다문자 획의 특성을 이용한 전처리 기법

이수봉°, 김우생
광운대학교 컴퓨터공학부

Preprocessing Technique Using a Feature of Character's Stroke

Su-Bong Lee, Woosaeng Kim
School of Computer Engineering, Kwangwoon Univ.

요 약

온라인 문자 인식 기술은 PDA, 테블릿 PC등 많은 새로운 응용에서 사용되고 있으나, 인식 기술은 아직 이러한 첨단 도구들을 자연스럽게 이용하기에는 못 미치는 실정이다. 따라서 본 논문에서는 인식률을 높이기 위해 전처리 과정에서 문자를 구성하는 획수를 통해 인식 시 해당 HMM 모델들에게만 적용하여 인식 시간을 줄이고 동시에 오류도 줄이고자 한다. 제안하는 방법들의 타당성은 실험을 통해서 검증하였다.

1. 서론

전자 펜은 휴대성과 편의성의 장점을 지녀 PDA, palm, pocket PC, 스마트폰, 테블릿 PC 등의 많은 응용에 사용될 수 있기에 높은 인식률의 온라인 문자 인식 방법이 더욱 필요하다[1][2].

온라인 문자 인식 시스템은 전처리를 한 후에 정합을 하게 되는데, 이러한 정합 연구들에 쓰인 방법들 중에서 은닉 마르코프 모델 (HMM)은 음성 인식 분야뿐 아니라 문자 인식에서도 성공적으로 적용되고 있다. HMM은 잘 정의된 수학적 이론을 배경으로 하며, 전이 확률과 관찰 확률의 이중 확률 구조로 온라인 문자 입력의 특성인 시간에 따른 획의 변이와 공간적인 모양 변이를 동시에 모델링 하기에 적합한 특성을 보인다[3][4].

본 논문에서는 문자를 구성하는 획들의 특성을 이용해 인식률을 높이는 방법에 대해서 연구하고자 한다. 영문자와 숫자는 최대 4개 획의 조합으로 하나의 문자가 구성되기 때문에, 본 논문에서는 문자의 이러한 특성을 통해 전처리 과정에서 문자를 구성하는 획수를 구한 후에, 인식 시 해당 HMM 모델들에게만 처리를 하게 함으로써 인식 시간을 줄이고 동시에 오류도 줄이는 방법을 제안한다. 제안하는 방법들의 타당성을 효율적으로 검증하기 위해서 본 논문에서는 외부 분할 된 영어 알파벳 문자와 숫자만을 인식의 대상으로 한다.

본 논문의 구조는 다음과 같다. 먼저 2장에서는 은닉 마르코프 모델을 소개하고, 3장에서는 문자를 구성하는 획의 조합수를 이용한 전처리 방법을 설명한다. 4장에서는 HMM 모델의 구조와 훈련 방법, 5장

에서는 실험 및 결과를 보이며 6장에서 결론을 맺는다.

2. 은닉 마르코프 모델

은닉 마르코프 모델은 유한개의 상태와 각 상태 사이를 방향성 있게 연결하는 전이의 집합으로 구성된 네트워크로 정의된다. 상태간 전이 확률은 마르코프 프로세스를 따르며 출력 확률은 시간에 독립적으로 각 상태에 지정되어 있다. 일반적으로 HMM은 λ = (A, B, π)의 기호로 표현되며 A는 상태전이 (state transition) 확률, B는 관측심볼 (observation symbol)의 출력확률, π는 초기 상태전이 (initial state transition) 확률을 나타낸다. 한 문자에 대해 하나의 모델이 구성되고 서로 다른 n개의 문자에 대해 대응하는 각각의 모델을 , i =1,2,.., n으로 표기 할 때, 관측열 O = O1, O2, ···., OT에 대응하는 미지의 문자는 식 (1)을 만족하는 모델에 대응하는 문자로 인식된다.

$$\lambda_j = \max_{i=1..n} P(O \mid \lambda_i) \quad \text{---------- (1)}$$

3. 문자 구성하는 획의 조합 수를 이용한 전처리 방법

영어 문자와 숫자의 경우는 최대 4개 획의 조합으로 하나의 문자가 구성된다. 예를 들어, 문자 'C' 와 같은 일부의 문자들은 사람에 관계없이 한 가지 종류인 1 획의 조합으로만 만들어 진다고 볼 수 있다. 그러나 많은 문자들은 사람에 따라 두, 세 가지 종류의 획의 조합으로 만들어 질 수 있다. 예를 들어 문자 'E' 의 경우는 사람에 따라 외곽의 'ㄷ' 과 '- '의 2획, 또는 맨 위의 '- '와 중간의 '- '와 나머지 'ㄴ' 의 3획, 또는 '|' 와 맨 위의 '- '와 중간의 '- '와 맨 아래의 '- '의 4획의 세가지 종류의 획의 조합으로 만들 수 있다.

영어 대문자의 경우, 문자를 구성하는 획 조합의 경우 수가 <표 1>에 나타나 있다. 표 1에서 'O' 표시는 문자가 해당 획 조합으로 구성 될 가능성이 많은 경우를 의미하며 'Δ' 기호는 가능성은 희박하지만 존재 할 수 도 있는 경우를 의미한다. 따라서 'Δ' 의 경우로 문자가 인식이 될 경우는, 후처리 작업 등에서 인식이 제대로 되었는지에 대한 검증 대상이 될 수 있다[5][6].

문자	1획	2획	3획	4획	문자	1획	2획	3획	4획
A	Δ	O	O		N	O	O	O	
B	Δ	O	O		O	O			
C	O				P	Δ	O		
D	Δ	O			Q	O	O		
E	Δ	O	O	O	R	Δ	O	Δ	
F		O	O		S	O			
G		O	O		T	Δ	O		
H			O		U	O			
I	O		O		V	O	Δ		
J	O	O			W	O	O		Δ
K		O	O		X		O		
L	O				Y		O	O	
M	O	O	Δ	Δ	Z	O	O		

<표 1>

반면에 영어 소문자의 경우는 최대 2개 획의 조합으로 하나의 문자가 구성된다. 즉, 'f' , 'p' , 'z' 는 1개 또는 2개의 획의 조합으로 쓸 수 있고, 'i' , 'j' , 't' , 'x' 의 경우는 오직 2개 획의 조합으로 쓸 수 있으며, 나머지 19개의 영어 소문자는 오직 1개 획으로만 쓸 수 있다. 숫자의 경우도 표 2와 같이 최대 2개의 획의 조합으로 쓸 수 있다.

숫자	1획	2획	숫자	1획	2획
0	O		5	O	O
1	O	O	6	O	
2	O		7	O	O
3	O		8	O	O
4	Δ	O	9	O	

<표 2>

따라서 앞에서 설명 된 문자 당 획 조합의 경우

수를 이용하면 영문자와 숫자에 대응하는 HMM의 모델들을 <표3>와 같은 4개의 그룹으로 구분을 할 수 있다. 즉, 한 획으로 필기 할 수 있는 문자들로부터 네 획으로까지 필기 할 수 있는 문자들로 구분이 된다. 단, 한 문자는 여러 그룹에 동시에 포함될 수도 있다. 따라서 사용자가 문자 'A' 를 3획으로 필기를 하였다면, 시스템은 입력된 문자를 영어와 숫자의 62 가지의 모든 모델에 적용할 필요 없이 관련이 있는 12 가지의 3획 모델에만 적용을 하게 되므로 인식 시간을 줄이며 동시에 인식 오류를 줄일 수 있게 된다. 따라서 시스템에서는 어떤 문자가 입력되면 우선 몇 획으로 필기 된 문자인가를 결정한 후, 해당하는 모델들에만 적용하여 인식을 시도한다.

	영어문자	숫자문자
1획 HMM 모델 (50문자)	a,b,c,d,e,f,g,h,k,l,m, m,o,p,q,r,s,u,v,w,y,z, A,B,C,D,I,J,L,M,N,O,P,Q,R ,S,T,U,V,W,Z	0,1,2,3,5, 6,7,8,9
2획 HMM 모델 (29문자)	f,i,j,p,t,x,z,A,B,D,E,F, G,J,K,M,N,P,Q,R,T,V, W,X,Y,Z	1,4,5,7,8
3획 HMM 모델 (12문자)	A,B,E,F,G,H,I,K, M,N,R,Y	
4획 HMM 모델 (2문자)	E,M,W	

<표 3>

각 문자 당 획 조합의 경우 수가 균등하게 발생한다고 가정 하면, 예를 들어 'A' 문자의 경우 평균 30개의 문자에 해당하는 HMM 모델에서만 인식을 적용하면 된다. 따라서 영문자와 숫자만을 고려했을 때 같은 원리로 각 문자 당 평균 42개의 HMM 모델에 적용하면 되므로 기존의 방법에 비해 개략 2/3의 시간 속도와 오인식률을 갖게 된다.

4. HMM 구조 및 훈련

초기 본 논문에서 사용된 HMM 모델의 구조는 left-to-right 모델이며, 상태 전이의 수는 자기 전이와 다음 상태까지만 전이 할 수 있는 구조를 갖는다. 각 문자에 대응하는 HMM의 상태 수는 10개로 하였고, 각 상태에서의 각 심볼의 발생 가능성은 같게 하였으며 초기 전이 확률 분포는 균일 확률 분포로 하였다.

각 문자는 인터페이스의 박스 당 한 문자씩 필기 하게 하여 외부 분할 방식으로 분리되게 하였으며, 테블릿으로부터 입력되는 데이터는 먼저 거친 점 제거와 평활화 등의 전처리 과정을 거친 후 코드화하기 위해 일정 시간 간격으로 샘플링 된 일련의 점들을 8 방향 체인코드로 만들어 사용하였다[7]. 본 논문에서는 훈련을 위해 Baum-Welch 알고리즘을 사용하였다[3].

5. 실험 및 결과 분석

제안한 방식들의 성능을 분석하기 위해 TMC300XCi 테블릿 피시를 사용하였으며, C#과 Microsoft Tablet PC Platform SDK API로 구현하였다[8]. 본 논문에서 문자 훈련은 5명의 데이터를 사용하였고, 각 사람은 각 문자를 3번 필기하도록 해, 각 모델은 평균적으로 15개의 해당 문자를 가지고 훈련되었다. 인식 실험은 훈련에 참여하지 않은 5명의 데이터를 사용하였으며, 전처리 기법을 적용한 방식과 적용 하지 않은 방식에 실험 데이터를 사용하여 양 측의 성능을 비교 하였다.

5.1 획의 조합 수를 이용한 전처리 기법 인식률 분석

문자를 인식할 때 본 논문에서 제안한 전처리 방법을 적용했을 때와 적용하지 않았을 때의 인식률을 평가하였다. <표 4>에서 볼 수 있듯이 전처리 방법을 적용한 결과 그렇지 않았을 때 보다 인식률이 3.4% 가량 상승하였다.

	전처리 기법 적용	전처리 기법 적용하지 않음
총 문자	320	320
오인식	7	18

인식률	97.8%	94.4%

<표 4>

5.2 획의 조합 수를 이용한 전처리 기법 인식 속도 분석

문자를 인식할 때 본 논문에서 제안한 전처리 방법을 적용했을 때와 적용하지 않았을 때의 인식 속도를 평가하였다. 인식 속도는 학습 이후 문자 당 인식에 걸린 시간을 측정하여 평균하였다. <표 5>에서 볼 수 있듯이 전처리 방법을 적용한 결과 그렇지 않았을 때 보다 32%의 인식에 드는 시간이 감소하였음을 알 수 있다.

	전처리 방법 적용	전처리 방법 적용하지 않음
평균 문자 인식 속도	0.0051초	0.0076초

<표 5>

6. 결론 및 향후 연구

본 논문에서는 기존 HMM 모델의 인식률을 높이고 인식 시간을 줄이기 위해 문자 획의 특성을 이용한 전처리 기법을 제안 하였으며, 인식률의 향상과 인식 시간의 감소를 실험을 통해 보였다. 향후 연구로써는 제안한 전처리 기법을 한글 문자에도 적용해 보고자 한다.

[참고문헌]

[1] 이성환, "온라인 필기 인식 기술의 현황," 한국 정보과학회지, 제 9권 제 1호, 1991년 2월, pp. 54-63

[2] Alesxander Wolfe, "Putting Pen to Screen On Tablet PCs", IEEE Spectrum, October 2002.

[3] 이성환, "문자인식-이론과 실제 1, 2권, 홍릉과학출판사", 1993.

[4] L. R. Labiner, "A Tutorial on HMM and Selected Applications in Speech Recognition," Proc. IEEE, 1989

[5] S. N. Srihari, J. J. Hull and R. Choudari, "Integrating Diverse Knowledge Sources in Text Recognition," ACM Trans. on Office Information Systems. 1983.

[6] R. Shinghal, "A Bottom-up and Top-down Approach to Using Context in Text Recognition," Int. Journal of Man-Machine Studies, 1979.

[7] 구본석, 전변환, 박명수, 김성훈, 안진모, 김재희, "On-line 문자 인식을 위한 전처리 기법, 한국 정보과학회 인공지능 연구회 추계 학술발표 논문집, 서울, 1991년 11월, pp. 78-82

[8] Rob Jarrett, Philip Su, "Building Tablet PC Applications," Microsoft Press, 2003