matrix factorization

神戸大学理学系研究科数学専攻 修士 2 年 高須航平

matrix factorization とは $M \times N$ 行列 R を $M \times K$ 行列 P と $K \times N$ 行列 Q の積に近似する機械学習の手法の一つ. R の良い近似 $PQ = \hat{R}$ を求め, R に空白の成分があればそれを予想し代入することを目標とする.

(1) 最尤法による推定

最尤法とは観測値からその値が最も発生しやすい (最尤な) パラメータを求める推定法である.

まず初期値 P, Q をランダムに決定し, $P = \{p_{ij}\}, Q = \{q_{ij}\}$ とする. $R = \{r_{ij}\}$ と $QP = \{\hat{r}_{ij}\} = \{\sum_{k=1}^{K} p_{ik} q_{kj}\}$ の (i, j) 成分の二乗誤差 $\mathrm{error}_{ij}^2 = (r_{ij} - \hat{r}_{ij})^2$ を各 p_{ik}, q_{kj} で微分すると $\frac{\partial}{\partial p_{ik}} = -2\mathrm{error}_{ij} q_{kj}$, $\frac{\partial}{\partial q_{kj}} = -2\mathrm{error}_{ij} p_{ik}$ となる. すべての $1 \leq i \leq M$, $1 \leq j \leq N$, $1 \leq k \leq K$ について,

$$p'_{ik} = p_{ik} - \frac{\partial}{\partial p_{ik}} = p_{ik} + 2\mathrm{error}_{ij} q_{kj}$$
$$q'_{kj} = q_{kj} - \frac{\partial}{\partial q_{kj}} = q_{kj} + 2\mathrm{error}_{ij} p_{ik}$$

と成分を書き換えていく. 但し r_{ij} が不明値ならその i, j について更新は行わない. この更新によって得られた新たな行列を再び P, Q とおけば誤差の減少した $\hat{R} = PQ$ が得られる. この更新を十分な回数繰り返せば R の近似値 P, Q を求められる.

Example 2.1

以下の R に最尤法を適用する. ただし, ?になっている成分は不明値として扱う.

$$R = \begin{bmatrix} 5 & 3 & ? & 1 \\ 4 & ? & ? & 1 \\ 1 & 1 & ? & 5 \\ 1 & ? & ? & 4 \\ ? & ? & 5 & 4 \end{bmatrix} \approx \begin{bmatrix} -0.020361 & 2.553650 \\ 0.064687 & 2.061994 \\ 1.938369 & 0.985131 \\ 1.533984 & 0.881417 \\ 1.517997 & 0.987769 \end{bmatrix} \begin{bmatrix} -0.473544 & -0.089867 & 1.634774 & 2.370825 \\ 1.954209 & 1.173456 & 2.542364 & 0.410368 \end{bmatrix}$$
$$= \begin{bmatrix} 5.000006 & 2.998425 & 6.459020 & 0.999662 \\ 3.998934 & 2.413846 & 5.348087 & 0.999537 \\ 1.007248 & 0.981813 & 5.673354 & 4.999798 \\ 0.996064 & 0.896451 & 4.748601 & 3.998514 \\ 1.211469 & 1.022687 & 4.992850 & 4.004253 \end{bmatrix}$$

観測値と比べても良い近似が求まるが, matrix factorization では初期値 P_0, Q_0 によって得られる値が変化してしまう. これは誤差関数 (ここでは $\mathrm{error} = \sum_{i=1}^{M} \sum_{j=1}^{N} \mathrm{error}_{ij}^2$) が多峰型であれば局所的な最大値に近似していくためである. そこで多峰型の誤差関数のサンプリングが可能な MCMC 法を利用する.

(2)MCMC 法

1. 初期値 P, Q をランダムに決定.
2. P, Q の成分を一つ選びランダムに変化させる. ただし, 発生させる成分の乱数の最大値, 最小値, 精度はあらかじめ決めておく. この新たな行列を P', Q' とする.
3. $\frac{f(P'Q')}{f(PQ)}$ を r に代入. 但し, $f(PQ) = \exp(-1 * \theta * \mathrm{error}^4)$, θ : 正の実数, $\mathrm{error} = \sum_{i=1}^{M} \sum_{j=1}^{N} \mathrm{error}_{ij}^2$

4. $0 \leq R < 1$ の一様乱数 R に対し,
   - $r > R \Rightarrow P, Q$ に P', Q' を代入.
   - $r \leq R \Rightarrow P, Q$ は変化させない.
5. P, Q をサンプルとして出力. 試行回数が $STEP$ 未満なら 2 に行く.
6. 出力された回数の多い P, Q の組が周囲に比べて誤差の少ない行列となる.

MCMC 法による matrix factorization は複数の局所的最大値を求めることができるが, 最尤法よりも精度が悪くなる. そこで MCMC 法で得た結果を元に最尤法で精度の高い値を求めることになる.

Example 2.2

Example 2.1 と同じ R を用いる. 試行回数 $STEP$ を 1,000,000 回, $\theta = 10^{-7}$, P, Q の成分は $-1 \leq a \leq 3$, 小数点第 2 位以下切り捨てとして MCMC 法を行った. このとき P, Q として複数の候補が挙げられた. この中から 2 つ例を挙げると,

$$\begin{bmatrix} -0.9 & 2.3 \\ 1.2 & 1.5 \\ -0.8 & 1.7 \\ 1.2 & -0.3 \\ 1.1 & 0.7 \end{bmatrix} \begin{bmatrix} 0.8 & 1.0 & 1.1 & 2.4 \\ 2.5 & 2.2 & 2.9 & 2.2 \end{bmatrix} = \begin{bmatrix} 5.03 & 4.16 & 5.68 & 2.90 \\ 4.71 & 4.50 & 5.67 & 6.18 \\ 3.61 & 2.94 & 4.05 & 1.82 \\ 0.21 & 0.54 & 0.45 & 2.22 \\ 2.63 & 2.64 & 3.24 & 4.18 \end{bmatrix}$$

$$\begin{bmatrix} 0.0 & 2.2 \\ 0.8 & -0.9 \\ 1.3 & -0.2 \\ 2.9 & 0.6 \\ 1.7 & -0.7 \end{bmatrix} \begin{bmatrix} 0.4 & -0.6 & 1.0 & 1.7 \\ 1.6 & -0.3 & 0.0 & -0.6 \end{bmatrix} = \begin{bmatrix} 3.52 & -0.66 & 0.00 & -1.32 \\ -1.12 & -0.21 & 0.80 & 1.90 \\ 0.20 & -0.72 & 1.30 & 2.33 \\ 2.12 & -1.92 & 2.90 & 4.57 \\ -0.44 & -0.81 & 1.70 & 3.31 \end{bmatrix}$$

である.

得られた P, Q それぞれに対して最尤法を用いると,

$$\begin{bmatrix} -1.178534 & 2.306891 \\ -0.865868 & 1.851985 \\ 1.611216 & 0.616871 \\ 1.226469 & 0.584429 \\ 1.181560 & 0.649782 \end{bmatrix} \begin{bmatrix} -0.174907 & 0.102575 & 2.155210 & 2.456764 \\ 2.078067 & 1.352846 & 3.775740 & 1.688585 \end{bmatrix}$$

$$= \begin{bmatrix} 5.000010 & 2.999981 & 6.170232 & 1.000001 \\ 3.999996 & 2.416635 & 5.126486 & 1.000000 \\ 1.000086 & 0.999802 & 5.801652 & 5.000015 \\ 0.999964 & 0.916448 & 4.849950 & 4.000003 \\ 1.143628 & 1.000254 & 4.999919 & 4.000026 \end{bmatrix}$$

$$\begin{bmatrix} 0.762553 & 2.284638 \\ 0.717601 & 1.788431 \\ 2.733931 & -0.485778 \\ 2.196135 & -0.289381 \\ 2.202557 & -0.218494 \end{bmatrix} \begin{bmatrix} 0.712393 & 0.565554 & 2.229127 & 1.799896 \\ 1.950752 & 1.124351 & -0.412924 & -0.163052 \end{bmatrix}$$

$$= \begin{bmatrix} 5.000000 & 3.000000 & 0.756443 & 1.000000 \\ 4.000000 & 2.416667 & 0.861137 & 1.000000 \\ 1.000000 & 1.000000 & 6.294869 & 5.000000 \\ 1.000000 & 0.916667 & 5.014956 & 4.000000 \\ 1.142857 & 1.000000 & 5.000000 & 4.000000 \end{bmatrix}$$

という行列を得た. 元々の P, Q の形を残しつつも error の値が小さい行列が求まった.