クラスタリングと非負行列の因数分解

情報科学類 3 年 江畑 拓哉 (201611350)

Outline

ある行列 $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ が与えられ、非負の要素を持ように制限された k ランクの近似を行いたいと考えた時、言い換えれば $\boldsymbol{W} \in \mathbb{R}^{m \times k}$ と $\boldsymbol{H} \in \mathbb{R}^{k \times n}$ を仮定して以下の式を解きたい場合について考える。

$$\min_{\boldsymbol{W} \geq 0, \boldsymbol{H} \geq 0} ||\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}||_F$$

$$where\ ||\boldsymbol{A}||_F = \sqrt{\Sigma_i \Sigma_j |a_{ij}|^2}\ means\ Frobenius\ norm \tag{9.3}$$

同時に $\boldsymbol{W}$ と $\boldsymbol{H}$ を最適化しようとした時、この問題は非線形になる。

しかし、行列の一つが既にわかっている場合、例えば $\boldsymbol{W}$ がわかっている場合で言えば、この $\boldsymbol{H}$ を求める問題は、非負の制限がついた右側の行列についての最小二乗問題であると言える。

従って元の問題に対する最も一般的な解決法は、交互最小二乗法 (ALS) を使うことである。

Alternating nonnegative least squares のアルゴリズム

Alternating nonnegative least squares algorithm

1. 初期値 $\boldsymbol{W}^{(1)}$ を与える。
2. $k = 1, 2 \ldots$ と収束するまで繰り返す。
   - $\min_{H \geq 0} ||\boldsymbol{A} - \boldsymbol{W}^{(k)}\boldsymbol{H}||_F$ を解き、 $\boldsymbol{H}^{(k)}$ を得る
   - $\min_{W \geq 0} ||\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}^{(k)}||_F$ を解き、 $\boldsymbol{W}^{(k+1)}$

しかし、近似 $\boldsymbol{WH}$ は単一のものではない。ここで正の対角要素を持つ任意の対角行列 $\boldsymbol{D}$ とその逆行列を要素間に適用しすることが出来る。

$$\boldsymbol{WH} = (\boldsymbol{WD})(\boldsymbol{D}^{-1}\boldsymbol{H})$$

ある要素の増大と他の要素の減衰を防ぐために、毎反復ごとにそれらの一つを正規化する必要がある。一般的な正規化は、 $\boldsymbol{W}$ の各列の最大要素が 1 になるように $\boldsymbol{W}$ の列をスケーリングすることである。

$\boldsymbol{A}$ と $\boldsymbol{H}$ の列要素として、 $\boldsymbol{a_j}$ と $\boldsymbol{h_j}$ を置く。各行の要素を一つずつ書き下すと、この最小二乗問題は以下の n 個の独立したベクトルの最小二乗問題と等しいものとみなすことが出来る。

$$\min_{h_j \geq 0} ||\boldsymbol{a}_j - \boldsymbol{W}^{(k)}\boldsymbol{h}_j||_2 \ \textit{where}\ j = 1, 2, \ldots, n$$

これらは 23 章で紹介されるアクティブセットアルゴリズムで解決される。この行列を転置することで、 $\boldsymbol{W}$ を決定するの最小二乗問題は、独立な m このベクトルの最小二乗問題に変換される。即ち ALS アルゴリズムのコア部分は擬似的な MATLAB コードで以下のように表すことが出来る。

疑似 MATLAB コード

```
while (not converged)
    [W] = normalize(W);
    for i = 1:n
        H(:,i) = lsqnonneg(W, A(:,i));
    end
    for i = 1:m
        w = lsqnonneg(W, A(:, i));
        W(i,:) = w';
    end
end
```

非負値行列因子分解のためのアルゴリズムは多くの種類がある。前頁のアルゴリズムは、非負な最小二乗法のためのアクティブセット法にかなり時間がかかってしまうという欠点がある。より簡易な代替手段として、部分 QR 分解 $\boldsymbol{W} = \boldsymbol{QR}$ を用いることで非負という制約がない最小二乗解を得ることが出来る。そして $\boldsymbol{H}$ におけるすべての負の要素はゼロと等しいとみなすことが出来る。これは $\boldsymbol{W}$ の計算においても同様の議論をすることが出来る。

$$\boldsymbol{H} = \boldsymbol{R}^{-1}\boldsymbol{Q}^{T}\boldsymbol{A}$$

疑似 MATLAB コード (上の例に基づけば $\boldsymbol{V} = \boldsymbol{A}$)

```
while (not converged)
    W = W.*(W >= 0);
    H = H.*(W'*V)./((W'*W)*H+epsilon);
    H = H.*(H>=0);
    W = W.*(V*H')./(W*(H*H')+epsilon);
    [W,H] = normalize(W, H);
end
```

epsilon は極小の値であり、これはゼロ除算を避けるために用いられている。 $.*$ や $./$ で表される行列操作はそれぞれの構成要素についての演算で、

$$\boldsymbol{H}_{ij} := \boldsymbol{H}_{ij} \frac{(\boldsymbol{W}^T \boldsymbol{A})_{ij}}{(\boldsymbol{W}^T \boldsymbol{W} \boldsymbol{H})_{ij} + \epsilon}, \ \boldsymbol{W}_{ij} := \boldsymbol{W}_{ij} \frac{(\boldsymbol{A} \boldsymbol{H}^T)_{ij}}{(\boldsymbol{W} \boldsymbol{H} \boldsymbol{H}^T)_{ij} + \epsilon}$$

尚このアルゴリズムは勾配降下法と考えることが出来る。

非負値行列分解には非常に多くの重要な用途があるため、これのアルゴリズム開発は活発に行われている。例えば、反復法を用いた終了基準を見つける問題は未だ良い解決策を見つけたとは言い難い。

非負値行列分解 $\boldsymbol{A} \approx \boldsymbol{WH}$ はクラスタリングにも用いられている。各データを表すベクトル $\boldsymbol{a}_j$ は、もし $\boldsymbol{h}_{ij}$ が $\boldsymbol{H}$ の j 列の最大の要素であるなら、それはクラスタ i に割り当てられる。

さらにこの分解法は以下のような分野でも用いられている。

- 文書分類
- 電子メールの監視
- 音楽の記譜 (楽譜にすること)
- バイオインフォマティクス
- スペクトル分析

初期化

非負値行列分解のアルゴリズムにはいくつかの問題がある。それは全体での最適解が求まる保証がないということである。このアルゴリズムではしばしば収束が遅いことや順最適解 (厳密解ではない) になってしまうことがある。良好な初期の近似を計算するための効率的な手法として、 A の SVD に基づいて行うというものがある。最初の k 個の特異の三つの組 $(\boldsymbol{\sigma}_i, \boldsymbol{u}_i, \boldsymbol{v}_i)_{i=1}^{k}$ は、フロベニウスノルムにおいて A の最適なランク k の近似を与える。

もし A が非負な行列であったならば、 $\boldsymbol{u}_i$ や $\boldsymbol{v}_1$ が非負であることは明らか。(Section 6.4)

つまりもし $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ が $\boldsymbol{A}$ の SVD であるならば、特異ベクトル $\boldsymbol{u}_1$ を $\boldsymbol{W}^{(1)}$ の最初の列であるとすることが出来る。 (同様に以降のアルゴリズムのため、 $\boldsymbol{v}_1^T$ を初期近似 $\boldsymbol{H}^{(1)}$ の 第 1 行 であるする。)

次の最良なベクトル $\boldsymbol{u}_2$ は直交性が満たされているために負の成分を有する可能性が非常に高い。しかし行列 $\boldsymbol{C}^{(2)} = \boldsymbol{u}_2\boldsymbol{v}_2^T$ を計算しすべての負の成分をゼロにすることで非負な行列 $\boldsymbol{C}_+^{(2)}$ を得ると、この行列の最初の特異ベクトルは非負であることがわかる。さらにそれは、これが $\boldsymbol{u}_2$ の合理的で良い近似であると考えることが出来るので、 $\boldsymbol{W}^{(1)}$ の第 2 列として取り上げることが出来る。

前頁の手続きを MATLAB を使って簡潔に書き下すと以下のようになる。

* $[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}] = svds(A, k)$ は Lanczos 法を用いることで、k 個の最大特異値及び対応する特異ベクトルとを計算する。標準な SVD 関数である $svd(A)$ はすべての分解を計算するがこれはかなり遅く、特に行列が大きな疎行列のときはより遅くなる。

MATLAB

```
[U,S,V] = svds (A, k) % Compute only the k largest singular
W (:,1) = U (:,1);    % values and the corresponding vectors
for j = 2:k
    C = U (:,j)*V (:,j)';
    C = C .* (C>=0);
    [u, s, v] = svds (C, 1);
    W (:,j) = u;
end
```

例 9.4

ランク 2 の行列 $\boldsymbol{A}$ の非負値分解の例を図 9.3 に示す。ここでは初期化をランダムな値で行ったものと、 SVD ベースで行ったものを比較している。ランダムな値で初期化したものは収束がより遅くなており、10 回反復させても収束したとは言い難い。これに対して SVD ベースで初期化したものの相対近似誤差は 0.574 であることがわかる。 (尚 k-means 法においてこの誤差は 0.596 であった) 更にいくつかのケースでランダムに初期化したものは最適でない準最適な値に収束してしまった。

図 9.3

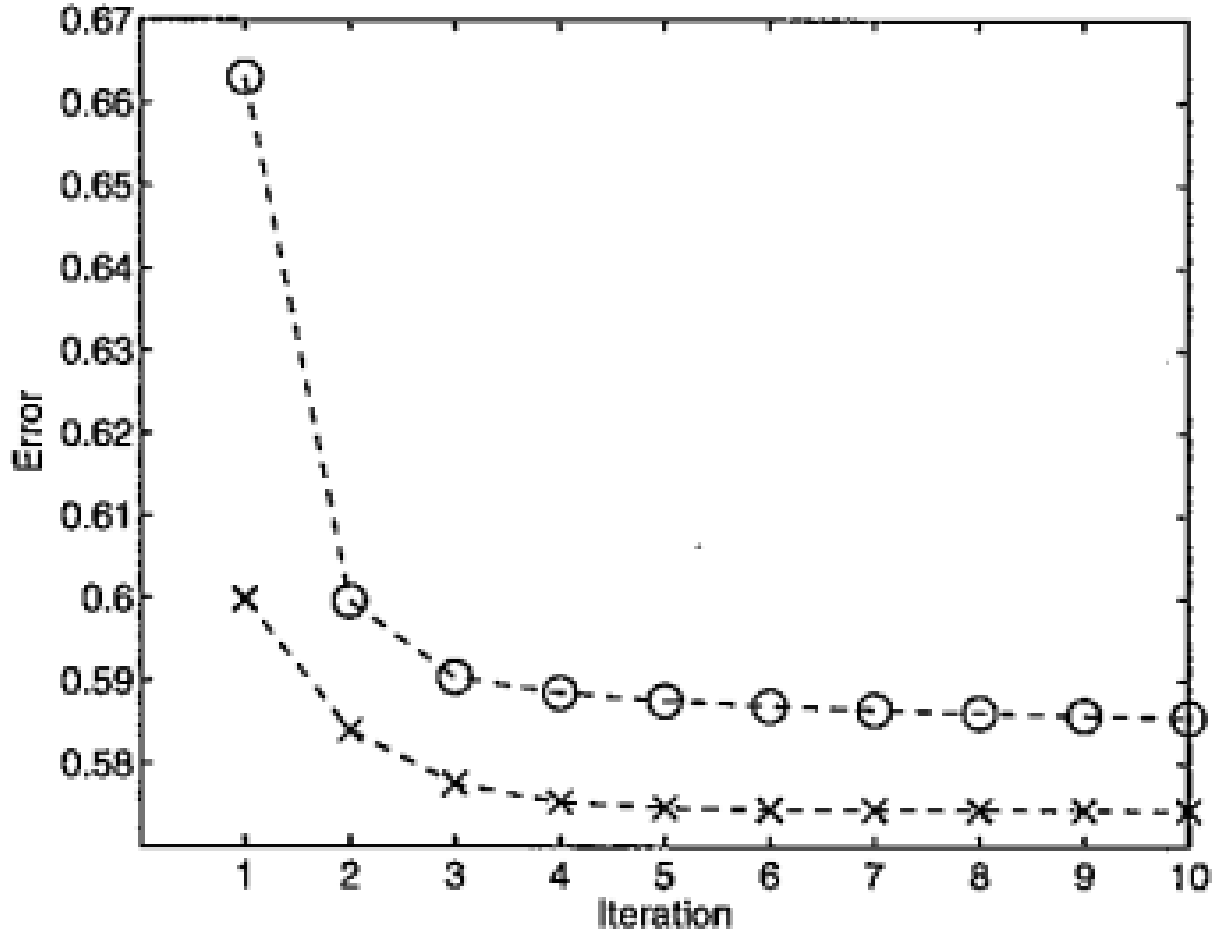

Figure: 9.3 反復回数を横軸とした相対近似誤差のグラフ。上のカーブは初期化をランダムに行ったもの、下のカーブは SVD ベースで初期化を行ったものである。

SVD ベースの初期化をするこの分解は具体的に以下のようになった。

$$\boldsymbol{WH} = \begin{pmatrix} 0.3450 & 0 \\ 0.1986 & 0 \\ 0.1986 & 0 \\ 0.6039 & 0.1838 \\ 0.2928 & 0 \\ 0 & 0.5854 \\ 1.0000 & 0.0141 \\ 0.0653 & 1.0000 \\ 0.8919 & 0.0604 \\ 0.0653 & 1.0000 \end{pmatrix} \begin{pmatrix} 0.7740 & 0 & 0.9687 & 0.9120 & 0.5251 \\ 0 & 1.0863 & 0.8214 & 0 & 0 \end{pmatrix}$$

前頁のそれは分解の処理を打ち切ることが出来る。最初の四つの文書は基底ベクトルによって表されており、これは Google-related keywords のための大きな要素を持っている。これに対して最後の文書は 1 つめの基底ベクトルによって表されているが、この座標値は先述の四つの文書に比べて小さくなっている。

この手法では、ランク 2 の近似は Google-related contents を強調するが、 "football-document" は強調しない。

11 章では、他の低ランクの近似について学ぶが、（例えば SVD をベースとしたもの）これらも類似の効果を確認することが出来る。

対して、ランク 3 の近似を計算した時には以下の値を得ることが出来る。

$\boldsymbol{W}$ の三番目のベクトルは、本質的に "football" についての基底であり、その一方で他の 2 つのベクトルは Google-related document を示している基底である。

$$\boldsymbol{WH} = \begin{pmatrix} 0.2516 & 0 & 0.1633 \\ 0 & 0 & 0.7942 \\ 0 & 0 & 0.7942 \\ 0.6924 & 0.1298 & 0 \\ 0.3786 & 0 & 0 \\ 0 & 0.5806 & 0 \\ 1.0000 & 0 & 0.0444 \\ 0.0589 & 1.0000 & 0.0007 \\ 0.4237 & 0.1809 & 1.0000 \\ 0.0589 & 1.0000 & 0.0007 \end{pmatrix} \begin{pmatrix} 1.1023 & 0 & 1.0244 & 0.8045 & 0 \\ 0 & 1.0815 & 0.8315 & 0 & 0 \\ 0 & 0 & 0.1600 & 0.3422 & 1.1271 \end{pmatrix}$$