制約ボルツマンマシン

清家大嗣

平成 31 年 11 月 1 日

1 制約ボルツマンマシン (RBM: Restricted Boltzmann Machine) について ([1] の 8 章を参考)

制約ボルツマンマシンとは, その名が示す通りボルツマンマシンの 1 種であり, ユニット間の結合に制約があるものをいう. 例えば, 図 1 が示す制約ボルツマンマシンは, 隠れ変数 (Hidden Variable) を持つユニット, 可視変数 (Visible Variable) を持つユニットは, 各々同種のユニット同士では結合を持たない. また, 可視変数を持つユニットは全ての隠れ変数を持つユニットとの結合を持つという特徴を持っている (この制約ボルツマンマシンを用いたディープビリーフネットワークをベースをきっかけとして, 今日の深層学習は成功した). 本稿では, 一般のボルツマンマシンや, 制約ボルツマンマシンの持つ性質について理解することを目的とする.

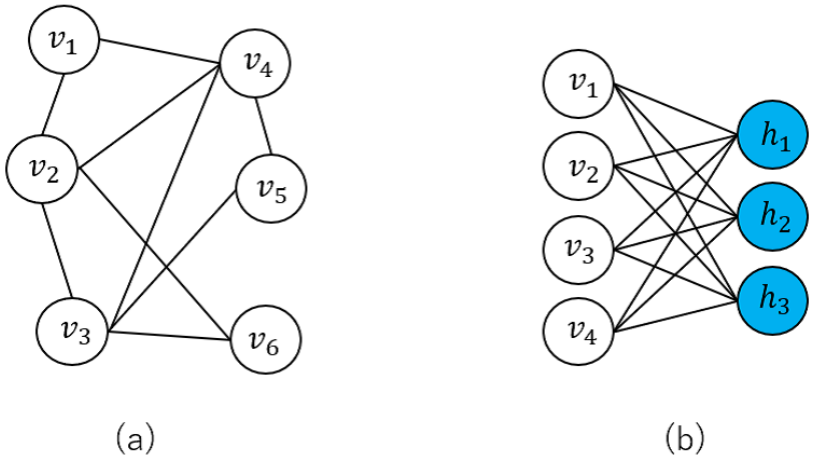

図 1: (a) 一般のボルツマンマシン (b) 制約ボルツマンマシンの一例. h_i は隠れ変数を表し, 水色をしたユニットは隠れ変数を持つユニットである.

1.1 ボルツマンマシンについて

ボルツマンマシンでは, 複数のユニットが向きを持たない結合により結びついている無向グラフについて考える. 各ユニットは 2 種 (0, 1) どちかの値を持ち i 番目のユニット状態を $x_i(= 0 \text{ or } 1)$

とし, 全ユニットの状態を M 次元ベクトル $\boldsymbol{x}$ により表す. この状態 $\boldsymbol{x}$ は, ボルツマンマシンでは次式の確率分布により与えられる.

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\{-\Phi(\boldsymbol{x}, \boldsymbol{\theta})\}. \tag{1}$$

ここで, $\Phi(\boldsymbol{x}, \boldsymbol{\theta})$ はエネルギー関数と呼ばれる. エネルギー関数 Φ は次式により定義される.

$$\Phi(\boldsymbol{x}, \boldsymbol{\theta}) = -\boldsymbol{b}^T\boldsymbol{x} - \sum_{(i,j)\in\boldsymbol{\epsilon}} w_{ij}x_ix_j \tag{2}$$

$\boldsymbol{\epsilon}$ はユニット間の結合であるエッジ集合である. (2) 式より, $\boldsymbol{\theta}$ はバイアスベクトル $\boldsymbol{b}$, エッジの重み集合 $\{w_{i,j}|(i,j) \in \boldsymbol{\epsilon}\}$[1] により表現される. $Z(\boldsymbol{\theta})$ は規格化定数[2]であり, 分配関数と呼ばれる.

各ユニットが 0, 1 の 2 種類の状態を持つため, 全状態数は 2^M となる. 従って, (1, 2) 式により 2^M 通りの状態の生起確率を与えられる. また, 一般に (2) 式の形の分布をボルツマン分布 (またはギブス分布) と呼ぶ.

1.1.1 (1, 2) 式に従うボルツマン分布の学習

$p(\boldsymbol{x}, \boldsymbol{\theta})$ が真の分布 $p_g(\boldsymbol{x})$ に近くなるように, データ集合 $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$ から最尤推定により $\boldsymbol{\theta}$ を推定する. (1) 式を用いて全データに関する結合確率 $L(\boldsymbol{\theta})$ の対数を取り, それを最大化するような $\boldsymbol{\theta}$ を求める. つまり, 次式を最大化する $\boldsymbol{\theta}$ を求める.

$$\log L(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \{-\Phi(\boldsymbol{x}_n, \boldsymbol{\theta}) - \log Z(\boldsymbol{\theta})\} \tag{3}$$

この対数尤度関数 (log-likelihood function) のパラメータ $\boldsymbol{\theta}$ に関する勾配は次式により計算できる[3].

$$\begin{aligned}
\frac{\partial \log L(\boldsymbol{\theta})}{\partial b_i} &= \sum_{n=1}^{N} \left\{ x_{ni} - \frac{1}{Z(\boldsymbol{\theta})} \cdot \frac{\partial Z(\boldsymbol{\theta})}{\partial b_i} \right\} = \sum_{n=1}^{N} \left\{ x_{ni} - \frac{1}{Z(\boldsymbol{\theta})} \cdot \sum_{\boldsymbol{x}} \exp\{-\Phi(\boldsymbol{x}, \boldsymbol{\theta})\} x_i \right\} \\
&= \sum_{n=1}^{N} \left\{ x_{ni} - \sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{\theta}) x_i \right\} = \sum_{n=1}^{N} x_{ni} - N E_{\boldsymbol{\theta}}[x_i]
\end{aligned} \tag{4}$$

$$\begin{aligned}
\frac{\partial \log L(\boldsymbol{\theta})}{\partial w_{ij}} &= \sum_{n=1}^{N} \left\{ x_{ni}x_{nj} - \frac{1}{Z(\boldsymbol{\theta})} \cdot \frac{\partial Z(\boldsymbol{\theta})}{\partial w_{ij}} \right\} = \sum_{n=1}^{N} \left\{ x_{ni}x_{nj} - \frac{1}{Z(\boldsymbol{\theta})} \cdot \sum_{\boldsymbol{x}} \exp\{-\Phi(\boldsymbol{x}, \boldsymbol{\theta})\} x_i x_j \right\} \\
&= \sum_{n=1}^{N} \left\{ x_{ni}x_{nj} - \sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{\theta}) x_i x_j \right\} = \sum_{n=1}^{N} \{x_{ni}x_{nj}\} - N E_{\boldsymbol{\theta}}[x_i x_j]
\end{aligned} \tag{5}$$

ここで, $E_{\boldsymbol{\theta}}[\cdot]$ は $\boldsymbol{\theta}$ が与えられた場合に定まるモデル分布から計算される $\cdot$ の期待値である. 従って, 勾配法を実行する際, その時点での $\boldsymbol{\theta}$ から $p(\boldsymbol{x}|\boldsymbol{\theta})$ は計算可能なように思える. しかし, M が大きくなると期待値計算に必要な計算回数が指数関数的に増加する (例えば $M = 10$ の場合, 全ての $p(\boldsymbol{x}|\theta)$ を計算し終えるまでに 1,048,576 回の施行が必要となる). 従って, (4, 5) 式を用いて, 素朴に勾配法を実行するのは現実問題として無理である.

1.1.2 ギブスサンプリングについて

前節のように組み合わせ爆発により期待値計算が困難な場合に, 近似的に期待値を計算する手法について考える. 一般に多変数確率分布からそれに従う変数を生成することは困難であるが, ボル

[1] 無効グラフであるため $w_{ij} = w_{ji}$ となる.
[2] $Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \Phi(\boldsymbol{x}, \boldsymbol{\theta})$ $(\sum_{\boldsymbol{x}} = \sum_{x_1=0,1} \sum_{x2=0,1}) \cdots \sum_{x_M=0,1})$
[3] ただし, x_{ni} とは n 番目のデータの第 i 成分要素とする

ツマンマシンは局所マルコフ性という性質があり, ギブスサンプリング (Gibbs Sampling) と呼ばれる手法で効率的に確率変数ベクトル $\boldsymbol{x}$ をサンプリングできる.

ユニット i を除いた全ユニットの変数を並べたベクトルを $\boldsymbol{x}_{-i}$ と表現する. あるパラメータ $\boldsymbol{\theta}$ から生成されるモデル分布 $p(\boldsymbol{x}|\boldsymbol{\theta})$ から $\boldsymbol{x}$ を効率的にサンプルするため, 各ユニットがベクトル $\boldsymbol{x}_{-i}$ の値を取る場合のユニット i に関する事後分布について考える. ユニット i の値が x_i となる確率は次式で表される.

$$\begin{aligned} p(x_i|\boldsymbol{x}_i, \boldsymbol{\theta}) &= \frac{p(\boldsymbol{x}, \boldsymbol{\theta})}{\sum_{x_i=0,1} p(\boldsymbol{x}, \boldsymbol{\theta})} \\ &= \frac{\exp\{(b_i + \sum_{j \in \boldsymbol{N}_i} w_{ij} x_j) x_i\}}{1 + \exp\{(b_i + \sum_{j \in \boldsymbol{N}_i} w_{ij} x_j)\}} \end{aligned} \tag{6}$$

従って, モデルのパラメータベクトル $\boldsymbol{\theta}$ とユニット i 以外のユニットの値さえ与えられていれば, ユニット i を (6) 式の分布に従う確率変数としてサンプリングすることができる. これを $i = 1, ..., M$ に対して順番に繰り返してサンプリングする手法がギブスサンプリングと呼ばれる. 例えば, このステップを t 純目まで繰り返した場合, その時のユニット i は次式の分布に従う確率変数としてサンプリングされる.

$$p(x_i|\boldsymbol{x}_{-i}^{(}t), \boldsymbol{\theta}) = p(x_i|x_1^{(t)}, ..., x_{i-1}^{(t)}, x_{i+1}^{(t-1)}, ..., x_M^{(t-1)}, \boldsymbol{\theta}) \tag{7}$$

このサンプリングした値は t を十分に大きくとれば, モデル分布 $p(\boldsymbol{x}, \boldsymbol{\theta})$ に従う [2]. また, このサンプリング列から複数のサンプル $\boldsymbol{x}^{(t_1)}, \boldsymbol{x}^{(t_2)}$ を得たい場合, サンプリングする間隔 $|t_2 - t_1|$ は十分に離れている必要がある (独立にするため).

(6, 7) 式に従うプロセスを繰り返すことで, (4, 5) 式の期待値を近似的に計算できる. つまり $O(a^M)$ $(a > 0)$ かかる計算コストを $O(MT)$ (T はギブスサンプリングの繰り返し回数) に減らすことができる[4].

1.2 隠れ変数を持つボルツマンマシン

図 1 (b) にあるような, 全変数がシステム外部に公開されていないボルツマンマシンもある. つまり, 可視変数ベクトル $\boldsymbol{v}$, 隠れ変数ベクトル $\boldsymbol{h}$ が与えられた場合, 前節の $\boldsymbol{x}$ に当たるものが $\boldsymbol{v}, \boldsymbol{h}$ であるが, $\boldsymbol{h}$ は観測不可能というケースである.

1.2.1 隠れ変数を持つボルツマンマシンの学習

最初にモデル分布 $p(\boldsymbol{v}, \boldsymbol{h}|\boldsymbol{\theta})$ を最大にする最尤法を観測可能な可視変数に対して用いるため, 観測していない隠れ変数 $\boldsymbol{h}$ を周辺化する[5].

$$p(\boldsymbol{v}|\boldsymbol{\theta}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h}|\boldsymbol{\theta}) \tag{8}$$

従って, 尤度関数の対数を取ることで次式を最大化する問題ととらえることができる.

$$\log L(\boldsymbol{\theta}) \propto \frac{1}{N} \sum_{n=1}^{N} \left[\log \sum_{\boldsymbol{h}} \exp\{-\Phi(\boldsymbol{v}_n, \boldsymbol{h}, \boldsymbol{\theta})\} - \log Z(\boldsymbol{\theta}) \right] \tag{9}$$

[4] それでもギブスサンプリングのコストは高いことには注目しておきたい.
[5] こうすることで, $\boldsymbol{v}$ のみの分布を最大化する問題として置き換える

右辺第一項を $\boldsymbol{\theta}$ で微分すると次式のようになる (ここで z_i, z_j は $\boldsymbol{v}, \boldsymbol{h}$ を連結したベクトル $\boldsymbol{z}$ の成分である).

$$\frac{\partial}{\partial w_{ij}} \log \sum_{\boldsymbol{h}} \exp\{-\Phi(\boldsymbol{v}_n, \boldsymbol{h}, \boldsymbol{\theta})\} = \frac{\sum_{\boldsymbol{h}} z_i z_j \exp\{-\Phi(\boldsymbol{v}_n, \boldsymbol{h}, \boldsymbol{\theta})\}}{\sum_{\boldsymbol{h}} \exp\{-\Phi(\boldsymbol{v}_n, \boldsymbol{h}, \boldsymbol{\theta})\}}$$
$$= \sum_{\boldsymbol{h}} z_i z_j p(\boldsymbol{h}|\boldsymbol{v}_n, \boldsymbol{\theta}) \tag{10}$$

従って, (9) 式は次のように書き換えることができる.

$$\log L(\boldsymbol{\theta}) \propto \mathrm{E}_{\boldsymbol{\theta}}[z_i z_j]_{data} - \mathrm{E}_{\boldsymbol{\theta}}[z_i z_j]_{model} \tag{11}$$

ただし, $\mathrm{E}_{\boldsymbol{\theta}}[z_i z_j]_{data} = 1/N \sum_{n=1}^{N} \sum_{\boldsymbol{h}} z_i z_j p(\boldsymbol{h}|\boldsymbol{v}_n, \boldsymbol{\theta})$ である. $\boldsymbol{h}$ の次元数を H とした場合の第一項の計算量を考慮すると, 上式の勾配計算のための期待値計算の計算量は $O(a^H + b^M)$ となる. 従って, (4, 5) 式より更に計算が困難となる.

1.3 制約ボルツマンマシン (RBM: Restricted Boltzmann Machine)

制約ボルツマンマシンは, 隠れ変数を持つボルツマンマシンの一種である. 図 1 にあるように, 可視変数間, 隠れ変数間の結合はなく, 任意の可視変数と隠れ変数の間には必ず結合がある. RBM ではこの特別な構造により, 効率的なギブスサンプリングが可能となる.

1.3.1 RBM のグラフ構造を利用したギブスサンプリング

RBM の分布関数で用いる新たな記法について説明する. RBM のエネルギー関数と状態の分布関数 $p(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{\theta})$ は次式により定義される (i, j は各々, 可視変数と隠れ変数のインデックスである. a_i, b_j は各々のユニットのバイアスである).

$$\Phi(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{\theta}) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j w_{ij} v_i h_j \tag{12}$$

$$p(\boldsymbol{v}, \boldsymbol{h}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\{-\Phi(\boldsymbol{v}, \boldsymbol{h}, \boldsymbol{\theta})\} \tag{13}$$

従って, 可視変数ベクトル $\boldsymbol{v}$ による隠れ変数ベクトルの条件付き分布は次式で定義される[6].

$$p(\boldsymbol{h}|\boldsymbol{v}, \boldsymbol{\theta}) = \prod_j p(h_j|\boldsymbol{v}, \boldsymbol{\theta})$$
$$= \frac{\exp\{(b_j + \sum_i w_{ij} v_i h_j) h_j\}}{1 + \exp(b_j + \sum_i w_{ij} v_i)} \tag{14}$$

つまり, 各ユニット変数 h_j, v_i は次のように, $\boldsymbol{v}$ により定まるベルヌーイ分布に従うと考えられる ($\sigma(x) = 1/(1 + \exp(-x))$ はロジスティック関数である).

$$\prod_j p(h_j = 1|\boldsymbol{v}, \boldsymbol{\theta}) = \frac{\exp(b_j + \sum_i w_{ij} v_i)}{1 + \exp(b_j + \sum_i w_{ij} v_i)}$$
$$= \sigma\left(1 + \exp(b_j + \sum_i w_{ij} v_i)\right) \tag{15}$$

可視変数の場合も同様にして, 次式で表せる.

$$\prod_j p(v_i = 1|\boldsymbol{h}, \boldsymbol{\theta}) = \sigma\left(1 + \exp(a_i + \sum_j w_{ij} h_j)\right) \tag{16}$$

[6] 隠れ変数同士に結合がない (独立) なため, $\boldsymbol{v}$ が与えられていることと合わせて 2 段目のような変形が可能である. 分母の第一項は $h_j = 0$ のケースを考え, 第二項は $h_j = 1$ のケースについて考えている項である.

従って, $\boldsymbol{v}^{(0)}$ を初期値として, (15, 16) 式を用いて逐次的に次のようにサンプリングしていくことができる.

$$\boldsymbol{v}^{(0)} \to \boldsymbol{h}^{(0)} \to \boldsymbol{v}^{(1)} \to \boldsymbol{h}^{(1)} \to ... \tag{17}$$

このサンプリング手法では, $\boldsymbol{v}^{(t)}, \boldsymbol{h}^{(t)}$ の値が変わらないことから並列計算が可能であり, 幾分か計算効率を改善することが可能である. しかし, (6) 式と (15, 16) 式を比較しても分かるように結合の数だけ積の計算をする必要があるのは変わらないため, それほど大きな改善にはならないと思われる.

1.3.2 RBM の学習

(17) 式を用いて次のように RBM の勾配を計算することができる ($\boldsymbol{h}$ による周辺化に注意. 隠れ変数同士の結合がないため, $p(\boldsymbol{h}|\boldsymbol{v}, \boldsymbol{\theta}) = \prod_j p(h_j|\boldsymbol{v}, \boldsymbol{\theta})$. (10) 式を参照).

$$\begin{aligned}\frac{1}{N}\frac{\partial \log L}{\partial w_{ij}} &= \frac{1}{N}\sum_{n=1}^{N}\left\{\sum_{h_j=0,1} v_{ni}h_j p(h_j|\boldsymbol{v}_n, \boldsymbol{\theta})\right\} - \sum_{\boldsymbol{v},\boldsymbol{h}} v_i h_j p(\boldsymbol{v}, \boldsymbol{h}|\boldsymbol{\theta}) \\ &= \frac{1}{N}\sum_{n=1}^{N}\left\{v_{ni}p(h_j = 1|\boldsymbol{v}_n, \boldsymbol{\theta})\right\} - \sum_{\boldsymbol{v},\boldsymbol{h}} v_i h_j p(\boldsymbol{v}, \boldsymbol{h}|\boldsymbol{\theta})\end{aligned} \tag{18}$$

同様にして, バイアスも次のように計算できる.

$$\frac{1}{N}\frac{\partial \log L}{\partial a_i} = \frac{1}{N}\sum_{n=1}^{N} v_{ni} - \sum_{\boldsymbol{v},\boldsymbol{h}} v_i p(\boldsymbol{v}, \boldsymbol{h}|\boldsymbol{\theta}) \tag{19}$$

$$\frac{1}{N}\frac{\partial \log L}{\partial b_j} = \frac{1}{N}\sum_{n=1}^{N}\sum_{h_j=0,1} h_j p(h_j|\boldsymbol{v}_n, \boldsymbol{\theta}) - \sum_{\boldsymbol{v},\boldsymbol{h}} h_j p(\boldsymbol{v}, \boldsymbol{h}|\boldsymbol{\theta}) \tag{20}$$

つまり, 各重みとバイアスの更新式は次式により表せるので, (17) 式を用いて学習が実行可能である.

$$\Delta w_{ij} \propto \mathrm{E}_{\boldsymbol{\theta}}[v_i h_j]_{data} - \mathrm{E}_{\boldsymbol{\theta}}[v_i h_j]_{model} \tag{21}$$

$$\Delta a_i \propto \mathrm{E}_{\boldsymbol{\theta}}[v_i]_{data} - \mathrm{E}_{\boldsymbol{\theta}}[v_i]_{model} \tag{22}$$

$$\Delta b_j \propto \mathrm{E}_{\boldsymbol{\theta}}[h_j]_{data} - \mathrm{E}_{\boldsymbol{\theta}}[h_j]_{model} \tag{23}$$

参考文献

[1] 岡谷貴之. 2015. 『深層学習』 講談社.

[2] K. P. Murphy. *Machine Learning: A probabilistic perspective. MIT Press, 2012.*