

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

阿部興¹ (発表者) ・ 島村徹平¹

2024 年 10 月 25 日

¹東京科学大学 総合研究院 難治疾患研究所

動機: 精神面

次元の呪い \longrightarrow 次元の祝い²

○ 高次元の積分は難しい

△ 高次元のデータはややこしい

◎ 見えるものが多いとうれしい
たとえそのほとんどが “0” だったとしても

疎 (sparse) なデータ \neq 汚いデータ

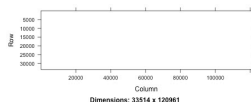
²Gelman, A. (2004). “The blessing of dimensionality”
https://statmodeling.stat.columbia.edu/2004/10/27/the_blessing_of/

動機: 分析対象

Bischoff et al. (2021)³ の肺がんに関する単一細胞 RNA 発現量データ

- 行 (遺伝子) : 33, 514
- 列 (細胞) : 120,961
- 非ゼロ要素: 239,634,370 (全体の 5%程度)

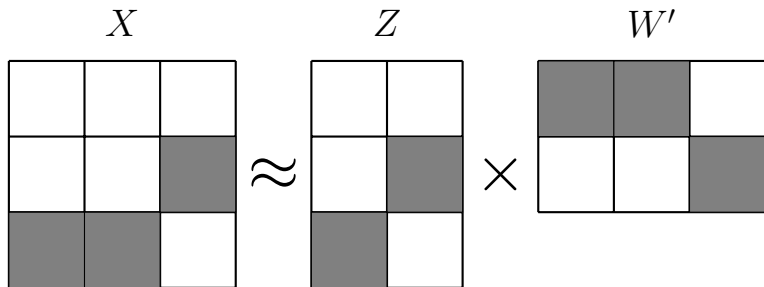
単一細胞解析の分野では, この研究のデータが特に大きいわけではない



³Bischoff, P., Trinks, A., Obermayer, B. et al. Single-cell RNA sequencing reveals distinct tumor microenvironmental patterns in lung adenocarcinoma. *Oncogene* 40, 6748 – 6758 (2021). <https://doi.org/10.1038/s41388-021-02054-3>

非負行列因子分解 (NMF)

- 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
- 非負制約により解釈がしやすい



$$P(I = i, J = j) = \sum_l \underbrace{P(I = i | L = l)}_Z \underbrace{P(J = j | L = l)}_W P(L = l)$$

疎行列と NMF

- 観測のゼロ過剰や過分散をモデル化したケース⁴
- 分解で得られる行列を疎にしようとした議論⁵

本研究: 疎であることを積極的に利用して計算効率を高める

⁴e.g. Abe, H., & Yadohisa, H. (2017). A non-negative matrix factorization model based on the zero-inflated Tweedie distribution. *Computational Statistics*, 32(2), 475-499.

Gouvert, O., Oberlin, T., & Févotte, C. (2020). Negative binomial matrix factorization. *IEEE Signal Processing Letters*, 27, 815-819.

⁵e.g. 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.

Kim, H., & Park, H. (2007). Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12), 1495-1502.

疎行列の形式

(i, j) 成分の値 x_{ij} を $\mathcal{D}_n = (i, j, x_{ij}) = (r_n, c_n, x_n)$ で表す.
 $x_{ij} = 0$ となる (i, j) は省略し, $\mathcal{D} = (\mathcal{D}_1, \dots, \mathcal{D}_{N_1})$ とする.

Tab: A と B は同じ情報を持つ

Tab: A

1	0	2
0	0	2
4	1	0

Tab: B

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

対数尤度の評価

指数型分布族:

$$p(x|\theta) = \exp(T(x)'\eta(\theta))$$

$T(x) = (T_1(x), T_0)'$, $\eta(\theta) = (\eta_1(\theta), \eta_0(\theta))'$ とする.

$T_1(x)$: x に依存する. T_0 : x に依存しない.

$T_1(0) = 0$ のとき,

$$\begin{aligned} \sum_i \log p(x_i|\theta) \\ = \left(\sum_{i \in \text{nonzero part}} T_1(x)' \eta_1(\theta_i) \right) + \left(\sum_{j \in \text{all of the data}} T_0 \eta_0(\theta_j) \right) \end{aligned}$$

例: ベルヌーイ分布

$$T_1(x) = (x), \quad T_0(x) = (1), \quad \eta_1(\theta) = (\log(\theta/(1-\theta))), \quad \eta_0(\theta) = (\log(1-\theta))$$

モデル

これよりポアソン分布の行列分解モデルを考える

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

事前分布:

$$z_{il} \sim \text{Gamma}(a, b), \quad w_{jl} \sim \text{Gamma}(a, b).$$

Note: x_{ij} は次と同値

$$x_{ij} \mid z, w \sim \text{Pois} \left(\sum_l z_{il} w_{lj} \right),$$

モデルの対数尤度

$$\ell(Z, W) = \left\{ \sum_{n=1}^{N_1} \boxed{u_{nl} \log(z_{r_{nl}} \cdot w_{c_{nl}})} - \log(u_{nl}!) \right\} \\ + \left\{ \sum_{i=1}^R \sum_{j=1}^M \{ - \boxed{z_{il} \cdot w_{jl}} \} \right\}.$$

第1項: ゼロ要素にアクセスしていないことに注意

第2項: Z, W を所与としたときデータの値に依存しない

参考: ガンマ分布

$$\log(\text{Gamma}(z|a, b)) = \underline{(a-1) \log(z)} - \underline{bz} + \log(b^a / \Gamma(a)).$$

擬似コード: 形状パラメータの更新

```
1: Function updateA( $\mathcal{D}$ ,  $Z$ ,  $W$ ,  $a$ )
2:  $\alpha_{il}^z \leftarrow a$ ;  $\alpha_{jl}^w \leftarrow a$  ( $i = 1, \dots, R$ ,  $j = 1, \dots, C$ )
3: for  $n \in \{1, \dots, N_1\}$  do
4:   for  $l \in \{1, \dots, L\}$  do
5:      $U_{nl} \leftarrow \frac{x_n \exp(E_q[\log z_{r_{nl}}] + E_q[\log w_{c_{nj}}])}{\sum_{l=1}^L \exp(E_q[\log z_{r_{nl}}] + E_q[\log w_{c_{nl}}])}$ 
6:      $\tilde{\alpha}_{r_{nl}}^z \leftarrow \alpha_{r_{nl}}^z + U_{nl}$ 
7:      $\tilde{\alpha}_{c_{nl}}^w \leftarrow \alpha_{c_{nl}}^w + U_{nl}$ 
8:   end for
9: end for
10: Return  $\alpha_{il}^z$ ,  $\alpha_{jl}^w$  ( $i = 1, \dots, R$ ,  $j = 1, \dots, C$ )
11: EndFunction
```

3 行目: ゼロ要素にアクセスしていないことに注意

5 行目: 和で条件付けたポアソン分布は多項分布

擬似コード: レートパラメータの更新

```
1: Function updateB( $Z, W, b$ )  
2: for  $l \in 1, \dots, L$  do  
3:    $\tilde{\beta}_l^z \leftarrow b + (\sum_j E_q[w_{jl}])$   
4:    $\tilde{\beta}_l^w \leftarrow b + (\sum_i E_q[z_{il}])$   
5: end for  
6: Return  $\beta_l^z, \beta_l^w$  ( $l = 1, \dots, L$ )  
7: EndFunction
```

1 行目: \mathcal{D} に依存しないことに注意

▶ updateA と updateB を収束するまで繰り返す

数値例: 設定

データ: 非ゼロ要素については「ポアソン乱数 +1」としてランダム行列を作成

- 非ゼロ要素の割合: 0.1, 0.2, 0.3, 0.4, 0.5
- 行 R : 100, 1000, 5000
- 列 C : 2000
- 分解のランク L : 2, 5, 10
- アルゴリズムのイテレーション: 100 回 (固定)

各 10 回繰り返した

計算量は,

- 通常の NMF のアルゴリズム: $R \cdot C \cdot L$ に比例
- 提案法: (非ゼロ要素の数) $\cdot L = R \cdot C \cdot (\text{非ゼロ要素の割合}) \cdot L$ に比例

数値例: 結果

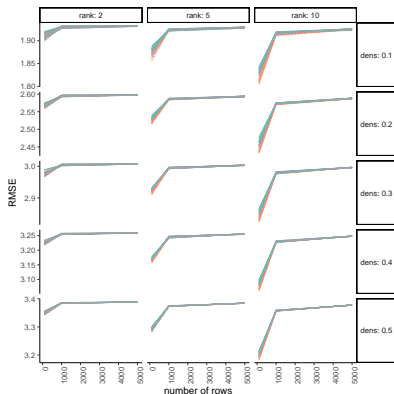


Fig: Root mean squared error: X と推定された ZW の平均 2 乗誤差

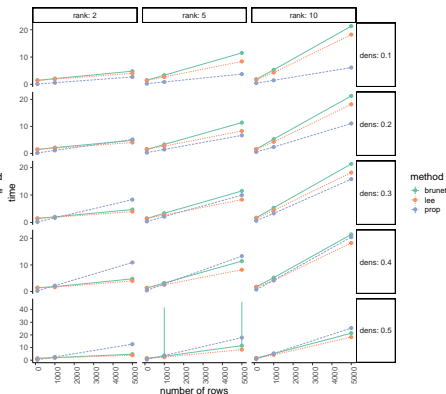


Fig: 計算時間 (秒): 点は中央値, エラーバーは 80% 区間

確率的変分ベイズ法

方針: 毎回すべてのデータセットを使うのではなく, リサンプルして更新を繰り返す

サンプル 1 つあたりの尤度:

$$\ell_n(Z, W) \approx E_{(i,j)}[\ell_n(Z, W)]$$

(i, j) が独立な離散一様分布とすると,

$$\begin{aligned} \ell_n(Z, W) \approx & \{u_{nl} \log(z_{r_{nl}} \cdot w_{c_{nl}}) - \log(u_{nl}!)\} \\ & + \frac{1}{RC} \left\{ \left(\sum_{i=1}^R z_{il} \right) \cdot \left(\sum_{j=1}^C w_{jl} \right) \right\}. \end{aligned}$$

※サイズ S のミニバッチを取るとき第 2 項は S 倍

欠損値のある場合

欠損の有無が行列の値と独立な場合, 欠損の部分の尤度は 1

サンプル 1 つあたりの尤度:

$$E_{(i,j)}[\ell_n(Z, W)] \approx \{u_{nl} \log(z_{rnl} \cdot w_{cnl}) - \log(u_{nl}!)\} \\ + \left\{ \left(\sum_{i=1}^R p(i) z_{il} \right) \cdot \left(\sum_{j=1}^C p(j) w_{jl} \right) \right\}.$$

$p(i)$, $p(j)$ は経験分布を用いる. 独立性は仮定.

擬似コード: 確率的変分ベイズ法

レートパラメータを更新する関数のみ変更:

Function updateB_s($Z, W, b, S_k/RC$)

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

$$\tilde{\beta}_l^z \leftarrow b + (S_k/RC)(\sum_j E_q[w_{jl}])$$

$$\tilde{\beta}_l^w \leftarrow b + (S_k/RC)(\sum_i E_q[z_{il}])$$

end for

return $\tilde{\beta}_l^z, \tilde{\beta}_l^w$ ($l = 1, \dots, L$)

EndFunction

\mathcal{D} をサイズ S_k のミニバッチ $\mathcal{D}^{(k)}$ に分割 ($k = 1, \dots, K$)

- $\mathcal{D}^{(k)}$ に対して updateA と updateB_s により $\tilde{\theta}$ を得る
- $\theta \leftarrow (1 - \eta_t)\theta + \eta_t \tilde{\theta}$

変分パラメータ ($\alpha_{il}^z, \beta_l^z, \alpha_{jl}^w, \beta_l^w$) をまとめて θ と書いた.

学習率: $\eta_t = (N_S/N_1) \cdot (t + \tau)^{-\kappa}$, ($\tau \geq 0, \kappa \in [0.5, 1]$)

データ分析: Bischoff et al. (2021)

肺がんに関する単一細胞 RNA 発現量データ (再掲)

- 行 (遺伝子) : 33, 514
- 列 (細胞) : 120,961
- 非ゼロ要素: 239,634,370 (全体の 5%程度)

```
R> res <- NMF::nmf(as.matrix(MM), rank = 2)
Error in h(simpleError(msg, call)) :
  error in evaluating the argument 'x' in selecting a method for function 'nmf':
  vector memory limit of 24.0 Gb reached, see mem.maxVSize()
In addition: Warning message:
In asMethod(object) :
  sparse->dense coercion: allocating vector of size 30.2 GiB
```

すべてをメモリ上に展開するのは無理がある

細胞（列）の特徴量 W をプロット

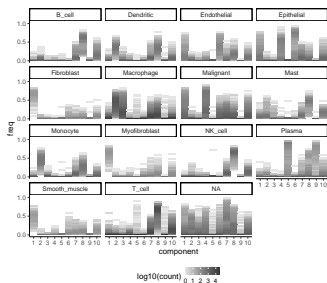


Fig: NMF

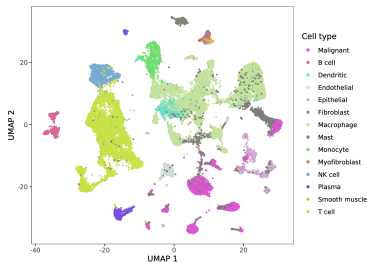
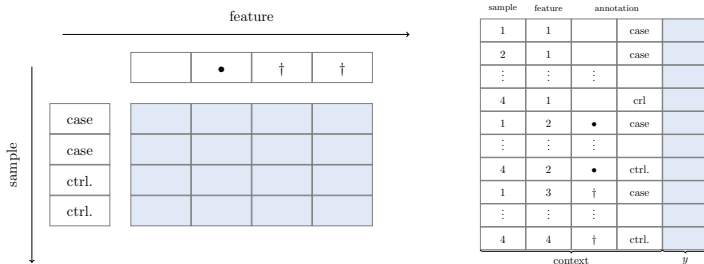


Fig: UMAP

拡張: Blessing of dimensionality へ向けて

Abe & Shimamura (2023) のモデル⁶を次のように表記すると本報告との関係がわかる:

$$y_{ij} = \sum_{l=1} u_{nl}, \quad u_{nl} \sim \text{Pois} \left(\sum_l \prod_{d=1}^D v_{x_{nd}, l}^{(d)} \right)$$



⁶Abe, 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>

まとめと議論

- 疎行列に適した非負値行列因子分解のアルゴリズムを提案した
- 疎であるほど計算量やメモリ効率の点で有利
- わずかな変更で応用の可能性（一般化線形モデル, 混合分布, ...）

本報告の実装: <https://github.com/abikoushi/VBsNMF>