Volume 3, Issue 10, October 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Image compression using Constrained Non-Negative Matrix Factorization

Srilakshmi Inuganti, Veerraju Gampala
Information Technology, GMRIT,
Rajam, AP, India-532127

Abstract*— Non-negative matrix factorization (NMF) is a recently developed method to obtain a representation of data using non-negativity constraints. In this paper existing techniques for Non-negative matrix factorization are studied and a constrained non-negative matrix factorization (CNMF) for image compression is proposed. The results of experiments are reported to show the effectiveness of the proposed method, in which CNMF converges fast when compared to multiplicative update and gradient descent methods of NMF. Additionally, we provide simple MATLAB code for both standards NMF and for our extension.

Keywords: *image compression, non-negative matrix factorization,Constrained non-negative matrix factorization*

I. INTRODUCTION

Image Compression, an important area in the field of digital image processing, deals with techniques for reducing the storage required for saving an image or the bandwidth required for transmitting it [1]. The objective of Image Compression is to reduce irrelevance and redundancy of the image data thereby optimizing the storage space and increasing the transmission rate over WebPages. This paper focuses on Non-Negative Matrix factorization (NMF) which is superior over standard factorization of data matrix uses singular value decomposition (SVD). For many dataset such as images and text, the original data matrices are nonnegative. A factorization such as SVD contains negative entries and thus has difficulty for interpretation. Nonnegative matrix factorization (NMF) [2,3] has many advantages over standard PCA/SVD based factorizations. In contrast to cancellations due to negative entries in matrix factors in SVD based factorizations; the no negativity in NMF ensures factors contain coherent parts of the original data (images).Rest of the paper is constituted as follows: Section II focuses on basic Non-Negative matrix factorization algorithm, Section III focuses on application of Constrained Non-Negative matrix factorization in image compression contains algorithm and Mat Lab implementation, Results obtained and comparisons shown in Section IV. Finally, conclusion of the paper presented in Section V.

II. NON-NEGATIVE MATRIX FACTORIZATION

The nonnegative matrix factorization (NMF,cf.[3,4]) consists of reduced rank nonnegative factors $W \epsilon R^{m\times k}$ and $H \epsilon R^{k\times n}$ with $k<<\min\{m,n\}$ that approximate a given nonnegative data matrix $A \epsilon R^{m\times n}$: $A \approx WH$. The nonnegativity constraints require that all entries in A,W and H are zero or positive. Although the product WH is only an approximate factorization of A of rank at most k,WH is called a nonnegative matrix factorization of A. The nonlinear optimization problem underlying NMF can generally be stated as

$$\min_{W,H} f(W,H)=1/2\|A-WH\|^2_F \qquad (1)$$

where ||.||F is the Frobenius norm ($(\|A\|F=(\sum|aij|^2)1/2)$. Frobenius norm is commonly used to measure the error between the original data A and WH. Unlike SVD, the NMF is not unique, and the convergence is not guaranteed for all NMF algorithms.

Due to its non-negativity constraints, NMF produces so-called "additive parts based" representations of the data(in contrast to many other representations such as SVD,PCA or ICA).

This is an important benefit of NMF,since it makes the interpretation of the NMF factors much easier than for factors containing positive and negative entries, and enables NMF a non-suitable combination of the parts to form a whole[3]. Another favorable consequences of the non negativity constraints is that both factors W and H are often naturally sparse.

1. Algorithms for computing NMF

Most existing NMF algorithms in the literature can be assigned to one of three general classes: multiplicative update (MU), alternating least squares (ALS) and gradient descent (GD) algorithms.A review of these can be found in,for example in [5]

2. Pseudo code for the general structure of all NMF algorithms is given below:

Algorithm1:Pseudocode for general NMF algorithm

```
Given matrix AϵRm×n and k<<min{m,n}
for iter=1 to maxrep do
W=rand(m,k)
        H=rand(k,n);
for t=1 to maxiter do
update W and H
check termination criterion
end for
end for
```

The variable maxrepition specifies the number of iterations of the complete algorithm for the case of randomly initialized W and H. In each repetition, NMF update steps are processed iteratively until a maximum no. of iterations is reached. The approach for the NMF with multiplicative update algorithm is given below. If the approximation error of the algorithm drop below a predefined threshold, or if the change between two successive iterations is very small, the alg. may terminate before maximum iterations are processed.

3. Multiplicative Update(MU) Algorithm

The prototypical multiplicative algorithm originated with Lee and Seung (2001) in [6]. Their multiplicative update algorithm with the mean squared error objective function is provided below

---

Algorithm 2:Pseudocode for Multiplicative Update algorithm for NMF

```
W=rand(m,k);   %initialize W as random dense matrix
 H=rand(k,n);    %initialize H as random dense matrix
for i=1:maxtier
(MU)  H=H*(WTA)/(WTWH+10-9);
(MU)  W=W*(AHT)/(WHHT+10-9);
End
```

The 10^{-9} in each update rule is added to avoid division by zero. Lee and Seung used the gradient and properties of continual descent to claim that the above algorithm converges to local minimum which was later shown to be incorrect. [7,8,9,10]. In fact, the proof by Lee and Seung merely shows a continual descent property which does not preclude to a saddle point. When the algorithm has converged to a limit point in the interior of the feasible region, this point is a stationary point. This stationary point may or may not to be a local minimum[5]. When the point lies on the boundary of the feasible region, its stationary cannot be determined. This is the first well-known NMF algorithm the Lee and Seung multiplicate update algorithms have become baseline for the many new algorithms. It has been proved that, when they converge, are slow to converge. They require many iterations, and the work per iteration is high since each iteration requires O(mnk) work.

III. CONSTRAINED NON-NEGATIVE MATRIX FACTORIZATION IN IMAGE PROCESSING

The NMF problem formulation given in equation 1 is sometimes extended to include auxiliary constraints on W and/or H. This is often done to compensate for uncertainties in the data, to enforce desired characteristics in the computed solution, or to impose prior knowledge about the application at hand. Penalty terms are typically used to enforce auxiliary constraints, extending the cost function of Eq. (1) as follows:

$$f(W,H)=\|A-WH\|^2_F + \alpha J_1(W)+\beta J_2(H) \qquad (2)$$

Here $J_1(W)$ and $J_2(H)$ are the penalty terms introduced to enforce certain application-dependent constraints, and α and β are small regularization parameters that balance the trade-off between the approximation error and the constraints.Smoothness constraints are often enforced to regularize the computed solutions in the presence of noise in the data.

For example the term,

$$J_1(W)=\|W\|^2_F \qquad (3)$$

penalizes W solutions of large Frobenius norm. Notice that this term is implicitly penalizing the columns of W since $\|W\|^2_F = \sum i \|wi\|22$. In practice, the columns of W are often normalized to add up to one in order to maintain W away

from zero. This form of regularization is known as Tikhonov regularization in the inverse problems community. More generally, one can rewrite (3) as J1(W)=||LW||22,,where L is a regularization operator. Other choices than the identity for L include Laplacian operators. Smoothness constraints can be applied likewise to H, depending on the application needs. For example, temporal smoothness can be enforced in the columns of H by defining [11]as

$$J_2(H) = 1/n\sum_i ||(I-T)h_i^T||^2_F, \qquad (4)$$

where n is the total number of columns in the data matrix A and T is an appropriately defined convolution operator.

Sparsity constraints on either W or H can be similarly imposed. The notion of sparsity refers sometimes to a representational scheme where only a few features are effectively used to represent data vectors [12,13].It also appears to refer at times to the extraction of local rather than global features, the typical example being local facial features extracted from the CBCL and ORL face image databases [13]. Measures for sparsity include, for example, the lp norms for $0<p\leq1$[14] and Hoyer's measure,

sparseness(x) $=(\sqrt{n} - ||x||1 / ||x||2)/\sqrt{n} - 1$

The latter can be imposed as a penalty term of the form

$$J_2(H) = (\omega||vec(H)_2 - ||vec(H)_1)^2, \qquad (5)$$

where $\omega=\sqrt{kn} - (\sqrt{kn} - 1)\gamma$ and vec(.) is the vec operator that transforms a matrix into a vector by stacking its columns. The desired sparseness in H is specified by setting γ to a value between 0 and 1.

In certain applications such as hyper spectral imaging, a solution pair (W,H) must comply with constraints that make it physically realizable[15]. One such physical constraint requires mixing coefficients h_{ij} to sum to one, i.e.,

$\sum_i h_{ij} = 1$ for all j. Enforcing such a physical constraint can significantly improve the determination of inherent features[18], when the data are in fact linear combinations of these features. Imposing additivity to

one in the columns of H can be written as a penalty term in the form

$$J_2(H) = ||H^T e_1 - e_2||, \qquad (6)$$

where e1 and e2 are vectors with all entries equal to 1. This is the same as requiring that H be column stochastic[16] or alternatively that the minimization of (3) seek solutions W whose columns form a convex set containing the data vectors in A. Notice, however, that full additivity is often not achieved since

$H^T e1 \approx e2$ depending on the value of the regularization parameter β.

Of course, the multiplicative update rules for W and H in the alternating gradient descent mechanism of Lee and Seung change when the extended cost function (2) is minimized. In general assuming that J1(W) and J2(H) have partial derivatives with respect to wij and hij , respectively, the update rules can be formulated as wij

$$W_{ij}^{(t)} = W_{ij}^{(t-1)} \cdot \frac{(AH^T)_{ij}}{(W^{(t-1)}HH^T)_{ij}+\alpha(\partial J_1(W)/\partial w_{ij})}$$

$$H_{ij}^{(t)} = H_{ij}^{(t-1)} \cdot \frac{(W^T A)_{ij}}{(W^T W H^{(t-1)})_{ij}+\beta(\partial J_2(H)/\partial h_{ij})}$$

The extended cost function is non-increasing with these update rules for sufficiently small values of α and β [17].We modified NMF with the extended cost function (3) and with smoothness constraints as in (4) to applications in the field of data analysis and image processing.This modified update rules converges fast than the original update rules.

The algorithm, denoted CNMF [18] is specified below for completeness.

Algorithm 3:Pseudocode for Constrained Nonnegative Matrix Factorization

---

```
W=rand(m,k);   %initialize W as random dense matrix
H=rand(k,n);    %initialize H as random dense matrix
for i=1:maxtier
H=H*(W^TA)/(W^TWH+βH+10^-9);
W=W*(AH^T)/(WHH^T+αW+10^-9);
End
```

IV. RESULTS AND DISCUSSIONS

1. Implementation

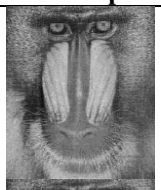
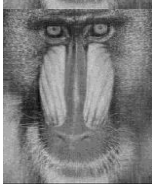
The following is the code for CNMF in MATLAB.

```
a=imread('images\puppy.tif');
maxiter = 100;
tol=0.6
a=rgb2gray(a);
a = im2double(a);
imshow(a)
[n,m]=size(a);
k=min(n,m);
w0=rand(n,k);
h0=rand(k,m);
x=10^-2;
for j=1:maxiter
        numer = w0'*a;
        h = h0 .* (numer) ./ ((w0'*w0)*h0+x*h0 +eps(numer));
        h=h.*(h>0);
        numer = a*h';
        w = w0 .* (numer) ./ (w0*(h*h')+x*w0+eps(numer));
        w=w.*(w>0);
        w0 = w; h0 = h;
        if(norm(a-w*h)<tol)
        break;
        end
end
v=w*h;
figure,imshow(v)
imwrite(v,'images\cnfpuppy.tif')
```

Results

Four grayscale images, which are commonly used in the community of image processing and compression, were used to do experiment in this paper. The evolution of the proposed method is made on the basis of the results of compression, image quality and CPU processing time on the four images.

The compression ratio is defined as the ration of the number of bits in the original image file to the no.of bits in the compressed file. The quality of image is determined by the peak signal-to-noise ratio (PSNR) which determines the difference between the two images. It is defined as PSNR= 20log10 (b/rms) db, where b is the largest possible values of the signal or pixl value, and the rms is the root-mean-square difference between two images. The PSNR is given in units of decibel (dB), which measures the ration of the peak signal and difference between two images. That is to say, the higher the value, the better the image quality. For each image, the results are measured for all the four methods. It is observed that CNMF methods consistently converges faster than the other methods by maintaining compression ration and image quality.

Original image				
Compressed images				
NMF(MU)				
CNMF				

PSNR value				
NMF(MU)	81.4625	74.5902	80.6327	83.7712
CNMF	81.5718	73.9736	80.1075	83.5732
Compression Ratio				
NMF(MU)	0.7187	0.9083	0.7826	0.2616
CNMF	0.7062	0.841	0.7826	0.2591
Speed(sec)				
NMF(MU)	9.023311	1.689906	5.84489	1.244554
CNMF	4.336461	0.779088	2.81434	0.560941

Fig.1 Original images, compressed images of multiplicative update NMF and proposed CNMF compressed images with their PSNR value, Compression ratio and convergence speed

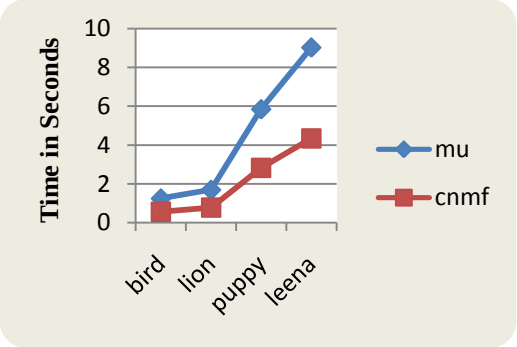


Fig.2.convergence comparision of Existing method vs propsed method

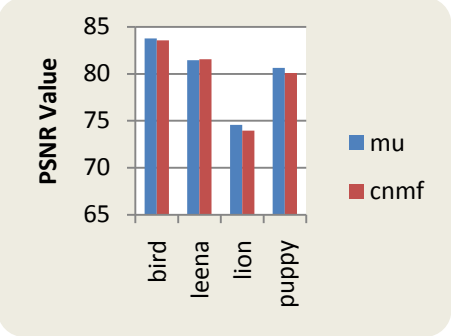


Fig.3.PSNR comparision of Existing method vs propsed method

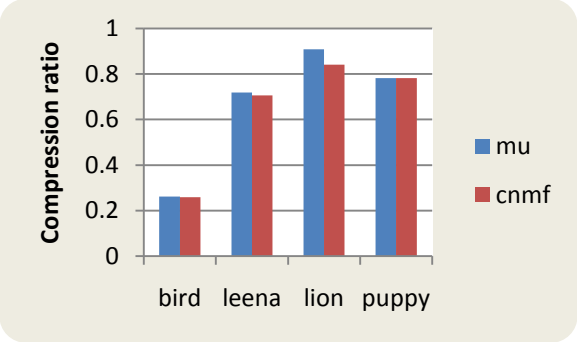


Fig.4. compression ratio comparision of Existing method vs propsed method

V. CONCLUSION

In this paper we reported image compression based on constrained nonnegative matrix factorization. This CNMF(constrained NMF) increases the compression ratio by 10% than other NMF methods. Furthermore, this converges fast. The compression results were experimentally shown. The proposed method can be paralyzed to converge fast further.

REFERENCES

[1] Rafael C.Gonzales and Richard E.Woods, Digital image processing, Pearson Education, 2001, 2nd edition

[2] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401:788–791, 1999.

[3] D.D. Lee and H.S. Seung. Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems, 13. 2001.

[4] Paatero, P.—Tapper, U.: Positive Matrix Factorization: A Nonnegative Factor Model With Optimal Utilization of Error Estimates of Data Values. Environmetrics,Vol. 5, 1994, No. 2, pp. 111–126.

[5] 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, Vol. 52, 2007, No. 1, pp. 155–173.

[6] Lee DD, Seung HS (2001) Algorithms for non-negative matrix factorization. In: Proceedings of Neural Information Processing Systems, vol 13, pp 556–562

[7] Chu, M., Diele, F., Plemmons, R., Ragni, S., 2004. Optimality, computation, and interpretations of nonnegative matrix factorizations, available at http://www.wfu.edu/∼plemmons.

[8] Finesso, L., Spreij, P., 2004. Approximate nonnegative matrix factorization via alternating minimization. In: Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems, Leuven, Belgium, July 5–9.

[9] Gonzalez, E., Zhang, Y., 2005. Accelerating the Lee–Seung algorithm for nonnegative matrix factorization. Technical Report TR-05-02, Rice University.

[10] Lin, C.-J., 2005b. Projected gradient methods for non-negative matrix factorization. Technical Report Information and Support Services Technical Report ISSTECH-95-013, Department of Computer Science, National Taiwan University.

[11] Chen, Z., Cichocki, A., 2005. Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints, preprint.

[12] Hoyer, P., 2002. Non-negative sparse coding. In: Proceedings of the IEEEWorkshop on Neural Networks for Signal Processing. Martigny, Switzerland

[13] Hoyer, P., 2004. Non-negative matrix factorization with sparseness constraints. J. Mach. Learning Res. 5, 1457–1469.

[14] Karvanen, J., Cichocki, A., 2003. Measuring sparseness of noisy signals. In: Proceedings of the Fourth International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), Nara, Japan.

[15] Keshava, N., 2003. A survey of spectral unmixing algorithms. Lincoln Laboratory J. 14 (1), 55–77.

[16] Berman, A., Plemmons, R., 1994. Non-Negative Matrices in the Mathematical Sciences. SIAM Press Classics Series, Philadelphia, PA.

[17] Chen, Z., Cichocki, A., 2005. Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints, preprint.

[18] Pauca, P., Piper, J., Plemmons, R., 2006a. Nonnegative matrix factorization for spectral data analysis. Linear Algebra Appl. 416 (1), 29–47.

