2013-08-25

PRML復々習レーン

8.4.5 max-sumアルゴリズム

@tomerun

やること

前節では、因子グラフで表される同時分布から、周辺分布を効率的に計算するための積和アルゴリズムを導出した。

この節では、積和アルゴリズムを変形し、同時分布の最大値と、その最大値を与える変数の値を求めるための手法、max-sumアルゴリズムを得る。

同時分布の最大化とは

例えば次のような分布を考える。

	x=0	x=1	x=2	p(y)
y=0	0.0	0.2	0.2	0.4
y=1	0.2	0.1	0.0	0.3
y=2	0.2	0.0	0.1	0.3
p(x)	0.4	0.3	0.3	

これをx,y両方について同時に最大化する。

同時というのは難しいので「一番ありそうなx」「一番ありそうなy」を別々に探してしまうと…

同時分布の最大化とは

	x=0	x=1	x=2	p(y)
y=0	0.0	0.2	0.2	0.4
y=1	0.2	0.1	0.0	0.3
y=2	0.2	0.0	0.1	0.3
p(x)	0.4	0.3	0.3	

yについて周辺化したp(x)を最大化するxは0
p(y)を最大化するyも0　だが、p(x=0,y=0) は 0

→各変数を別々に考えてはダメ。全変数を同時に考慮して最大化しないといけない。だがそれはとても大変
→前節でやったのと同じように、因子グラフの表す構造を使って効率的に計算する

max-productアルゴリズム

まず、同時分布の最大値を計算する方法を考える。（それを与える変数の値を求める方法はその後で）

求めたいものは、

$$\max_{\mathbf{x}} p(\mathbf{x}) = \max_{x_1} \ldots \max_{x_M} p(\mathbf{x})$$

p(x)がx_Mを含む項と含まない項に因数分解されるなら、

$$\max_{x_1} \ldots \max_{x_M} p(\mathbf{x}) = \max_{x_1} \ldots \max_{x_{M-1}} (\underbrace{p_{M-1}(x_1, \ldots x_{M-1})}_{x_M\text{を含まない項}} \max_{x_M} \underbrace{p_M(x_1, \ldots x_M)}_{x_M\text{を含む項}})$$

つまり、max演算は和演算のΣと同様に、関係しない項については前に出せる
→前節と全く同じ議論が成立する。単純に、Σをmaxで置き換えれば良い

max-productアルゴリズム

(8.66)のΣをmaxで置き換えて、

$$\mu_{f_s \to x}(x) = \max_{x_1} \ldots \max_{x_M} f_s(x, x_1, \ldots x_M) \prod_{m \in ne(f_s) \setminus x} \mu_{x_m \to f_s}(x_m)$$

(8.69)は変わらず

$$\mu_{x_m \to f_s}(x_m) = \prod_{l \in ne(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m)$$

前節の積和アルゴリズムと同様、これらのメッセージを葉ノードから根ノードへ送っていき、最後に根ノードで最大化を行うと、同時確率の最大値が得られる。

これがmax_productアルゴリズムと呼ばれる。

max-sumアルゴリズム

多くの小さな項の積はアンダーフローを引き起こすため、logを取って積を和に変換するのが常套手段である。また、logは単調関数なのでmax演算と交換可能。

$$\ln(\max_{\mathbf{x}} p(\mathbf{x})) = \max_{\mathbf{x}} \ln p(\mathbf{x})$$

前ページに記載したメッセージの式について、両辺にlnを適用して、lnとmaxの順序を交換し、ln$\mu \rightarrow \mu$ と書き直すと、次のメッセージの表現を得る。

$$\mu_{f_s \rightarrow x}(x) = \max_{x_1} \ldots \max_{x_M} [\ln f_s(x, x_1, \ldots x_M) + \sum_{m \in ne(f_s) \setminus x} \mu_{x_m \rightarrow f_s}(x_m)]$$

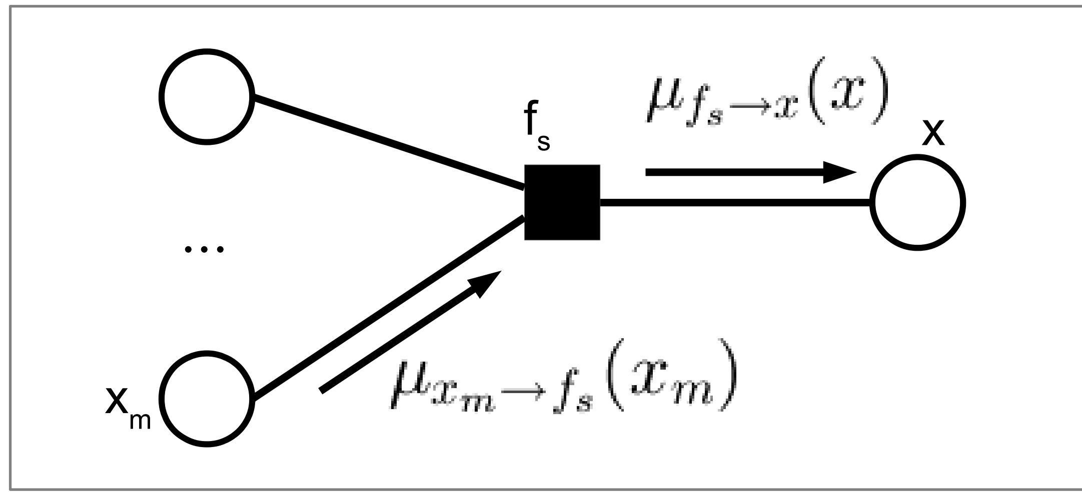
$$\mu_{x_m \rightarrow f_s}(x_m) = \sum_{l \in ne(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)$$

max-sumアルゴリズム

葉ノードから次の式でメッセージの伝達を始め、前ページの式に従って次のノードへ送っていく。

$$\mu_{x \to f}(x) = 0$$

$$\mu_{f \to x}(x) = \ln f(x)$$



根ノードでは、次の式で同時分布の最大値が得られる。

$$p^{max} = \max_x \left[\sum_{s \in ne(x)} \mu_{f_s \to x}(x) \right]$$

これが<u>max-sumアルゴリズム</u>である。

最大値を与える変数の値

根ノードで同時確率の最大値を与える変数の値は、次で求められる。

$$x^{max} = \arg\max_{x}[\sum_{s \in ne(x)} \mu_{f_s \to x}(x)]$$

他の変数についても、同時確率の最大値を与える値を求めるにはどうすれば良いだろうか。

前節で各変数についての周辺分布を求めたときと同じように、根から葉に向かって逆向きのメッセージパッシングを行い、それぞれのノードで根ノードでやったのと同じようにして値を求めるという方法が考えられるが、これはうまくいかない。

最大値を与える変数の値

問題点：
最大の同時分布を与える変数値の組が複数通りある場合に、各変数について別々に考えていると、違う組に属する値を選んでしまい、全体として最大値を与える変数値の組にならない場合がある。

例として右の分布を考える。

	x=0	x=1
y=0	0.0	0.5
y=1	0.5	0.0

最大の同時分布を与える変数の組は (x,y) = (0,1)または(1,0) だが、xについてx=0を選び、yについてy=0を選んでしまうと、これらを合わせた p(x=0,y=0) は0になってしまう。

バックトラック

この問題点を解消するために、次のようにする。

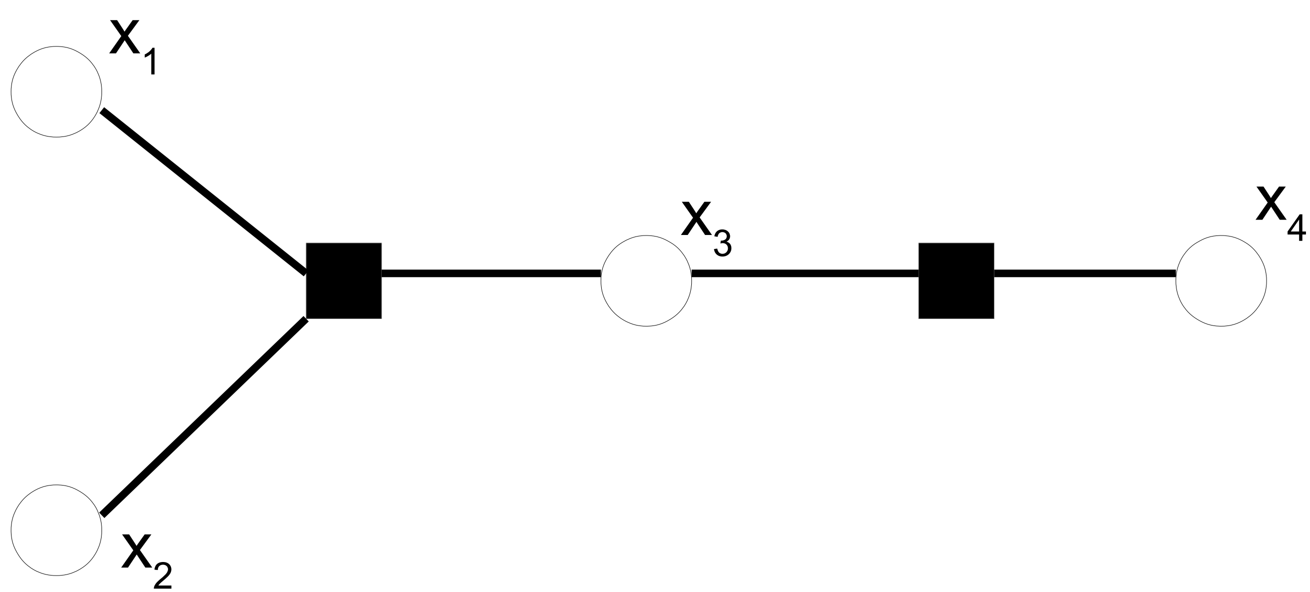
各変数ノードx_{n-1}で、親の変数の値x_n各々について、その場合に最大値を与えるx_{n-1}の値を記録しておく。
式で書くと以下の値（注：これはx_nの関数）。

$$\phi(x_n) = \arg\max_{x_{n-1}}[\ln f_{n-1,n}(x_{n-1}, x_n) + \mu_{x_{n-1}\to f_{n-1,n}}(x_{n-1})]$$

根ノードで最大の同時確率を与える変数の値を決めたら、この関係を逆向きに辿っていけば、全変数の値がわかる。

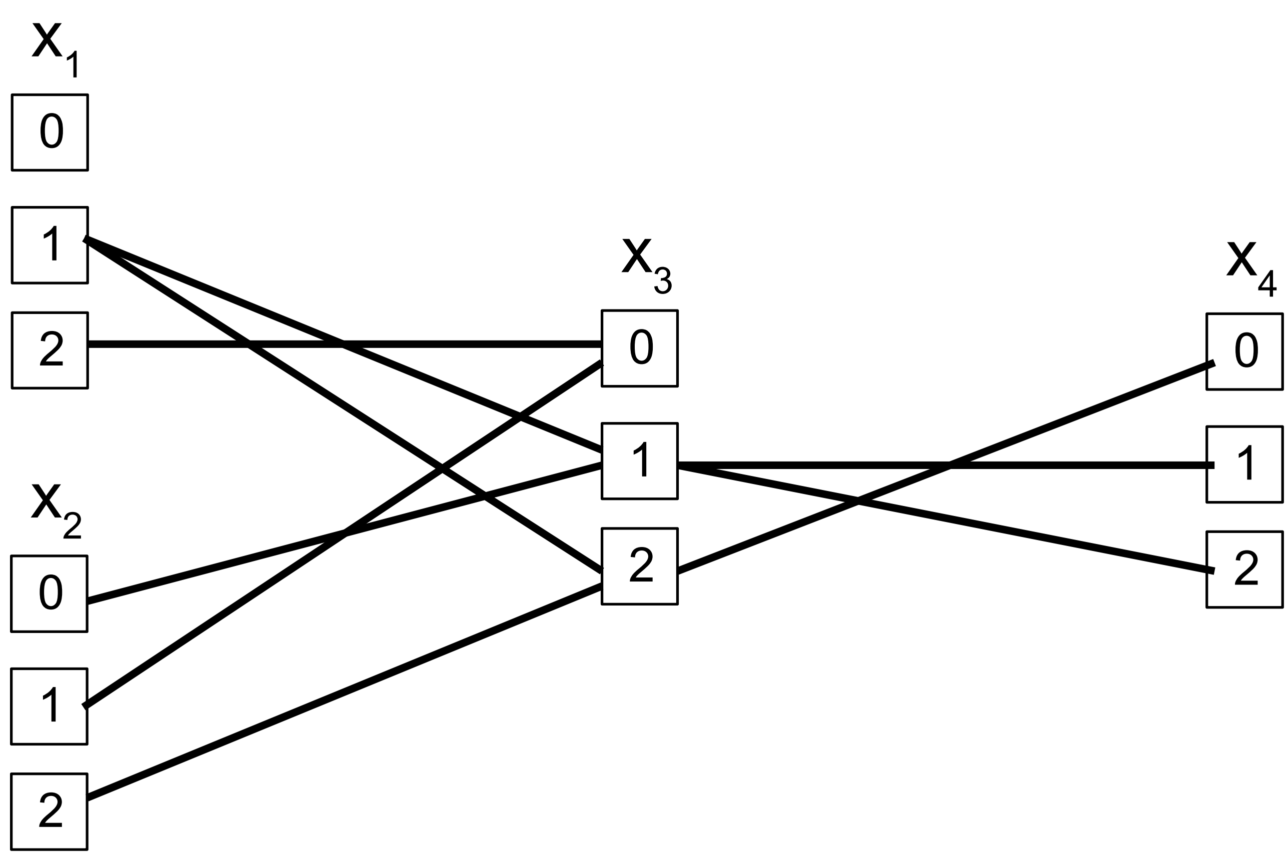
バックトラック

例として、4つの変数があり、それぞれ{0,1,2}の3状態を取るケースを考える。
次のようなグラフでこの確率分布が表されるとする。



バックトラック

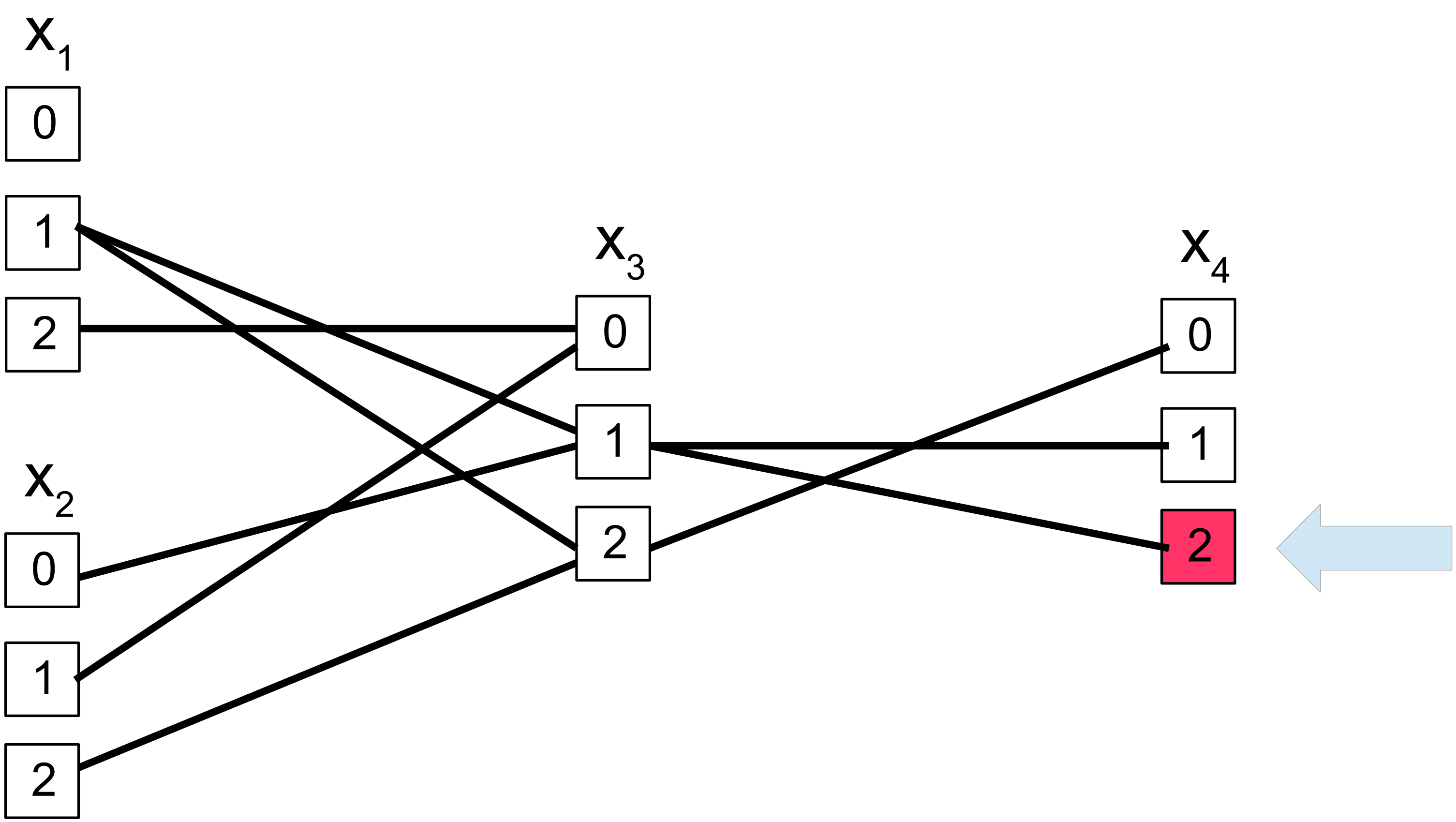
各ノードで、親ノードの値ごとに最大値を与える変数値を表にすると、次のようになる。



これは格子図やトレリス図と呼ばれる。

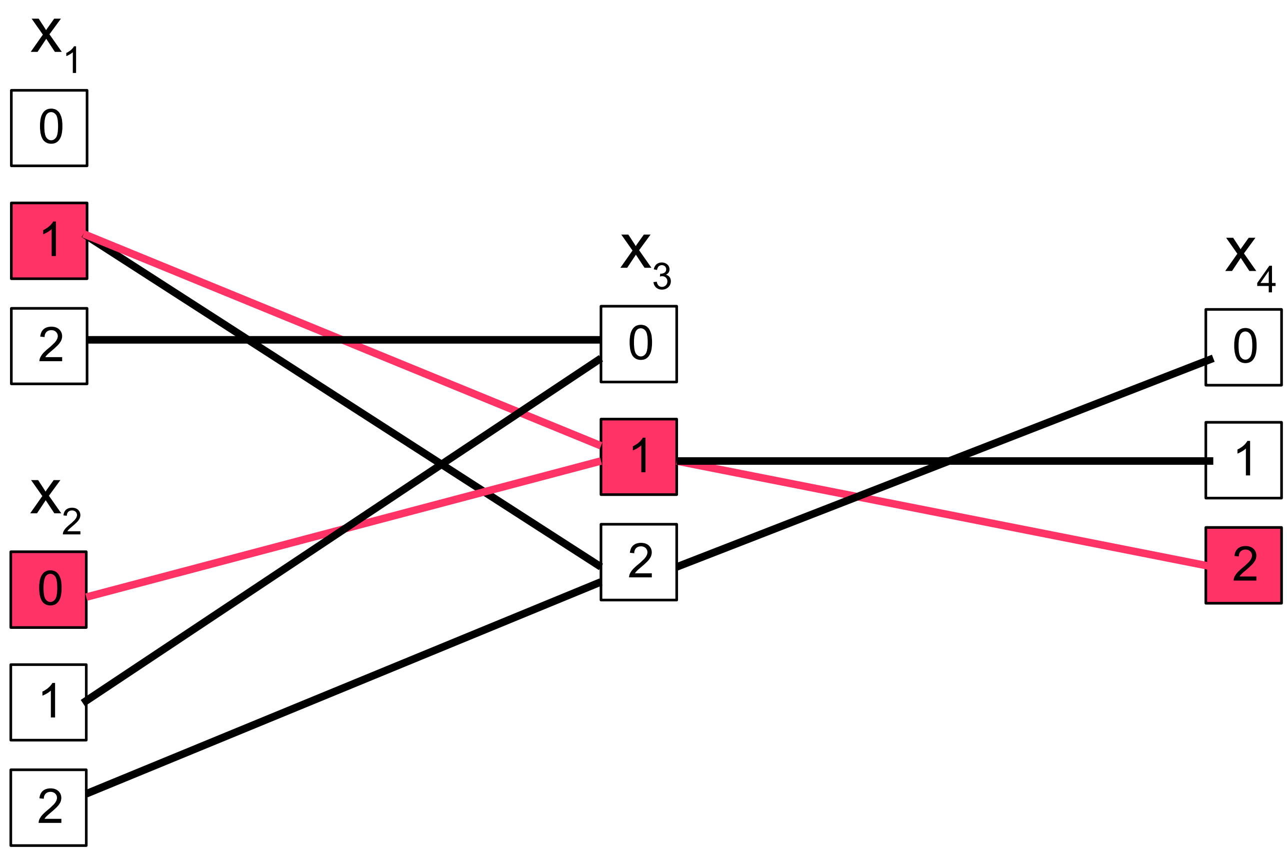
バックトラック

根ノードでx_4=2の場合に最大の同時確率が得られるとすると、



バックトラック

葉ノードに向かって表を辿っていけば、全変数で同時確率最大値を与える値がわかる。



これは<u>動的計画法</u>の応用例。

コメント

- 複数の変数が集まる因子ノードの所では、結局複数変数の同時最適化をがんばって行う必要はある
- 離散じゃなくて連続変数の場合どうなるの