Fast Orthogonal Nonnegative Matrix Tri-Factorization for Simultaneous Clustering

Zhao Li[1], Xindong Wu[1,2], and Zhenyu Lu[1]

[1] Department of Computer Science, University of Vermont, Unite States
[2] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, PR China
{zhaoli,xwu,zlu}@cems.uvm.edu

Abstract. Orthogonal Nonnegative Matrix Tri-Factorization (ONMTF), a dimension reduction method using three small matrices to approximate an input data matrix, clusters the rows and columns of an input data matrix simultaneously. However, ONMTF is computationally expensive due to an intensive computation of the Lagrangian multipliers for the orthogonal constraints. In this paper, we introduce Fast Orthogonal Nonnegative Matrix Tri-Factorization (FONT), which uses approximate constants instead of computing the Lagrangian multipliers. As a result, FONT reduces the computational complexity significantly. Experiments on document datasets show that FONT outperforms ONMTF in terms of clustering quality and running time. Moreover, FONT is further accelerated by incorporating Alternating Least Squares, and can be much faster than ONMTF.

Keywords: Nonnegative Matrix Factorization, Orthogonality, Alterative Least Square.

1 Introduction

Dimension reduction is a useful method for analyzing data of high dimensions so that further computational methods can be applied. Traditional methods, such as principal component analysis (PCA) and independent component analysis (ICA) are typically used to reduce the number of variables and detect the relationship among variables. However, these methods cannot guarantee nonnegativity, and are hard to model and interpret the underlying data. Nonnegative matrix factorization (NMF) [7,8], using two lower-rank nonnegative matrices $W \in \mathbb{R}^{m \times k}$ and $H \in \mathbb{R}^{k \times n}$ to approximate the original data $V \in \mathbb{R}^{m \times n}$ $(k \ll min(m, n))$, has gained its popularity in many real applications, such as face recognition, text mining, signal processing, etc [1].

Take documents in the vector space model for instance. The documents are encoded as a term-by-document matrix V with nonnegative elements, and each column of V represents a document and each row a term. NMF produces k basic topics as the columns of the factor W and the coefficient matrix H. Observed from H, it is easy to derive how each document is fractionally constructed by

M.J. Zaki et al. (Eds.): PAKDD 2010, Part II, LNAI 6119, pp. 214–221, 2010.

the resulting k basic topics. Also, the factor H is regarded as a cluster indicator matrix for document clustering, each row of which suggests which documents are included in a certain topic. Similarly, the factor W can be treated as a cluster indicator matrix for word clustering. Traditional clustering algorithms, taking k-means for instance, require the product between the row vectors or column vectors to be 0 that only one value exists in each row of H; thus each data point only belongs to one cluster, which leads to a hard clustering. It was proved that orthogonal nonnegative matrix factorization is equivalent to k-means clustering [4]. Compared to rigorous orthogonality of k-means, relaxed orthogonality means each data point could belong to more than one cluster, which can improve clustering quality [5,10]. Simultaneous clustering refers to clustering of the rows and columns of a matrix at the same time. The major property of simultaneous clustering is that it adaptively performs feature selection as well as clustering, which improves the performance for both of them [2,3,6,12]. Some applications such as clustering words and documents simultaneously for an input term-by-document matrix, binary data, and system log messages were implemented [9]. For this purpose, Orthogonal Nonnegative Matrix Tri-Factorization (ONMTF) was proposed [5]. It produces two nonnegative indictor matrices W and H, and another nonnegative matrix S such that $V \approx WSH$. Orthogonality constraints were imposed on W and H to achieve relaxed orthogonality. However, in their methods, to achieve relaxed orthogonality, Lagrangian multipliers have to be determined for the Lagrangian function of ONMTF. Solving the Lagrangian multipliers accounts for an intensive computation of update rules for the factors, especially the factor W whose size is larger than other factors. In this paper, we introduce Fast Orthogonal Nonnegative Matrix Tri-Factorization (FONT), whose computational complexity is decreased significantly by setting the Lagrangian multipliers as approximate constants. In addition, FONT is further accelerated by using Alternating Least Squares [11], which leads to a fast convergence.

The rest of the paper is organized as follows. In Section 2, related work is reviewed, including NMF and ONMTF. Section 3 introduces our methods in detail, followed by the experiments and evaluations in Section 4. Finally, conclusions are described in Section 5.

2 Related Work

Given a data matrix $V = [v_1, v_2, ..., v_n] \in \mathbb{R}^{m \times n}$, each column of which represents a sample and each row a feature. NMF aims to find two nonnegative matrices $W \in \mathbb{R}^{m \times k}, H \in \mathbb{R}^{k \times n}$, such that $V \approx WH$, where $k \ll min(m, n)$. There is no guarantee that an exact nonnegative factorization exists. Iterative methods become necessary to find an approximate solution to NMF which is a nonlinear optimization problem with inequality constraints. To find an approximate factorization of NMF, an objective function has to be defined by using some measurements of distance. A widely used distance measurement is the Euclidean distance which is defined as:

$$\min_{W,H} \|V - WH\|^2 \quad s.t. \quad W \geq 0, \quad H \geq 0 \tag{1}$$

where the $\|\cdot\|$ is Frobenius norm. To find a solution to this optimization problem, the multiplicative update rules were first investigated in [8] as follows:

$$W := W * (VH^T)/(WHH^T) \tag{2}$$

$$H := H * (W^TV)/(W^TWH) \tag{3}$$

where * and / denote elementwise multiplication and division, respectively. ONMTF was conducted for the application of clustering words and documents simultaneously by imposing additional constraints on W and/or H. The objective function for ONMTF can be symbolically written as:

$$F = \min_{W,S,H \geq 0} \|V - WSH\|^2 \quad s.t. \quad HH^T = D, \quad W^TW = D \tag{4}$$

where D is a diagonal matrix. By introducing the Lagrangian multipliers the Lagrange L is:

$$L = \|V - WSH\|^2 + Tr[\lambda_w(W^TW - D)] + Tr[\lambda_h(HH^T - D)] \tag{5}$$

The multiplicative update rules for (8) can be computed as follows:

$$W = W * (VH^TS^T)/(W(HH^T + \lambda_w)) \tag{6}$$

$$S = S * (W^TWH^T)/(W^TSHH^T) \tag{7}$$

$$H = H * (S^TW^TV)/((W^TW + \lambda_h)H) \tag{8}$$

By solving the minimum $W^{(t+1)}$ and $H^{(t+1)}$, respectively [5]. λ_w and λ_h can be approximately computed as follows:

$$\lambda_w = D^{-1}W^TVH^TS^T - HH^T \tag{9}$$

$$\lambda_h = D^{-1}S^TW^TVH^TH - W^TW \tag{10}$$

Substituting λ_w and λ_h in (7) and (8) respectively, we obtain following update rules:

$$W = W * (VH^TS^T)/(WW^TVH^TS^T) \tag{11}$$

$$H = H * (S^TW^TV)/(S^TW^TVH^TH) \tag{12}$$

Based on matrix multiplication, the computational complexity of NMF based on the Euclidean distance metric at each iteration is $O(mnk)$, and that of ONMTF is $O(m^2n)$. The computation of the Lagrangian multipliers accounts for an intensive computation. It becomes worse when m increases, which represents the number of words in a vector space model.

3 Fast Orthogonal Nonnegative Matrix Tri-Factorization

It was proved that the Lagrange L is monotonically non-increasing under the above update rules by assuming $W^TW + \lambda_w \geq 0$ and $HH^T + \lambda_h \geq 0$ [5]. We note that λ_w and λ_h are approximately computed under these assumptions, and from (9) and (10) we can see that λ_w and λ_h are symmetric matrices of size $k * k$. Since achieving relaxed orthogonality is the purpose of orthogonal matrix factorization in this paper, and computing the lagrangian multipliers accounts for an intensive computation, we would use constants for λ_w and λ_h for decreasing computational complexity. By normalizing each column vector of W and each row vector of H to unitary Euclidean length at each iteration, λ_w and λ_h can be approximately denoted by minus identity matrix ($\lambda_w = \lambda_h = -I$). Thus, we introduce our method *Fast Orthogonal Nonnegative Matrix Tri-Factorization* (FONT).

3.1 FONT

The Lagrange L is rewritten as:

$$L = \min_{W,S,H \geq 0} (\|V - WSH\|^2 + Tr(I - W^TW) + Tr(I - HH^T)) \tag{13}$$

where I is the identity matrix. Noting $\|V - WSH\|^2 = Tr(VV^T) - 2Tr(WSHV^T) + Tr(WSHH^TS^TW^T)$, the gradient of L with respect to W and H are:

$$\partial L/\partial W = -2VH^TS^T - 2W + 2WSHH^TS^T \tag{14}$$

$$\partial L/\partial H = -2S^TW^TV - 2H + 2S^TW^TWSH \tag{15}$$

By using the Karush-Kuhn-Tucker conditions the update rules for W and H can be inferred as follows:

$$W = W * (VH^TS^T + W)/(WSHH^TS^T) \tag{16}$$

$$H = H * (S^TW^TV + H)/(S^TW^TWSH) \tag{17}$$

Because of no orthogonality constraint on S, the update rule for S, in both FONT and ONMTF, is the same. The computational complexity of FONT, at each iteration, is $O(mnk)$, far less than $O(m^2n)$ because $k \ll min(m, n)$.
Now we give the convergence of this algorithm by using the following theorem (we use H as an example here, and the case for W can be conducted similarly):

Theorem 1. *The Lagrange L in (13) is non-increasing under the update rule in (17).*

To prove this theorem, we use the auxiliary function approach [8]. $G(h, h')$ is an auxiliary function for $F(h)$ if the conditions $G(w, w') \geq F(w)$ and $G(w, w) = F(w)$ are satisfied. If G is an auxiliary function, then F is nonincreasing under the updating rule $w^{t+1} = arg\min_w G(w, w^t)$. Because $F(w^{t+1}) \leq G(w^{t+1}, w^t) \leq G(w^t, w^t) = F(w^t)$ [8]. So it is crucial to find an auxiliary function. Now we show that

$$G(h, h^t_{ij}) = L_{ij}(h^t_{ij}) + L'_{ij}(h^t_{ij})(h - h^t_{ij}) + (h - h^t_{ij})^2 * (S^T W^T WSH)_{ij}/h^t_{ij} \quad (18)$$

is an auxiliary function for L.

Proof. Apparently, $G(h, h) = L_{ij}(h)$, so we just need to prove that $G(h, h^t_{ij}) \geq L_{ij}(h)$. We expand $L_{ij}(h)$ using Taylor series.

$$L_{ij}(h) = L_{ij}(h^t_{ij}) + L'_{ij}(h^t_{ij})(h - h^t_{ij}) + [(S^T W^T WS)_{ii} - 1](h - h^t_{ij})^2 \quad (19)$$

Meanwhile,

$$\begin{aligned}(S^T W^T WSH)_{ij} &= \textstyle\sum_{p=1}^{k} (S^T W^T WS)_{ip} H_{pj} \\ &\geq (S^T W^T WS)_{ii} H_{ij} > ((S^T W^T WS)_{ii} - 1)h^t_{ij}\end{aligned} \quad (20)$$

Thus we have $G(h, h^t_{ij}) \geq L_{ij}(h)$. Theorem 1 then follows that the Lagrangian L is nonincreasing.

3.2 FONT + ALS

However, we still note that the factor W accounts for a larger computation than the other two factors, thus we consider to compute W by using Alternating Least Squares (ALS). ALS is very fast by exploiting the fact that, while the optimization problem of (1) is not convex in both W and H, it is convex in either W or H. Thus, given one matrix, the other matrix can be found with a simple least squares computation. W and H are computed by equations $W^T WH = W^T V$ and $HH^T W^T = HA^T$, respectively. Reviewing (4) for W, F can be rewritten as:

$$(W^T W + SHH^T S^T)W^T = W^T + SHV^T \quad (21)$$

Then an approximate optimal solution to W is obtained by using ALS. To maintain nonnegativity, all negative values in W should be replaced by zero.

4 Experiments

5 document databases from the CLUTO toolkit (`http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download`) were used to evaluate our algorithms. They are summarized in Table 1. Considering the large memory requirement for the matrix computation, we implemented all algorithms in Matlab R2007b on a 7.1 teraflop computing cluster which is an IBM e1350 with an 1 4-way (Intel Xeon MP 3.66GHz) shared memory machine with 32GB. The memory was requested between 1G to 15G for different datasets. 15G memory was required for the ONMTF algorithm to run on 2 largest datasets *la12* and *class*.

Table 1. Summary of Datasets

Dataset	Source	# Classes	# Documents	# Words
classic	CACM/CISI/Cranfield/Medline	4	7094	41681
reviews	San Jose Mercury(TREC)	5	4069	18483
klb	WebACE	6	2340	21839
la12	LA Times(TREC)	6	6279	31472
ohscal	OHSUMED-233445	10	11162	11465

4.1 Evaluation Metrics

We also use purity and entropy to evaluate the clustering performance [5]. Purity gives the average ratio of a dominating class in each cluster to the cluster size and is defined as:

$$P(k_j) = \frac{1}{k_j} max(h(c_j, k_j)) \tag{22}$$

where $h(c, k)$ is the number of documents from class c assigned to cluster k. The larger the values of purity, the better the clustering result is.
The entropy of each cluster j is calculated using the $E_j = \sum_i p_{ij} log(p_{ij})$, where the sum is taken over all classes. The total entropy for a set of clusters is computed as the sum of entropies of each cluster weighted by the size of that cluster:

$$E_C = \sum_{j=1}^{m} (\frac{N_j}{N} \times E_j) \tag{23}$$

where N_j is the size of cluster j, and N is the total number of data points. Entropy indicates how homogeneous a cluster is. The higher the homogeneity of a cluster, the lower the entropy is, and vice versa.

4.2 Performance Comparisons

In contrast to document clustering, there is no prior label information for word clustering. Thus, we adopt the class conditional word distribution that was used in [5]. Each word belongs to a document class in which the word has the highest frequency of occurring in that class. All algorithms (FONT$_{ALS}$ stands for FONT combined with ALS) were performed by using the stopping criterion $1 - F^{t+1}/F^t \leq 0.01$, where F is $\|V - WSH\|^2$ and calculated by every 100 iterations. The comparison for both word and document clustering are shown in Table 2 and Table 3 respectively. All results were obtained by averaging 10 independent trails.

We observe that the FONT algorithms (including FONT$_{ALS}$) achieve better purity than ONMTF for both words and documents clustering. It also shows that FONT obtains lower entropy for word clustering than ONMTF. But for document clustering, the clusters obtained by ONMTF are more homogenous than FONT and FONT$_{ALS}$. Meanwhile, in Table 4, it is shown that FONT and FONT$_{ALS}$ are significantly faster than ONMTF. In particular, the running time

Table 2. Comparison of Word Clustering

Dataset	Purity			Entropy		
	ONMTF	FONT	FONT$_{ALS}$	ONMTF	FONT	FONT$_{ALS}$
classic	0.5077	0.5153	0.5577	0.6956	0.6881	0.6309
reviews	0.5905	0.6298	0.6001	0.6850	0.6656	0.7003
klb	0.7258	0.7356	0.7335	0.4546	0.4486	0.4478
la12	0.4612	0.4823	0.4721	0.7693	0.7569	0.7794
ohscal	0.3740	0.4056	0.2991	0.7601	0.7297	0.8331

Table 3. Comparison of Document Clustering

Dataset	Purity			Entropy		
	ONMTF	FONT	FONT$_{ALS}$	ONMTF	FONT	FONT$_{ALS}$
classic	0.5484	0.5758	0.6072	0.6246	0.6359	0.6661
reviews	0.7312	0.7635	0.7540	0.7775	0.8097	0.8126
klb	0.8021	0.8095	0.8118	0.8317	0.8389	0.8366
la12	0.4978	0.5176	0.5379	0.5665	0.5883	0.6063
ohscal	0.3581	0.3983	0.3616	0.4305	0.4682	0.4340

of FONT$_{ALS}$ is 12.16 seconds on the largest dataset *classic*, compared to 36674 seconds ONMTF used and 1574.2 seconds NMTF used, which indicates that the FONT algorithms are effective in terms of clustering quality and running time.

Table 4. Comparison of Running Time (s)

Dataset	ONMTF	FONT	FONT$_{ALS}$
classic	3.6674e+4	1.5985e+3	**12.16**
reviews	2.0048e+4	370.36	**25.12**
klb	1.0852e+4	275.94	**14.07**
la12	4.5239e+4	1.0496e+3	**41.45**
ohscal	1.0051e+4	767.86	**37.29**

5 Conclusions

The Orthogonal Nonnegative Matrix Tri-Factorization algorithm needs a large computation to achieve relaxed orthogonality, which makes it infeasible for clustering large datasets in terms of computational complexity and a large requirement of memory. In our research, to achieve relaxed orthogonality, we have introduced our method Fast Nonnegative Matrix Tri-Factorization (FONT). By using unitary matrix to estimate the Lagrangian multipliers, the computational complexity is reduced and clustering quality is improved as well. Meanwhile, by using Alternating Least Squares, FONT is further accelerated, which leads to a significant decrease of running time.

Acknowledgements

This research is supported by the US National Science Foundation (NSF) under grant CCF-0905337 and EPS-0701410, the National Natural Science Foundation of China (NSFC) under award 60828005, and the 973 Program of China under award 2009CB326203.

References

1. Berry, M.W., Browne, M., Langville, A.N., Pauca, P.V., Plemmons, R.J.: Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics & Data Analysis 52(1), 155–173 (2007)
2. Boley, D.: Principal direction divisive partitioning. Data Min. Knowl. Discov. 2(4), 325–344 (1998)
3. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 89–98. ACM, New York (2003)
4. Ding, C., He, X., Simon, H.D.: On the equivalence of nonnegative matrix factorization and spectral clustering. In: SIAM Data Mining Conference (2005)
5. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix tri-factorizations for clustering. In: KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 126–135. ACM, New York (2006)
6. Koyutürk, M., Grama, A., Ramakrishnan, N.: Nonorthogonal decomposition of binary matrices for bounded-error data compression and analysis. ACM Trans. Math. Softw. 32(1), 33–69 (2006)
7. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–791 (1999)
8. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Neural Information Processing Systems, pp. 556–562 (2001)
9. Li, T., Ding, C.: The relationships among various nonnegative matrix factorization methods for clustering. In: ICDM '06: Proceedings of the Sixth International Conference on Data Mining, Washington, DC, USA, pp. 362–371. IEEE Computer Society, Los Alamitos (2006)
10. Li, Z., Wu, X., Peng, H.: Nonnegative matrix factorization on orthogonal subspace. Pattern Recognition Letters (to appear, 2010)
11. Paatero, P., Tapper, U.: Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics 5(2), 111–126 (1994)
12. Zhang, Z., Li, T., Ding, C., Zhang, X.: Binary matrix factorization with applications. In: ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, Washington, DC, USA, pp. 391–400. IEEE Computer Society, Los Alamitos (2007)