Deep Learning 輪読会 2017

第16章　深層学習のための構造化確率モデル

2017.12.18

東京大学

中村藤紀

構成

16.1 非構造化モデルの課題

16.2 グラフを使用したモデル構造の記述

16.3 グラフィカルモデルからのサンプリング

16.4 構造化モデリングの利点

16.5 依存関係の学習

16.6 推論と近似推論

16.7 構造化確率モデルへの深層学習のアプローチ

はじめに

- **構造化確率モデル** (**structured probabilistic model**)
  - 確率分布を記述する方法。
  - 確率分布内のどの確率変数が互いに直接相互作用するかを記述するためにグラフを使用。
  - モデルの構造をグラフで定義 = **グラフィカルモデル** (**graphical model**)

16.1 非構造化モデルの課題

確率モデルが解くことができるタスク

- **密度推定 (Density estimation)**
  - 入力 x が与えられた下で、データ生成分布の下での真の密度 p(x) を推定
- **ノイズ除去 (Denoising)**
  - 損傷していたり、不完全に観測されていたりする入力が与えられた下で、元の正しい x を推定
- **欠損値補完 (Missing value imputation)**
  - x のいくつかの要素の観測が与えられた下で、モデルは x の観測されない要素の一部またはすべてにわたる推定量か確率分布を返す
- **サンプリング (Sampling)**
  - モデルは分布 p(x) から新しいサンプルを生成

16.1 非構造化モデルの課題

- テーブル形式のアプローチの限界
  - k 個の値を取る n 個の離散変数からなる確率ベクトル x の分布をモデル化するために、K^n 通りのパラメータが必要

- メモリ (表現を格納するコスト)
  - n, k が非常に小さい数である場合を除き、非常に多くのパラメータを保存することになる。
- 統計的効率
  - 非常に多くのパラメータを学習するためには、非常に多くの訓練データを必要とする。
- 実行時間 (推論コスト, サンプリングコスト)
  - 同時分布から周辺分布や条件付き分布を計算するために、テーブル全体を合計する必要がある。

16.2 グラフを使用したモデル構造の記述
16.2.1 有向モデル

- **有向グラフィカルモデル (directed graphical model)**
  - 信念ネットワーク (belief network), ベイジアンネットワーク (Bayesian network)
  - 局所条件付き確率分布 (local conditional probability distributions) の集合で定義される。

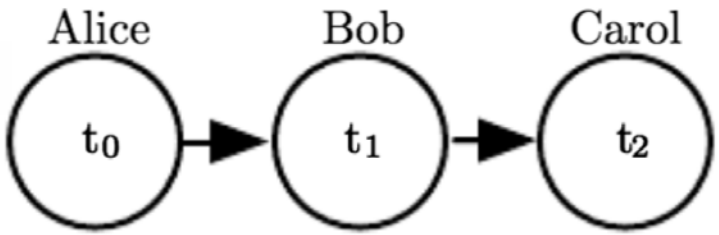
X_i の親ノード

$$p(\mathbf{x}) = \Pi_i \, p(\mathrm{x}_i \mid Pa_{\mathcal{G}}(\mathrm{x}_i)).$$

局所条件付き確率

$$p(\mathrm{t}_0, \mathrm{t}_1, \mathrm{t}_2) = p(\mathrm{t}_0)p(\mathrm{t}_1 \mid \mathrm{t}_0)p(\mathrm{t}_2 \mid \mathrm{t}_1).$$

グラフで表現



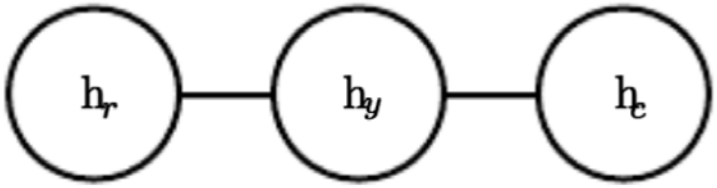
リレー競走の例。
アリスのゴール時間 t_0 はボブのゴール時間 t_1 に影響。
ボブのゴール時間 t_1 はキャロルのゴール時間 t_2 に影響。

16.2.2 無向モデル

- **無向モデル** (**undirected model**)
  - マルコフ確率場 (Markov random fields, MRFs), マルコフネットワーク (Markov networks)
  - 相互作用がそもそも方向を持たなかったり、両方向に作用する場合、無向グラフを使用する方が適切かもしれない。
  - **因子** (factor) (or **クリークポテンシャル** (clique potential)) は、そのクリーク内の変数が取りうる結合状態のそれぞれに対して、その変数の親和性を測定。非負。

非正規化確率分布　　**クリークポテンシャル**

$$\tilde{p}(\mathbf{x}) = \Pi_{\mathcal{C} \in \mathcal{G}} \phi(\mathcal{C}).$$



ルームメイトの健康状態 h_r, あなたの健康状態 h_y, 同僚の健康状態 h_c が互いにどのように影響しているかを表す無向グラフ

16.2.3 分配関数

- **分配関数** (partition function)
  - 総和もしくは積分が1となるとは限らない非正規化確率分布から、Z を使って正規化し、有効な確率分布を得る。

$$p(\mathbf{x}) = \frac{1}{Z}\tilde{p}(\mathbf{x}) \qquad Z = \int \tilde{p}(\mathbf{x})d\mathbf{x}$$

  - Z は **x** のすべての取りうる同時割り当てについて積分または総和を取るので、しばしば計算が困難。近似アルゴリズムが必要 (→ 18章)
- (注意) Z が存在しないような方法で因子を指定することが可能。
  - 例：Φ(x) = x^2 でスカラー変数 x をモデル化

$$Z = \int x^2 dx$$

は発散し、対応する確率分布は得られない。

16.2.4 エネルギーベースモデル

- **エネルギーベースモデル** (energy-based model, EBM)

$$\tilde{p}(\mathbf{x}) = \exp(-E(\mathbf{x}))$$

エネルギー関数 (energy function)

  - この形式の分布は **ボルツマン分布** (Boltzmann distribution) と呼ばれる。
    - 多くのエネルギーベースモデルは **ボルツマンマシン** と呼ばれる。

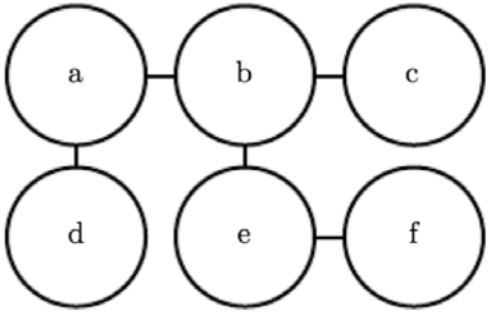
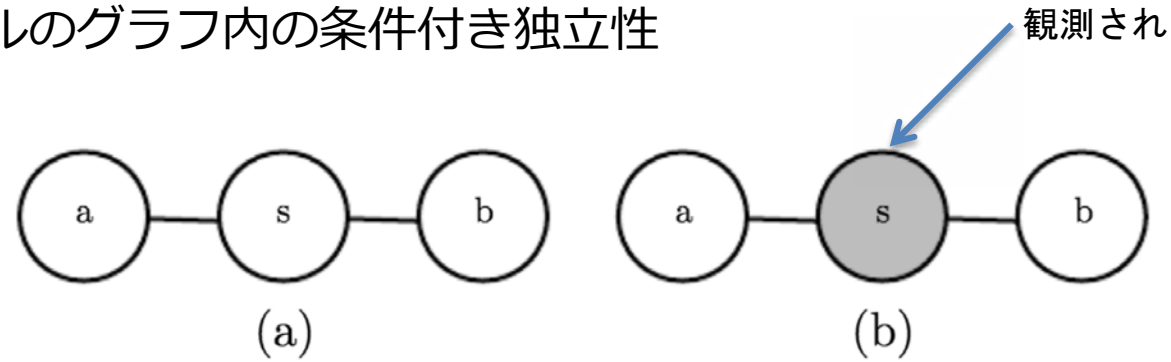


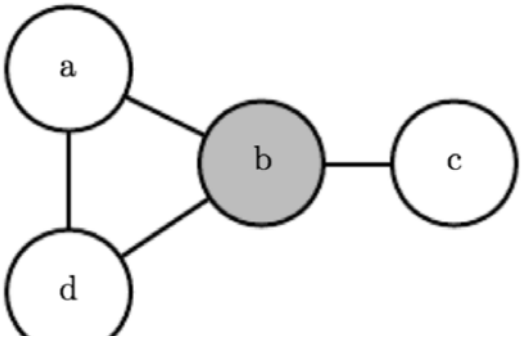
Figure 16.5: This graph implies that $E(\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{e}, \mathrm{f})$ can be written as $E_{\mathrm{a,b}}(\mathrm{a}, \mathrm{b}) + E_{\mathrm{b,c}}(\mathrm{b}, \mathrm{c}) + E_{\mathrm{a,d}}(\mathrm{a}, \mathrm{d}) + E_{\mathrm{b,e}}(\mathrm{b}, \mathrm{e}) + E_{\mathrm{e,f}}(\mathrm{e}, \mathrm{f})$ for an appropriate choice of the per-clique energy functions. Note that we can obtain the ϕ functions in figure 16.4 by setting each ϕ to the exponential of the corresponding negative energy, e.g., $\phi_{\mathrm{a,b}}(\mathrm{a}, \mathrm{b}) = \exp(-E(\mathrm{a}, \mathrm{b}))$.

16.2.5 分離と d 分離

- グラフィカルモデルが明示的に示す直接的な相互作用に加えて、どの変数が間接的に相互作用するか知りたい。
  - どの変数が互いに条件付き独立か？
- **分離** (separation)
  - 無向モデルのグラフ内の条件付き独立性



a, b間の経路はいずれの方向もアクティブ　　分離。経路は非アクティブ

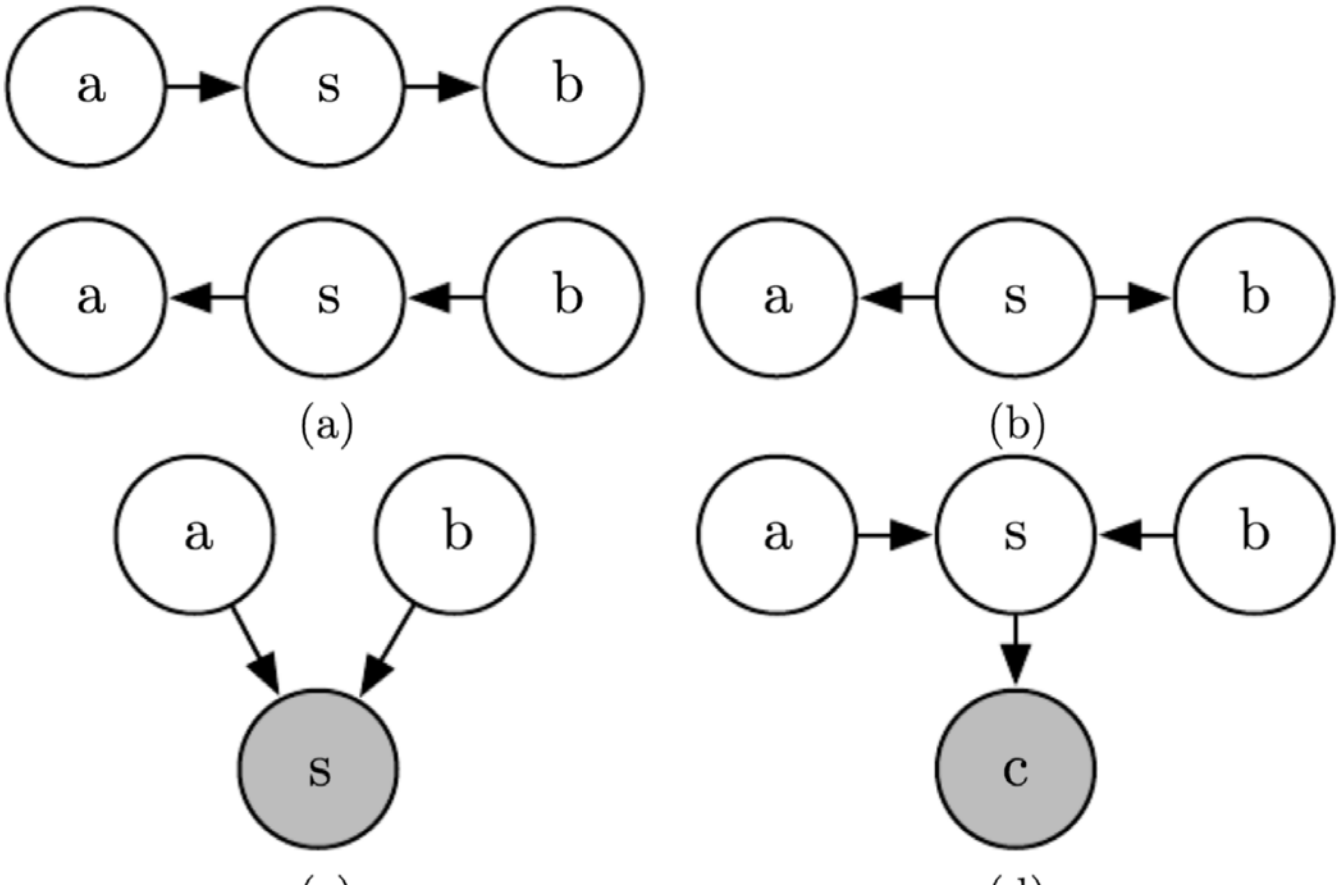


b が観測された下で、

- a, c は分離
- a, d は、その間の1つの経路が遮断されるが、別のアクティブな経路があるため、分離されない。

16.2.5 分離と d 分離

- **d 分離** (有向分離, d-separation)
  - 有向モデルにおける条件付き独立性
  - 以下のグラフでは、a から b への経路はすべてアクティブ。



16.2.5 分離と d 分離

- 問題

  右のグラフについて、

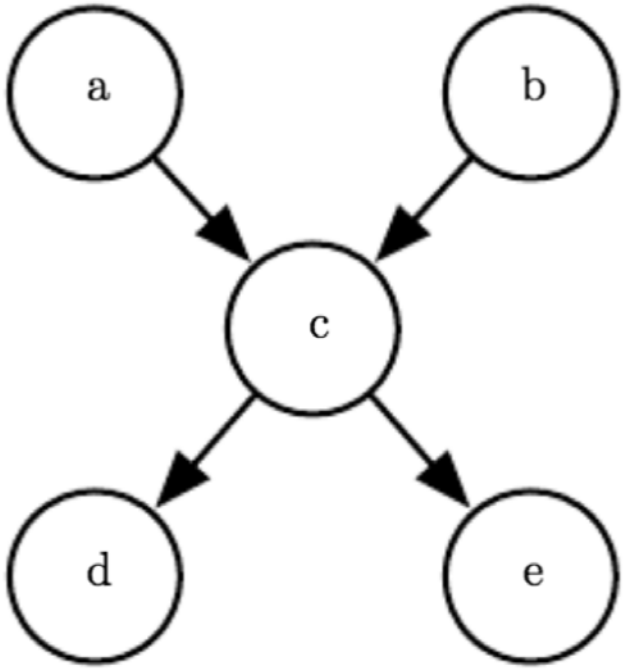
  – a と b は空集合が与えられた下で _____。

  – a と e は c が与えられた下で _____。

  – d と e は c が与えられた下で _____。

  – a と b は c が与えられた下で _____。

  – a と b は d が与えられた下で _____。

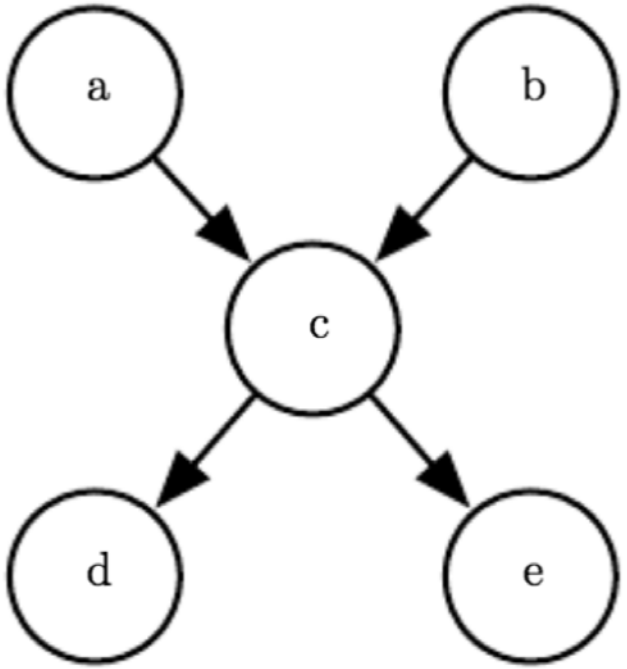


16.2.5 分離と d 分離

- 解答

  右のグラフについて、

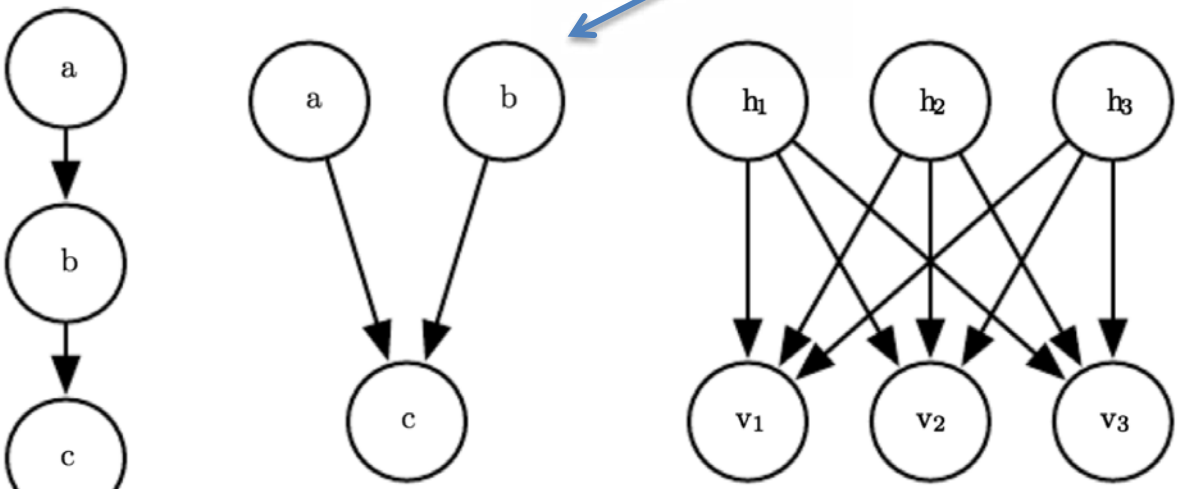
  – a と b は空集合が与えられた下で d 分離される。
  – a と e は c が与えられた下で d 分離される。
  – d と e は c が与えられた下で d 分離される。
  – a と b は c が与えられた下で d 分離されない。
  – a と b は d が与えられた下で d 分離されない。



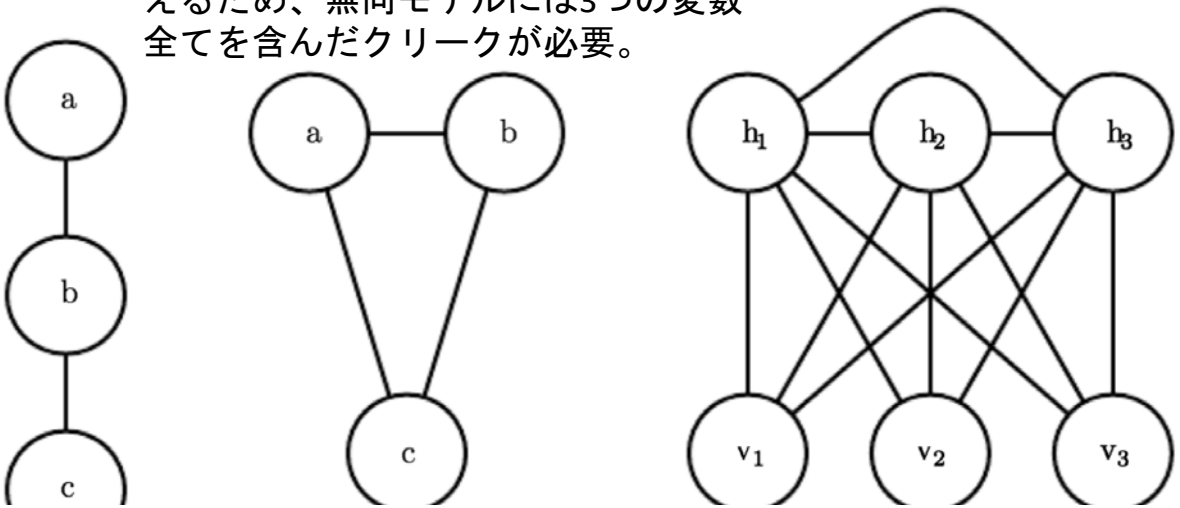
16.2.6 有向グラフと無向グラフの変換

- 有向モデルを無向モデルに変換：モラル化

非モラル (immorality)
a, b は c の親であり、両者を直接接続する辺がどちらの向きにもない。



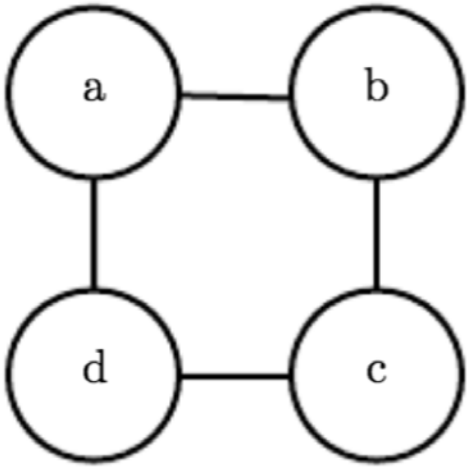
c が観測された下で a,b がアクティブな経路によって接続されることを捉えるため、無向モデルには3つの変数全てを含んだクリークが必要。

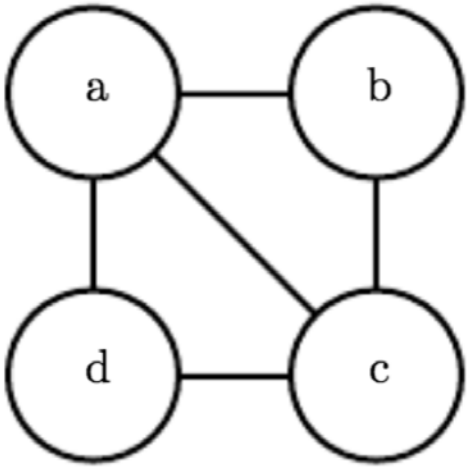


16.2.6 有向グラフと無向グラフの変換

- 無向モデルを有向モデルに変換
  - ループ：無向辺で接続された変数の連なり。連なりの最後の変数は最初の変数に接続されている。
  - 弦　　：ループを定義する連なり内の任意の2つの連続していない変数間の接続。

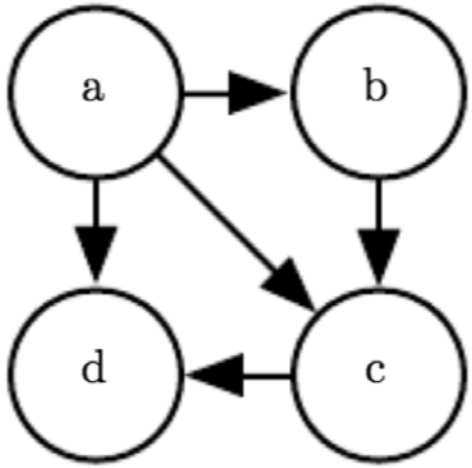
弦がない長さ 4 のループを持つため、
有向モデルに変換できない。





長さ 4 以上のループが弦を持つように、
グラフを三角化。
a, c を接続 or b, d を接続。

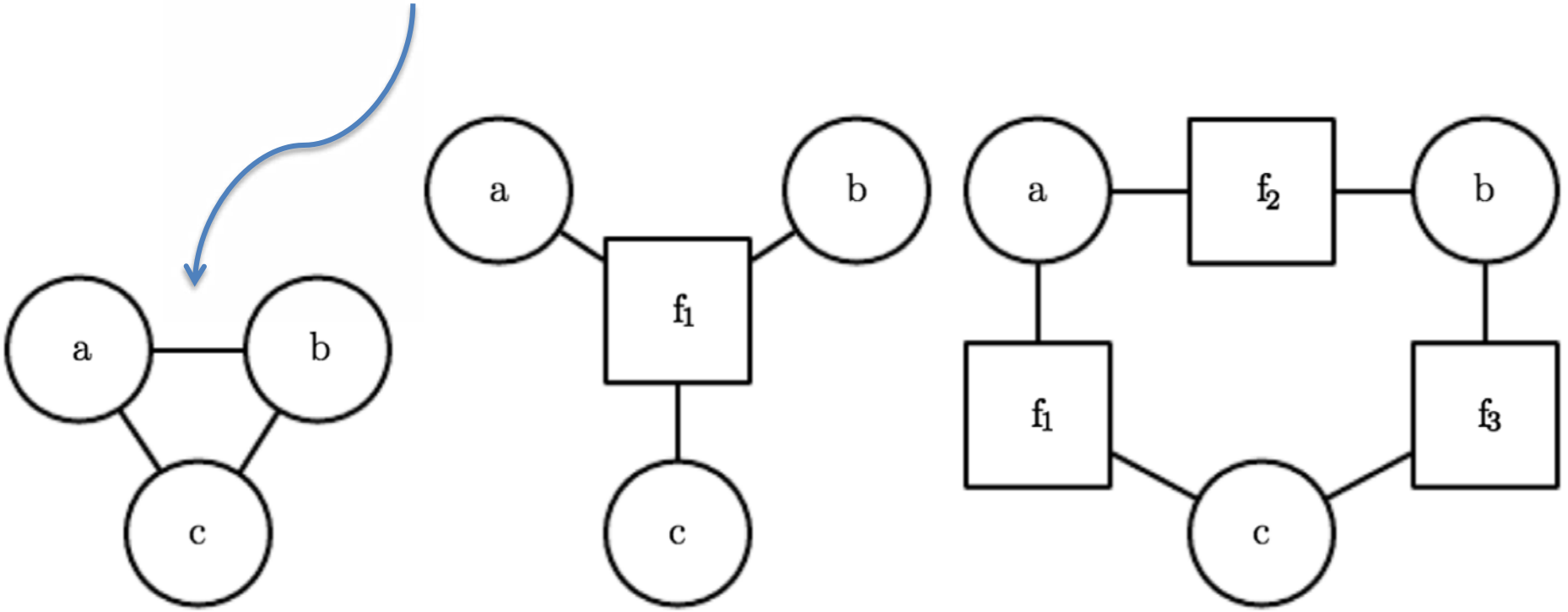
有向巡回を作らないように、
各辺に方向を割り当てる。



16.2.7 因子グラフ

- **因子グラフ** (factor graphs) は、無向グラフにおける曖昧性を解決する。

ポテンシャルを、 3変数 (a, b, c) に対して計算するのか、変数の各ペア (a, b), (b, c), (c, a) に対して計算するのか曖昧。



因子グラフによって曖昧性が解決される。

16.3 グラフィカルモデルからのサンプリング

- 有向モデルからのサンプリング
  - **伝承サンプリング** (**ancestral sampling**)
    - 親ノードが所与の下で各ノードを順番にサンプリングしていく。
- 無向モデルからのサンプリング
  - **ギブスサンプリング** (**Gibbs sampling**)
    - 各変数 x_i を順番に他のすべての変数で条件付けられた $p(x_i \mid x_{-i})$ からサンプリング。これを繰り返す。
  - より詳細な説明は17章を参照。

16.4 構造化モデリングの利点

- 構造化確率モデルを用いる最も重要な利点
  - 確率分布を表現するコストだけではなく、
  - 学習と推論のコストも劇的に削減できること。
  - いくらかの相互作用をモデル化しないように選択しているため、短い実行時間と少ないメモリで実行可能。
- 構造化確率モデルを使うことによる定量化できない利益
  - 「知識表現」と「知識の学習もしくは既存の知識が与えられた下での推論」を明示的に分離することができる。
    - 幅広いクラスのグラフに適用可能な学習アルゴリズムと推論アルゴリズムを設計・分析・評価することができる。
    - それとは独立に、データの中で重要だと考えられる関係を捉えるモデルを設計することができる。

    → モデルの開発とデバックが容易になる。

16.5 依存関係の学習

- 観測変数 **v** における分布を正確に捉えるために、潜在変数 **h** を導入
  - モデルは、v_i と **h** 間の直接的な依存関係と、**h** と v_j 間の直接的な依存関係とを介して、間接的に v_i と v_j 間の依存関係を捉えることができる。
  - **h** は **v** の別の表現を提供。潜在変数を学習することで、特徴学習を達成。

- 潜在変数を使わないアプローチ: **構造学習** (**structure learning**)
  - 観測変数のうち、密に結びついた変数を接続、他の変数間の辺を省略するように設計。
    - いくつかの構造を試す。
    - どの構造がより良いか決定。
    - ベストな構造に少数の辺を追加もしくは削除した候補構造を提案。
    - 再び探索。

16.6 推論と近似推論

- **推論 (inference)**
  - p(**h** | **v**) を計算したい。

    多くの場合 最尤原理を使ってモデルを学習するため。

    $$\log p(\boldsymbol{v}) = \mathbb{E}_{\mathbf{h} \sim p(\mathbf{h}|\boldsymbol{v})} \left[\log p(\boldsymbol{h}, \boldsymbol{v}) - \log p(\boldsymbol{h} \mid \boldsymbol{v}) \right]$$

  - 多くの深層モデルでは、構造化グラフィカルモデルを利用したとしても困難。

  → 近似推論を使用 (深層学習では通常 変分推論)

    - 変分推論では、できるだけ真の分布に近い近似分布 q(**h** | **v**) を求めることで、真の分布 p(**h** | **v**) を近似。
    - 詳細は19章。

16.7 構造化確率モデルへの深層学習のアプローチ

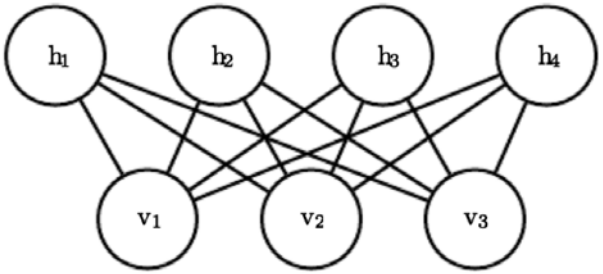
- 深層学習では、つねに深いグラフィカルモデルが必要とされるわけではない。
  - 計算グラフの深さ ≠ グラフィカルモデルの深さ
- 深層学習では、つねに分散表現の考え方が使われている。
  - 観測変数より潜在変数の方が多い ↔ 伝統的なグラフィカルモデルの潜在変数は通常少ない。
- 深層学習での潜在変数は、前もって何か特定の意味を取るようには意図されていない。
  - 伝統的なグラフィカルモデルでの潜在変数は、特定の意味を念頭に置いて設計される。
- 深層学習では接続があまりスパースではない。
  - 伝統的なグラフィカルモデルは、接続がスパース。
  - スパースに接続されたグラフでたいていうまく機能する **ループあり確率伝播法** (loopy belief propagation) が、深層学習ではほとんど使われない。
  - 代わりに、ギブスサンプリングや変分推論を効率的にするように設計される。

16.7.1 例：制限付きボルツマンマシン

- **制限付きボルツマンマシン** (restricted Boltzmann machine, RBM

  or **ハーモニウム** (harmonium)

  – 詳細は20.2節



- Binary RBM

  – 二値の可視ユニットと隠れユニットを持つエネルギーベースモデル

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{b}^\top \boldsymbol{v} - \boldsymbol{c}^\top \boldsymbol{h} - \boldsymbol{v}^\top \boldsymbol{W} \boldsymbol{h}$$

  – どの2つの可視ユニット間にも、どの2つの隠れユニット間にも直接的な相互作用はない

  → 「制限付き」

  – 条件付き分布が陽に求められる。

$$p(\mathbf{h} \mid \mathbf{v}) = \Pi_i p(\mathrm{h}_i \mid \mathbf{v}) \quad P(\mathrm{h}_i = 1 \mid \mathbf{v}) = \sigma\left(\mathbf{v}^\top \boldsymbol{W}_{:,i} + b_i\right)$$

$$p(\mathbf{v} \mid \mathbf{h}) = \Pi_i p(\mathrm{v}_i \mid \mathbf{h}) \quad P(\mathrm{h}_i = 0 \mid \mathbf{v}) = 1 - \sigma\left(\mathbf{v}^\top \boldsymbol{W}_{:,i} + b_i\right)$$

  → 効率的なブロックギブス (block Gibbs) サンプリングが可能。

参考文献

- Deep Learning
  - Ian Goodfellow, Yoshua Bengio, Aaron Courville
  - 日本語版
    https://www.amazon.co.jp/%E6%B7%B1%E5%B1%A4%E5%AD%A6%E7%BF%92-Ian-Goodfellow/dp/4048930621