Topic Modeling with SVD and NMF

April 29, 2020

0.1 Non-negative Matrix Factorization

In this section, we will briefly introduce the non-negative matrix factorization problem and algorithms in solving this problem.

0.1.1 Non-negative Matrix Factorization Problem

The non-negative matrix factorization (NMF for short) problem can be expressed formally as the following:

> **NMF problem:** Given a non-negative matrix $A \in R^{m \times n}$ and a positive integer $k < minm, n$, find non-negative matrices $W \in R^{m \times k}$ and $H \in R^{k \times n}$ to minimize the cost function
>
> $$f(W, H) = \frac{1}{2}\|A - WH\|_F^2 \tag{1}$$

The product WH is called a non-negative matrix factorization of A. Through NMF, we get a low-rank (at most rank k) approximation, WH, of the original matrix A.

There are two note-worthy properties about NMF that we would like to mention at this moment. First is that, NMF can be written column by column as

$$a_i = Wh_i, \;\; i \in 1, 2, ..., n \tag{2}$$

Where a_i is the i-th column vector of matrix A, and h_i is the corresponding column vector of matrix H. That is to say, each data vector a_i is approximated by a linear combination of the columns of W, weighted by the components of h_i. Therefore, W can be regarded as a basis that is optimized for the linear approximation of data A. Usually, the parameter k is small compare to the dimension of A, so NMF will be a good approximation if there exists latent structures in vectors of data matrix A.

Second property of NMF is that, negative entries are not allowed in W and H. For this reason, NMF is intuitively a part-based representation, since the whole is formed by combining the parts. These two features make NMF a good method for text classification, we will discuss this with details in later sections.

0.1.2 Basic Algorithms for Solving NMF Problems

Though NMF problems has a simple form, it has been proven that finding the exact solution of these problems in general is NP-hard. Nonetheless, it is possible to approximately solve the NMF problems using iterative measures. In this section, we will introduce three fundamental algorithms based on which other advanced algorithms are built. They are multiplicative update algorithm, gradient descent algorithm and alternating least squares algorithm.

Multiplicative Update Algorithm for NMF:

$W = rank(m, k)$; initialize W as random dense matrix
$H = rank(m, k)$; initialize H as random dense matrix
for $i = 1 : maxiter$
    $H_{ij} \leftarrow H_{ij} \frac{(W^T A)_{ij}}{(W^T W H)_{ij} + \epsilon}$
    $W_{ij} \leftarrow W_{ij} \frac{(A H^T)_{ij}}{(W H H^T)_{ij} + \epsilon}$
end

Basic Gradient Descent Algorithm for NMF:

$W = rank(m, k)$; initialize W
$H = rank(m, k)$; initialize H
for $i = 1 : maxiter$
  $H_{ij} \leftarrow H_{ij} - \epsilon_H \frac{\partial f}{\partial H}$
  $W_{ij} \leftarrow W_{ij} - \epsilon_W \frac{\partial f}{\partial W}$
end

Basic Alternative Least Squares (ALS) Algorithm for NMF:

$W = rank(m, k)$; initialize W as random dense matrix
for $i = 1 : maxiter$
  (LS) $H \leftarrow (W^T W)^{-1} W^T A$
  (NONNEG) Set all negative elements in H to 0
  (LS) $W \leftarrow ((HH^T)^{-1} WA)^T$
  (NONNEG) Set all negative elements in W to 0
end

All above three algorithms can only find local solution but not global optimal, due to the fact that the cost function $f = \frac{1}{2}\|A - WH\|_F^2$ is not convex in both variables W and H together. In addition, for the last two algorithms, we need to imply a "projection" step to ensure the non-negative property of W and H, that is, to set all non-negative entries to 0 after each iteration step. Due to limitation of space, we are not going to extend our discussion into details on the algorithms. The presentation here is simply to provide some basic understanding on how to deal with the NMF problem. Of course, there exist more advanced and faster algorithms, and will we be learning to use tool-kit *scikit-learn* to solve the NMF problem in later sections.

0.1.3 Application of NMF on Text Classification

As we have seen in previous sections, a collection of documents can be represented a data matrix A, each column vector represents a document in word-frequency space. And we have learned how to classify it using singular value decomposition. In this section, we will learn to apply NMF in text classification, and compare the method of NMF with SVD.

The NMF problem for text classification can be expressed as the following:

> Given a collection of document represented by a data matrix $A \in R^{m \times n}$, find the non-negative matrix factorization
>
> $$A_{m \times n} \approx W_{m \times k} H_{k \times n} \tag{3}$$
>
> that minimize the cost function in (1). In above equation, k is a positive integer that $k \ll min(m, n)$.

Remember that the non-negative matrix factorization can be written column by column, it leads to a natural interpretation for W and H:

W – basis document matrix
H – weight matrix

To finish up this section, we will briefly compare NMF and SVD in topic modeling problems.

- Both SVD and NMF can be used to solve text classification problems. The interpretation of decomposed matrices as result are similar.

- NMF only allows non-negative entries, which leads to the interpretation that it uses parts to represent the whole. While SVD gives a more "deeper" factorization basing on the structural features of the entire data set.

- The solution to SVD is unique; while the solution to NMF is not. Since topic modeling problems are often presented as un-supervised learning problems, one might need to try different rank approximations (different k) using NMF.

0.1.4 Warm-Up Questions

1. Prove that NMF is non-unique.
(Hint: Insert matrix (BB^{-1}) between W and H.)

2. Suppose that, in the NMF problem described in equation (1), the matrix W is fixed. Prove that the solution to the least square problem

$$min_H \|A - WH\|_F^2$$

is $H = (W^T W)^{-1} W^T A$.
(Hint: Use the property that NMF can be written column by column.)