

# NMF와 EMD를 이용한 영문자 활자체 폰트분류

이창우\*, 강현\*, 정기철\*\*, 김학준\*  
 \*경북대학교 컴퓨터공학과 인공지능연구실  
 {cwlee, hkang, khkim}@ailab.knu.ac.kr  
 \*\*승실대학교 정보과학대학 미디어 학부  
 kcjung@ssu.ac.kr

## Font Classification using NMF and EMD

Chang Woo Lee\*, Hyun Kang\*, Keechul Jung\*\*, and Hang Joon Kim\*

\*Dept. of Computer Engineering, Kyungpook National Univ., Daegu, South Korea

\*\*School of Media, College of Information Science, Soongsil Univ., Seoul, South Korea

### 요 약

최근 전자화된 문서 영상을 효율적으로 관리하고 검색하기 위한 문서구조분석 방법과 문서의 자동 분류에 관한 많은 연구가 발표되고 있다. 본 논문에서는 NMF(non-negative matrix factorization) 알고리즘을 사용하여 폰트를 자동으로 분류하는 방법을 제안한다. 제안된 방법은 폰트의 구분 특징들이 공간적으로 국부성을 가지는 부분으로 표현될 수 있다는 가정을 바탕으로, 전체의 폰트 이미지들로부터 각 폰트들의 구분 특징인 부분을 학습하고, 학습된 부분들을 특징으로 사용하여 폰트를 분류하는 방법이다. 학습된 폰트의 특징들은 계층적 군집화 알고리즘을 이용하여 템플릿을 생성하고, 테스트 패턴을 분류하기 위하여 템플릿 패턴과의 EMD(earth mover's distance)를 사용한다. 실험결과에서 폰트 이미지들의 공간적으로 국부적인 특징들이 조사되고, 그 특징들의 폰트 식별을 위한 적절성을 보였다. 제안된 방법이 기존의 문자인식, 문서 검색 시스템들의 전처리기로 사용되면, 그 시스템들의 성능을 향상시킬 것으로 기대된다.

### 1. 서론

기존 OCR 시스템에 폰트 식별기(font recognizer)가 전처리기로 사용될 경우, 그 성능이나, 정확도 및 처리 시간 면에서 상당한 개선을 기대할 수 있다. 또한, 대부분의 문자 인식 시스템에서 폰트 식별은 다양한 글자들의 변형을 처리해야 하는 부담을 줄여주는 장점이 있다[1].

폰트 식별이 문서 이미지 처리나 인식 시스템에서 필수 불가결한 요소임에도 불구하고, 아직 많은 시스템들이 이를 채택하지 않고 있으며, 최근에는 폰트 식별에 대해 많은 국내외 연구가 보고 되고 있다[1-5]. 이전 폰트 분류 방법들은 문서 이미지 전체적인 특징(global property)[2, 3, 5]과 국부적인 특징(local property)를 사용하는 방법[4, 6]으로 구분할 수 있고, 글자 단위의 분류 방법[6]과 단어나 문장의 길이 단위의 분류[2], 블록단위의 분류 방법[5]들로 크게 구분할 수 있다.

본 논문에서는 NMF를 사용한 폰트 식별 방법을 제안한다. 제안한 방법은 이미지 공간에서 각각의 폰트들이 차지하고 있는 공간적 지역성(spatial locality)이 다르다는 것을 폰트를 분류하는 특징으로 간주한 방법으로써, 객체의 부분기반 표현들을 학습할 수 있는 NMF 알고리즘을 사용하여 폰트 이미지들 부분별로 분리하여 학습한다. 제안된 폰트 분류 방법은 그림 1에서 보듯이 폰트 이미지들의 집합으로부터 각 폰트를 구별하기 위한 지역적 특징들을 자동적으로 추출해 주는 NMF 알고리즘을 사용하여 특징을 추출한다. 추출된 특징들은 각 폰트를 대표하는 템플릿을 형성하기 위해 사용되고, 테스트 패턴의 분류를 위해 EMD를 사용한다.

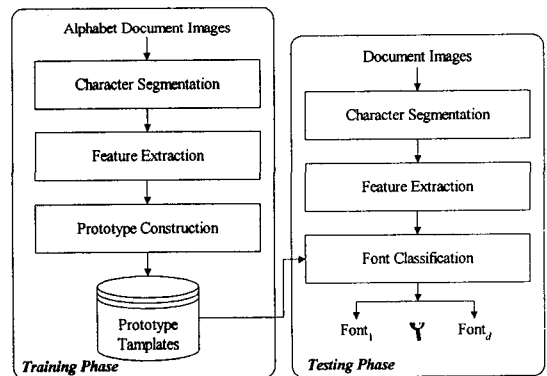


그림 1. 제안된 폰트분류 방법의 흐름도.

### 2. NMF 알고리즘

최근 Lee 와 Seung 에 의해 발표된 NMF 알고리즘은 주어진 행렬을 베이스스 행렬과 엔코딩 행렬로 분해하는 방법이다 [7]. 이 알고리즘이 주성분분석(Principal Component Analysis, PCA) 방법이나 벡터양자화(Vector Quantization, VQ) 방법과 같은 행렬분해 알고리즘들과 가장 큰 차이는 분해된 각 행렬의 요소들이 모두 0 이상의 값들이라는 것이다. NMF 알고리즘은 수식(1)에서처럼  $n \times m$ 의 행렬  $V$ 를  $n$  차원의 비음수 벡터  $r$ 개로 구성된 베이스스( $W$ )와  $r$  차원의 비음수 벡터  $m$ 개로 구성된 엔코딩( $H$ )으로 분해한다. 여기서 랭크

(rank)  $r$  은 일반적으로  $(n+m)r < nm$  의 범위내의 임의의 수로 정한다. NMF 알고리즘은  $W$  와  $H$  행렬의 랜덤 초기화로부터 시작하여, 두 행렬의 곱이 원 행렬에 수렴될 때까지 반복적으로 갱신(update)한다. 알고리즘의 수렴 정도를 측도하기 위해 수식(2)와 같은 평가 함수(objective function)가 필요하고, 이 평가 함수는  $H$  와  $W$  행렬의 곱이 원 행렬  $V$  를 생성하기 위한 유사도(likelihood)와 연관된다.

$$V_{i\mu} \approx (WH)_{i\mu} = \sum_{\alpha=1}^r W_{i\alpha} H_{\alpha\mu} \quad \text{식(1)}$$

$$F = \sum_{i=1}^n \sum_{\mu=1}^m V_{i\mu} \log(WH)_{i\mu} - (WH)_{i\mu} \quad \text{식(2)}$$

NMF 알고리즘은 반복적 알고리즘으로 곱셈만으로 이루어진 갱신 규칙(update rule)을 가지고 있다. 특히 이 알고리즘은 gradient-descent 알고리즘의 변형으로 간주될 수 있다 [8]. NMF 알고리즘은 수식(3)에서와 같이 곱셈 갱신 규칙에 따라  $W$  와  $H$  를 반복적으로 갱신한다.

$$H_{\alpha\mu} \leftarrow H_{\alpha\mu} \frac{(W^T V)_{\alpha\mu}}{(W^T W H)_{\alpha\mu}}, \quad W_{i\alpha} \leftarrow W_{i\alpha} \frac{(V H^T)_{i\alpha}}{(W H H^T)_{i\alpha}}, \quad W_{i\alpha} \leftarrow W_{i\alpha} \frac{W_{i\alpha}}{\sum_j W_{j\alpha}} \quad \text{식(3)}$$

### 3. 제안된 폰트 분류 방법

제안된 방법은 NMF 알고리즘에 의해 생성된 훈련 샘플들의 엔코딩을 각 폰트를 대표하는 특징으로 간주한 방법이다. 각 폰트를 대표하는 템플릿 엔코딩을 생성하기 위해 훈련단계의 엔코딩들을 군집화하여 각 폰트의 대표 특징으로 설정한다. 군집화 알고리즘은 계층적 군집화 방법을 사용하고, 이때 사용된 거리 척도는  $d_{\min}(D_i, D_j) = EMD(m_i, m_j)$  을 사용한다. 이때,  $D_i, D_j$  는 병합의 대상이 되는 클러스터들이고,  $m_i$  와  $m_j$  는 각 클러스터의 중심값이다.

EMD 는 잘 알려진 *Transportation problem* 의 해결책에 기반한 거리 척도로서, 아래와 같이 *linear programming problem* [9-11]으로 공식화할 수 있다. 두 패턴은 다음과 같은 시그니처(signature)로 표현한다:  $p = \{(p_1, w_{p_1}), \dots, (p_m, w_{p_m})\}$  와  $q = \{(q_1, w_{q_1}), \dots, (q_m, w_{q_m})\}$ . 본 논문에서  $p$  와  $q$  는 각각 클러스터의 중심값과 테스트 이미지의 엔코딩 값이다. 그리고,  $w_{p_i}, w_{q_j}$  는 각 포인트들의 가중치 값들이다. 수식 (4)와 같이 EMD 는 전체 비용을 최소화 하는  $f_{ij}$  의 집합을 찾는 것이다.  $f_{ij}$  는  $q_j$  와 매치되는 일의 량( $p_i$ )를 의미하기도 한다.

$$EMD(p, q) = \min \frac{\sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij}}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}} \quad \text{식(4)}$$

여기서,

$$f_{ij} \geq 0, \quad p_i \geq 0, \quad q_j \geq 0, \quad 1 \leq i \leq m, 1 \leq j \leq n \quad \text{식(5)}$$

$$\sum_{j=1}^n f_{ij} \leq w_{p_i}, \quad 1 \leq j \leq n \quad \text{식(6)}$$

$$\sum_{i=1}^m f_{ij} \leq w_{q_j}, \quad 1 \leq i \leq m \quad \text{식(7)}$$

$$\sum_{i=1}^m \sum_{j=1}^n f_{ij} = \min(\sum_{i=1}^m w_{p_i}, \sum_{j=1}^n w_{q_j}), \quad 1 \leq i \leq m, 1 \leq j \leq n \quad \text{식(8)}$$

$$\sum_{i=1}^m w_{p_i} \leq \sum_{j=1}^n w_{q_j} \quad \text{식(9)}$$

비용  $d_{ij}$  는  $p_i$  와  $q_j$  사이의 *ground distance* 이고, 본 논문에서는 이를 두 베이스스 벡터들 사이의 *tanimoto distance* 로 정의한다. 식(5)에서 식(9)까지는 전체 비용을 최소화 하는  $f_{ij}$  를 찾기 위한 제약 조건들이다.

NMF 알고리즘에서 베이스스의 최소성의 결과로써, 한 이미지의 엔코딩 또한 투영된 공간에서 최소하게 분포된다. 결과적으로, 폰트 이미지들의 지역적 부분들을 학습하는 NMF 의 베이스스는 각 폰트의 특징들을 표현하고, 한 폰트 이미지가 베이스스(폰트이미지의 부분) 이미지의 일부분만을 사용하여 표현된다는 성질을 이용하여 폰트를 분류한다. [그림 1]에서 처럼, 제안된 폰트 분류 방법은 훈련 테스트 이미지들을 이용하여 템플릿 엔코딩을 만들고, 폰트의 분류를 위해 문서 이미지로부터 얻어진 테스트 이미지와 비교하는 방법으로 구성된다. 폰트의 분류는 아래의 수식(7)과 같이 테스트 이미지로부터 추출한 엔코딩과 템플릿 엔코딩들을 비교하여 가장 가까운 거리에 있는 템플릿의 폰트  $F_i$  로 분류한다.  $T_c$  와  $h$  는  $i$  번째 폰트 타입의  $c$  번째 템플릿 엔코딩과 테스트 이미지의 엔코딩을 의미한다.

$$F_i = \underset{c}{\operatorname{argmin}} EMD(T_c, h), \quad i = 1, \dots, d \quad c = 1, \dots, l_i \quad \text{식(10)}$$

### 4. 실험 결과

실험에 사용된 폰트는 빈번하게 사용되어지는 영어 폰트 6 개 (Courier New (CN), Comic Sans MS (CS), Impact (IM), MS Gothic (MG), Microsoft Sans Serif (MS), Times New Roman (TR))이다. 본 논문에서 분류는 6 개의 폰트×4 개의 폰트 스타일(굵게, 보통, 기울기, 굵게+기울기)×2 개의 형태(대문자, 소문자) 중에 하나로 분류하였다. 즉, 테스트 영상은 48 개의 글꼴 유형들 중 하나로 분류된다.

글자 크기를 14 픽셀로 노이즈가 없는 폰트 이미지들과 각각 100, 200 그리고, 300 dpi 의 해상도로 스캔된 이진 영상들로부터 획득한 4992 개의 문자 이미지들을 훈련 이미지로 채택하였다. 훈련에 사용된 이미지는 모두 28×28 크기의 이미지로 정규화하여 사용하였고, 모든 폰트 이미지를 열벡터로 하나의 전체 행렬을 구성하였다. 이렇게 구성된 전체 행렬을 NMF 알고리즘을 이용하여, 베이스스  $W$  와 엔코딩  $H$  로 분해하였다. 테스트 단계에서 엔코딩들은 각 폰트 유형을 표현하는 템플릿 엔코딩들로서 사용되었고, 테스트 데이터는 인쇄된 144 개의 문서 이미지를 스캔하여 사용하였다. 즉, 각각 100, 200, 300dpi 의 해상도를 가진 이진 영상으로써, 실험 대상이 된 모든 글꼴유형에 대해 워드프로세서를 이용하여 생성, 인쇄, 그리고 스캔하여 사용하였다.

실험에 사용된 폰트 영상들은 각각 Courier New 폰트 10368×3(3 종류의 해상도)개의 문자 이미지, Comic Sans 폰트 8544×3 개의 문자 이미지, Impact 폰트 11586×3 개의 문자 이미지, MS Gothic 폰트 10416×3 개의 문자 이미지, Microsoft Sans Serif 폰트 11464×3 개의 문자 이미지, 그리고 Times New Roman 폰트 11232×3 개의 문자 이미지로 구성된다. 테스트와 훈련단계에서 모든 문서들에 대해, 글자는 흰색으로 나머지부분은 검정색으로 표현하였다.

그림 2 는 랭크 수에 따라 실험에 사용된 베이스스 집합을 보여준다. 훈련 단계에서 사용된 샘플 이미지들을 이용한 행렬분해 시, 각 샘플들이 모두 분해된 엔코딩들과 일대일 상용관계를 가지므로, 이를 이용하여 템플릿 엔코딩을 형성한다. [그림 6]에서 보여지는 것처럼 베이스스 이미지의 수가 증가할수록 베이스스 영상들과 이미지 엔코딩들은 더욱 최소화되고, 반대로 베이스스 영상의 수가 감소할수록 하나의 베이스스 이미지가 한 폰트의 지역적인 특징을 포괄하여야 하기 때

문에 베이스스 이미지와 각 폰트 이미지에 상응하는 엔코딩들의 최소성이 감소함을 볼 수 있다. 이처럼 랭크의 수에 따라 지역적인 특징을 학습한 베이스스의 모양이 달라지며, 본 논문에서는 부분적으로 랭크의 수에 따른 폰트 분류율에 대하여 조사하였다.

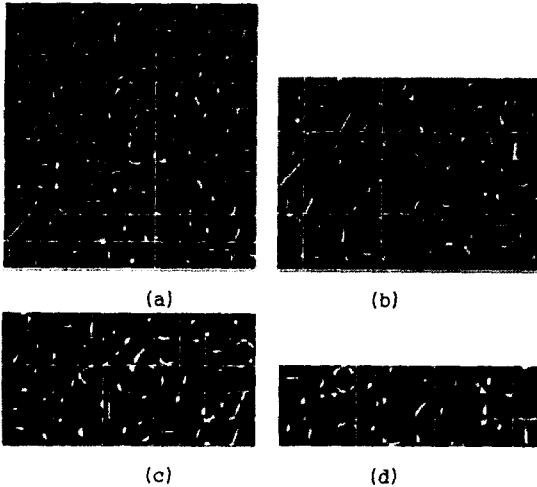


그림 2. 랭크 수에 따른 각 폰트의 국부적인 특징들 (a) 100 랭크 (b) 70 랭크 (c) 50 랭크 (d) 30 랭크.

아래 표 1 은 표 1 은 프로토타입 템플릿을 생성하는 단계에서 계층적 군집화의 결과를 보여준다. 계층적 군집화 알고리즘의 수행 시 설정된 각 임계값(Threshold values)은 30, 50, 70, 과 100 랭크로 분해된 엔코딩마다 각각 2.2, 2.5, 2.8 과 3.2 였다. 이는 각 클래스마다 프로토타입의 수와 계산상의 효율, 그리고 분류율을 고려하여 실험적으로 정하였다.

아래 표 2 는 제안된 방법으로 폰트를 분류하였을 때의 정확도를 나타낸다.

표 2. EMD와 L1-norm을 사용한 평균 폰트 분류율 (%)

30 랭크		50 랭크		70 랭크		100 랭크	
EMD	L1	EMD	L1	EMD	L1	EMD	L1
92.29	89.8	96.1	91.43	96.0	91.8	98.06	92.73

랭크의 수가 작을수록 베이스스 이미지들이 사이의 상관관계(correlation)과 커지므로, 제안된 방법에서 랭크수가 작을수록 비교적 낮은 분류율을 나타내었다. 비록 훈련단계에 사용된 모든 엔코딩들을 템플릿으로 사용할 수 있지만, 분류대상이 많아질수록 계산량의 증가와 대용량의 데이터베이스를 유지해야 하는 부담이 존재한다. 또한, 거리척도로서 EMD 와 L1 norm 을 사용하여 제안된 방법을 평가한 결과 NMF 공간에서

의 거리척도로 EMD 가 L1 보다 좀더 적절한 것으로 나타났다.

5. 결론

제안된 논문에서는 한 장의 영상을 부분의 결합으로 표현할 수 있는 NMF 기술을 사용하여 다양한 폰트를 분류하는 새로운 방법을 제안하였다. 제안된 폰트 분류 방법은 한 폰트의 공간적으로 지역적인 특징을 학습하여 그 폰트의 특징으로 사용하는 방법이다. 실험은 문자 레벨로 실험되었고, 이러한 방식이 단어나, 블록단위의 폰트 분류 시스템들보다 문자 인식 시스템의 전처리기에 보다 적합한 방법이다. 또한, 단어 단위의 실험을 사용한다면 보다 좋은 분류 결과를 얻을 수 있을 것이라 기대된다. 본 논문의 초점은 NMF 기술이 개별적 글꼴들을 표현하는 특징들을 학습할 수 있다는 것을 검증하는 것이다. 향후 과제로 NMF 공간에서의 랭크의 수와 거리척도에 대한 연구를 수행하여 좀더 일반화된 폰트 분류 방법을 만들기 위해 노력할 것이다.

참고 문헌

- [1] G. Nagy, "Twenty Years of Document Image Analysis in PAMI," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 38-62, 2000.
- [2] S. Khoubryari, J. J. Hull, "Font and Function Word Identification in Document Recognition," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 66-74, 1996.
- [3] H. Shi, T. Pavlidis, "Font Recognition and Contextual Processing for More Accurate Text Recognition," *Proceedings of Document Analysis and Recognition 97*, pp. 39-44, 1997.
- [4] A. Zramdini, R. Ingold, "Optical Font Recognition Using Typographical Features," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 877-882, 1998.
- [5] Y. Zhu, T. Tan, Y. Wang, "Font Recognition Based on Global Texture Analysis," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1192-1200, 2001.
- [6] C. W. Lee, H. Kang, K. Jung, H. J. Kim, "Font Classification Using NMF," *Lecture Notes in Computer Science 2756*, pp. 470-477, 2003.
- [7] D. D. Lee, H. S. Seung, "Learning the Parts of Objects by Non-Negative Matrix Factorization," *Nature* 401, pp. 788-791, 1999.
- [8] D. D. Lee, H. S. Seung, "Algorithms for non-negative matrix factorization," *In Advances in Neural Information Processing Systems*, 13, pp. 556-562, 2001.
- [9] Y. Rubner, C. Tomasi and L. J. Guibas, "The Earth Mover's Distance, Multi-Dimensional Scaling, and Color-Based Image Retrieval," *Proc. ARPA Image Understanding Workshop (1997)* 661-668.
- [10] Y. Rubner, C. Tomasi and L. J. Guibas, "A Metric for Distributions with Applications to Image Databases," *Proc. IEEE Int. Conf. Computer Vision (1998)*
- [11] E. K. P. Chong and S. H. Żak, *An Introduction to Optimization*, A Wiley-Interscience, 2001.

표 1. 중심의 개수 (r = 30/50/70/100).

	Regular		Bold		Italic		Bold Italic	
	Capital	Small	Capital	Small	Capital	Small	Capital	Small
CN	28/29/25/25	31/28/28/24	30/30/27/23	26/26/24/23	33/31/28/28	32/31/26/28	32/37/33/26	33/32/32/28
CS	43/37/40/37	45/43/36/32	31/32/33/24	33/33/31/23	40/44/36/33	43/43/44/42	44/45/41/40	37/41/32/41
IM	25/21/23/16	23/22/16/16	16/16/16/16	16/16/16/16	16/16/16/16	16/17/16/16	16/19/19/16	19/16/16/16
MG	22/24/23/22	29/32/31/31	30/28/23/22	29/32/33/29	28/24/23/22	32/30/33/32	30/28/25/22	38/36/34/30
MS	22/21/21/19	26/26/26/23	30/24/23/25	27/32/33/28	26/24/25/22	29/28/27/28	27/23/24/21	30/30/28/26
TR	40/38/29/31	36/34/31/28	34/38/31/26	41/43/41/34	41/44/38/33	41/40/38/34	44/43/43/33	49/41/49/44