온라인 한글 인식을 위한 HMM 상태 수의 최적화

하진영*, 신봉기**
*강원대학교 컴퓨터공학과, **한국통신 멀티미디어연구소

Optimization of Number of States in HMM for On-line Hangul Recognition

Jin-Young Ha*, Bong-Kee Sin**
*Dept. of Computer Eng., Kangwon National University
**Multimedia Research Labs, Korea Telecom

요 약

온라인 문자 인식을 위해 시도된 여러 방법 중 은닉 마르코프 모델(HMM)이 우수한 성능을 보이고 있다. 영숫자 인식은 물론 한글 인식에 있어서도 HMM은 최근 널리 사용되고 있는데, HMM을 이용해서 모델링 할 때 해결해야 할 문제 중의 하나는 HMM의 구조를 어떻게 최적화 하느냐이다. 본 논문에서는 HMM을 이용한 온라인 한글 인식 시스템에서 HMM의 최적화를 통해 인식률을 향상시키고자 한다. 특히 HMM의 상태(state) 수를 어떻게 정할 것인가에 초점을 맞춰, 실험을 통해 최적의 HMM 상태 수를 찾고자 한다.

1. 서 론[1)]

문자 인식에 관한 연구는 지난 30 여년 동안 많은 연구가들에 의해 연구되었고, 다양한 방법론이 적용되어 왔다[1]. 특히 최근 우수한 성능을 보이고 있는 은닉 마르코프 모델(HMM hidden Markov model)은 음성 인식 분야에 먼저 적용되어 그 모델링의 우수성이 입증되었고, 1980년대 후반부터 문자 인식에 적용되어 좋은 결과를 내기 시작했다[2,3].

HMM을 온라인 문자 인식에 적용할 때의 모델 구조는 대부분 왼쪽에서 오른쪽으로 전이(transition)가 허용되는 모델(left-to-right model)로서 전이의 형태는 크게 다르지 않지만, 상태(state)의 수를 결정하는 것은 어려운 문제 중의 하나이다. HMM의 입력으로 사용될 관측열(observation sequence)의 평균 길이의 1.5배로 상태 수를 정하는 휴리스틱과 데이터의 복잡도와 반복 구조 분석을 통해 적절한 상태 수를 정하는 방법, 그리고 상태 수를 점차 변화시켜 가면서 인식률이 최고로 되는 상태 수를 찾는 방법 등이 사용되었다[2-4].

본 논문에서는 HMM 네트워크 기반의 온라인 한글 인식기에서 HMM의 최적 상태 수를 찾아내기 위한 방법에 대해 연구하고, 실험을 통해 그 결과를 비교하고자 한다.

2. HMM 네트워크 기반의 한글 인식기

본 논문에서는 온라인 한글 인식에 성능을 보인 BongNet을 기반으로 인식기를 개발하였다. BongNet은 각 자소 모델과 연결획 모델을 기반으로, 이를 한글의 글자 조합 원리를 이용하여 연결함으로써 한글 필기를 모델링한 네트워크 구조이다[3]. <그림 1>은 BongNet의 구조를 보여주고 있다. 한글에는 초성 19자, 중성 21자, 종성 27자 등 총 67개의 자소가 있으며, 각각의 자소에 발생하는 필기 습관, 필기 상태, 전체 자소의 결합 형태에 따른 다양한 변형을 흡수하기 위하여 각각을 은닉 마르코프 모델로 모델링 하였다. 또한 한글 필기 시에 발생하는 자소 간의 흘림을 위한 연결획(ligature) 모델 개념을 도입하여, 연결획의 시작부분과 마침 부분의 상대적인 위치에 따라 모델을 나누어 각각을 은닉 마르코프 모델로 모델링 하였다.

한글을 필기할 때에는 초성, 중성, 종성의 순서로 필기한다. 추가적으로 연결획을 고려하면, 한글 한 음절은

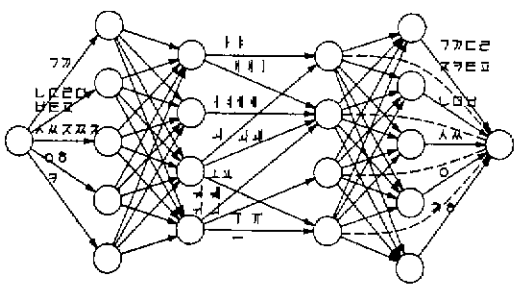
(초성)+(연결획)-(중성)

(초성)+(연결획)-(중성)+(연결획)-(종성)

과 같은 확장된 자소열로 표현할 수 있다. 이러한 방법으로 각각의 자소 모델과 연결획 모델을 연결한 것이 BongNet이다. 네트워크의 시작 노드에서 종료 노드로의 각 경로는 하나의 글자에 해당한다. 인식은 주어진 글씨에 대해서 그 글씨가 표현하고자 하는 글자의 경로를 찾는 작업이라고 할 수 있다. 즉 통계적으로 가장 유사한 경로를 찾는 문제로, Viterbi 알고리즘을 사용하여 효율적으로 구할 수 있다[2,3].

3. HMM의 구조와 훈련

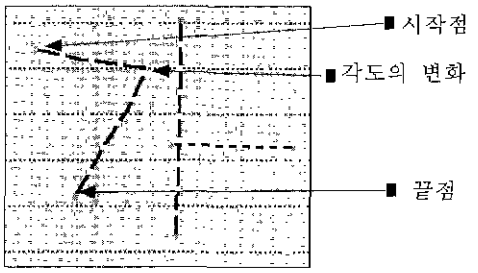
3.1 관측열의 생성



<그림 1> BongNet의 구조

1) 본 연구는 정보통신부의 정보통신연구관리단의 지원에 의한 강원대학교 멀티미디어 특화연구센터의 지원을 일부 받았음을 밝힙니다.

BongNet에서는 필기 입력 데이터를 16-방향 체인코드로 변환해서 관측열로 사용했지만, 정서체 인식률과 인식 속도 향상을 위해 구조적 정보를 포함하는 새로운 코드를 사용하였다[5]. 그래픽 테블렛으로부터 입력되는 데이터는 먼저 거친점 제거(wild point reduction)와 평활화(smoothing)등의 전처리 과정을 거친 후 계산량을 줄이기 위해 일정한 간격으로 다시 샘플링한다. 각 획의 시작점으로부터 끝점까지의 각 점 사이의 각도의 변화에 따라 <그림 2>와 같이 특징점을 추출한다. 각도의 변화에 의한 특징점 뿐만 아니라 시작점과 끝점도 특징점에 포함시켰다.

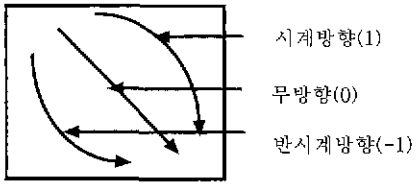


<그림 2> 특징점의 추출

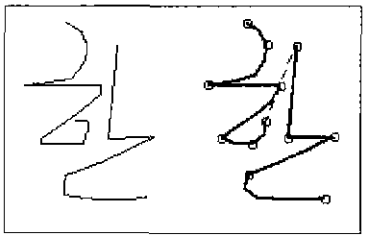
이와 같은 특징점 추출 방법은 정서체 뿐만 아니라 흘려쓴 대부분의 필기 한글의 자소 간 경계점을 누락시키지 않고 찾아낼 수 있다

구조적 정보를 포함하는 코드열을 생성하기 위해, 각도의 변화에 따른 특징점 사이의 각 부분획으로부터 다음과 같은 5차원의 벡터를 생성한다

- Distance: 특징점 사이의 각 점 간 거리의 합(입력문자의 높이를 100으로 정규화)
- Straightness: 부분획의 곧은 정도로서 시작점과 끝점 사이의 직선 거리를 누적거리로 나눈 비율(완전한 직선일 경우 100%)
- Direction: 시작점에서 끝점으로 향하는 각도(0도~360도)
- Real: 실제획이면 1, 가상획이면 0
- Rotation: 부분획의 굴곡 방향을 시계방향(1), 반시계방향(-1), 그리고 무방향(0)의 3가지로 분류 (<그림 3> 참조)



<그림 3> Rotation



<그림 4 > 특징점 추출의 예

특징 벡터들을 비지도 학습(unsupervised learning)에 널리 사용되는 K-Means Clustering 알고리즘[6]을 사용하여 64개의 클러스터로 분류한 후 그 중심(center)의 평균을 코드북(codebook)에 기록하여 새로운 입력 벡터를 관측열(observation sequence)로 변환시킬 때 사용한다[5]. <그림 4>는 한글 필기 입력 데이터 '한'에서 특징점을 찾은 예를 보여주고 있다. 특징점 사이의 선분을 특징 벡터로의 변환한 후 구조 코드 관측열로 변환한 결과를 <표1>에 표시하였다.

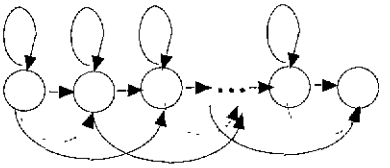
<표 1> 특징 벡터와 구조 코드

Distance	Straightness	Direction	Real	Rotation
18.213	96.828	46 813	1	1
45 934	87 675	146.230	1	1
41.238	99.997	0 301	1	0
45.796	97 941	138.221	1	1
18 299	98 820	8.777	1	0
16 531	88 824	298.276	1	-1
45.957	100.000	293 199	0	0
52 815	99 999	95.421	1	0
26.917	99.961	356.488	1	0
53 588	99 881	156.188	1	0
56.840	80.187	17 449	1	-1

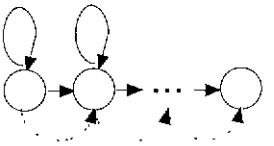
==>(관측열) 3 37 0 37 0 31 54 11 28 15 5

3.2 HMM의 구조

각 자소 모델과 각 연결획 모델의 기본 구조는 <그림 5>에 나와 있다. <그림 5>에서의 파선은 관측열의 소모 없이도 진이가 가능한 널-진이(null-transition)을 나타낸다 자소 HMM에서 두 번째와 세 번째 상태 사이에는 널-전이를 허용하지 않았다. 이것은 널-전이만을 통해 첫 번째 상태로부터 마지막 상태로의 전이를 허용하지 않게 하기 위함이다 자소 HMM의 상태 수는 3개에서 20개까지 변화시키며 실험하도록 하였고, 연결획 HMM은 2개에서 4개까지 변화시켰다



(a) 자소 HMM



(b) 연결획 HMM

<그림 5> HMM 구조

3.3 HMM의 훈련 및 인식

HMM의 훈련을 위해 문자 단위의 한글 데이터를 수작업으로 자소 데이터와 연결획 데이터로 분할한다. HMM의 파라미터(parameter) 수가 많기 때문에 충분한 데이터의 확보가 요구된다. 모델 파라미터의 훈련은 널리 쓰이는 Baum-Welch 알고리즘을 수행한다[2]. 모델의 상태 수를 변화시키면서 각각의 경우에 대한 모델의 파라미터를 훈련시킨다. 상태 수의 변화에 따른 모델링 정도는 훈련된 HMM을 이용하여 훈련 데이터에 대한 모델링 확률의 평균을 구함으로써 측정할 수 있다.

모델의 훈련이 완료되면, <그림 1>과 같이 구성된 HMM의 네트

워크에서 주어진 필기 데이터에 대한 최대 확률을 갖는 경로를 Viterbi 알고리즘을 이용하여 구할 수 있다. 자세한 방법은 논문[3]를 참조 바란다.

4. 실험 및 결과 분석

한글 모델의 훈련과 인식 실험을 위해 KAIST의 온라인 문자 DB를 사용하였다. 필기 문자의 획득은 WACOM SD-510C 디지타이저를 이용하였으며 총 41명이 필기한 것을 이용하였다. 한글 자소 훈련을 위해 총 214,176 개의 자소 데이터를 사용하였고, 연결획 모델의 훈련을 위해 총 150,341 개의 연결획 데이터를 사용하였다

Sun Ultra-2 200MHz 워크스테이션 상에서 훈련 데이터와 무관한 8명이 필기한 6,308자의 정서체 한글에 대해 각 상태 수에 따라 인식률을 측정하였다 <표 2>는 상태 수가 4일 때 최고의 인식률 93.74%를 얻을 수 있다는 것을 보여준다. 이 결과는 모든 자소 모델의 상태 수를 동일하게 적용하는 것이 부적합하다는 것을 말해주고 있다 한글의 자소는 중성 'ㅣ'의 경우 매우 단순하고, 초성 'ㅃ'과 종성 'ㄹㅂ', 'ㄹㅌ', 'ㅂㅅ'의 경우는 매우 복잡하기 때문에 동일한 수의 상태 수로 모델링하는 것보다는 각 자소의 복잡도에 따라 다른 상태 수를 갖게 하는 것이 더 적합하다

본 논문에서는 각 자소 모델마다의 최적의 상태 수를 구하기 위해, 훈련 데이터에 대한 모델 확률이 최대가 되는 상태 수를 찾는 방법을 적용하였다 가장 단순한 자소인 중성 'ㅣ'의 경우 상태 수가 3일 때 최대의 확률을 보였고, 가장 복잡한 자소인 종성 'ㄹㅂ', 'ㄹㅌ', 'ㅂㅅ'의 경우에는 20개의 상태 수일 때 최대 확률값을 관측할 수 있었다. 비교적 단순한 자소인 초성 'ㄱ'은 4, 'ㄴ'은 5일 때, 중성 'ㅏ'는 5, 종성 'ㅅ'은 6일 때 각각 최대 확률값을 관측할 수 있었다(<그림 6> 참조). 같은 방법으로 한글 모든 자소에 대해 HMM이 훈련 데이터의 최대 확률값을 갖는 상태 수를 구한 후 인식률을 다시 측정했는데, 97.26%의 평균 인식률을 얻을 수 있었다. 이것은 동일한 상태 수를 갖는 자소 모델을 사용할 때보다 3.52%의 인식률 향상을 보인 것이다(<표 3> 참조). 연결획 모델에 대한 모델링 확률값은 <그림 7>과 같다. 모든 연결획에 대해서 상태가 2일 때 최대 확률값을 갖는다는 것을 관측할 수 있었다

<표 3> 인식률의 비교

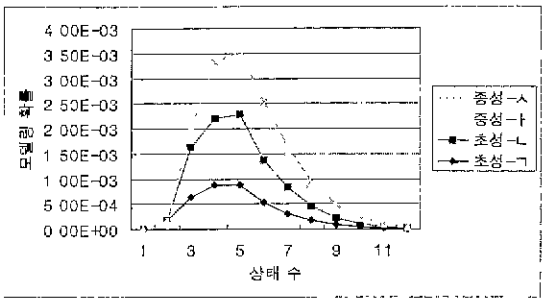
	최대 인식률 동일 상태수	자소 모델마다의 최대 확률 상태수	증감
인식률	93.74 %	97.26 %	+3.52 %

5. 결론

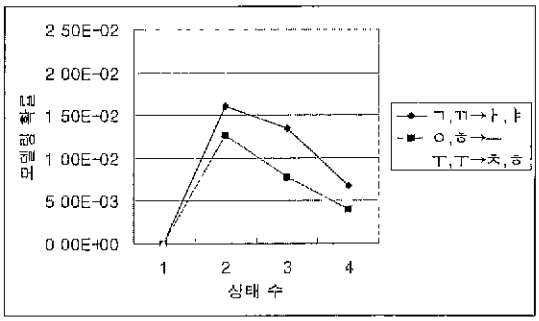
HMM 기반의 온라인 한글 인식기의 성능을 높이고자 하는 목적으로 HMM 최적화에 대해 실험하였다. 각 자소 모델마다 동일한 상태 수를 갖는 HMM으로 구성할 때보다 각 자소 HMM의 훈련 데이터에 대한 최대 모델링 확률값을 갖도록 각각의 자소 HMM의 상태 수로 HMM의 구조를 정하는 것이 좋다는 결론을 얻을 수 있었다. 이렇게 하는 것은 한글 자소의 복잡도의 다양성을 자연스럽게 반영할 수 있는 방법이기도 하다.

<표 2 > 자소 모델의 상태 수에 따른 인식률

상태 수	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
인식률(%)	84.95	**93.74**	93.33	91.19	91.54	90.23	92.02	90.90	90.55	89.68	88.83	87.94	86.57	86.89	86.43	85.44	84.93	83.84



<그림 6> 일부 자소에 대한 모델링 확률값



<그림 7> 일부 연결획에 대한 모델링 확률값

향후 연구 과제로는 HMM 구조의 최적화를 위한 수학적 모델링이 필요하다

참고문헌

[1] 이성환, 문자인식 - 이론과 실제, 홍릉과학출판사, 1993.

[2] L.R Labiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Porc. IEEE, Vol. 77, 1989, pp 257-285.

[3] B.-K. Shin and Jin H Kim, "Ligature Modeling for online cursive script recognition," IEEE Trans. Pattern Recognition and Machine Intelligence, Vol 19, No 6, 1997 pp 623-633

[4] J. Hu, S G Lim, M.K. Brown, "HMM Based Writer Independent On-Line Handwritten Character And Word Recognition," Proc. Int Workshop on Frontiers in Handwriting Recognition, Taejon, Korea, 1998, pp 143-155

[5] 히진영, "HMM 네트워크 기반의 한글 인식기를 위한 구조 특성열의 적용," 제 10회 한글 및 한국어 정보처리 학술대회, 1998(to appear).

[6] R. Schalkoff, Pattern Recognition - Statistical, Structural, and Neural Approaches, John Wiley & Sons, Inc., 1992