Multigrid NMF

Andersen Ang

Mathématique et recherche opérationnelle
UMONS, Belgium

manshun.ang@umons.ac.be Homepage: angms.science

First draft : Feburary 20, 2020
Last update : March 10, 2020

NMF on very fat matrices

- Given a matrix $\mathbf{M} \in \mathbb{R}_+^{m \times n}$, NMF is the problem to find $\mathbf{W} \in \mathbb{R}_+^{m \times r}$ and $\mathbf{H} \in \mathbb{R}_+^{r \times n}$ such that $\mathbf{M} \approx \mathbf{W}\mathbf{H}$.
- If $n > m$ and n is big[1], we have a short fat matrix.
- If we treat the columns of $\mathbf{M}$ as data points, m is the dimension of the feature and n is the number of data points.
- In this case we have many data points, if we run NMF algorithm on $\mathbf{M}$, it may takes a very long time.
- We can instead run NMF algorithm on $\mathbf{M}'$ which has far fewer number of columns than n.
- There are multiple way to generate $\mathbf{M}'$. Randomized approach includes : select columns of $\mathbf{M}$ by random, or take $\mathbf{M}' = \mathbf{M}\mathbf{X}$, where $\mathbf{X}$ is a random matrix of size n-by-n' with $n' < n$, so that $\mathbf{M}'$ is m-by-n'.

[1] Big means $\geq 10^6$

NMF on big data with tall matrices

- What about the case m is large?
- If $m > n$ and m is large, in this case we have a thin tall matrix.
- In the case with big n, size reduction can be performed as $\mathbf{M}' = \mathbf{M}\mathbf{X}$. Now we can do the same as $\mathbf{M}' = \mathbf{X}\mathbf{M}$. That is, we perform a left-multiplication on $\mathbf{M}$ to change m to m', $m' < m$.
- If $\mathbf{X}$ is not random but designed derministically, we arrived at the *multi-grid method*.

Multi-grid

- Multi-grid methods were initially used to develop fast numerical solver for boundary value problems in differential equation.
- The word "grid" means the discretization of continuous smooth function f by choosing a set of points.
- The word "multi" means there are different levels of approximation
  - a fine grid means a higher number of points is used for the discretizaton
  - a coarse grid means a low number of point is used for the discretization
- The solution process of multi-grid method is as follows
  - Perform discretization on the problem, get a smaller sized problem
  - Perform iterative method to get the solution of the small problem
  - Get the solution of the original big problem from the solution of the small problem

Restriction and Interpolation / Prolongation

- Restriction operator $\mathcal{R}$

$$\mathbf{R} : \mathbb{R}_+^m \to \mathbb{R}_+^{m'} : \mathbf{x} \rightarrowtail \mathcal{R}(\mathbf{x}) = \mathbf{R}\mathbf{x}, \quad \mathbf{R} \in \mathbb{R}_+^{m \times m'}, \ m < m'$$

- Interpolation operator $\mathcal{I}$

$$\mathcal{I} : \mathbb{R}_+^{m'} \to \mathbb{R}_+^{m} : \mathbf{x} \rightarrowtail \mathcal{I}(\mathbf{x}) = \mathbf{J}\mathbf{x}, \quad \mathbf{J} \in \mathbb{R}^{m' \times m}, \ m < m'$$

Remarks

- Symbol $\mathbf{I}$ is reserved for the identity matrix, so we use $\mathbf{J}$ for $\mathcal{I}$.
- $\mathbf{R}$ and $\mathbf{J}$ are nonnegative and preserve nonnegativity on multiplication.
- $\mathbf{R}$ is short fat matrix and $\mathbf{J}$ is thin tall matrix
- The two operators are defined on vector $\mathbf{x}$. For matrix $\mathbf{X}$, we apply the operator columnwise :

$$\mathcal{R}(\mathbf{X}) = \mathcal{R}([\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]) = [\mathcal{R}(\mathbf{x}_1)\mathcal{R}(\mathbf{x}_2) \ldots \mathcal{R}(\mathbf{x}_n)]$$

Condition for multi-grid to work

We want the information loss during transistion from one level to another level to be small : the reconstruction $\mathcal{I}(\mathcal{R}(\mathbf{x}))$ must be close to the original $\mathbf{x}$. i.e., s is small

$$s_{\mathbf{x}} := \frac{\|\mathbf{x} - \mathcal{I}(\mathcal{R}(\mathbf{x}))\|_2}{\|\mathbf{x}\|_2}.$$

In matrix case $\mathbf{X}$,

$$s_{\mathbf{X}} := \frac{\|\mathbf{X} - \mathcal{I}(\mathcal{R}(\mathbf{X}))\|_F}{\|\mathbf{X}\|_F}.$$

Using $\mathbf{R}$ and $\mathbf{J}$, we have

$$s_{\mathbf{x}} := \frac{\|\mathbf{x} - \mathbf{JRx}\|_2}{\|\mathbf{x}\|_2}, \quad s_{\mathbf{X}} := \frac{\|\mathbf{X} - \mathbf{JRX}\|_F}{\|\mathbf{X}\|_F}.$$

A "bad" upper bound

It seems natural to factor out $\mathbf{x}$ in $s_{\mathbf{x}}$ and get

$$\|\mathbf{x} - \mathcal{I}(\mathcal{R}(\mathbf{x}))\|_2 = \|\mathbf{x} - \mathbf{JRx}\|_2 = \|(\mathbf{I} - \mathbf{JR})\mathbf{x}\|_2 \leq c\|\mathbf{x}\|_2,$$

where $c = \|\mathbf{I} - \mathbf{JR}\|_2$.

For the matrix case :

$$\begin{aligned}
\|\mathbf{X} - \mathcal{I}(\mathcal{R}(\mathbf{X}))\|_F &= \|\mathbf{X} - \mathcal{I}([\mathbf{Rx}_1 \ \mathbf{Rx}_2 \ \ldots \ \mathbf{Rx}_n])\|_F \\
&= \|\mathbf{X} - [\mathbf{JRx}_1 \ \mathbf{JRx}_2 \ \ldots \ \mathbf{JRx}_n])\|_F \\
&= \|[(\mathbf{I} - \mathbf{JR})\mathbf{x}_1 \ (\mathbf{I} - \mathbf{JR})\mathbf{x}_2 \ \ldots \ (\mathbf{I} - \mathbf{JR})\mathbf{x}_n])\|_F \\
&= \|(\mathbf{I} - \mathbf{JR})[\mathbf{x}_1 \ \mathbf{x}_2 \ \ldots \ \mathbf{x}_n]\|_F \\
&= \|(\mathbf{I} - \mathbf{JR})\mathbf{X}\|_F \\
&\leq c\|\mathbf{X}\|_F.
\end{aligned}$$

So we have $s_{\mathbf{x}}$ and $s_{\mathbf{X}}$ both upper bounded by the constant c.

But, c is a bad upper bound for s.

Why c is a bad upper bound

- By design $\mathbf{JR}$ is very far away from $\mathbf{I}$, so $c = \|\mathbf{I} - \mathbf{JR}\|_2$ is a large number.
- Factorizing $\mathbf{x}$ in the expression of s removes the role of $\mathbf{x}$ in s. Note that there are some vectors $\mathbf{x}$ that $s_{\mathbf{x}}$ is zero.

  For example, the vector of all-ones gives $\mathbf{x} - \mathbf{JRx} = 0$. And in fact, there are subsets of $\mathbb{R}^n$ for which the upper bound c will be far from good.

Therefore, we should stick with the definition of s, not doing any simplification.

The multi-grid NMF process

Given $\mathbf{M} \in \mathbb{R}^{m \times n}$, the goal is to solve $\mathbf{M} \approx \mathbf{W}\mathbf{H}$.

1. Given $\mathbf{M} \in \mathbb{R}^{m \times n}$, and two matrices $\mathbf{W}_0 \in \mathbb{R}_+^{m \times r}$, $\mathbf{H}_0 \in \mathbb{R}_+^{r \times n}$.
2. Compute $\mathbf{M}' = \mathcal{R}(\mathbf{M})$ and $\mathbf{W}_0' = \mathcal{R}(\mathbf{W}_0)$, so now we have $\mathbf{M}'$ and $\mathbf{W}_0'$ with smaller size (fewer rows).
3. Compute NMF for $\mathbf{M}'$ using $\mathbf{W}_0', \mathbf{H}_0$ as initial estimate.
   i.e. we have $\mathbf{M}' \approx \mathbf{W}'\mathbf{H}$.
4. Get $\mathbf{W}$ back by $\mathcal{I}(\mathbf{W}')$ from the last step.
   Now we have $\mathbf{W}, \mathbf{H}$ that approximately solve NMF of $\mathbf{M}$.
5. The solution can be improved further by using $\mathbf{W}, \mathbf{H}$ as input in other NMF algorithm.

The above describes a single-level grid process. To have "multi"-grid, repeats steps 2 to 4.

Note that the NMF computation in step 3 is cheap due to smaller size.

Why multi-grid NMF works

- Operators $\mathcal{R}$, $\mathcal{I}$ both preserve nonnegativity.
- The error between $\mathbf{M}$ and $\mathcal{I}(\mathbf{W}')\mathbf{H}$ is small.

$$\begin{aligned}
\|\mathbf{M} - \mathcal{I}(\mathbf{W}')\mathbf{H}\|_F &= \|\mathbf{M} - \mathcal{I}(\mathcal{R}(\mathbf{M})) + \mathcal{I}(\mathcal{R}(\mathbf{M})) - \mathcal{I}(\mathbf{W}')\mathbf{H}\|_F \\
&\leq \|\mathbf{M} - \mathcal{I}(\mathcal{R}(\mathbf{M}))\|_F + \|\mathcal{I}(\mathcal{R}(\mathbf{M})) - \mathcal{I}(\mathbf{W}')\mathbf{H}\|_F \\
&= s_{\mathbf{M}}\|\mathbf{M}\|_F + \|\mathcal{I}(\mathcal{R}(\mathbf{M}) - \mathbf{W}'\mathbf{H})\|_F \\
&= s_{\mathbf{M}}\|\mathbf{M}\|_F + \|\mathbf{J}(\mathcal{R}(\mathbf{M}) - \mathbf{W}'\mathbf{H})\|_F \\
&\leq s_{\mathbf{M}}\|\mathbf{M}\|_F + \|\mathbf{J}\|_F\|\mathbf{M}' - \mathbf{W}'\mathbf{H}\|_F
\end{aligned}$$

As $\mathbf{M}' \approx \mathbf{W}'\mathbf{H}$, and both $\mathbf{M}', \mathbf{W}'$ has smaller size, thus it is not difficult for algorithm to achieve high accuracy on $\mathbf{M}' \approx \mathbf{W}'\mathbf{H}$, so $\|\mathbf{M}' - \mathbf{W}'\mathbf{H}\|_F$ can be very small (says 10^{-9}).

So if $s_{\mathbf{M}}$ is small, then $\|\mathbf{M} - \mathcal{I}(\mathbf{W}')\mathbf{H}\|_F$ is small, and the grid approximation works.

When does multi-grid NMF works

A part from the design of $\mathcal{R}$ and $\mathcal{I}$, the matrix $\mathbf{M}$ itself also contribute to $\mathbf{s_M}$, and therefore whether multi-grid works also depends on $\mathbf{M}$.

- For some matrices, $\mathbf{s_M}$ is small thus the multi-gird approach works. These matrices are those containing a lots of *low frequency component*. In image, these component corresponds to a large region of slowly changing pixels. At those regions, as the pixel values change slowly, their pixel value can be well approixmated by the pixel values of its neighbors.

- If the matrix has lots of sudden changes (high frequency component), then multi-grid may not work.

Last page - summary

- NMF on big m
- Multigrid method
- Multigrid NMF

Reference

- Gillis, Nicolas, and Franois Glineur. "A multilevel approach for nonnegative matrix factorization." Journal of Computational and Applied Mathematics 236.7 (2012).

End of document