

텐서의 비음수 Tucker 분해 (Nonnegative Tucker Decomposition)

김 용 덕 [†] 최 승 진 ^{**}
(Yong-Deok Kim) (Seungjin Choi)

요 약 최근에 개발된 Nonnegative tensor factorization(NTF)는 비음수 행렬 분해(NMF)의 multiway(multilinear) 확장형이다. NTF는 CANDECOMP/PARAFAC 모델에 비음수 제약을 가한 모델이다. 본 논문에서는 Tucker 모델에 비음수 제약을 가한 nonnegative Tucker decomposition(NTD)라는 새로운 텐서 분해 모델을 제안한다. 제안된 NTD 모델을 least squares, I-divergence, α -divergence를 이용한 여러 목적함수에 대하여 fitting하는 multiplicative update rule을 유도하였다.

키워드 : 부공간 분석, 비음수 행렬 분해, 텐서 분해

Abstract Nonnegative tensor factorization (NTF) is a recent multiway (multilinear) extension of nonnegative matrix factorization (NMF), where nonnegativity constraints are imposed on the CANDECOMP/PARAFAC model. In this paper we consider the Tucker model with nonnegativity constraints and develop a new tensor factorization method, referred to as nonnegative Tucker decomposition (NTD). We derive multiplicative updating algorithms for various discrepancy measures: least square error function, I-divergence, and α -divergence.

Key words : Subspace Analysis, Nonnegative Matrix Factorization, Tensor Factorization

- 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원 사업의 연구결과로 수행되었음(IITA-2008-C1090-0801-0045)
- 이 논문은 제34회 추계학술대회에서 '텐서의 비음수 Tucker 분해'의 제목으로 발표된 논문을 확장한 것임

[†] 학생회원 : 포항공과대학교 컴퓨터공학과
karma13@postech.ac.kr

^{**} 종신회원 : 포항공과대학교 컴퓨터공학과 교수
seungjin@postech.ac.kr

논문접수 : 2007년 12월 14일

심사완료 : 2008년 3월 5일

Copyright©2008 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제3호(2008.5)

1. 서론

부공간 분석(subspace analysis), 주성분 분석(principal component analysis, PCA)은 데이터 행렬 $\mathbf{X}=[x_1 x_2 \cdots x_l] \in \mathbb{R}^{m \times l}$ 를 분석하기 위해 널리 쓰이는 선형 모델이다. 이 선형 모델은 목적 함수 $\|\mathbf{X}-\mathbf{AS}\|^2$ ($\|\cdot\|$ 은 행렬의 Frobenious norm을 의미)을 최소화 하는 기저(basis) 행렬 $\mathbf{A}=[a_1 a_2 \cdots a_R] \in \mathbb{R}^{m \times R}$ 과 계수(encoding variable) 행렬 $\mathbf{S} \in \mathbb{R}^{R \times l}$ 을 찾는 것을 목표로 한다. PCA의 변형된 형태인 독립성분 분석(independent component analysis, ICA)[1] 역시 널리 사용되는 선형 모델이다. PCA는 계수 행렬의 가로 벡터들이 통계적으로 비상관적이 되는 선형 모델 $\mathbf{X}=\mathbf{AS}$ 을 학습한다. 반면 ICA는 계수 행렬의 가로 벡터들의 통계적인 독립성을 최대화하는 선형모델을 학습하는 것을 목표로 한다. 데이터 행렬의 원소가 모두 비음수인 경우에는, 비음수 행렬 분해(NMF)[2]가 부분국부적 표현(parts-based representation)과 특징 추출(feature extraction)에 유용하게 사용된다.

컴퓨터 비전에서 다루는 이미지, 비디오 데이터는 텐서(tensor)라고 불리는 3-way 혹은 multi-way 데이터 배열로 잘 표현될 수 있다. Multi-way 구조는 가로, 세로, RGB(혹은 HSV) 색상 채널, 시간 등을 반영한다.

기존의 부공간 분석이나 행렬 분해 방법들은 2D 이미지 데이터를 1D 이미지 벡터로 변환시켰다. 이 과정에서 2D 이미지 데이터의 공간 구조가 버려지게 되었다. 최근 들어 2D-PCA[3]와 2D-NMF[4]와 같은 부공간 분석 방법의 2D 확장이 제안되었다.

Multilinear algebra와 텐서 분해는 부공간 분석 방법들의 2D 확장과 그보다 더 일반적인 multiway 구조를 반영하는 프레임워크이다. 대표적인 텐서 분해 방법으로는 (1) Tucker 모델(multilinear SVD 혹은 N-mode SVD로도 불림)[5,6] ; (2) PARAFAC 모델[7]; (3) PARAFAC 모델 혹은 PARAFAC2 모델에 비음수 제약을 가한 nonnegative tensor factorization(NTF)[8,9] 등이 있다.

기존의 NTF는 PARAFAC 모델에 기반하여 NMF를 multiway 모델로 확장 시켰다. 본 논문에서는 Tucker model을 이용하여 NMF를 multiway 모델로 확장하고자 한다. 본 논문에서는 Tucker 모델을 구성하는 core 텐서와 mode 행렬들 모두에게 비음수 제약을 두는 것을 고려한다. 그리고 multiplicative update rule을 통한 model fitting 방법을 제안한다. 기존의 NTF와의 혼동을 피하기 위해 제안된 방법을 nonnegative Tucker decomposition(NTD)라 명명한다. NTD의 core 텐서가 특정 구조를 갖게끔 고정시킴으로써 NMF, NTF,

nsNMF[10]를 NTD의 특정한 경우로 표현할 수 있으며, 앞서 언급한 방법들의 multiplicative updating rule 역시 NTD의 updating rule로부터 재유도 할 수 있다.

2. 배경: Multilinear algebra

N-way 텐서 $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ 는 N개의 인덱스 (i_1, i_2, \dots, i_N) 을 가지며 그 원소들은 x_{i_1, i_2, \dots, i_N} 로 표현된다. $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ 을 mode-n 행렬화 한다는 것은 $\underline{\mathbf{X}}$ 의 원소들을 재배열하여 행렬 $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_{n+1} I_{n+2} \dots I_{N-1} I_1 I_2 \dots I_{n-1}}$ 로 만드는 것을 의미한다. 행렬의 열 부분에 해당하는 인덱스 $I_{n+1} I_{n+2} \dots I_{N-1} I_1 I_2 \dots I_{n-1}$ 는 n 다음의 cyclic order이다. 여기서 맨 앞에 있는 인덱스 i_{n+1} 이 가장 빠르게 변하고 맨 뒤에 있는 i_{n-1} 이 가장 천천히 변한다. N-way tensor $\underline{\mathbf{X}}, \underline{\mathbf{Y}}$ 간의 내적 $\langle \underline{\mathbf{X}}, \underline{\mathbf{Y}} \rangle$ 은 $\sum_{i_1, i_2, \dots, i_N} x_{i_1, i_2, \dots, i_N} y_{i_1, i_2, \dots, i_N}$ 로 정의된다. 텐서 $\underline{\mathbf{X}}$ 의 Frobenious norm $\|\underline{\mathbf{X}}\|$ 은 $\sqrt{\langle \underline{\mathbf{X}}, \underline{\mathbf{X}} \rangle}$ 로 정의된다.

2.1 PARAFAC 모델

N-way 텐서 $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ 가 다음과 같이 N개 벡터들의 outer product로 표현되는 경우, rank 1이라 한다.

$$\underline{\mathbf{X}} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \dots \circ \mathbf{a}^{(N)}, \quad (1)$$

where $\mathbf{a}^{(n)} \in \mathbb{R}^{I_n}$ for $n = 1, \dots, N$.

이 때, 텐서의 원소는 $x_{i_1, i_2, \dots, i_N} = a_{i_1}^{(1)} a_{i_2}^{(2)} \dots a_{i_N}^{(N)}$ 과 같이 계산된다. 텐서의 rank는 텐서를 (2)와 같이 rank 1 텐서들의 선형 합으로 표현하는데 필요한 최소한의 rank 1 텐서들의 개수이다. $R = \text{rank}(\underline{\mathbf{X}})$ 는 텐서 $\underline{\mathbf{X}}$ 의 rank가 R임을 뜻한다.

$$\underline{\mathbf{X}} = \sum_{r=1}^R \mathbf{A}_{:,r}^{(1)} \circ \mathbf{A}_{:,r}^{(2)} \circ \dots \circ \mathbf{A}_{:,r}^{(N)}. \quad (2)$$

여기서 $\mathbf{A}_{:,r}^{(n)}$ 은 mode-n 행렬 $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ 의 r 번째 열벡터를 나타낸다.

PARAFAC 모델은 텐서 $\underline{\mathbf{X}}$ 를 (3)과 같이 rank R로 근사한다.

$$\underline{\mathbf{X}} \approx \sum_{r=1}^R \mathbf{A}_{:,r}^{(1)} \circ \mathbf{A}_{:,r}^{(2)} \circ \dots \circ \mathbf{A}_{:,r}^{(N)}. \quad (3)$$

위의 식을 텐서와 mode 행렬들의 원소들만을 이용해 표기하면 다음과 같다.

$$x_{i_1, i_2, \dots, i_N} = \sum_{r=1}^R a_{i_1, r}^{(1)} a_{i_2, r}^{(2)} \dots a_{i_N, r}^{(N)}. \quad (4)$$

PARAFAC 모델은 mode-n 행렬화와 Khatri-Rao product를 이용해 (5) 같이 표현할 수도 있다.

$$\mathbf{X}_{(n)} \approx \mathbf{A}^{(n)} [\mathbf{A}^{(n-1)} \odot \dots \odot \mathbf{A}^{(2)} \odot \mathbf{A}^{(1)} \odot \mathbf{A}^{(N)} \odot \dots \odot \mathbf{A}^{(n+2)} \odot \mathbf{A}^{(n+1)}]^\top. \quad (5)$$

2.2 Tucker 모델

텐서 $\underline{\mathbf{S}} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_n \times \dots \times J_N}$ 와 행렬 $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ 사이의 mode-n product $\underline{\mathbf{S}} \times_n \mathbf{A}^{(n)} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times I_n \times \dots \times J_N}$ 는 다음과 같이 정의된다.

$$[\underline{\mathbf{S}} \times_n \mathbf{A}^{(n)}]_{j_1 \dots j_{n-1} i_n j_{n+1} \dots j_N} = \sum_{j_n=1}^{J_n} s_{j_1 \dots j_{n-1} j_n j_{n+1} \dots j_N} a_{j_n, i_n}^{(n)} \quad (6)$$

행렬을 분해할 때 자주 등장하는 $\mathbf{X} = \mathbf{U} \mathbf{S} \mathbf{V}^\top$ 는 mode-n product를 이용해 $\mathbf{X} = \mathbf{S} \times_1 \mathbf{U} \times_2 \mathbf{V}$ 와 같이 표기된다. 그밖에도 mode-n product는 다음과 같은 성질을 가지고 있다.

$$(\mathbf{S} \times_n \mathbf{U}) \times_m \mathbf{V} = (\mathbf{S} \times_m \mathbf{V}) \times_n \mathbf{U} \quad (7)$$

$$(\mathbf{S} \times_n \mathbf{U}) \times_n \mathbf{V} = \mathbf{S} \times_n (\mathbf{V} \mathbf{U}). \quad (8)$$

Tucker 모델은 텐서 $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ 를 (9)와 같이 core 텐서 $\underline{\mathbf{S}} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_N}$ 와 N개의 mode 행렬 $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ 들 사이의 mode-n product로 근사한다.

$$\underline{\mathbf{X}} \approx \underline{\mathbf{S}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \dots \times_N \mathbf{A}^{(N)}. \quad (9)$$

위의 식을 텐서와 mode 행렬들의 원소들만을 이용해 표기하면 다음과 같다.

$$x_{i_1, i_2, \dots, i_N} = \sum_{j_1, j_2, \dots, j_N} s_{j_1, j_2, \dots, j_N} a_{i_1, j_1}^{(1)} a_{i_2, j_2}^{(2)} \dots a_{i_N, j_N}^{(N)} \quad (10)$$

Tucker 모델은 mode-n 행렬화와 Kronecker product를 이용해 (11) 같이 표현할 수도 있다.

$$\mathbf{X}_{(n)} \approx \mathbf{A}^{(n)} \mathbf{S}_{(n)} [\mathbf{A}^{(n-1)} \otimes \dots \otimes \mathbf{A}^{(2)} \otimes \mathbf{A}^{(1)} \otimes \mathbf{A}^{(N)} \otimes \dots \otimes \mathbf{A}^{(n+2)} \otimes \mathbf{A}^{(n+1)}]^\top. \quad (11)$$

3. Nonnegative Tucker decomposition

3.1 Problem formulation

Nonnegative Tucker decomposition(NTD)은 주어진 비음수 N-way 텐서 $\underline{\mathbf{X}} \in \mathbb{R}_+^{I_1 \times I_2 \times \dots \times I_N}$ 를 (9)와 같은 형태로 분해하는데 있어서, core tensor $\underline{\mathbf{S}} \in \mathbb{R}_+^{J_1 \times J_2 \times \dots \times J_N}$ 와 모든 mode 행렬 $\mathbf{A}^{(n)} \in \mathbb{R}_+^{I_n \times J_n}$ 에 비음수 제약을 둔다. 데이터 $\underline{\mathbf{X}}$ 와 모델 $\hat{\underline{\mathbf{X}}}$ 사이의 차이를 나타내는 목적 함수로는 다음과 같이 LS-error function, I-divergence, α -divergence가 사용되었다.

$$\mathcal{J}_{LS} = \|\underline{\mathbf{X}} - \hat{\underline{\mathbf{X}}}\|^2, \quad (12)$$

$$\mathcal{J}_I = \sum_{i_1, i_2, \dots, i_N} x_{i_1, i_2, \dots, i_N} \log \frac{x_{i_1, i_2, \dots, i_N}}{\hat{x}_{i_1, i_2, \dots, i_N}} - x_{i_1, i_2, \dots, i_N} + \hat{x}_{i_1, i_2, \dots, i_N}, \quad (13)$$

$$\mathcal{J}_\alpha = \frac{\sum_{i_1, i_2, \dots, i_N} \alpha x_{i_1 i_2 \dots i_N} + (1-\alpha) \hat{x}_{i_1 i_2 \dots i_N} - x_{i_1 i_2 \dots i_N}^\alpha + \hat{x}_{i_1 i_2 \dots i_N}^{1-\alpha}}{\alpha(1-\alpha)} \quad (14)$$

3.2 Multiplicative updating rule

NTD의 core 텐서와 mode 행렬들을 위한 multiplicative updating rule은 NMF의 그것과 비슷하다. Tucker model을 mode-n 행렬화와 Kronecker product를 이용해 표현한 (11)과 다음의 Kronecker product와 $\text{vec}(\cdot)$ 연산의 성질은 updating rule을 유도하는데 있어서 매우 중요한 역할을 한다.

$$\text{vec}(\mathbf{UBV}^T) = (\mathbf{U} \otimes \mathbf{V}) \text{vec}(\mathbf{B}), \quad (15)$$

$$(\mathbf{U} \otimes \mathbf{V})^T = \mathbf{U}^T \otimes \mathbf{V}^T, \quad (16)$$

$$(\mathbf{U} \otimes \mathbf{V})(\mathbf{B} \otimes \mathbf{C}) = \mathbf{UB} \otimes \mathbf{VC}. \quad (17)$$

또한 표현의 간결성을 위해 다음과 같은 표기를 새로 정의한다.

$$\mathbf{A}^{(\wedge n)} = \mathbf{A}^{(n-1)} \otimes \dots \otimes \mathbf{A}^{(2)} \otimes \mathbf{A}^{(1)} \otimes \mathbf{A}^{(M)} \otimes \dots \otimes \mathbf{A}^{(n+2)} \otimes \mathbf{A}^{(n+1)}. \quad (18)$$

3.2.1 Mode 행렬의 multiplicative updating rule

(11)로부터 다음과 같이 NMF 모델과 똑같은 형태를 얻을 수 있다.

$$\mathbf{X}_{(n)} = \mathbf{A}^{(n)} \mathbf{S}_{(n)} \mathbf{A}^{(\wedge n)T} = \mathbf{A}^{(n)} \mathbf{S}_A^{(n)}. \quad (19)$$

따라서 $\mathbf{A}^{(n)}$ 의 updating rule은 NMF에서의 기저 행렬 \mathbf{A} 의 updating rule과 동일하다.

• LS:

$$\mathbf{A}^{(n)} \leftarrow \mathbf{A}^{(n)} \star \frac{\mathbf{X}_{(n)} \mathbf{S}_A^{(n)T}}{\mathbf{A}^{(n)} \mathbf{S}_A^{(n)} \mathbf{S}_A^{(n)T}}, \quad (20)$$

• I-divergence:

$$\mathbf{A}^{(n)} \leftarrow \mathbf{A}^{(n)} \star \frac{[\mathbf{X}_{(n)} / (\mathbf{A}^{(n)} \mathbf{S}_A^{(n)})] \mathbf{S}_A^{(n)T}}{\mathbf{11}^T \mathbf{S}_A^{(n)T}}, \quad (21)$$

• a-divergence:

$$\mathbf{A}^{(n)} \leftarrow \mathbf{A}^{(n)} \star \left\{ \frac{[\mathbf{X}_{(n)} / (\mathbf{A}^{(n)} \mathbf{S}_A^{(n)})]^\alpha \mathbf{S}_A^{(n)T}}{\mathbf{11}^T \mathbf{S}_A^{(n)T}} \right\}^{\frac{1}{\alpha}}, \quad (22)$$

위의 식에서 \star 는 Hadamard product, $/$ 는 element-wise division이다.

행렬 $\mathbf{S}_A^{(n)}$ 를 구할 때 계산량을 줄이기 위해서는 Kronecker product의 정의에 기반 한 방법이 아니라 mode-n product를 이용하여 다음과 같이 계산되어야 한다.

$$\begin{aligned} \mathbf{S}_A^{(n)} &= \mathbf{I} \mathbf{S}_{(n)} \mathbf{A}^{(\wedge n)T} \\ &= [\mathbf{S} \times_1 \mathbf{A}^{(1)} \dots \times_{n-1} \mathbf{A}^{(n-1)} \times_n \mathbf{I} \times_{n+1} \mathbf{A}^{(n+1)} \dots \times_N \mathbf{A}^{(N)}]_{(n)} \\ &= [\mathbf{S} \times_1 \mathbf{A}^{(1)} \dots \times_{n-1} \mathbf{A}^{(n-1)} \times_{n+1} \mathbf{A}^{(n+1)} \dots \times_N \mathbf{A}^{(N)}]_{(n)} \\ &= [\mathbf{S} \times_{m \neq n} \mathbf{A}^{(m)}]_{(n)}. \end{aligned} \quad (23)$$

LS-error function을 사용한 경우에는 다음과 같은 방법을 통해 계산량을 더욱 줄일 수 있다.

$$\mathbf{A}^{(n)} \mathbf{S}_{(n)} \mathbf{A}^{(\wedge n)T} = [\mathbf{S} \times_{m \neq n} \mathbf{A}^{(m)T} \mathbf{A}^{(m)}]_{(n)} \mathbf{S}_{(n)}^T \quad (24)$$

$$\mathbf{X}_{(n)} \mathbf{S}_A^{(n)T} = [\mathbf{X} \times_{m \neq n} \mathbf{A}^{(m)T}]_{(n)} \mathbf{S}_{(n)}^T. \quad (25)$$

3.2.1 Core 텐서의 multiplicative updating rule

Core 텐서의 경우 (15)~(19)를 이용하여 다음과 NMF와 똑같은 형태를 얻을 수 있다.

$$\text{vec}(\mathbf{X}_{(n)}) = \text{vec}(\mathbf{A}^{(n)} \mathbf{S}_{(n)} \mathbf{A}^{(\wedge n)T}) = (\mathbf{A}^{(\wedge n)} \otimes \mathbf{A}^{(n)}) \text{vec}(\mathbf{S}_{(n)}) \quad (26)$$

따라서 core 텐서 \mathbf{S} 의 update rule을 NMF의 계수 행렬 \mathbf{S} 에 대한 update rule을 이용해 구할 수 있다.

LS: NMF의 updating rule에 따라

$$\text{vec}(\mathbf{S}_{(n)}) \leftarrow \text{vec}(\mathbf{S}_{(n)}) \star \mathbf{K}_n, \quad (27)$$

where

$$\mathbf{K}_n = \frac{[\mathbf{A}^{(\wedge n)} \otimes \mathbf{A}^{(n)}]^T \text{vec}(\mathbf{X}_{(n)})}{[\mathbf{A}^{(\wedge n)} \otimes \mathbf{A}^{(n)}]^T [\mathbf{A}^{(\wedge n)} \otimes \mathbf{A}^{(n)}] \text{vec}(\mathbf{S}_{(n)})}. \quad (28)$$

(28)의 분자에 (15)~(17)을 적용하면

$$\begin{aligned} [\mathbf{A}^{(\wedge n)} \otimes \mathbf{A}^{(n)}]^T \text{vec}(\mathbf{X}_{(n)}) &= \text{vec}(\mathbf{A}^{(n)T} \mathbf{X}_{(n)} \mathbf{A}^{(\wedge n)}) \\ &= \text{vec}([\mathbf{X} \times_1 \mathbf{A}^{(1)T} \dots \times_N \mathbf{A}^{(N)T}]_{(n)}). \end{aligned} \quad (29)$$

같은 방식으로 (28) 분모는

$$\begin{aligned} &[\mathbf{A}^{(\wedge n)} \otimes \mathbf{A}^{(n)}]^T [\mathbf{A}^{(\wedge n)} \otimes \mathbf{A}^{(n)}] \text{vec}(\mathbf{S}_{(n)}) \\ &= [\mathbf{A}^{(\wedge n)T} \mathbf{A}^{(\wedge n)} \otimes \mathbf{A}^{(n)T} \mathbf{A}^{(n)}] \text{vec}(\mathbf{S}_{(n)}) \\ &= \text{vec}(\mathbf{A}^{(n)T} \mathbf{A}^{(n)} \mathbf{S}_{(n)} \mathbf{A}^{(\wedge n)T} \mathbf{A}^{(\wedge n)}) \\ &= \text{vec}([\mathbf{S} \times_1 \mathbf{A}^{(1)T} \mathbf{A}^{(1)} \dots \times_N \mathbf{A}^{(N)T} \mathbf{A}^{(N)}]_{(n)}). \end{aligned} \quad (30)$$

(29), (30)을 통해 (27)을 다시 정리하면

$$\mathbf{S} \leftarrow \mathbf{S} \star \frac{\mathbf{X} \times_1 \mathbf{A}^{(1)T} \dots \times_N \mathbf{A}^{(N)T}}{\mathbf{S} \times_1 \mathbf{A}^{(1)T} \mathbf{A}^{(1)} \dots \times_N \mathbf{A}^{(N)T} \mathbf{A}^{(N)}}. \quad (31)$$

위와 같은 방식으로 I-divergence, a-divergence에 대해서도 updating rule을 유도할 수 있다.

I-divergence:

$$\mathbf{S} \leftarrow \mathbf{S} \star \frac{(\mathbf{X}/\hat{\mathbf{X}}) \times_1 \mathbf{A}^{(1)T} \dots \times_N \mathbf{A}^{(N)T}}{\mathbf{E} \times_1 \mathbf{A}^{(1)T} \dots \times_N \mathbf{A}^{(N)T}}. \quad (32)$$

a-divergence:

$$\mathbf{S} \leftarrow \mathbf{S} \star \left[\frac{(\mathbf{X}/\hat{\mathbf{X}})^\alpha \times_1 \mathbf{A}^{(1)T} \dots \times_N \mathbf{A}^{(N)T}}{\mathbf{E} \times_1 \mathbf{A}^{(1)T} \dots \times_N \mathbf{A}^{(N)T}} \right]^{\frac{1}{\alpha}}. \quad (33)$$

(32), (33)에서 \mathbf{E} 는 모든 원소가 1인 텐서이다.

NTD의 core 텐서를 학습하지 않고, 미리 특정한 구조를 갖는 텐서로 고정시켜 놓은 채, mode 행렬들만 학습하게 될 경우에는 NMF, nsNMF, NTF가 NTD의 특별한 경우로 표현이 된다. 표 1을 NMF, nsNMF, NTF와 NTD와의 관계를 정리해 놓은 것이다. 또한 NTD의

표 1 Core 텐서를 각각 identity 행렬 I , smoothing 행렬 M , unit super diagonal tensor ($\delta_{i_1 i_2 \dots i_N}$)로 설정하여 NMF, nsNMF, NTF를 NTD의 특별한 경우로 설명할 수 있다.

모델	Original representation	NTD representation
NMF	$X \approx AS$	$X = I \times_1 A \times_2 S^T$
nsNMF	$X \approx AMS$	$X = M \times_1 A \times_2 S^T$
NTF	$X \approx \sum_{r=1}^R A_{:,r}^{(1)} \circ A_{:,r}^{(2)} \circ \dots \circ A_{:,r}^{(N)}$	$X \approx \underline{A} \times_1 A^{(1)} \times_2 A^{(2)} \dots \times_N A^{(N)}$

표 2 최소자승법을 사용했을 때의 NMF, nsNMF, NTF, NTD의 multiplicative updating rule. nsNMF에서 $S_M = MS$, $A_M = AM$; NTF에서 $S_A^{(n)} = [A^{(n-1)} \odot \dots \odot A^{(2)} \odot A^{(1)} \odot A^{(N)} \odot \dots \odot A^{(n+2)} \odot A^{(n+1)}]^T$

모델	Multiplicative updating rule
NMF	$A \leftarrow A \star \frac{XS^T}{ASS^T}, S \leftarrow S \star \frac{A^T X}{A^T A S}$
nsNMF	$A \leftarrow A \star \frac{XS_M^T}{AS_M S_M^T}, S \leftarrow S \star \frac{A_M^T X A_M^T}{A_M^T A_M S}$
NTF	$A^{(n)} \leftarrow A^{(n)} \star \frac{X^{(n)} S_A^{(n)T}}{A^{(n)} S_A^{(n)} S_A^{(n)T}}$
NTD	$A^{(n)} \leftarrow A^{(n)} \star \frac{[X \times_{m \neq n} A^{(m)}]_{(n)} S_{(n)}^T}{[S \times_{m \neq n} A^{(m)T} A^{(m)}]_{(n)} S_{(n)}^T}$ $S \leftarrow S \star \frac{X \times_1 A^{(1)T} \dots \times_N A^{(N)T}}{S \times_1 A^{(1)T} A^{(1)} \dots \times_N A^{(N)T} A^{(N)}}$

mode 행렬들의 updating rule에 표 1의 특정 형태의 core 텐서를 삽입하게 되면 NMF, nsNMF, NTF의 본래 updating rule이 다시 유도 된다. 표 2는 LS error function을 사용할 경우에 각각의 알고리즘들의 multiplicative updating rule을 정리해 놓은 것이다.

4. 실험 결과 및 분석

본 실험에서는 NTD와 NMF가 ‘pepper & salt’ 노이즈에 어떻게 반응하는지를 비교하였다. 실험에는 AT&T Cambridge University, ORL dataset이 사용되었다. 총 400장의 이미지를 48×48 pixel로 정규화하였다. NTD는 48×48×400인 텐서를 다루고 NMF는 48·48×400인 행렬을 다룬다. NTD의 경우 core 텐서의 사이즈는 24×24×30, NMF의 경우 축소된 차원은 30이었다.

그림 1은 NMF와 NTD의 기저(basis) 영상이다. NTD의 기저 영상이 NMF의 그것보다 상대적으로 덜 훼손되었다. 그림 2는 NMF와 NTF의 복구된 영상이다. 복구된 영상 역시 NTD가 NMF보다 노이즈의 효과가 적게 나타난다. Average peak signal to noise ratio (PSNR) 역시 NMF는 22.16dB, NTD는 24.12db로 NTD가 좋게 나온다. 이미지를 구성하는 대부분의 픽셀들은 인접 픽셀들과 비슷한 값을 가지고 있다. ‘pepper

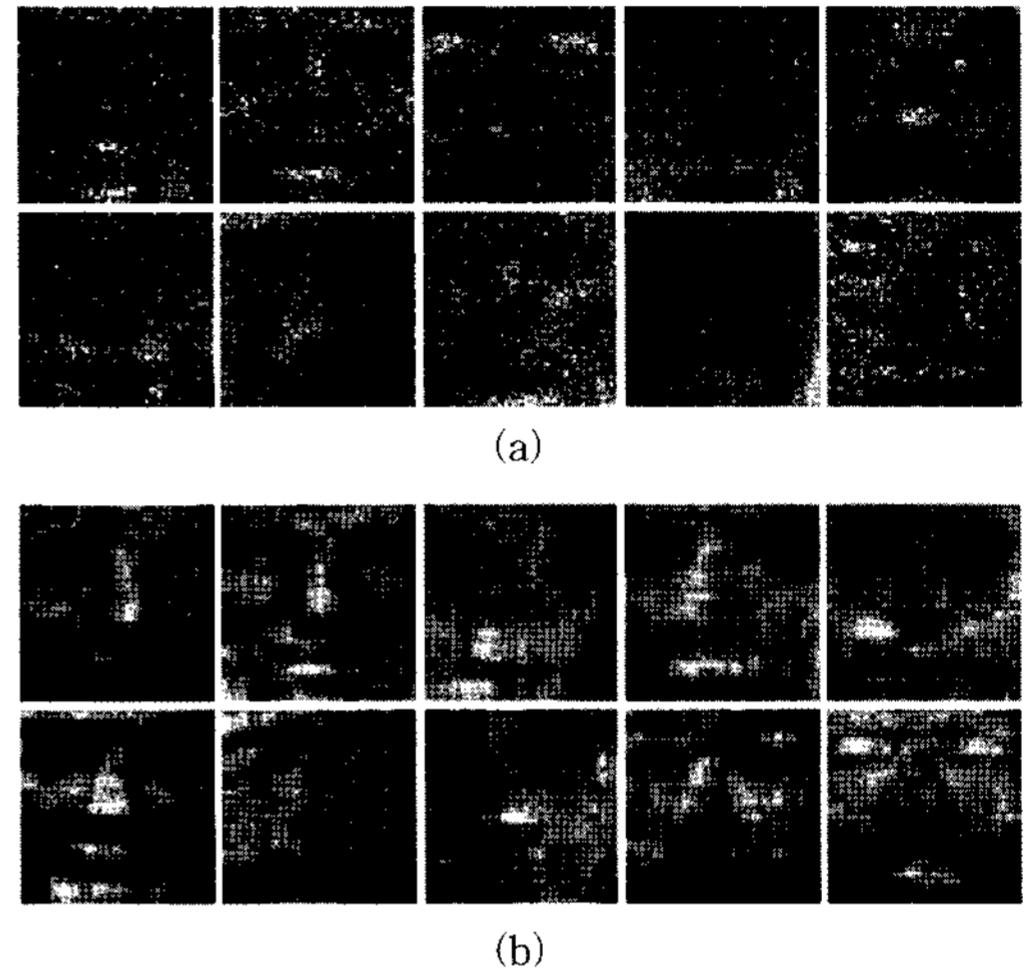


그림 1 ‘pepper & salt’ 노이즈로 훼손된 영상에서 얻은 기저 영상. (a) NMF; (b) NTD

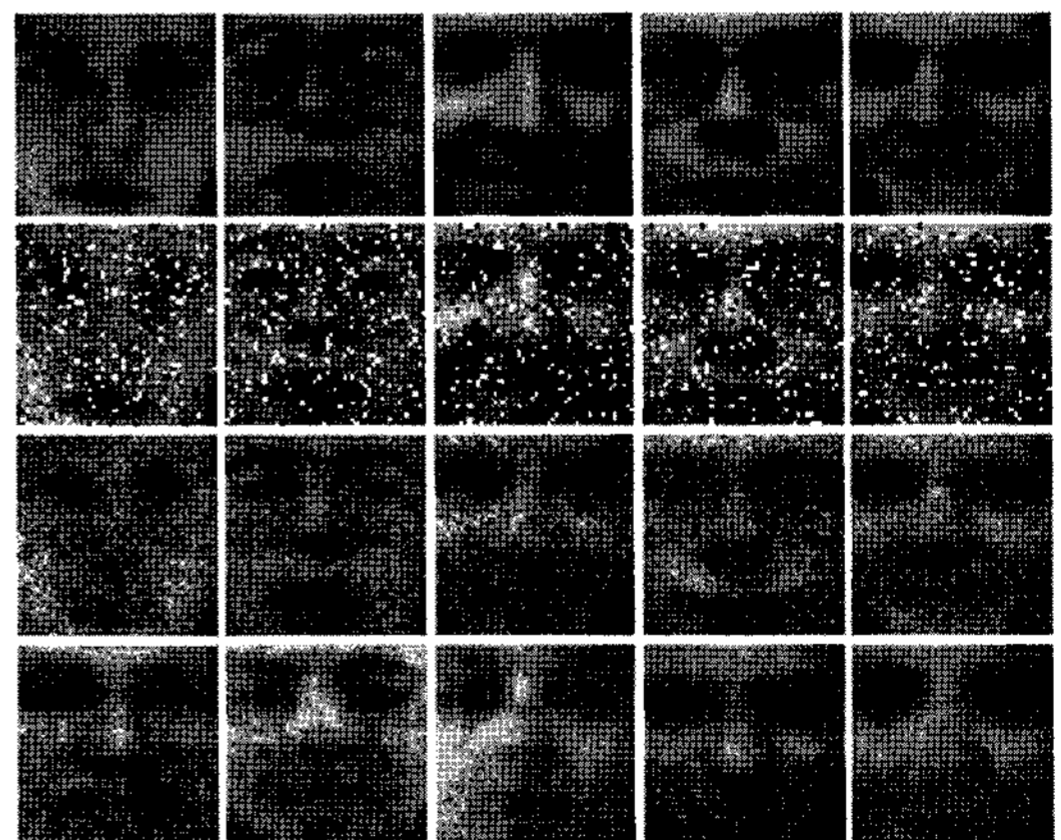


그림 2 위에서부터 차례대로 (1) 원본 영상; (2) ‘pepper & salt’ 노이즈에 훼손된 영상; (3) NMF로 복구된 영상 ; (4) NTD로 복구된 영상

& salt’ 노이즈는 이런 성질을 벗어나는 일종의 극단값 (outlier)로 볼 수 있다. NTD는 이미지의 2D 구조를 유지하여 NMF 보다 훨씬 적은 파라미터를 이용해 데이터를 모델링한다. 그 결과 학습 데이터의 양이 적고 극

단 값이 존재하더라도 보다 안정적인 성능을 보인다.

5. 결론 및 향후 연구

본 논문에서는 Tucker 모델에 비음수 제약을 가한 NTD를 제안하고 여러 목적함수에 대한 multiplicative updating rule을 유도하였다. NTD는 NMF, nsNMF, NTF를 포함하는 일반적인 모델이기 때문에 multiway 데이터를 다루는데 매우 유용한 방법이 될 것이다.

Multiplicative updating rule은 구현이 간편한 장점이 있지만 수렴속도가 느리다는 단점을 지니고 있다. 이러한 문제를 해결하기 위해 좋은 시작점을 찾는 방법, 최적화에 기반한 보다 빠른 updating rule이 연구되어야 한다.

참고 문헌

- [1] A. Hyvärinen, J. Karhunen, and E. Oja, "Independent Component Analysis," John Wiley & Sons, Inc., 2001.
- [2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, 401:788.791, 1999.
- [3] J. Yang, D. Zhang, A. F. Frangi, and J. Y. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(1):131.137, 2004.
- [4] D. Zhang, S. Chen, and Z. H. Zhou, "Two-dimensional non-negative matrix factorization for face representation and recognition," In *ICCV-2005 Workshop on Analysis and Modeling of Faces and Gestures*, 2005.
- [5] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, 31:279.311, 1966.
- [6] L. de Lathauwer, B. de Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, 21(4):1253.1278, 2000.
- [7] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an Exploratory multi-modal factor analysis," *UCLA Working Papers in Phonetics*, 16:1.84, 1970.
- [8] A. Shashua and T. Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," In *Proceedings of International Conference on Machine Learning*, Bonn, Germany, 2005.
- [9] A. Cichocki, R. Zdunek, S. Choi, R. J. Plemmons, and S. Amari, "Non-negative tensor factorization using alpha and beta divergences," In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Honolulu, Hawaii, 2007.
- [10] A. Pascual-Montano, J. M. Carazo, K. K. D. Lehmann, and R. D. Pascual-Margui, "Nonsmooth nonnegative matrix factorization (nsNMF)," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(3):403.415, 2006.