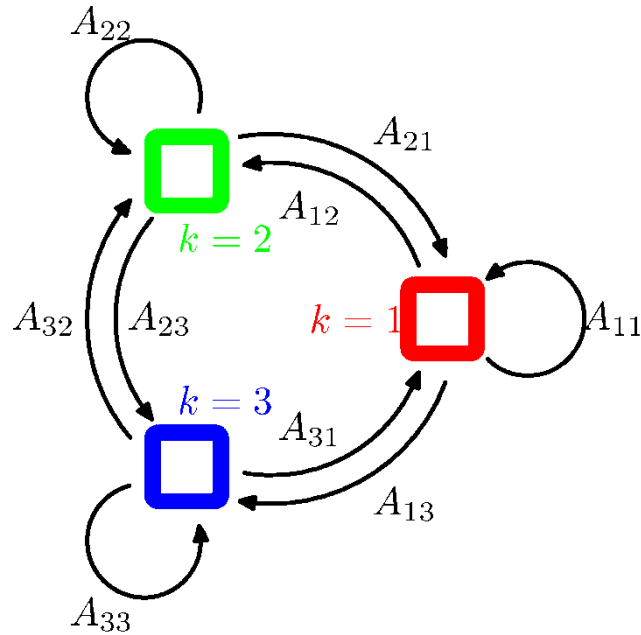
$$p(z_1 | \pi) = \prod_{k=1}^K \pi_k^{z_{1k}}$$

ただし、 $\sum_k \pi_k = 1$  である

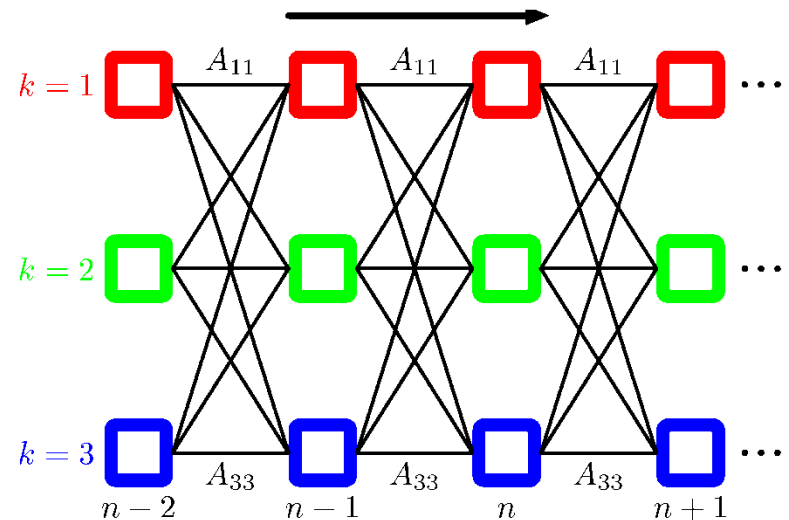
# 2.1 隠れマルコフモデルの記述

## ■ 遷移行列の表し方いろいろ

### ① 状態遷移図



### ② 格子図 (トレリス図)



$$p(\mathbf{z}_n | \mathbf{z}_{n-1}, \mathbf{A}) = \prod_{k=1}^K \prod_{j=1}^K A_{jk}^{z_{n-1,j} z_{n,k}}$$

# 2.1 隠れマルコフモデルの記述

目標②：観測変数の条件付き確率分布  $p(\mathbf{x}_n | \mathbf{z}_n, \phi)$  を定義する

※  $\phi$  は分布を支配するパラメータの集合

$p(\mathbf{x}_n | \mathbf{z}_n, \phi)$ ：出力確率(emission diagram)

$\mathbf{x}_n$  は観測されるので、 $\phi$  の値が与えられたとき、分布  $p(\mathbf{x}_n | \mathbf{z}_n, \phi)$  は二値ベクトル  $\mathbf{z}_n$  の  $K$  個の可能な状態に対応した  $K$  個の要素をもつベクトルからなり、出力確率は以下の形式で表現される。

$$p(\mathbf{x}_n | \mathbf{z}_n, \phi) = \prod_{k=1}^K p(\mathbf{x}_n | \phi_k)^{z_{nk}}$$

モデルが均一(homogeneous)であるとき、つまり、 $\mathbf{A}$  と  $\phi$  が全時刻を通じて一定のとき、潜在変数と観測変数の同時確率分布は以下のようなになる。

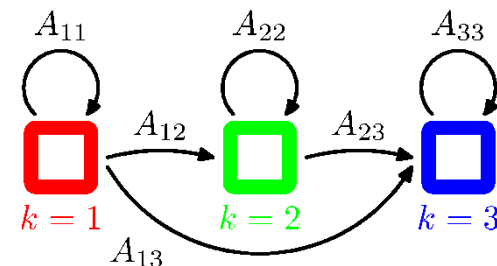
$$p(\mathbf{X}, \mathbf{Z} | \theta) = p(\mathbf{z}_1 | \pi) \left[ \prod_{n=2}^N p(\mathbf{z}_n | \mathbf{z}_{n-1}, \mathbf{A}) \right] \prod_{m=1}^N p(\mathbf{x}_m | \mathbf{z}_m, \phi). \quad (13.10)$$

## ■ この間の話題

- 生成モデルの観点から見た理解
- 遷移行列に制限を加えて変異形を作れる

Ex. Left to right HMM

これを用いた数字の識別を通して、HMMには時間軸上での局所的な伸縮にたいしてある程度まで普遍性を保つ能力があることを見る



## 2.2 最尤推定法

データ集合  $\mathbb{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  が観測された時、HMMのパラメータを最尤推定で決定することができる。

尤度関数は、同時分布の式(13.10)を潜在変数について周辺化して得られるので、

・同時分布の式

$$p(\mathbf{X}, \mathbf{Z} | \theta) = p(\mathbf{z}_1 | \pi) \left[ \prod_{n=2}^N p(\mathbf{z}_n | \mathbf{z}_{n-1}, \mathbf{A}) \right] \prod_{m=1}^N p(\mathbf{x}_m | \mathbf{z}_m, \phi). \quad (13.10)$$

・尤度関数

$$p(\mathbf{X} | \theta) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} | \theta). \quad (13.11)$$

これは潜在変数  $\mathbf{z}_n$  についての和演算(格子図にある全経路について和を取ることに相当)

→ 変数が  $N$  個、それぞれが  $K$  個の状態をもつため、和を取る対象は全部で  $K^N$  項!

⇒ EMアルゴリズムを使って効率的に尤度関数を最大化しよう!

①モデルパラメータ $\theta$ をある初期集合 $\theta^{old}$ に限定する

②Eステップ

2-1.  $\theta^{old}$ の値を用いて、潜在変数 $p(\mathbf{Z}|\mathbf{X}, \theta^{old})$ の事後分布を求める

2-2. 求めた事後分布を用いて、パラメータ集合 $\theta$ の関数として、完全データに対する尤度関数の対数の期待値 $Q(\theta, \theta^{old})$ を求める

$$Q(\theta, \theta^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{old}) \ln p(\mathbf{X}, \mathbf{Z}|\theta). \quad (13.12)$$

これを求めるために以下を定義し、これらを効率的に求めることを考える

$$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{X}, \theta^{old}) \quad (13.13)$$

$$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X}, \theta^{old}). \quad (13.14)$$

③Mステップ

$\gamma(\mathbf{z}_n), \xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ を定数と見なし、パラメータ $\theta = \{\pi, A, \phi\}$ に関して $Q(\theta, \theta^{old})$ を最大化し、 $\theta^{new}$ へ更新

## 2. Eステップ

Eステップを説明する流れ

- ①先ほど定義した $\gamma(\mathbf{z}_n)$ ,  $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ を使って目的関数 $Q(\theta, \theta^{old})$ を表す
- ②フォワード・バックワードアルゴリズム（特に $\alpha - \beta$ アルゴリズム）により、 $\gamma(\mathbf{z}_n)$ ,  $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ を効率的に求められることを確認する

ということで、

- ① $\gamma(\mathbf{z}_n)$ ,  $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ を使って目的関数 $Q(\theta, \theta^{old})$ を表す

$$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n | \mathbf{X}, \theta^{old}) \quad (13.13)$$

$$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_{n-1}, \mathbf{z}_n | \mathbf{X}, \theta^{old}). \quad (13.14)$$

$\gamma(\mathbf{z}_n)$  は潜在変数 $\mathbf{z}_n$ の周辺事後分布であり、

$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ は2つの連続した潜在変数に対する同時周辺事後分布

$\gamma(\mathbf{z}_n)$  は潜在変数  $\mathbf{z}_n$  の周辺事後分布なので、各々の  $n$  の値について、 $\gamma(\mathbf{z}_n)$  は和が 1 となる  $K$  個の非負の数の集合を用いて記憶できる

$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$  は 2 つの連続した潜在変数に対する同時周辺事後分布なので、和が 1 となるような非負の数を成分とする  $K \times K$  行列を用いて記憶することができる

ここで、

$\gamma(z_{nk}) : z_{nk} = 1$  の条件付き確率

$\xi(z_{n-1,j}, z_{nk}) : z_{n-1,j} = 1$  かつ  $z_{nk} = 1$  の条件付き確率

とする。

$z_{nk}$  は二値の確率変数なので、それらの期待値は単にそれが値 1 をもつ確率であり、

$$\gamma(z_{nk}) = \mathbb{E}[z_{nk}] = \sum_{\mathbf{z}_n} \gamma(\mathbf{z}_n) z_{nk} \quad (13.15)$$

$$\xi(z_{n-1,j}, z_{nk}) = \mathbb{E}[z_{n-1,j} z_{nk}] = \sum_{\mathbf{z}_{n-1}, \mathbf{z}_n} \xi(\mathbf{z}_{n-1}, \mathbf{z}_n) z_{n-1,j} z_{nk}. \quad (13.16)$$



$$p(\mathbf{X}, \mathbf{Z}|\theta) = p(\mathbf{z}_1|\pi) \left[ \prod_{n=2}^N p(\mathbf{z}_n|\mathbf{z}_{n-1}, \mathbf{A}) \right] \prod_{m=1}^N p(\mathbf{x}_m|\mathbf{z}_m, \phi). \quad (13.10)$$

$$Q(\theta, \theta^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\theta). \quad (13.12)$$

$$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{X}, \theta^{\text{old}}) \quad (13.13)$$

$$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_{n-1}, \mathbf{z}_n|\mathbf{X}, \theta^{\text{old}}). \quad (13.14)$$

$$\gamma(z_{nk}) = \mathbb{E}[z_{nk}] = \sum_{\mathbf{z}_n} \gamma(\mathbf{z}_n) z_{nk} \quad (13.15)$$

$$\xi(z_{n-1,j}, z_{nk}) = \mathbb{E}[z_{n-1,j} z_{nk}] = \sum_{\mathbf{z}_{n-1}, \mathbf{z}_n} \xi(\mathbf{z}_{n-1}, \mathbf{z}_n) z_{n-1,j} z_{nk}. \quad (13.16)$$

これらの式を使って(13.12)を整理すると、次式が得られる

$$\begin{aligned} Q(\theta, \theta^{\text{old}}) = & \sum_{k=1}^K \gamma(z_{1k}) \ln \pi_k + \sum_{n=2}^N \sum_{j=1}^K \sum_{k=1}^K \xi(z_{n-1,j}, z_{nk}) \ln A_{jk} \\ & + \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk}) \ln p(\mathbf{x}_n|\phi_k). \end{aligned} \quad (13.17)$$

Eステップ第1段階クリア！

## 2. Eステップ

次の目標は、

潜在変数の周辺事後確率 $\gamma(z_n), \xi(z_{n-1}, z_n)$ の値を効率的に求めること

⇒フォワード・バックワードアルゴリズム（特に $\alpha - \beta$ アルゴリズム）を使う！

注意：出力密度 $p(x|z)$ の関数形は何でも良い。

必要なのは各 $n$ に対する $p(x_n|z_n)$ の「値」のみ

本節ではモデルパラメータ $\theta^{old}$ は一定なので、

このパラメータへの依存性は明示しない

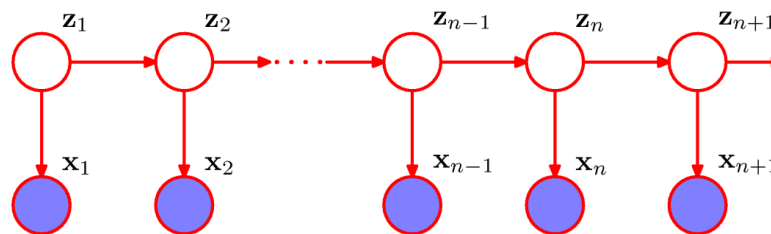
ステップ①:グラフィカルモデルの有向分離を用いて条件付き独立性を書き下す

ステップ②:新たに定義する条件付き同時確率 $\alpha(z_n), \beta(z_n)$ を再帰的に求める

ステップ③:条件付き独立性と $\alpha(z_n), \beta(z_n)$ を用いて $\gamma(z_n), \xi(z_{n-1}, z_n)$ を求める

## 2. Eステップ —条件付き独立性の書き下し—

- 有向分離の考え方より、  
以下の式がかける



$$p(\mathbf{X}|\mathbf{z}_n) = p(\mathbf{x}_1, \dots, \mathbf{x}_n | \mathbf{z}_n) \times p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_n) \quad (13.24)$$

$$p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | \mathbf{x}_n, \mathbf{z}_n) = p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | \mathbf{z}_n) \quad (13.25)$$

$$p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | \mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | \mathbf{z}_{n-1}) \quad (13.26)$$

$$p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_n, \mathbf{z}_{n+1}) = p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_{n+1}) \quad (13.27)$$

$$p(\mathbf{x}_{n+2}, \dots, \mathbf{x}_N | \mathbf{z}_{n+1}, \mathbf{x}_{n+1}) = p(\mathbf{x}_{n+2}, \dots, \mathbf{x}_N | \mathbf{z}_{n+1}) \quad (13.28)$$

$$p(\mathbf{X} | \mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | \mathbf{z}_{n-1}) \times p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_n) \quad (13.29)$$

$$p(\mathbf{x}_{N+1} | \mathbf{X}, \mathbf{z}_{N+1}) = p(\mathbf{x}_{N+1} | \mathbf{z}_{N+1}) \quad (13.30)$$

$$p(\mathbf{z}_{N+1} | \mathbf{z}_N, \mathbf{X}) = p(\mathbf{z}_{N+1} | \mathbf{z}_N). \quad (13.31)$$

# Eステップ — $\alpha(\mathbf{z}_n), \beta(\mathbf{z}_n)$ の定義 —

求めたいものを  
確認！

$$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n | \mathbf{X}, \theta^{\text{old}}) \quad (13.13)$$

$$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_{n-1}, \mathbf{z}_n | \mathbf{X}, \theta^{\text{old}}). \quad (13.14)$$

$\gamma(\mathbf{z}_{nk})$  :  $z_{nk} = 1$  の条件付き確率

$\xi(\mathbf{z}_{n-1,j}, \mathbf{z}_{nk})$  :  $z_{n-1,j} = 1$  かつ  $z_{nk} = 1$  の条件付き確率

ベイズの定理より

$$\gamma(\mathbf{z}_n) = p(\mathbf{z}_n | \mathbf{X}) = \frac{p(\mathbf{X} | \mathbf{z}_n) p(\mathbf{z}_n)}{p(\mathbf{X})} \quad (13.32)$$

ここで  $p(\mathbf{X})$  はパラメータ  $\theta^{\text{old}}$  に依存することから、尤度関数である  
(13.24) 式を  $p(\mathbf{X} | \mathbf{z}_n)$  に代入し、確率の乗法定理を用いることで、

$$\gamma(\mathbf{z}_n) = \frac{p(\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{z}_n) p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_n)}{p(\mathbf{X})} = \frac{\alpha(\mathbf{z}_n) \beta(\mathbf{z}_n)}{p(\mathbf{X})} \quad (13.33)$$

ただし、

$$\alpha(\mathbf{z}_n) \equiv p(\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{z}_n) \quad (13.34)$$

$$\beta(\mathbf{z}_n) \equiv p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_n) \quad (13.35)$$

# Eステップ — $\alpha(\mathbf{z}_n)$ を求める —

$\alpha(\mathbf{z}_n)$ は時刻nまでの全ての観測データと $\mathbf{z}_n$ の値の同時確率。  
条件付き独立性・加法定理・乗法定理より再帰的に定義できる

$$\begin{aligned}
 \alpha(\mathbf{z}_n) &= p(\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{z}_n) \\
 &= p(\mathbf{x}_1, \dots, \mathbf{x}_n | \mathbf{z}_n) p(\mathbf{z}_n) \\
 &= p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | \mathbf{z}_n) p(\mathbf{z}_n) \\
 &= p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, \mathbf{z}_n) \\
 &= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1}, \mathbf{z}_n) \\
 &= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, \mathbf{z}_n | \mathbf{z}_{n-1}) p(\mathbf{z}_{n-1}) \\
 &= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | \mathbf{z}_{n-1}) p(\mathbf{z}_n | \mathbf{z}_{n-1}) p(\mathbf{z}_{n-1}) \\
 &= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, \mathbf{z}_{n-1}) p(\mathbf{z}_n | \mathbf{z}_{n-1}). \\
 \alpha(\mathbf{z}_n) &= p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1}) p(\mathbf{z}_n | \mathbf{z}_{n-1}) \tag{13.36}
 \end{aligned}$$

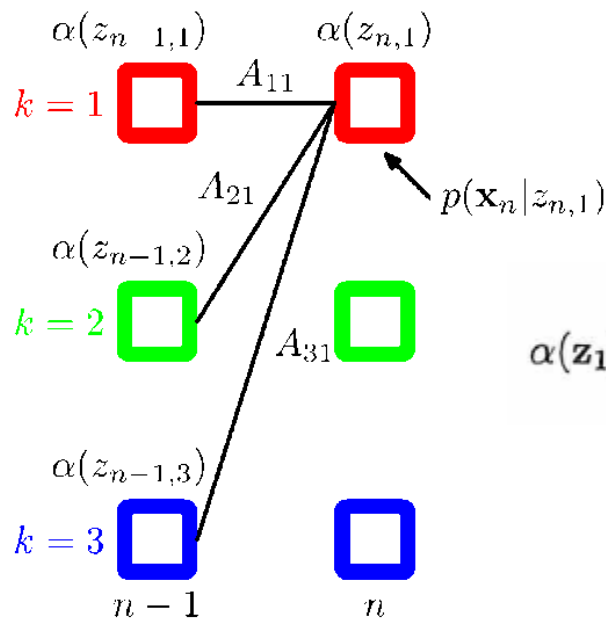
$$\begin{aligned}
 p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | \mathbf{x}_n, \mathbf{z}_n) &= p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | \mathbf{z}_n) \\
 p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | \mathbf{z}_{n-1}, \mathbf{z}_n) &= p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | \mathbf{z}_{n-1})
 \end{aligned}$$

# Eステップ — $\alpha(\mathbf{z}_n)$ を求める —

$z_{nk} = 1$ に対応する $\alpha(\mathbf{z}_n)$ の値を $\alpha(z_{nk})$ と表す.

$$\alpha(\mathbf{z}_n) = p(\mathbf{x}_n | \mathbf{z}_n) \sum_{\mathbf{z}_{n-1}} \alpha(\mathbf{z}_{n-1}) p(\mathbf{z}_n | \mathbf{z}_{n-1}) \quad (13.36)$$

和を取る項：右辺の $\mathbf{z}_n$ の $K$ 個の値それぞれについて求めた $\mathbf{z}_{n-1}$ の $K$ 個の状態数



↓  
α再帰の各ステップは $O(K^2)$ オーダーの計算量

再帰の開始に必要な初期値は以下で与える.

$$\alpha(\mathbf{z}_1) = p(\mathbf{x}_1, \mathbf{z}_1) = p(\mathbf{z}_1) p(\mathbf{x}_1 | \mathbf{z}_1) = \prod_{k=1}^K \{\pi_k p(\mathbf{x}_1 | \phi_k)\}^{z_{1k}}. \quad (13.37)$$

鎖の長さを $N$ とすると、α再帰により鎖全体の値を求めるコストは  
 $O(NK^2)$

# Eステップ — $\beta(\mathbf{z}_n)$ を求める —

同様に $\beta(\mathbf{z}_n)$ に関する再帰式を求める.

$\beta(\mathbf{z}_n)$ は $\mathbf{z}_n$ の値を与えたときの、時刻 $n+1$ から $N$ までの全てのデータの条件付き確率・

$$\begin{aligned}\beta(\mathbf{z}_n) &= p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_n) \\ &= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N, \mathbf{z}_{n+1} | \mathbf{z}_n) \\ &= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_n, \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n) \\ &= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n) \\ &= \sum_{\mathbf{z}_{n+1}} p(\mathbf{x}_{n+2}, \dots, \mathbf{x}_N | \mathbf{z}_{n+1}) p(\mathbf{x}_{n+1} | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n).\end{aligned}$$

$$\begin{aligned}p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_n, \mathbf{z}_{n+1}) &= p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_{n+1}) \\ p(\mathbf{x}_{n+2}, \dots, \mathbf{x}_N | \mathbf{z}_{n+1}, \mathbf{x}_{n+1}) &= p(\mathbf{x}_{n+2}, \dots, \mathbf{x}_N | \mathbf{z}_{n+1})\end{aligned}$$

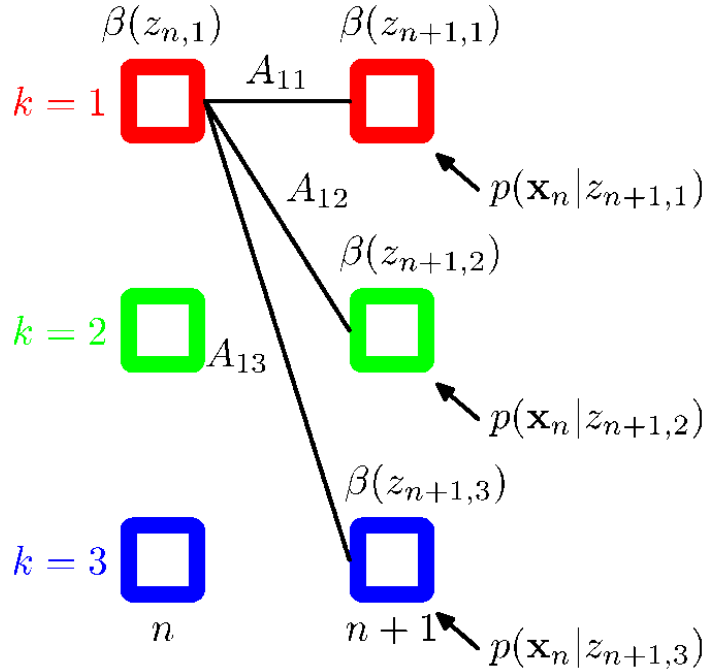
$$\beta(\mathbf{z}_n) = \sum_{\mathbf{z}_{n+1}} \beta(\mathbf{z}_{n+1}) p(\mathbf{x}_{n+1} | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n). \quad (13.38)$$

$$p(\mathbf{z}_N | \mathbf{X}) = \frac{p(\mathbf{X}, \mathbf{z}_N) \beta(\mathbf{z}_N)}{p(\mathbf{X})}. \quad (13.39)$$

# Eステップ — $\beta(\mathbf{z}_n)$ を求める —

$z_{nk} = 1$ に対応する $\beta(\mathbf{z}_n)$ の値を $\beta(z_{nk})$ と表す

$$\beta(\mathbf{z}_n) = \sum_{\mathbf{z}_{n+1}} \beta(\mathbf{z}_{n+1}) p(\mathbf{x}_{n+1} | \mathbf{z}_{n+1}) p(\mathbf{z}_{n+1} | \mathbf{z}_n). \quad (13.38)$$



初期値は、

$$\gamma(\mathbf{z}_n) = \frac{p(\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{z}_n) p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_n)}{p(\mathbf{X})} = \frac{\alpha(\mathbf{z}_n) \beta(\mathbf{z}_n)}{p(\mathbf{X})} \quad (13.33)$$

(13.33)式について $n=N$ とし、 $\alpha(\mathbf{z}_n)$ の定義により得られる以下の式から得られる。

$$p(\mathbf{z}_N | \mathbf{X}) = \frac{p(\mathbf{X}, \mathbf{z}_N) \beta(\mathbf{z}_N)}{p(\mathbf{X})}. \quad (13.39)$$

$\mathbf{z}_n$ の全ての値について、 $\beta(\mathbf{z}_n) = 1$ とすると、この式は正しい式となる。



# Eステップ — $\gamma(\mathbf{z}_n), \xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ を求める—

25

最後に $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ を求める.

$\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) = (\mathbf{z}_{n-1}, \mathbf{z}_n)$ の $K \times K$ 個の値の組各々についての条件付き確率 $p(\mathbf{z}_{n-1}, \mathbf{z}_n | \mathbf{X})$

ベイズの定理より、以下のように展開できる.

$$\begin{aligned}\xi(\mathbf{z}_{n-1}, \mathbf{z}_n) &= p(\mathbf{z}_{n-1}, \mathbf{z}_n | \mathbf{X}) \\ &= \frac{p(\mathbf{X} | \mathbf{z}_{n-1}, \mathbf{z}_n) p(\mathbf{z}_{n-1}, \mathbf{z}_n)}{p(\mathbf{X})} \\ &= \frac{p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | \mathbf{z}_{n-1}) p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_n) p(\mathbf{z}_n | \mathbf{z}_{n-1}) p(\mathbf{z}_{n-1})}{p(\mathbf{X})} \\ &= \frac{\alpha(\mathbf{z}_{n-1}) p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{z}_n | \mathbf{z}_{n-1}) \beta(\mathbf{z}_n)}{p(\mathbf{X})}.\end{aligned}\tag{13.43}$$

→再帰の結果を使って $\xi(\mathbf{z}_{n-1}, \mathbf{z}_n)$ を計算できる！

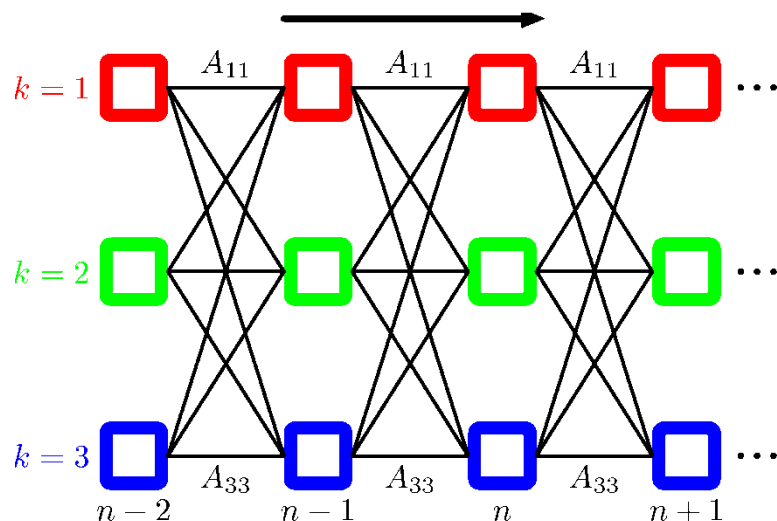
$$\gamma(\mathbf{z}_n) = \frac{p(\mathbf{x}_1, \dots, \mathbf{x}_n, \mathbf{z}_n) p(\mathbf{x}_{n+1}, \dots, \mathbf{x}_N | \mathbf{z}_n)}{p(\mathbf{X})} = \frac{\alpha(\mathbf{z}_n) \beta(\mathbf{z}_n)}{p(\mathbf{X})}\tag{13.33}$$

→Eステップの目標クリア！

- $\alpha$ 再帰を使えば、鎖の長さに対して線形の計算量で尤度を計算できる！

$$p(\mathbf{X}) = \sum_{\mathbf{z}_n} \alpha(\mathbf{z}_n) \beta(\mathbf{z}_n). \quad (13.41)$$

$$p(\mathbf{X}) = \sum_{\mathbf{z}_N} \alpha(\mathbf{z}_N). \quad (13.42)$$



## 2. Mステップ — $Q(\theta, \theta^{old})$ の最大化—

最大化の目的関数は、Eステップによって以下のように書ける

$$\begin{aligned} Q(\theta, \theta^{old}) = & \sum_{k=1}^K \gamma(z_{1k}) \ln \pi_k + \sum_{n=2}^N \sum_{j=1}^K \sum_{k=1}^K \xi(z_{n-1,j}, z_{nk}) \ln A_{jk} \\ & + \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk}) \ln p(\mathbf{x}_n | \phi_k). \end{aligned} \quad (13.17)$$

$\pi$ と $A$ に関してはラグランジュ乗数を使うことで簡単に求められ、

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^K \gamma(z_{1j})} \quad (13.18)$$

$$A_{jk} = \frac{\sum_{n=2}^N \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^K \sum_{n=2}^N \xi(z_{n-1,j}, z_{nl})}. \quad (13.19)$$

※初期値において0だった要素はずっと0のまま残ることに注意！

## 2. Mステップ<sup>o</sup> — $Q(\theta, \theta^{old})$ の最大化—

$\phi_k$ に関して最大化することを考える.

$$Q(\theta, \theta^{old}) = \sum_{k=1}^K \gamma(z_{1k}) \ln \pi_k + \sum_{n=2}^N \sum_{j=1}^K \sum_{k=1}^K \xi(z_{n-1,j}, z_{nk}) \ln A_{jk} \\ + \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk}) \ln p(\mathbf{x}_n | \phi_k). \quad (13.17)$$

最後の項のみ $\phi_k$ に依存している.

⇒  $\gamma(z_{nk})$ が負担率の役割を果たす混合分布に対するデータ依存項と同じ！

↓

パラメータ $\phi_k$ が成分ごとに独立な場合、

- ①  $k$ の各々の値ごとの項の和に分解し、それぞれを独立に最大化
- ② 重み $\gamma(z_{nk})$ で重みづけられた出力確率 $p(\mathbf{x} | \phi_k)$ の重み付き対数尤度関数を最大化

## 2. Mステップ — $Q(\theta, \theta^{old})$ の最大化—

出力確率がガウス密度分布

$$\mu_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^N \gamma(z_{nk})} \quad (13.20)$$

$$\Sigma_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \mu_k) (\mathbf{x}_n - \mu_k)^T}{\sum_{n=1}^N \gamma(z_{nk})}. \quad (13.21)$$

離散多項分布

$$p(\mathbf{x}|\mathbf{z}) = \prod_{i=1}^D \prod_{k=1}^K \mu_{ik}^{x_i z_k}. \quad (13.22)$$

$$\mu_{ik} = \frac{\sum_{n=1}^N \gamma(z_{nk}) x_{ni}}{\sum_{n=1}^N \gamma(z_{nk})}. \quad (13.23)$$

# EMアルゴリズムによる最尤推定のお話 30

- 観測データはEステップでの再帰式において、 $p(\mathbf{x}_n | \mathbf{z}_n)$ の形を通してしか影響を与えない
  - 観測変数の種類・次元・条件付き分布の形には影響されない
- 最尤推定法が有効なのは、データ点数がパラメータ数に比べて大きいとき
  - 学習系列が十分に長い場合に限り、最尤法による隠れマルコフモデルの学習が有効
    - ※ left-to-right アルゴリズムに注意

## ■ 予測分布

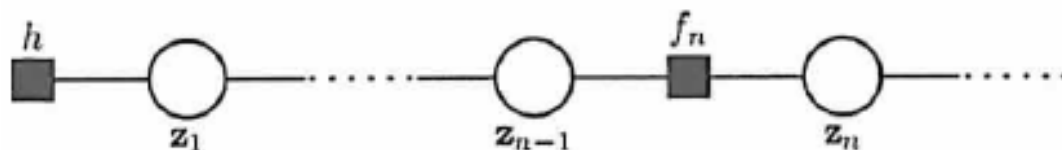
$$\begin{aligned} p(\mathbf{x}_{N+1}|\mathbf{X}) &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}, \mathbf{z}_{N+1}|\mathbf{X}) \\ &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1})p(\mathbf{z}_{N+1}|\mathbf{X}) \\ &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}, \mathbf{z}_N|\mathbf{X}) \\ &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}|\mathbf{z}_N)p(\mathbf{z}_N|\mathbf{X}) \\ &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}|\mathbf{z}_N) \frac{p(\mathbf{z}_N, \mathbf{X})}{p(\mathbf{X})} \\ &= \frac{1}{p(\mathbf{X})} \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}|\mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}|\mathbf{z}_N)\alpha(\mathbf{z}_N). \quad (13.44) \end{aligned}$$

$x_1$  から  $x_N$  までのデータ影響が  $\alpha(\mathbf{z}_N)$  の  $K$  個の値にまとめられている

⇒ 限られた記憶領域ですっと先まで予測分布の計算をし続けることができる

## 2.2 最尤推定法(積和アルゴリズムver) 32

- 積和アルゴリズムでもっと簡単に解けるのでその紹介  
出力確率を遷移確率因子に吸収することで、因子グラフを単純化



ここでの因子は以下で表せる

$$h(\mathbf{z}_1) = p(\mathbf{z}_1)p(\mathbf{x}_1|\mathbf{z}_1) \quad (13.45)$$

$$f_n(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_n|\mathbf{z}_{n-1})p(\mathbf{x}_n|\mathbf{z}_n). \quad (13.46)$$

最後の潜在変数 $\mathbf{z}_n$ を根ノードとし、葉ノード $h$ からメッセージを伝播

$$\mu_{\mathbf{z}_{n-1} \rightarrow f_n}(\mathbf{z}_{n-1}) = \mu_{f_{n-1} \rightarrow \mathbf{z}_{n-1}}(\mathbf{z}_{n-1}) \quad (13.47)$$

$$\mu_{f_n \rightarrow \mathbf{z}_n}(\mathbf{z}_n) = \sum f_n(\mathbf{z}_{n-1}, \mathbf{z}_n) \mu_{\mathbf{z}_{n-1} \rightarrow f_n}(\mathbf{z}_{n-1}). \quad (13.48)$$

変数ノードでは和を取ることもない。



## 2.2 最尤推定法(積和アルゴリズムver) 33

$$\mu_{\mathbf{z}_{n-1} \rightarrow f_n}(\mathbf{z}_{n-1}) = \mu_{f_{n-1} \rightarrow \mathbf{z}_{n-1}}(\mathbf{z}_{n-1}) \quad (13.47)$$

$$\mu_{f_n \rightarrow \mathbf{z}_n}(\mathbf{z}_n) = \sum f_n(\mathbf{z}_{n-1}, \mathbf{z}_n) \mu_{\mathbf{z}_{n-1} \rightarrow f_n}(\mathbf{z}_{n-1}). \quad (13.48)$$

上の二式から  $\mu_{\mathbf{z}_{n-1} \rightarrow f_n}(\mathbf{z}_{n-1})$  を消去すると、以下の再帰式が出てくる

$$\mu_{f_n \rightarrow \mathbf{z}_n}(\mathbf{z}_n) = \sum_{\mathbf{z}_{n-1}} f_n(\mathbf{z}_{n-1}, \mathbf{z}_n) \mu_{f_{n-1} \rightarrow \mathbf{z}_{n-1}}(\mathbf{z}_{n-1}). \quad (13.49)$$

ここで、 $\alpha$ を次のように定義する.

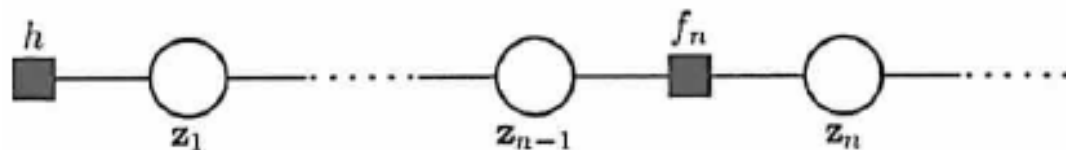
$$\alpha(\mathbf{z}_n) = \mu_{f_n \rightarrow \mathbf{z}_n}(\mathbf{z}_n)$$

因子の定義(13.46)と再帰式(13.49)より、  
先ほどの $\alpha$ 再帰と同一のものを得ることができる！

$$f_n(\mathbf{z}_{n-1}, \mathbf{z}_n) = p(\mathbf{z}_n | \mathbf{z}_{n-1}) p(\mathbf{x}_n | \mathbf{z}_n). \quad (13.46)$$

## 2.2 最尤推定法(積和アルゴリズムver) 34

根ノードから葉ノードへのメッセージの伝播( $\beta$ 再帰)を考える。



$\alpha$ 再帰と同様、演算を行わない変数ノードを消去して以下の再帰式を得る

$$\mu_{f_{n+1} \rightarrow f_n}(\mathbf{z}_n) = \sum_{\mathbf{z}_{n+1}} f_{n+1}(\mathbf{z}_n, \mathbf{z}_{n+1}) \mu_{f_{n+2} \rightarrow f_{n+1}}(\mathbf{z}_{n+1})$$

さらに以下のように $\beta(\mathbf{z}_n)$ を定義する。

$$\beta(\mathbf{z}_n) = \mu_{f_{n+1} \rightarrow \mathbf{z}_n}(\mathbf{z}_n)$$

因子の定義(13.46)と再帰式より、 $\beta$ 再帰と同一のものが得られる

このとき、同時確率は式(13.53)で表される。

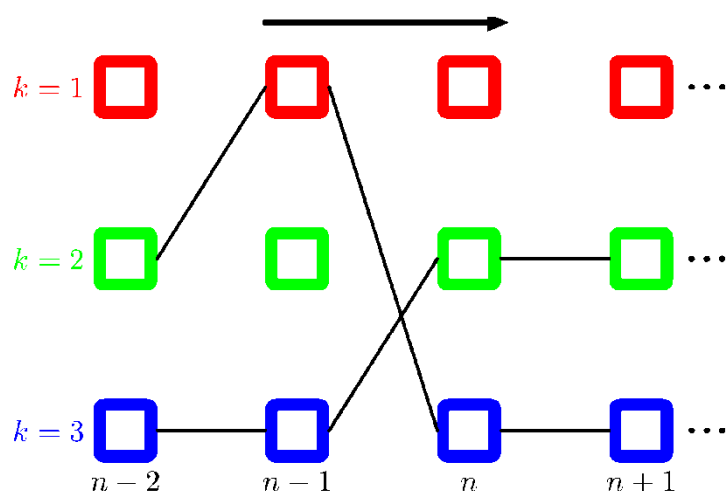
$$p(\mathbf{z}_n, \mathbf{X}) = \mu_{f_n \rightarrow \mathbf{z}_n}(\mathbf{z}_n) \mu_{f_{n+1} \rightarrow \mathbf{z}_n}(\mathbf{z}_n) = \alpha(\mathbf{z}_n) \beta(\mathbf{z}_n). \quad (13.53)$$

## 2.3 潜在変数の系列を知るには...

隠れマルコフモデルでは多くの場合、潜在変数に何らかの意味があると思われる。

⇒与えられた観測系列に対して、隠れ状態の最も確からしい系列を求めたい！

⇒max-sumアルゴリズム



直感的な説明 (Dialっぽい)

葉ノードからメッセージを終点 $z_n$ まで伝播

→もっとも確からしい経路をたどるには

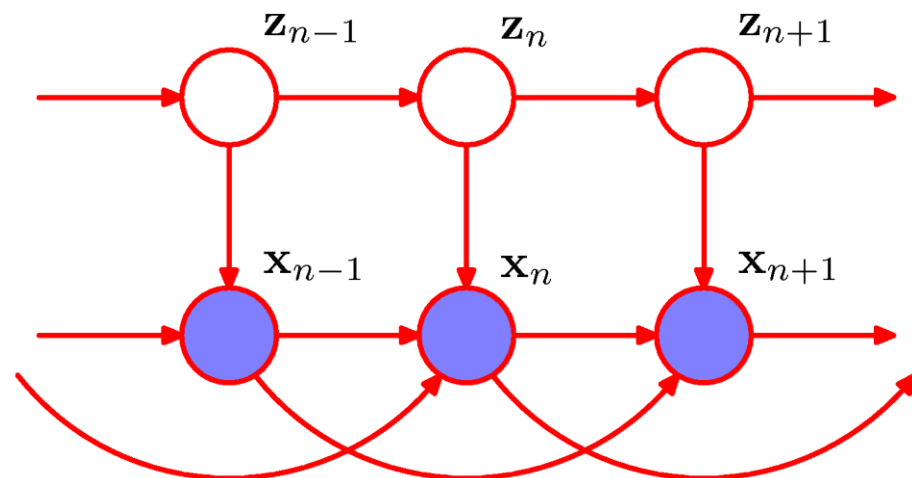
$z_{n-1}$ のどの状態数を通ればいいのかわかる

→  $z_{n-1}$ にとって、 $z_{n-2}$ のどの状態数から

の経路がもっともらしいかわかる

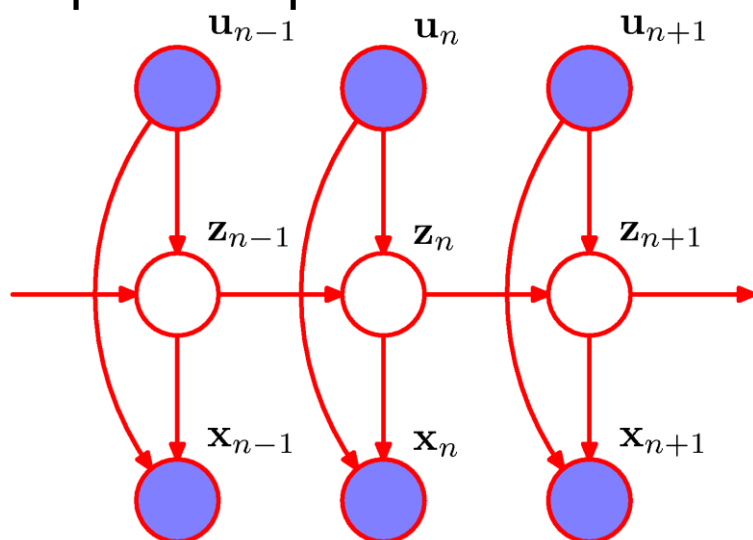
→以下繰り返す

## ■ 自己回帰隠れマルコフモデル



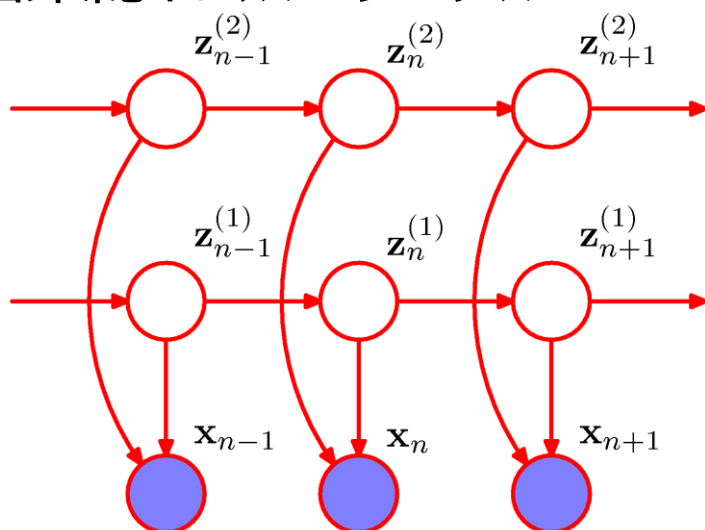
観測変数が2つ以前の観測変数と潜在変数に依存している  
⇒観測変数間のより長い期間の相関をとることができる

## ■ input-output隠れマルコフモデル



出力変数以外の観測変数系列が、潜在変数や出力変数に影響を与える  
潜在変数のマルコフ性に関して、条件付き独立性が成り立つため、  
計算量の面で効率的な学習アルゴリズムを組むことができる

## ■ 階乗隠れマルコフモデル



お互いに独立な、複数の潜在変数のマルコフ連鎖があり、与えられた時刻のステップの観測変数の分布が同じ時刻ステップにおけるすべての潜在変数の状態に依存するようなモデル

潜在状態をかなり少なく抑えることができるが、条件付き独立性が成り立たないため、学習が複雑になってしまう。

# 3. 線形動的システム

## ■ 線形動的システムとは？

- 状態空間モデルのうち、潜在変数と観測変数の両方が連続変数、特にガウス分布に従う、というモデル
- 隠れマルコフモデルとの重要なちがいは、潜在変数が離散か連続（ガウス分布）か、ということ。

## ■ 直感的な説明。

- 時間軸上で変化している未知の量  $\{z_1 \dots z_N\}$  の値（位置とか）を、平均がゼロのガウスノイズを載せた観測値  $\{x_1 \dots x_N\}$  で返すセンサ（GPSとか）で計測したいとき、 $z_n$  の値の推定のために、いちばん最近の数回の観測値を平均するとよいように思える...！
- もしセンサのノイズレベルが高い場合には、平均をとるときに比較的長い観測窓幅をとったほうがよいだろうし、ノイズレベルが低い場合には観測値を比較的そのまま使うほうがよいだろうし...というあたりを定式化したモデル。

# 3. 線形動的システム

遷移確率分布と出力確率分布はどちらもガウス分布であり、以下の一般的な形に書くことができる。

$$p(\mathbf{z}_n | \mathbf{z}_{n-1}) = \mathcal{N}(\mathbf{z}_n | \mathbf{A}\mathbf{z}_{n-1}, \mathbf{\Gamma})$$

$$p(\mathbf{x}_n | \mathbf{z}_n) = \mathcal{N}(\mathbf{x}_n | \mathbf{C}\mathbf{z}_n, \mathbf{\Sigma}).$$

このモデルパラメータ推定に対してはEMアルゴリズムを使うのが一般的である。

このEステップについては、線形動的システムにおけるフォワード-バックワード再帰のような効率的な推論アルゴリズムが積和アルゴリズムより導かれ、以下のように呼ばれる。

- フォワード再帰についてはカルマンフィルタ方程式
- バックワード再帰についてはカルマンスムーザ方程式あるいは**Rauch-Tung-Striebel (RTS)** 方程式

くわしいメッセージの形については省略。



- もちろん線形ガウスでない動的システムも考えられる。これは、以下のような状況下で効果を発揮する。
  - 時系列に急激な構造変化が見られるとき
  - 異常値が存在するとき
  - 分布に非対称性や非正規性があるとき
  - 非線形システム
- このような非線形・非ガウスの状態空間モデルに対する扱いやすい推論アルゴリズムとしてサンプリング手法による定式化を用いたものが粒子フィルタ（ブートストラップフィルタ、モンテカルロフィルタ等とも呼ばれる）。