THE UNIVERSITY OF TEXAS AT AUSTIN
Department of Statistics and Data Sciences
College of Natural Sciences

Stat models for Big Data

Topic models and NMF

Purnamrita Sarkar
Department of Statistics and Data Science
The University of Texas at Austin

`https://psarkar.github.io/teaching`

Matrix factorization : non-negative matrix factorization angle

- So, SVD returns directions or principal components
- But these are not interpretable.
- But what if we optimized the following?

$$\min_{\substack{U \in \mathbb{R}^{+}_{m\times k} \\ V \in \mathbb{R}^{+}_{n\times k}}} \|A - UV^T\|_F^2$$

Matrix factorization : non-negative matrix factorization angle

- So, SVD returns directions or principal components
- But these are not interpretable.
- But what if we optimized the following?

$$\min_{\substack{U \in \mathbb{R}^{+}_{m \times k} \\ V \in \mathbb{R}^{+}_{n \times k}}} \|A - UV^T\|_F^2$$

- Is this factorization unique?

Matrix factorization : non-negative matrix factorization angle

- So, SVD returns directions or principal components
- But these are not interpretable.
- But what if we optimized the following?

$$\min_{\substack{U \in \mathbb{R}^+_{m \times k} \\ V \in \mathbb{R}^+_{n \times k}}} \|A - UV^T\|_F^2$$

- Is this factorization unique?
- No – I could multiply U by a positive constant, and divide V by the same and that will give me the same UV^T

The non-negative matrix factorization angle

- Typically, the issues with uniqueness can be resolved by putting constraints on norm or sparsity.
- Despite that, we now have a non-convex loss. There a variety of algorithms, most of them based on alternating minimization type methods.

The non-negative matrix factorization angle

- Typically, the issues with uniqueness can be resolved by putting constraints on norm or sparsity.
- Despite that, we now have a non-convex loss. There a variety of algorithms, most of them based on alternating minimization type methods.
- Here is the loss function minimized by the buit-in NMF code in scikit-learn

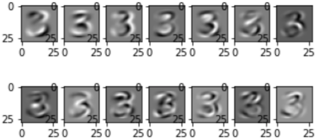
$$\min_{\substack{U \in \mathbb{R}^+_{m \times k} \\ V \in \mathbb{R}^+_{n \times k}}} \|A - UV^T\|_F^2 + \alpha\beta \left(\|\mathrm{vec}(W)\|_1 + \|\mathrm{vec}(H)\|_1\right)$$

$$+ \frac{1}{2}\alpha(1-\beta)\left(\|W\|_F^2 + \|H\|_F^2\right)$$

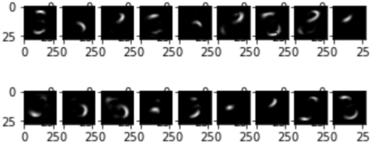
- α, β are regularization parameters

Why Non-negative matrix factorization

- Let us compare the basis vectors obtained using NMF and matrix factorization.
- Look at the right singular vectors or the V in the aforementioned optimization problem with $k = 20$.



PCA basis



NMF basis with 20 components

- Take five minutes to think how the two are different.
- Drumrolls——

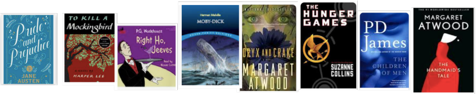
Why Non-negative matrix factorization

- The basis vectors from SVD are global, they are picking up a linear combination of the individual pixel values (which are the features)
- On the other hand, NMF is actually picking up the different parts of the threes, which can be thought of as pieces which are combined together in different ways to give many different handwritten 3's.

Why Non-negative matrix factorization

- The basis vectors from SVD are global, they are picking up a linear combination of the individual pixel values (which are the features)
- On the other hand, NMF is actually picking up the different parts of the threes, which can be thought of as pieces which are combined together in different ways to give many different handwritten 3's.
- So NMF is interpretable, and columns of U and V are not orthogonal.
- But we need conditions to make sure that algorithms return the global optima, and one needs to also think about uniqueness.

Matrix completion - NMF angle



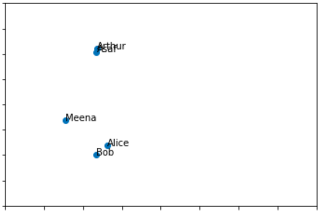
Alice	4	3	5	4	1	1	1	2
Bob	4	5	4	5	1	2	2	1
Meena	4	5	4	4	4	5	5	3
Asaf	1	1	1	1	4	4	4	5
Arthur	2	1	1	1	5	4	4	4

- Remember our user-book rating matrix?
- We random pick 5 elements and set them to zero (think missing).

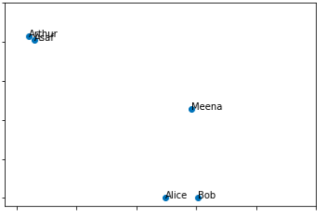
4	0	5	4	1	1	0	2
4	4	4	5	1	0	2	1
4	5	4	4	0	5	5	3
1	1	1	1	4	4	4	5
2	1	0	1	5	4	4	4

Matrix completion - NMF angle

- We will do SVD to get $Y = U_1 V_1^T$
- We will do NMF to get $Y = U_2 V_2^T$ Now we will use U_1 and U_2 to embed the users as we had before.



(A)



(B)

Table 1: (A) embedding with SVD, (B) embedding with NMF

- Take a few minutes to ponder over why these two are different and which one is more interpretable and why.

Matrix completion - NMF angle

- We will do SVD to get $Y = U_1 V_1^T$
- We will do NMF to get $Y = U_2 V_2^T$ Now we will use U_1 and U_2 to embed the users as we had before.

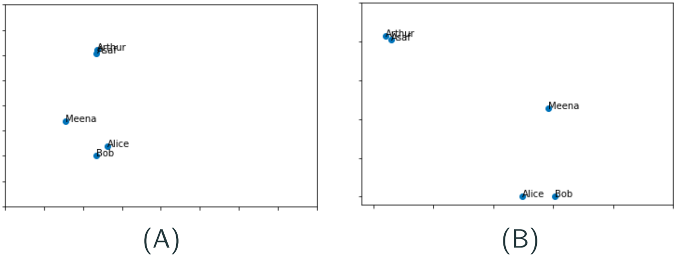


Table 2: (A) embedding with SVD, (B) embedding with NMF

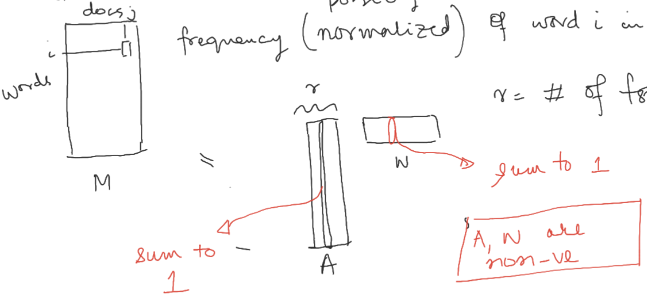
- NMF is more interpretable, because Alice/Bob are placed on the X axis (approximately) and Arthur/Asaf on the Y axis, so its almost like the different directions are for the different genres of books, classics and dystopian fiction.

Topic models - NMF angle

Topic models

frequency (possibly normalized) of word i in doc j

r = # of topics

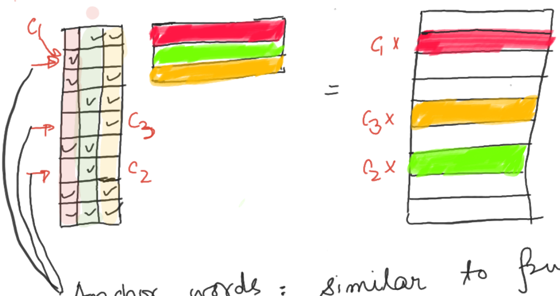


Prev work : It is NP-hard to compute NMF

But if we make an assumption, then there is a simple polynomial time algorithm.

Separability : A matrix A is separable, if for every column of A, ∃ a row of A whose only non-zero entry is in that column.

Topic models - NMF angle



Anchor words : similar to pure nodes

* If an anchor word appears in a doc, it has some representation of that topic

Topic models - NMF angle

So, in previous example, the rows in W appear as rows in M (upto scaling).

- Say I have the anchor words
- I know $W = \begin{bmatrix} M(2,:)\ \alpha_1 \\ M(5,:),\ \alpha_2 \\ M(7,:),\ \alpha_3 \end{bmatrix}$
- columns of W sum to 1.

Topic models - NMF angle

Question 1: how do I find anchor words?

Question 2: how do I recover A ?

First,

$$MM^T = A\underbrace{WW^TA^T}$$
$$= AY$$

A: separable

Y: non-negative

Topic models - NMF angle

$Q = MM^T$

$Q = V \times V$ $\Longrightarrow$ Row normalize $\Longrightarrow$ $\bar{Q}$

$Q_{ij} = P(w_1 = i,\ w_2 = j)$

$\bar{Q}_{ij} = P(w_2 = j \mid w_1 = i)$

Every row of $\bar{Q}$ lies in the convex hull of rows indexed by anchor words.

Topic models - NMF angle

$$
\begin{aligned}
\bar{Q}_{i,j} &= P(w_2 = j \mid w_1 = i) \\
&= \sum_{\ell} P(w_2 = j,\ t_1 = \ell \mid w_1 = i) \\
&= \sum_{\ell} P(w_2 = j \mid t_1 = \ell, w_1 = i)\, P(t_1 = \ell \mid w_1 = i) \\
&= \sum_{\ell} \underbrace{P(w_2 = j \mid t_1 = \ell)}\, P(t_1 = \ell \mid w_1 = i)
\end{aligned}
$$

For $i = S_k$ $\boxed{\text{anchor word for topic } k}$

$$\bar{Q}_{S_k, j} = P(w_2 = j \mid t_1 = k)$$

$$\bar{Q}_{ij} = \sum_{\ell} \bar{Q}_{S_\ell, j}\, P(t_1 = \ell \mid w_1 = i)$$

= conv. comb of $\bar{Q}_{S_\ell, j}$

Topic models - NMF angle

Finding anchor words is again finding corners of a convex hull of V points in V dims.

→ similar algorithms to one we saw in class exist.

→ Robust to noise & fast.

If I know anchor node set S,
how do we get $A_{ik} = P(w=i \mid t=k)$?

$$\bar{Q}_{ij} = \sum_k \underbrace{P(t_1=k \mid w_1=i)}_{C_{ik}} \bar{Q}_{s_k,j}$$

$$C \in \mathbb{R}^{V \times r}$$

$$p_i = \sum_j Q_{ij}$$

$$\bar{Q} = C \bar{Q}_{[S,:]}$$

$$C\mathbf{1} = \mathbf{1}$$

$$C = \bar{Q}\bar{Q}_S^T \left(\bar{Q}_S \bar{Q}_S^T\right)^{-1}$$

$$A_{ik} = \frac{P(t=k \mid w=i)\, P(w=i)}{\sum_k P(t=k \mid w=i)\, P(w=i)} = \frac{C_{ik}\, p_i}{\sum_k C_{ik} p_i}$$