疎行列の非負値行列因子分解のための効率的な近似推定法

東京科学大学総合研究院難治疾患研究所　阿部興
東京科学大学総合研究院難治疾患研究所　島村徹平

1　はじめに

行列分解は行列の形式で得られたデータを低次元に射影して圧縮することでパターンを抽出する手法としてよく知られている. なかでも, 非負値行列因子分解（nonnegative matrix factorization; NMF）は非負性の制約により解釈がしやすい長所があり, 負の値を持たないデータに対して広く利用されている.

NMF をカウント（0 以上の整数）データに適用する場面では, 離散性の強く疎な（ゼロ要素が多い）行列が頻繁に見られる. 具体的に想定すると, 生命科学における細胞 × 遺伝子発現量の行列や自然言語を分析する際の文書 × 単語の行列といった例があげられる.

高次元の疎行列に対してはメモリ効率の観点から適した形式がある. 一方で, 確率モデルや推定論の観点からは, データが疎であることを積極的に活用して計算の効率を高めようとする議論はあまりされてこなかった. 観測のゼロ過剰や過分散をモデル化したケース（e.g. Abe & Yadohisa, 2017）や, 分解で得られる行列を疎にしようとした議論（e.g. Hyunsoo & Haesun, 2007）においてもこの点は考察されていない.

そこで本報告ではポアソン分布を用いて行列のゼロ要素を省略して計算効率を高める方法を提案する. 確率的勾配降下法に代表される大規模なデータに対して効率的にパラメータ推定を成し得る計算の技術は重要で, ディープニューラルネットワークの成功の背景にもそれがあると考えられる. 確率的勾配降下法はデータセットからのリサンプリングを行う. しかし, 疎行列の形式で 0 を省略して格納されているデータにランダムアクセスすることを考えると, 素朴な実装は指定した要素がゼロか非ゼロかはファイルすべてを確認しないとわからないため, 非効率となる. 本報告はこの点を改善するものである.

2　手法

いま, 非負の整数からなる R 行, C 列の行列 X に対して $X \approx ZW'$ となる非負の実行列 Z（R 行, L 列）, W（C 行, L 列）を探したい. 分解された行列の次元 L はユーザーが指定する. この分解のもとで, Z は各行について, W は各列についての特徴量とみなせる. この問題の確率モデルとしての定式化と変分ベイズ法による推定については Cemgil (2009) が詳細に議論した. ポアソン分布はカウントデータに対する基本的な分布であるため, Cemgil (2009) の設定に倣い, 確率的データ生成過程として次の組で表されるモデルを考える.

$$x_{ij} \mid z, w \sim \mathrm{Pois}\left(\sum_l w_{il} h_{lj}\right), \tag{2.1}$$

$$z_{il} \sim \mathrm{Gamma}(a, b), \tag{2.2}$$

$$w_{jl} \sim \mathrm{Gamma}(a, b). \tag{2.3}$$

ここで x_{ij}, z_{il}, w_{jl} はそれぞれ行列 X, Z, W の (i,j), (i,l), (j,l) 成分とした. また, $\mathrm{Pois}(\lambda)$ は平均 λ のポアソン分布, $\mathrm{Gamma}(a,b)$ は形状, レートパラメータがそれぞれ a, b のガンマ分布を表す. ガンマ分布は条件付き共役性によって選ばれた事前分布である.

その詳細は Cemgil (2009) に譲るが, 推定にあたっての主なアイデアは式 (2.1) と次の式 (2.4)

$$x_{ij} = \sum_{l=1} u_{ijl}, \quad u_{ijl} \sim \mathrm{Pois}\left(\sum_l z_{il} w_{jl}\right) \tag{2.4}$$

が確率分布として同値であることを用いて, 中間的な変数 u_{ijl} を利用することで簡明な平均場近似を導くことである.

さて, 次のポアソン分布についての対数尤度関数を考える.

$$\ell(\lambda) = \sum_n \left(u_{nl} \log(\lambda_{nl}) - \lambda_{nl} - \log(u_{nl}!)\right). \tag{2.5}$$

ここですべての λ_{nl} $(n = 1, \ldots, N;\ l = 1, \ldots, L)$ をまとめて λ と置いた. $y_n = 0$ のときサンプル 1 つあたりの尤度は $\log(p(y_n|\lambda_n)) = -\lambda_n$ という簡明な形を取る. このことより, 非ゼロの部分 $\xi_1 = \{n|y_n > 0\}$ と, ゼロの部分

表 1: 疎行列の例. (A) と (B) は同じ情報を持つ. (B) は本研究で扱う疎行列の形式である.

(A)

1	0	2
0	0	2
4	1	0

(B)

r_n	c_n	x_n
1	1	1
3	1	4
3	2	1
1	3	2
2	3	2

表 2: 変分事後分布の一覧

分布族	パラメータ
多項分布 $q(u_{nl}) = \mathrm{Multi}(y_n, \rho_n)$	$\rho_n = (\rho_{n1}, \ldots, \rho_{nL})',\ \ \rho_{nl} = \dfrac{\exp(E_q[\log z_{il}] + E_q[\log w_{jl}])}{\sum_{l=1}^{L} \exp(E_q[\log z_{il}] + E_q[\log w_{jl}])}$
ガンマ分布 $q(z_{il}) = \mathrm{Gamma}(\alpha_{il}^z, \beta_l^z)$	$\alpha_{il}^w = a + \sum_{n \in \{n \mid r_n = i\}} E_q[u_{nl}],\ \ \beta_l^w = b + \sum_j E_q[w_j]$
ガンマ分布 $q(w_{j,l}) = \mathrm{Gamma}(\alpha_{jl}^w, \beta_l^w)$	$\alpha_{jl}^w = a + \sum_{n \in \{n \mid c_n = j\}} E_q[u_{nl}],\ \ \beta_l^w = b + \sum_i E_q[z_i]$

$\xi_0 = \{n | y_n = 0\}$ を分けて考えることで, 式 (2.5) の対数尤度関数は次のように書ける.

$$\ell(\lambda) = \left\{ \sum_{n \in \xi_1} u_{nl} \log(\lambda_{nl}) - \lambda_{nl} - \log(u_{nl}!) \right\} + \left\{ \sum_{n \in \xi_0} \{-\lambda_{nl}\} \right\}. \tag{2.6}$$

疎行列のためのデータ形式は複数あるが, ここでは行列の (i, j) 成分の値 x_{ij} を 3 つ組の変数 $\mathcal{D}_n = (i, j, x_{ij}) = (r_n, c_n, x_n)$ で表すことを考える. ここで N_1 は行列の 0 でない要素の数である. $x_{ij} = 0$ となる (i, j) については省略すると約束し, 非ゼロ要素のみを記録し, $\mathcal{D} = (\mathcal{D}_1, \ldots, \mathcal{D}_{N_1})$ とする. その例を表 1 に示す. 疎行列を表す方法は一意ではなく, この 3 つ組に変数 $\mathcal{D}_n$ で表す形式をまず考えたのは計算上の便宜によるところが大きい. しかし, この形式は十分に実用に耐えるものである. 事実, R 言語の Matrix パッケージで `TsparseMatrix` クラスとして実装されているものと同値であり, また疎行列をテキストファイルに格納する際によく使われる Matrix Market 形式でも採用されている. 加えて, 他の形式の疎行列であってもゼロ要素を省略するという基本的なアイデアは変わらないため, 本報告で考察する内容がわずかな変更で適用できるものと考えられる.

再び式 (2.1) のモデルを考えると, 式 (2.6) の λ_n は内積 $\lambda_n = \sum_l z_{r_n l} \cdot w_{c_n l}$ である. よって式 (2.6) の対数尤度関数は次のように書ける.

$$\ell(\lambda) = \sum_n \ell_n(\lambda) = \left\{ \sum_{n \in \xi_1} u_{nl} \log(z_{r_n l} \cdot w_{c_n l}) - \log(u_{nl}!) \right\} + \left\{ \sum_{i=1}^{R} \sum_{j=1}^{C} \{-z_{il} w_{jl}\} \right\}. \tag{2.7}$$

式 (2.7) の第 1 項に必要なのは非ゼロ要素のみである. 第 2 項は Z, W を所与としたときデータの値に依存しない. 一方で式 (2.7) で表される尤度にガンマ事前分布（式 2.2-2.3）の密度をかけ合わせて考えると, Cemgil (2009) と同様の平均場近似の仮定から変分事後分布が導かれる. 得られる変分事後分布を表 2 に書き下しておく.

変分ベイズ法による NMF のアルゴリズムは u_{ijl} の変分事後分布である多項分布の更新と w_{il} と h_{jl} の変分事後分布であるガンマ分布の更新の繰り返しからなる. よって, この結果をまとめると Algorithm 1 を得る. ここでは変分事後分布による期待値を $E_q[\cdot]$ で表した. また, 変分事後分布のパラメータ $\alpha_{i,l}^z$, β_l^z, $\alpha_{j,l}^w$, β_l^w, $(i = 1, \ldots, R,$ $j = 1, \ldots, C, l = 1, \ldots, L)$ をそれぞれまとめて, A^z, B^z, A^w, B^w と置いた. さらに, 表記の節約のため, 求めたい

Algorithm 1 NMF の変分ベイズ法による推定. データのゼロ要素にアクセスする必要がない.

Require:

Require: 疎行列 $\mathcal{D}$, 次元 L, 事前分布のパラメータ a, b

Ensure: θ ▷ 変分事後分布のパラメータ A^z, B^z, A^w, B^w

  Z, W, $\log Z$, $\log W$ を初期化 ▷ $\log X$ は X の要素ごとの対数とする

  while 収束するまで **do**

    $A^z \leftarrow a$ ▷ 事前分布のパラメータによる初期化

    $A^w \leftarrow a$

    for $n \in \xi_1$ **do** ▷ ξ_1 は非ゼロ要素のみ

      for $l \in 1, \ldots, L$ **do**

        $U_{nl} \leftarrow \dfrac{x_n \exp(E_q[\log z_{r_n l}] + E_q[\log w_{c_n j}])}{\sum_{l=1}^{L} \exp(E_q[\log z_{r_n l}] + E_q[\log w_{c_n l}])}$ ▷ 多項分布の期待値である

        $\alpha^z_{r_n l} \leftarrow \alpha^z_{r_n l} + U_{nl}$ ▷ 十分統計量のインクリメント

        $\alpha^w_{c_n l} \leftarrow \alpha^w_{c_n l} + U_{nl}$

      end for

    end for

    for $l \in 1, \ldots, L$ **do**

      $\beta^z_l \leftarrow a + \sum_i E_q[z_{il}]$

      $\beta^w_l \leftarrow a + \sum_j E_q[w_{jl}]$

    end for

  end while

変分事後分布のパラメータをすべてまとめて表すときは $\theta = (A^z, B^z, A^w, B^w)$ と置いた. Algorithm 1 ではデータのゼロ要素にアクセスする必要がないことに注意してほしい.

次に, Hoffman, et al. (2013) に基づき確率的変分ベイズ法を考える. 狭義の確率的変分ベイズ法はデータセットからサンプルを 1 つずつ抜き出しながら行うものだが, ここでは応用上の重要性からミニバッチ変分ベイズ法を考えることにする. 確率的変分ベイズ法はデータセットからリサンプリングを行う. いま, (2.7) の第 2 項はすべての行と列のインデックスを参照した場合のものであるから, データセットを部分的に抜き出した場合には成り立たないことが問題となる. 一方で, 必要な行と列のインデックスを直接参照するにはゼロ要素にもアクセスすることになり, 疎行列の形式の利点は失われる. そこで行と列のインデックス i, j を確率変数とみなし, 期待値で近似する.

$$\ell_n(\lambda) \approx E_{(i,j)}[\ell_n(\lambda)] = \{y_n \log(z_{r_n l} \cdot w_{c_n l}) - \log(u_{nl}!)\} + E_{(i,j)}[z_{il} \cdot w_{jl}].$$

ここで $E_{(i,j)}$ は行列のインデックスの分布による期待値を表す. 通常, 行列ではすべての行, 列で要素の数が均等である（すなわち, 1 行目に比べ 2 行目が多いというようなことはない）ため, 以降 i, j の分布として離散一様分布の直積を用いることにする. このとき, サイズ N_S の非ゼロのサンプル S を抜き出したときの対数尤度の期待値は次式で与えられる.

$$E_{(i,j)}\left[\sum_{n \in S} \ell_n(\lambda)\right] = \left(\sum_{n \in S} \{u_{nl} \log(z_{r_n l} \cdot w_{c_n l}) - \log(u_{nl}!)\}\right) + (N_S/N_1)\left(\sum_{i=1}^{R}\sum_{j=1}^{C}\{-z_{il} \cdot w_{jl})\}\right).$$

結果, 目指していた Algorithm 2 を得る. Algorithm 2 ではアルゴリズムの現在のステップにおける更新の候補となる変分パラメータは $\tilde{\theta} = (\tilde{A}^z, \tilde{B}^z, \tilde{A}^w, \tilde{B}^w)'$ と表した. Algorithm 2 はミニバッチのサイズ N_S を 1 としたとき狭義の確率的変分ベイズ法となり, $N_S = N_1$ としたとき Algorithm 1 と一致する. 学習率 η_t については, Hoffman, et al. (2013) で推奨されている $\eta_t = (N_S/N_1) \cdot (t + \tau)^{-\kappa}$ を使用する. ここで $\tau \geq 0$ と $\kappa \in [0.5, 1]$ はユーザーが指定するパラメータである.

この提案法についての数値例は紙幅の都合により口頭発表で報告する.

Algorithm 2 NMF の確率的ミニバッチ変分ベイズ法による推定. データのゼロ要素にアクセスする必要がない.

Require: 疎行列 $\mathcal{D}$, 次元 L, 事前分布のパラメータ a, b, 学習率 η_t, 非ゼロ要素のインデックスをランダムに K 個 $S_k(k = 1, \ldots, K)$ に分割し, そのサイズを N_{S_k} とする.

Ensure: θ ▷ 変分事後分布のパラメータである A^z, B^z, A^w, B^w

Z, W, $\log Z$, $\log W$ を初期化 ▷ $\log X$ は X の要素ごとの対数とする

$t \leftarrow 0$

while 収束するまで **do**

$A^z \leftarrow a$ ▷ 事前分布のパラメータによる初期化

$A^w \leftarrow a$

for $k \in 1, \ldots K$ **do**

for $n \in S_k$ **do**

for $l \in 1, \ldots, L$ **do**

$$U_{nl} \leftarrow \frac{x_n \exp(E_q[\log z_{r_n l}] + E_q \log w_{c_n j})}{\sum_{l=1}^{L} \exp(E_q[\log z_{r_n l}] + E_q[\log w_{c_n l}])}$$ ▷ 多項分布の期待値である

$\tilde{\alpha}^z_{r_n l} \leftarrow \alpha^z_{r_n l} + U_{nl}$ ▷ 十分統計量のインクリメント

$\tilde{\alpha}^w_{c_n l} \leftarrow \alpha^w_{c_n l} + U_{nl}$

end for

end for

for $l \in 1, \ldots, L$ **do**

$\tilde{\beta}^z_l \leftarrow a + (N_{S_k}/N_1)(\sum_j E_q[w_{jl}])$ ▷ Algorithm 1 からの主要な変更点

$\tilde{\beta}^w_l \leftarrow a + (N_{S_k}/N_1)(\sum_i E_q[z_{il}])$

end for

end for

$\theta \leftarrow (1 - \eta_t)\theta + \eta_t \tilde{\theta}$

$t \leftarrow t + 1$

end while

3 議論とまとめ

本研究は行列因子分解以外の手法, 例を上げると回帰型のモデルや混合分布モデルにもわずかな変更で応用できる可能性がある. 特に Abe & Shimamura (2023) で提案されたモデルはデータの文脈と形式を分離したことにより, 変数の次元の増減にも対応でき, 階層モデル（ここでは『マルチレベルモデル』や『混合効果モデル』と同じ意味で用いた）に相当する分析を自由に行えるという利点があるため, これに応用することは重要と考える.

参考文献

[1] Abe, H., & Yadohisa, H. (2017). A non-negative matrix factorization model based on the zero-inflated Tweedie distribution. Computational Statistics, 32(2), 475-499.

[2] Hyunsoo, K. & Haesun, P. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis, Bioinformatics, 23 (12), 1495–1502, https://doi.org/10.1093/bioinformatics/btm134

[3] Cemgil, A. T.(2009). Bayesian Inference for Nonnegative Matrix Factorization Models, Computational Intelligence and Neuroscience, 785152, 17 https://doi.org/10.1155/2009/785152

[4] Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. Journal of Machine Learning Research.

[5] Abe, K. & Shimamura, T. (2023). UNMF: a unified nonnegative matrix factorization for multi-dimensional omics data, Briefings in Bioinformatics, 24(5), bbad253, https://doi.org/10.1093/bib/bbad253