Research Article **Impact Factor: 0.621** **ISSN: 2319-507X**
Vijay Kolte, IJPRET, 2014; Volume 2 (8): 423-435 **IJPRET**

INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

ONLINE NON-NEGATIVE MATRIX FACTORIZATION FOR CLUSTERING OF NEWS DOCUMENT

VIJAY S. KOLTE, RAJENDRA SAWANT, PRAMOD BHALERAO

KSIET, Hingoli.
Accepted Date: 15/02/2014 ; Published Date: 01/04/2014

Abstract: When document corpus is very large, we often need to reduce the number of features using technique such as NMF. It is not possible to apply conventional Non-negative Matrix Factorization (NMF) to find the factors for billion by million matrix as the matrix may not fit in memory. Here we propose Online NMF for updating the factors when new data comes in without recomputing from scratch. We will apply Online NMF on a small subset of documents and cluster all the documents in low dimensional space using means algorithm. We will experimentally show that by processing small subsets of documents we will be able to learn meaningful topics, which are similar and in some cases better than the NMF baseline. We will also give the mathematical proof.

Keywords: NMF, Clustering, Online NMF

PAPER-QR CODE

Corresponding Author: MR. VIJAY S. KOLTE

Access Online On:

www.ijpret.com

How to Cite This Article:

Vijay Kolte, IJPRET, 2014; Volume 2 (8): 423-435

INTRODUCTION

Document Clustering is the process of collecting similar documents into clusters. Many document clustering techniques have been developed because of application in information retrieval, web navigation, pattern recognition and organization of huge volumes of text documents. These techniques can be broadly divided into categories:

Hierarchical- The algorithms find successive clusters using the existing clusters in top-down or bottom-up manner.

Partitioning(flat clustering)-This approach divides the documents into disjoint clusters. The various methods in this category are : k-means clustering, probabilistic clustering using the Naive Bayes or Gaussian model, latent semantic indexing (LSI), spectral clustering, and Nonnegative Matrix Factorization(NMF).

If data is in high dimensions then applying traditional k-means or Hierarchical clustering is not feasible. Before applying any clustering we have to reduce the dimensionality of data. In the low-dimensional semantic space, the traditional clustering algorithms can be then applied. To serve this purpose, spectral clustering, clustering using LSI and clustering based on nonnegative matrix factorization are the most well-known techniques. Recent studies show that clustering based on Nonnegative Matrix Factorization outperforms the other two. The reason behind this is NMF method focuses on the global geometrical structure of document space while clustering using LSI does not.

The Nonnegative Matrix Factorization problem can be stated as follows:

Given a nonnegative $V \in R^{d \times n}$ and a positive integer k < min{d,n}, find nonnegative matrices $W \in R^{d \times k}$ and $H \in R^{n \times k}$ to minimize the function:

$$f(W, H) = \frac{1}{2} \| V - WH^T \|_F^2 \qquad (1)$$

The product WH^T is called a nonnegative matrix factorization of V. It is not possible to find a unique solution to the above problem as the function is non-convex in terms of W and H. Also, we can prove that $WDD^{-1}H^T$ is another solution for any non negative invertible matrix D4

A. Applications and Need of the Project

- **Text mining**

NMF can be used for text mining applications. In this process, a document-term matrix is constructed with the weights of various terms (typically weighted word frequency information) from a set of documents. This matrix is factored into a term-feature and a feature-document matrix. The features are derived from the contents of the documents, and the feature-document matrix describes data clusters of related documents. One specific application used hierarchical NMF on a small subset of scientific abstracts from PubMed. Another research group clustered parts of the Enron email dataset with 65,033 messages and 91,133 terms into 50 clusters. NMF has also been applied to citations data, with one example clustering Wikipedia articles and scientific journals based on the outbound scientific citations in Wikipedia. Arora, Ge and Moitra (2012) have given polynomial-time algorithms to learn topic models using NMF. The algorithm assumes that the topic matrix satisfies a separability condition that is often found to hold in these settings.

- **Spectral data analysis**

NMF is also used to analyze spectral data; one such use is in the classification of space objects and debris.

- **Scalable Internet distance prediction**

NMF is applied in scalable Internet distance (round-trip time) prediction. For a network with N hosts, with the help of NMF, the distances of all the N2 end-to end links can be predicted after conducting only O(N) measurements. This kind of method was firstly introduced in Internet Distance Estimation Service (IDES).Afterwards, as a fully decentralized approach, Phoenix network coordinate system is proposed. It achieves better overall prediction accuracy by introducing the concept of weight.

- **Bioinformatics**

NMF has been successfully applied to bioinformatics.

B. Preliminary Survey and Need Analysis through leading Technical; Literature

To find the approximate nonnegative factors W and H for V, there are many algorithms proposed, such as

1) **Multiplicative Update rule:** The most popular algorithm for the NMF problem uses the multiplicative rules suggested by Lee and Seung [4]. To formulate these rules, we choose to fix one of the factors (i.e. W or H) and try to minimize the cost function with respect to the other factor.

Algorithm 1 Multiplicative Rules

1: Initialize W^0, H^0 and $t = 0$
2: **repeat**
3: $W^{t+1} = W^t \otimes \frac{VH^t}{W^t(H^t)^T H^t}$
4: $H^{t+1} = H^t \otimes \frac{V^T W^{t+1}}{H^t(W^{t+1})^T W^{t+1}}$
5: $t = t + 1$
6: **until** stopping condition

Where $\otimes$ is element wise matrix multiplication.

2) Alternating Least Square: The first algorithm proposed for solving the nonnegative matrix factorization was the alternating least squares method. It is known that, fixing either W or H, the problem becomes a least squares problem with nonnegativity constraints.

Since the least squares problems in Algorithm 2 can be perfectly decoupled into smaller problems corresponding to the columns or rows of V, we can directly apply methods for the Nonnegative Least Square problem to each of the small problems.

Algorithm 2 Alternating Least Square (ALS)

1: Initialize W and H
2: **repeat**
3: Solve: $\min_{H \geqslant 0} \frac{1}{2}\|V - WH^T\|_F^2$
4: Solve: $\min_{W \geqslant 0} \frac{1}{2}\|V^T - HW^T\|_F^2$
5: **until** stopping condition

Many algorithms are proposed to solve ALS such as active set method [6].

3) Block principle pivoting method(BPB): This algorithm builds upon the block principal pivoting method for the nonnegativity constrained least squares problem that overcomes some limitations of active set methods [7].

4) Projected Gradient descent(PGD): The Projected Gradient Method techniques can be applied to NMF [5].

Algorithm 3 Projected Gradient descent(PGD)

```
1: Initialize W^0, H^0, t = 1, 0 < β, σ < 1, and α_0 = 1
2: repeat
3:   α_t = α_{t-1}
4:   y = max(0, W^k − α_t ∇_W f(W^t, H^t))
5:   if f(y, H^t) − f(W^t, H^t) > σ⟨∇_W f(W^t, H^t), y − W^t⟩
6:     repeat
7:       α_t = α_t.β
8:       y = max(0, W^k − α_t ∇_W f(W^t, H^t))
9:   until f(y, H^t) − f(W^t, H^t) ≤ σ⟨∇_W f(W^t, H^t), y − W^t⟩
10:  else
11:    repeat
12:      lasty = y
13:      α_t = α_t/β
14:      y = max(0, W^k − α_t ∇_W f(W^t, H^t))
15:  until f(y, H^t) − f(W^t, H^t) > σ⟨∇_W f(W^t, H^t), y − W^t⟩
16:    y = lasty
17:  end if
18:  W^{t+1} = y
19:  t = t + 1
20: until stopping condition
21: Let f(W^{t+1}, H^t) = ½‖V^T − W^{t+1}(H^t)^T‖_F^2, Then, H^{t+1}
    can be easily obtained by repeating steps similar to 2 − 20
```

where <.,.> is the sum of the elementwise product of two matrices, and $\nabla_W f(W^t, H^t)$ and $\nabla_H f(W^{t+1}, H^t)$ are the gradients of f with respect to W and H respectively.\

5) Graph Regularized NMF(GNMF): In GNMF, an affinity graph is constructed to encode the geometrical information, and to find a matrix factorization which respects the graph structure [3].

The low dimensional representation of V_j with respect to the new basis is $z_j = [V_{1j}, \ldots, V_{kj}]$. To measure the smoothness of the low dimensional representation, we can use

$$R = \frac{1}{2}\sum_{i,j} \|z_i - z_j\|^2 G_{ij}$$
$$= \sum_i z_i^T z_i D_{ii} - \sum_{i,j} z_i^T z_j G_{ij}$$
$$= Tr(V^T DV) - Tr(V^T GV) = Tr(V^T LV)$$

Where $\|z_i - z_j\|^2$ is the Euclidean distance, $Tr(\cdot)$ denotes the trace of a matrix, G is graph weight matrix, D is a diagonal matrix whose entries are row sums of G, $D_{ii} = \sum_j G_{ij}$, L= D - G which is called graph Laplacian.

For each data point in V, we find its p nearest neighbors and put edges between them. There are many choices to define the graph weight matrix G on the graph. The most commonly used are as follows:

1) 0-1 weighting $G_{ij} = 1$ if and only if nodes i and j are connected by an edge.

2) Heat kernel weighting If nodes i and j are connected, put

$$G_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\sigma}} \quad (2)$$

Heat kernel has an intrinsic connection to the Laplace

Beltrami operator on differentiable functions on a manifold[9].

3) Dot-product weighting If nodes i and j are connected,

put

$$G_{ij} = x_i^T x_j$$

The objective function is defined as follows,

$$\begin{aligned} O &= \|V - WH^T\|^2 + \lambda Tr(V^T LV) \\ &= Tr((V - WH^T)(V - WH^T)^T) + \lambda Tr(V^T LV) \\ &= Tr(VV^T) - 2Tr(VHW^T) + Tr(WH^T HW^T) \\ &\quad + \lambda Tr(V^T LV) \end{aligned}$$

Let ϕ_{ik} and ψ_{jk} be the Lagrange multipliers for constraint $w_{ik} \geq 0$ and $h_{jk} \geq 0$ respectively, and $\Phi = [\phi_{ik}]$, $\Psi = [\psi_{jk}]$, the Lagrange L is

$$\begin{aligned} L &= Tr(VV^T) - 2Tr(VHW^T) + Tr(WH^T HW^T) \\ &\quad + \lambda Tr(V^T LV) + Tr(\Phi W^T) + Tr(\Psi H^T) \end{aligned}$$

The partial derivatives of L with respect to W and H are:

$$\frac{\partial L}{\partial W} = -2VH + WH^T H + \Phi \quad (4)$$

$$\frac{\partial L}{\partial H} = -2V^T W + HW^T W + 2\lambda LH + \Psi \quad (5)$$

Using the KKT conditions $\phi_{ik} w_{ik} = 0$ and $\psi_{jk} h_{jk} = 0$, we get the following equations for w_{ik} and h_{jk}:

$$-(VH)_{ik} w_{ik} + (WH^T H)_{ik} w_{ik} = 0$$

$$-(V^T W)_{jk} h_{jk} + (HW^T W)_{jk} h_{jk} + \lambda (LH)_{jk} h_{jk} = 0$$

These equations lead to the following updating rules:

$$w_{ik} \leftarrow w_{ik} \frac{(VH)_{ik}}{(WH^T H)_{ik}}$$

$$h_{jk} \leftarrow h_{jk} \frac{(V^T W)_{jk} + \lambda GH}{(HW^T W)_{jk} + \lambda DH}$$

The GNMF algorithm is shown in algorithm 4.

Algorithm 4 GNMF

1: Initialize W^0, H^0, $\lambda \geqslant 1$, and $t = 0$
2: Construct G
3: $D_{ii} = \sum_j G_{ij}$
4: $D_{i \neq j} = 0$
5: **repeat**
6: $W^{t+1} = W^t \otimes \frac{VH^t}{W^t (H^t)^T H^t}$
7: $H^{t+1} = H^t \otimes \frac{V^T W^{t+1} + \lambda G H^t}{H^t (W^{t+1})^T W^{t+1} + \lambda D H^t}$
8: $t = t + 1$
9: **until** stopping condition

The objective function (O) of GNMF is gradient descent [12]. Gradient descent leads to the following additive update rules:

$$w_{ik} \leftarrow w_{ik} + \eta_{ik} \frac{\partial O}{\partial w_{ik}} \quad (6)$$

$$h_{jk} \leftarrow h_{jk} + \delta_{jk} \frac{\partial O}{\partial h_{jk}} \quad (7)$$

The η_{ik} and δ_{jk} are step size parameters. As long as η_{ik} and δ_{jk} are sufficiently small, the above updates should reduce O unless W and H are at a stationary point.

Let $\eta_{ik} = -w_{ik}/2(WH^TH)_{ik}$, we have

$$w_{ik} + \eta_{ik} \frac{\partial O}{\partial w_{ik}} = w_{ik} - \frac{w_{ik}}{2(WH^TH)_{ik}} \frac{\partial O}{\partial w_{ik}} \quad (8)$$

$$= w_{ik} - \frac{w_{ik}}{2(WH^TH)_{ik}} (-2(VH)_{ik} + 2(WH^TH)_{ik}) \quad (9)$$

$$= w_{ik} \frac{(VH)_{ik}}{(WH^TH)_{ik}}$$

Similarly, let $\delta_{jk} = -h_{jk}/2(HW^TW + \lambda DH)_{jk}$, we have

$$h_{jk} + \delta_{jk} \frac{\partial O}{\partial h_{jk}} = h_{jk} - \frac{h_{jk}}{2(HW^TW + \lambda DH)_{jk}} \frac{\partial O}{\partial h_{jk}} \quad (10)$$

$$= h_{jk} - \frac{h_{jk}}{2(HW^TW + \lambda DH)_{jk}} (-2(V^TW)_{jk} + 2(HW^TW)_{jk} + 2\lambda(LH)_{jk})$$

$$= h_{jk} \frac{(V^TW + \lambda GH)_{jk}}{(HW^TW + \lambda DH)_{jk}}$$

The **GNMF** multiplicative updating rules are special cases of gradient descent with an automatic step parameter selection.

C. Brief Outline of the current topic Or Work

Current research in nonnegative matrix factorization includes,

- **Algorithmic:** searching for global minima of the factors and factor initialization.
- **Scalability:** how to factorize million-by-billion matrices, which are commonplace in Web-scale data mining, e.g., see Distributed Nonnegative Matrix Factorization(DNMF).
- **Online:** how to update the factorization when new data comes in without recomputing from scratch.

D. Scope of the intended work

Online NMF (Non-negative matrix factorization) is a recently developed method for real time data analysis in an online context. Non-negative matrix factorization in the past has been used for static data analysis and pattern recognition. In the past it has been used for facial recognition and spectral data analysis, however due to the time and memory expensive nature of NMF algorithms, PCA, SVD, and Pearson correlation based methods have been used instead. However ,the fact that data can be recreated as a linear combination of the set of resolved "basis" data is advantageous in some lines of study. One such use is for collaborative filtering in

recommendation systems where it is advantageous to know not only how much two individuals are alike, which can be derived from the Pearson correlation, but also in what ways are they alike. The purpose of the online NMF algorithm is to perform rapid NMF analysis so that recommendations can be produced in real time.

PROPOSED WORK

As electronic documents become available in streams over time, their content contains a strong temporal ordering. Considering the time information is essential to better understand the underlying topics and track their evolution and spread within their domain. In addition, instead of analyzing large collections of time-stamped text documents as archives in an off-line fashion, it is more practical for genuine applications to analyze, summarize, and categorize the stream of text data at the time of its arrival. For example, as news arrives in

streams, organizing it as threads of relevant articles is more efficient and convenient. In addition, there is a great potential to rely on automated systems to track current topics of interest and identify emerging trends in online digital libraries and scientific literature.

Also in real-world applications with large-scale high dimension data, where nd is extremely large, it would be difficult to apply a conventional NMF algorithm directly. For example, 20NG dataset contains nearly n $\approx$ 20000 documents with dimension d $\approx$70000. Then the size of matrix V becomes $O(10^9)$, which clearly does not fit in memory. Specifically we are interested in online soft clustering of a large collection of documents in high dimensions.

BACKGROUND

If data is online or evolving, we have to process the data at once to find the combined factor. But the iterative update method for solving NMF problem is computationally expensive. Processing already processed data wastes resources. To overcome this problem, Bin Cao et.al.[1] suggested online NMF. Fei Wang et.al.[2] proposed another improved algorithm to serve the same purpose.

The loss function (1) can be decomposed as,

$$f(W,H) = \frac{1}{2}\|V - WH^T\|_F^2 = \frac{1}{2}\sum_{i=1}^{n}\|v^i - Wh^i\|_F^2$$

where h^i is the i-th column of H^T.
when W is fixed, the minimum value of $f(W,H)$ can be reached if and only if

$$f(W,h^i) = \frac{1}{2}\sum_{i=1}^{n}\|v^i - Wh^i\|_F^2$$

is minimized for all i.

Thus, we can solve n independent Nonnegative Least Square (NLS) problems and aggregate the solution as,

$$H = [h^1, h^2, \ldots, h^n]^T$$

Find h^i, W fixed by minimizing following loss function,

$$f(h^i) = min_{h_i \geqslant 0}\ \frac{1}{2}\sum_{i=1}^{n}\|v^i - Wh^i\|_F^2$$

Which is a typical nonnegative least square (NLS) problem. We can solve it using Algorithm 2 or 3 or BPB method. Update W by minimizing the following loss function,

$$\begin{aligned} f(W^t) &= \frac{1}{2}\sum_t \|v^t - W^t h^t\|_F^2 \\ &= \sum_t tr[(v^t)^T v^t - 2(v^t)^T W^t (h^t)^T \\ &\quad + (W^t(h^t)^T)^T W^t (h^t)^T] \end{aligned}$$

we can apply the projected gradient descent (PGD) as per Algorithm 3.

The gradient of $f(W^t)$ with respect to W is,

$$\nabla_W f(W^t) = -2\sum_t [v^t h^t - W^t (h^t)^T h^t]$$

Following two terms represent the first and second-order information at time t

$A^t = \sum_t v^t h^t$
$B^t = \sum_t (h^t)^T h^t$

A. First Order PGD

The first order PGD method updates W^t by the following

rule starting with some initial W^0.

$$W^{t+1} = P[W^t - \alpha_t \nabla_{W^t} f(W^t)]$$

where α_t is step size and $P[\cdot]$ is projection onto the nonnegative constraint set and is defined as

$$P[x] = \begin{Bmatrix} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{Bmatrix}$$

Then updation rule for W^{t+1} becomes

$$W^{t+1} = P[W^t + 2\alpha_t(v^t h^t - W^t (h^t)^T h^t)]$$

B. Second Order PGD

The Hessian matrix of $f(W^t)$ with respect to W is

$$H[f(W^t)] = 2\sum_t (h^t)^T h^t$$

One disadvantage of the first order PGD is that we need to carefully choose the step size and its convergence could be quite slow. To make the PGD algorithm parameter free with faster convergence, we can make use of the second-order information [13] by utilizing the Hessian with the following updating rule:

$$W^{t+1} = P[W^t - \nabla_{W^t} f(W^t) \cdot H^{-1}[f(W^t)]]$$

Where $H^{-1}[f(W)]$ is inverse of the Hessian matrix. Exact

calculation of the inverse of the Hessian matrix becomes time-consuming when the number of clusters (r) becomes large. There are two common strategies for approximating the Hessian inverse:

- **Diagonal Approximation (DA)** [10] It only uses the diagonal line of the Hessian matrix to approximate the whole matrix.

- **Conjugate Gradient (CG) [11]** This method exploits the fact that what we really need is the product of a matrix and the inverse of the Hessian.

The Online NMF(ONMF) algorithm proposed by Fei Wang et.al.[2] is shown in algorithm 5.

Algorithm 5 ONMF

Initialize W^0, $A^0 = 0$, $B^0 = 0$, and m
$S = \lceil \frac{n}{m} \rceil$ // where n is total no of document.
repeat
  for $t \leftarrow 1$ **to** S **do**
    Draw v^t (i.e. m data points) from V
    Compute h^t by solving $f(h^t)$ using Algorithm 2 or 3 or BPB method
    $A^t = A^{t-1} + v^t h^t$
    $B^t = B^{t-1} + (h^t)^T h^t$
    $W^t = W^{t-1}$
    repeat
      $\Delta = (A^t - W^t B^t)(B^t)^{-1}$
      $W^t = max(0, \Delta + W^t)$
    until stopping condition
  end for
until stopping condition
$W = W^S$
$H = [h^1, h^2, \ldots, h^S]$

where $H = [h^1, h^2, \ldots, h^S]$ is a concatenation of matrices.

CONCLUSION AND FUTURE WORK

- We implemented the existing Online NMF algorithm and got poor results.
- We will come up with a improved algorithm.
- We will try to give mathematical proof for our algorithm.

REFERENCES

1. B. Cao, D. Shen, J. Sun, X. Wang, Q. Yang, and Z. Chen. "Detect and Track Latent Factors with Online Nonnegative Matrix Factorization" In Proc. International Joint Conference on Artificial Intelligence(IJCAI),pages 2689-2694, 2007.

2. F. Wang, C. Tan, A. Konig, and P. Li. "Efficient Document Clustering via Online Nonnegative Matrix Factorizations" In Eleventh SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, 28 April 2011.

3. D. Cai, X. He, J. Han, and T. Huang. "Graph Regularized Non-negative Matrix Factorization for Data Representation" IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011.

4. D. Lee and H. Seung. “Learning the parts of objects by non-negative matrix factorization” Nature, 1999.

5. C. Lin, “Projected gradient methods for non-negative matrix factorization.”Neural Computation, 2007.

6. C.L. Lawson and R.J. Hanson. “Solving least squares problems” Prentice- Hall Englewood Cliffs, NJ, 1974.

7. J. Kim and H. Park. “Toward faster nonnegative matrix factorization: new algorithm and comparisons” In Proceedings of the 8th International Conference on Data Mining (ICDM), pages 353-362, 2008.

8. W. Xu, X. Liu, and Y. Gong. “Document Clustering Based on Non-Negative Matrix Factorization” Proc. Intl Conf. Research and Development in Information Retrieval, pp. 267-273, Aug. 2003.

9. M. Belkin and P. Niyogi. “Laplacian eigenmaps and spectral techniques for embedding and clustering” In Advances in Neural Information Processing Systems 14, pages 585591. MIT Press, Cambridge, MA, 2001.

10. C. M. Bishop. “Neural Networks for Pattern Recognition” Oxford University Press, 1995.

11. N. Andrei. “Accelerated conjugate gradient algorithm with finite difference hessian/vector product approximation for unconstrained optimization” Journal of Computational and Applied Mathematics, 2009.

12. J. Kivinen and M. K. Warmuth. “Additive versus exponentiated gradient updates for linear prediction” In STOC95: Proceedings of the twentyseventh annual ACM symposium on Theory of computing, pages 209218,1995.

13. D. P. Bertsekas. “Projected newton methods for opti- mization problems with simple constraints” SIAM J. Control and Optimization, 1982.

14. F. R. K. Chung. “Spectral Graph Theory” volume 92 of Regional Conference Series in Mathematics, AMS, 1997.

15. P.K. Chan, D.F. Schlag, and J.Y. Zien. “Spectral k-Way Ratio-Cut Partitioning and Clustering” IEEE Trans. Computer-Aided Design, 1994.

16. A.Y. Ng, M. Jordan, and Y. Weiss. "On Spectral Clustering: Analysis and an Algorithm" Advances in Neural Information Processing Systems 14, pp. 849-856, Cambridge, Mass.: MIT Press,2001.

17. H. Zha, C. Ding, M. Gu, X. He, and H. Simon. "Spectral Relaxation for k-Means Clustering" Advances in Neural Information Processing Systems 14, pp. 1057-1064, Cambridge, Mass.: MIT Press, 2001.

18. D. D. Lee and H. S. Seung. "Algorithms for non-negative matrix factorization" In Advances in Neural Information Processing Systems13,2001