봉네트 — 그후 일년

신 봉 기, 김 진 형
한국과학기술원 전산학과

BongNet — One Year After

Sin, Bongkee and Kim, Jin Hyung
Computer Science Department, KAIST

요 약

봉네트는 온라인 한글 필기 글씨 모델이다 [신92]. 글씨를 자소와 연결획의 결합구조로 보고, 각 자소 및 연결획 모델을 정의한 후, 이들을 제자 원리에 따라 네트워크 구조로 설계한 모델이다.

본 논문에서는 봉네트가 소개된 후 지난 일년 동안 수행되었던 실험 및 모델 검증의 결과와 앞으로도 계속될 개선책을 소개하고, 동 모델의 바탕이 된 통계적 인식 이론을 정립하고자 한다.

1 서 론

봉네트(*BongNet*, 그림 1)는 은닉 마르코프 모델(hidden Markov model, HMM)을 이용한 온라인 한글 필기 모델이다 [신92]. 글씨 패턴을 자모 및 연결획 HMM으로 분리 모델링하고, 이들을 다시 제자 원리에 따라 연결한 네트워크(finite state network, FSN) 구조의 한글 음절 필기 모델이다. 봉네트의 특징은 다음과 같다:

- HMM으로 다양한 필기를 모델링한다. 봉네트의 구성은 자소 단위의 모델을 기반으로 되어 있으며, 자소 모델은 모델링 능력이 뛰어난 HMM으로 표현한다.
- 연결획 모델이 있어 흘려쓴 글씨도 수용한다. 필기는 펜촉의 궤적으로 이루어 지고 자소와 자소 사이에는 여러가지 형태의 연결획이 존재하게 된다. 이러한 패턴을 연결획 모델이라는 독립적인 HMM으로 모델링한다. 훈련 샘플에 패턴에 적응하여 자소간 연결이 있는 흘림 필기를 모델링 한다.

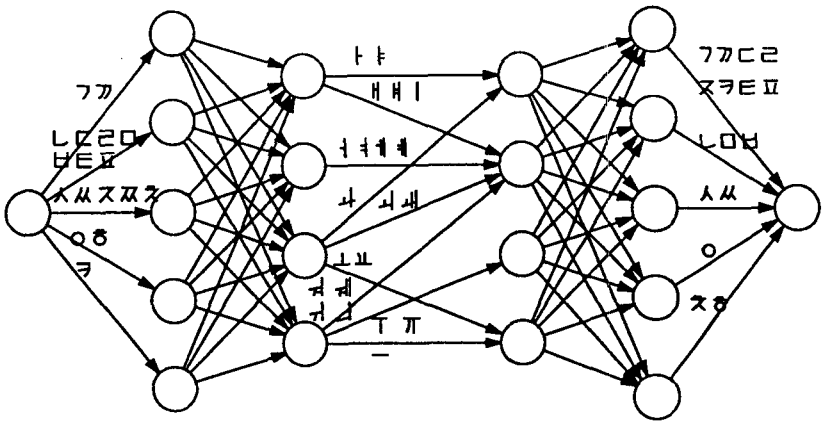

그림 1: 한글 필기 글자 모델 — 봉네트*(BongNet)*

- 제자 원리를 담은 네트워크 구조를 가지고 있다. 자소 모델과 연결획 모델을 글자 형성 규칙에 맞춰 네트워크로 구성한 한글 필기 오토마타이다.

- 통계 모델로써, 인식에 관련된 여러가지 정보의 결합이 용이하다. 네트워크 모델의 각 파라미터는 한글 필기의 통계적 특성을 표현하며, 입력 필기 패턴이 확률값으로 추정된다. 모든 지식과 정보원을 통계 모델로 정의함으로써 동일 측정선 상에서 정보의 결합이 이루어 질 수 있다.

- 네트워크상의 완전 경로가 글자를 구성하고, 입력 코드열과 비선형 배열된다. 봉네트의 시작 노드에서 최종 노드까지 연결된 모든 경로는 하나의 글자에 대응 된다. 주어진 입력 코드열을 이러한 모든 경로에 배열했을 때 국부적으로 왜곡이 있는 비선형 배열이 일어난다.

- 네트워크 디코딩으로 인식과 자모분할이 동시에 일어난다. 입력에 대해 확률적으로 최적 경로와 비선형 배열을 얻을 수 있으며, 그 경로상의 모델 경계점이 곧 입력 코드열상의 자소 경계를 결정한다.

- 일관된 표현 구조, 일관된 계산 구조를 갖고 있다. HMM이라는 유한 상태 네트워크, 그리고 이들의 네트워크 구조가 봉네트이다. 이 모델위에서 실현된 계산 구조는 Viterbi 알고리즘 [Vite67] 하나로 이루어져 있다.

2 인식 모델

2.1 인식기 모델

MAP (maximum *a posteriori*) 원리에 근거한 패턴 인식은 *a posteriori* 확률 $Pr(W|X)$에 따라서 결정하는 문제이다. [Duda73] 여기서 W는 모델이고 X는 입력 패턴을 나타낸다. MAP 원리란 결정에 따른 오류를 최소화하는 결정 이론인데

$$Pr(\hat{W}|X) = \max_{W} Pr(W|X). \tag{1}$$

으로 표현된다. 이는 다시 Bayes 정리에 따라서

$$Pr(W|X) = \frac{Pr(W,X)}{Pr(X)}. \tag{2}$$

로 변환할 수 있다. $Pr(X)$은 W와는 독립이므로 인식기의 모델이 고정되었을 때에는 상수가 된다. 그러므로 $Pr(\hat{W}|X)$를 최대값은 곧 확률 $Pr(W,X) = Pr(X|W)Pr(W)$를 최대로 하는 것이 되고, 최적 인식기는

$$Pr(\hat{W},X) = \max_{W} Pr(W,X). \tag{3}$$

에 근거하여 동작하게 된다. 그런데 실제 인식기는 $Pr(W,X)$의 정확한 분포를 알지 못하기 때문에 $Pr_m(W,X)$로 추정을 하게 된다. 그리고 인식 오류율을 최대한 줄이기 위해서, $Pr(W,X)$ 값이 커서 실제로 W가 가장 유망할 때, $Pr_m(W,X)$를 최대로 해주는 모델을 가져야 한다.

일반적으로 인식기 모델 m은 — 언어 정보 처리의 관점으로 국한을 시키도록 하면 — 언어 모델과 패턴 매치 모델의 두 부분으로 나뉜다. 이들 각각은

$$Pr(W,X) = Pr(X|W)Pr(W). \tag{4}$$

에서 $Pr(W)$와 $Pr(X|W)$를 추정하는 모델이다. $Pr(X|W)$는 글씨 생성자, 예를 들면, 사람의 손이 자신의 머릿속에 있는 정보를 전달하기 위하여 W로부터 출력(스트링) X를 생성할 확률이다. $Pr(X|W)$을 추정하기 위해서는 W에 패턴에 대한 확률 모델을 세우고, 그 형태와 다양성 등에 대한 특징을 표현하도록 해야 한다. HMM을 이용한 한글 필기 모델이 한 예에 해당한다. 그 네트워크에서 임의의 글자 W는 유일한 경로 하나에 대응한다. 인식시에는 각 경로에 대해서 $Pr_m(X|W)$이 계산된다. 모든 가능성을 열거해서 최대값 $Pr_m(\hat{W},X)$을 계산

하는 것은 일반적으로 지수 함수 복잡도를 가져 비현실적이다. 다행히 동적 프로그래밍 기법을 응용한 매우 효율적인 알고리즘이 있어 이론적, 실제적 돌파구는 마련되어 있다.

식 (4)의 $Pr(W)$은 글자 또는 단어의 *a priori* 확률로써 언어 정보 생성기 (text generator, 예를 들어 사람[의 머리])가 그 글자 또는 단어를 생성할 확률을 의미한다. 그러므로 $Pr(W)$에 대한 언어 모델은 글자 또는 단어에 대한 확률 모델이다. 간단히 말하면 W에 대한 확률 생성 모델이라고 할 수 있다. 아주 제한된 문제에서는 언어 모델이 단순하여, 종종 간단한 finite state 문법 또는 contex-free 문법을 정의하고 어렵지 않게 확률을 첨가할 수 있지만, 일반적인 문제에 대해서 $Pr(W)$를 추정하는 믿을만한 자연 언어 모델을 세우는 것은 거의 불가능하다. 현재 널리 쓰이고 있는 것은 언어 정보 생성기를 Markov source로 가정하고, 막대한 양의 텍스트를 모아서 파라미터를 추정해서 얻은 N-gram 모델이다.

2.2 네트워크 탐색

동적 프로그래밍 (DP) [Bell57] 기법에 의한 패턴 매칭은 70년대 초반 음성 인식 분야에 적용되기 시작했다 [Vint71] [Sako71] [Baker75]. 이후 DP에 의거한 여러가지 형태의 탐색 구조가 개발되었다 [Sako79] [Myer81] [Ney84]. 동적 프로그래밍이란 복잡하고 커다란 문제를 일련의 작은 문제로 분해하여 푸는 방식이다. 실제 구현의 관점에서 보면 DP는 가능한 모든 소문제에 대한 해답표(lattice)를 작성해가는 과정이라 할 수 있다.

[Baker75]의 연구 이래 음성 인식 문제는 유한개의 노드와 노드간 천이가 있고, 이로써 다양한 정보원(knowledge source, KS)과 그들간의 관계를 표현하는 유한 상태 네트워크(finite state network, FSN)의 계층 구조로 정립할 수 있게 되었다. 모든 KS를 FSN의 형태로 표현한다는 생각은 간단한 언어 모델과 더불어, 크지 않은 연속 음성 인식 문제 영역에서 상당한 성공을 거두었다고 평가된다. 일단 여러가지 KS를 포괄하는 FSN이 구성되면, 네트워크 상에서 최적 해답을 찾는 문제는 Bellman [Bell57]의 최적의 원리 *(principle of optimality)*에 따라 순차적인 디코딩 문제로 정립이 된다. 이 방식에 따르면 프레임 (또는 심볼) 단위로 디코딩이 가능한데, 각 단계에서 모든 local optimum 경로를 계산하고 이를 일련의 프레임 등에 따라서 순환 계산을 거치면 결국 global optimum에 도달한다는 것이다.

DP 적용에 들어간 중요한 개념은 미지의 입력 패턴을 각 모델과 비선형 배열(time-aligning)을 한다는 것이다. 연속 단어 인식에서 가장 큰 문제는 단어간 경계를 구하는 것이었는데 비선형 배열을 가능케한 DP가 문제 해결의 돌파구를 마련해 주었던 것이다. 흘려쓴 온라인 필기 인식에도 음성 인식문제와 유사하게 자소 분류, 비선형 배열, 그리고 자소간의 경계 탐색이라는 세 문제가 상호 간섭하는 것을 발견할 수 있다. 이점에 대한 DP 해결책을 얻기 위해서는 각 문제를 동시에 만족시키기 위한 최적화 기준(global criterion) 정의가 선행되어야 한다. FSN 모델 상에서 현재 가장 널리 적용되는 것은 "가장 좋은" 완전한 상태(노드) 열, 즉 경로를 찾는

것이다. 확률 모델의 틀 안에서 최적성은 곧 최고의 확률을 계산하는 경로를 의미한다. 그리고 다행히 이와 같은 최적 경로를 찾는, 형식화된 기법이 존재하는데, DP의 하나인 Viterbi 알고리즘이 그것이다 [Vite67].

2.3 인식 알고리즘

본 소절에서는 그림 1와 같은 두 계층 구조를 가진 네트워크 — 자소 HMM의 네트워크 — 탐색에 적용한 구체적인 Viterbi 알고리즘을 기술키로 한다. 여기서 설명될 알고리즘은 입력 코드에 동기 방식으로 진행되는 계산 구조로 되어있고, 입력 코드열의 길이에 정비례하는 계산 복잡도를 갖기 때문에 실시간 응용에 적당하다. 기본적으로 하나의 HMM에 적용한 Viterbi 알고리즘과 동일하다[Rabi89].

식 (3)과 (4)에 따라서 최적 인식기는

$$Pr(\hat{W}, X) = \max_{W} Pr(X|W)Pr(W). \tag{5}$$

에 따르게 된다. K를 임의 경로 W의 길이 또는 경로상의 모델 수라고 하면 $W = W_1 W_2 \cdots W_K$로 표현할 수 있다. 네트워크에서 W에 대한 $Pr(X|W)$는 입력-경로간 비선형 배열 계산을 수반하게 된다. W와 X에 대한 비선형 배열의 결과 입력 시퀀스의 어떤 분할(partition) $X = X(\tau_1)X(\tau_2)\cdots X(\tau_K)$,

$$X(\tau_k) = x_{t_{k-1}+1} \cdots x_{t_k}, \tag{6}$$

$$1 \le t_1 \le t_2 \le \cdots \le t_K = T \tag{7}$$

을 얻었다고 하면 $X(\tau_k)$ 각각은 모델 W_k 모델에 배열된 부분 코드열이 된다. 이와 같은 분할(또는 segmentation)을 $\tau = (\tau_1, \tau_2, \ldots, \tau_K)$ 라고 표시하자. 그러면 (5)식은

$$Pr(\hat{W}, X) = \max_{W,\tau} Pr(X(\tau_1)\cdots X(\tau_K)|W_1 \cdots W_K) \quad \times Pr(W_1 \cdots W_K) \tag{8}$$

이 된다. 각 연결획 모델은 앞뒤 두 자소간의 (초-중성 또는 중-종성간) 종속 관계를 충분히 모델링한다고 가정하자. 그러면 네트워크 각 경로의 일련의 모델 간에는 독립이 되며, 이에 따라

윗식을 다음과 같이 쓸 수 있다.

$$Pr(\hat{W}, X) = \max_{W,\tau} \prod_{k=1}^{K} [Pr(X(\tau_k)|W_k)Pr(W_k)] \tag{9}$$

이식은 글자를 인식하는 문제를 각 자소 및 연결획별 인식의 소문제로 분리할 수 있음을 보인 것이다. $X(\tau_k)$ 가 주어 졌을 때 자소 모델 내에서의 코드열-경로는 Viterbi 알고리즘으로 간단히 실현할 수 있다. 자소 HMM을 N-state left-to-right 모델이라고 하면, Viterbi 알고리즘의 계산은

$$\delta_t(j) = \max_{1 \le i \le j} \{\delta_{t-1}(i) a_{ij} b_{ij}(x_t)\} \tag{10}$$

에 따라 반복 계산으로 이루어 진다. 여기서 $\delta_t(j)$는 코드열 $x_1, x_2, \cdots, x_t$와 현재 모델의 상태 j에 도달한 경로와의 비선형 배열 확률을 나타낸다. 그리고 경로에 대한 정보는 $\psi_t^m(j)$에 저장된다. 이것은 경로상의 이전 상태를 가리키는 포인터를 가지고 있으며, 나중에 경로 역추적시 이용되는 정보이다. 이와 같은 반복 계산을그림 1와같은 계층 구조 네트워크에 적용한 디코딩 알고리즘은 그림 2에 제시하였다.

3 성능 평가 및 개선

3.1 모델 훈련

초, 중, 종성 자소 모델과 연결획 모델을 훈련시키기 위해서는 많은 데이타를 필요로 한다. 그것은 각 HMM에 추정해야할 파라미터가 많기 때문이다. 예를 들어 (이산) 은닉 마르코프 모델의 상태수가 10 이면 600 ~ 800개의 파라미터를 갖게 된다. 많은 파라미터를 신뢰도 높은 추정을 해내기 위해서는 충분한 양의 데이타를 모으는 것이 급선무다. 훈련 데이타가 불충분할 때에는 훈련 데이타에 편향되어 충분한 모델링 능력을 기대하기 어렵기 때문이다.

연결획 모델은 글자 안에서만 나타나기 때문에 훈련을 위한 글자에서 훈련 샘플을 모아야 한다. 이를 위해서는 흘려쓴 샘플에 대해서는 물론이고 또박또박 정서한 필기에 대해서도 훈련 샘플의 자소의 경계를 가르쳐 줄 필요가 있다. 본 인식 시스템에서는 사람의 가르침(manual segmentation)에 따라 각 자소, 연결획 부분을 모아서 HMM를 훈련시킨다. 훈련 과정은 아래와 같다. 각 훈련 샘플이 무슨 글자인지를 알 때:

(1) 봉넷에서 그 글자에 해당하는 경로를 찾고 그 경로에서 일련의 자소 및 연결획 모델 이름 C-L_{cj}-J-L_{jz}-Z 또는 C-L_{cj}-J를 결정한다. L_{cj} 및 L_{jz}는 봉넷에서의 초성-중성 및 중

For t = 1, ..., T, do
  For each arc (gl->gr), as of Figure 1, do
    For each model m labelling the arc, do
      Perform Viterbi Algorithm as:
        For each state $j = 1, \ldots, N_m$ do
        $\delta_t^m(j) = \max_{1 \le i \le j}\{\delta_{t-1}^m(i) a_{ij}^m] b_{ij}^m(x_t)\}$
        $\psi_t^m(j) = \arg\max_{1 \le i \le j}\{\delta_{t-1}^m(i) a_{ij}^m b_{ij}^m(x_t)\}$
      End
      $\Delta_t(gr) = \max_{m(gl->gr)} \delta_t^m(N_m)$
      $\Psi_t(gr) = \arg\max_{m(gl->gr)} \delta_t^m(N_m)$
  End
End
Perform BackTracking as:
$gr^* =$ the final node in the network
Do until gr^* is the start node
  output the label of the model pointed by $\Psi_t(gr^*)$
  $gr^* = gl$ from $\Psi_t(gr^*)$
End

그림 2: 인식 알고리즘

성-종성 사이의 연결획을 말한다.

(2) 글씨에서 연결획 모델을 포함한 자소 모델의 경계를 지정한다. 이 과정이 많은 양의 수작업이 들어가는 단계이다. 그리고 그 경계의 통계적 분포가 결국 나중에 인식 결과로 얻을 수 있는 자소 경계를 결정하게 된다.

(3) 시간적 순서에 따른 각 분할(segment)을 경로상의 모델에 대응하는 훈련 샘플 화일에 저장한다.

(4) 각 자소 및 연결획 모델에 대해서, HMM 훈련에 널리 쓰이는 Baum-Welch 알고리즘 [Rabi89]을 수행한다.

위 (2) 단계에서의 원칙은 각 자소 부분이 해당 자소의 기본 형태(base form) [Ward88]과 가장 가까운 모양을 갖도록 한다는 것이다. 이것은 각 자소 모델이 그 자소를 포함하는 모든 글자에서의 해당 자소끌을 모델링하기 위해서는 그 변이 또는 편차를 최소화한다는 의미에서이다. 만약 모양의 변화가 크다면 결국 그 모델의 모델링 능력이 이에 비례하여 줄어들게될 것이다.

3.2 자소 인식

각 자소의 분별력이 뛰어나다면 글자 인식률이 높을 것이라는 것은 쉽게 예측할 수 있다. 그러므로 필기 모델 및 인식기 성능 평가에 앞서, 그 하부 구조인 자소 및 연결획 모델의 질을 평가할 필요가 발생하였다. 자소 모델의 모델링 지표로서 분별력(discrimination)을 테스트하는 방법으로 초성, 중성, 종성별 인식률을 측정하였다 (그림 3.2 참조). 이것은 자소 종류(초설,중

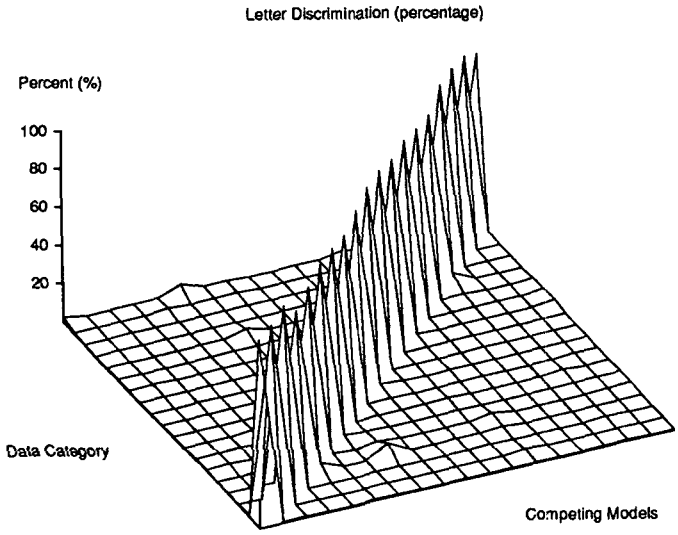

그림 3: 초성 모델의 인식률; 훈련 데이타를 다시 테스트한 결과; 모델 및 데이타 축은 축은 초성의 순서에 따른 것임.

설,종성)별로 상호 분별력을 시험한 것인데, 최우 추정법(maximum likelihood estimation)의 취약점을 보는데도 그 의의를 찾을 수 있다. 자소별로 뛰어난 모델링은 보다 나은 글자 인식 결과를 가져옴에 틀림없으나, 그것이 극적인 인식률 향상의 효과를 가져오지 못하는 것은 각 자소 모델을 독립적으로 파라미터 추정했기 때문이다 [Brow87] [Ephr89].

그림1의 네트워크 구조에 따르면 자소간의 인식률은 그 자소가 경쟁하는 자소 집합 내에서 그 의미를 갖긴 하지만 그 자체가 글자 인식률에 직접 반영되는 것은 아니다. Viterbi 알고리즘에서 최적화 기준이 각 아크 (그림1에서는 자소 및 연결획 모델)별 최적화가 아니라 시작 노드로부터의 완전한 경로의 최적화이기 때문이다. 하지만 상태 천이에 따라 네트워크상의 각 노드에서 만나는 경로의 각 가능성(hypotheses)은 현재 노드로 천이하는 마지막 아크사이의 경쟁이 큰 역할을 담당한다고 가정한다면 — 실제 글자 인식의 결과 후보 글자가 비슷한 모양의

글자가 나타난다 — 어느 정도 그 의의를찾을 수 있을 것이다.

3.3 언어 모델

글자 인식은 주어진 입력 X에 대하여

$$Pr(\hat{W}|X) = \max_{W} \frac{Pr(X|W)Pr(W)}{Pr(X)}$$

을 만족하는 $\hat{W}$를 구하는 과정이다. $Pr(X)$는 상수 이므로 무시하면

$$Pr(X|W)Pr(W)$$

를 최대로 하는 $\hat{W}$를 구하면 된다. 여기서 $Pr(W)$는 언어 모델에서 얻을 수 있는 사전(a $priori$) 확률이다. 실험에 사용된 한글 언어 모델은 bigram 글자 언어 모델을 채택하였는데, 일차 마르코프 가정에 따라서 글자 $W = cvj$ (c = 초성, v = 중성, z = 종성)의 확률 $Pr(cvz)$을 $Pr(c)Pr(v|c)Pr(z|v)$으로 추정한다. 전이 확률 $Pr(v|c)$ 와 $Pr(z|v)$는 각각 초-중성, 중-종성 간의 연결획의 사전 확률인 동시에 자소간 전이 확률이다.

3.4 글자 인식

모델 훈련에 이용된 데이타는 약간 수정된 단편 소설 40 명분의 필기를 위주로 하였고 일부 희귀 자소 및 연결획에 대하여는 약간 보충을 하였다. 한편 인식 실험에 사용된 데이타(K)는 8명이 쓴 국민교육헌장 필기를 사용하였다. 모든 데이타는 WACOM SD-520C 디지타이저로 수집한 것이다. 훈련 데이타는 고교생이 쓴 것이고, 테스트용 데이타는 대학 및 대학원생으로부터 수집한 것 중의 일부이다.

[베이스라인 모델]

베이스라인 모델로 8 방향 코딩 모델 (Λ_8)과 16 방향 코딩 모델 (Λ_{16})의 두가지를 훈련하였다. 이것은 파라미터가 많은 16 방향 모델의 모델링 능력과 훈련 데이타 부족 문제를 동시에 테스트 해보고자는 것이다. 표 1는 훈련 데이타와 전혀 상관이 없는 Data set K와, 쓴 글씨의 내용, 즉 텍스트가 같다는 것을 제외하면 역시 훈련 데이타와 전혀 상관이 없는 Data set H에 대한 실험을 요약한 것이다. H 데이타에 대해서는 Λ_{16}이 더 나은 성능을 보여준다. 그러나 K 데이타의 경우에는 Λ_8 의 인식률이 일정한 반면 Λ_{16} 의 인식률은 Λ_8 이하로 떨어진다. 이 실험은 필기자 독립 실험이므로 결과를 훈련 샘플의 부족으로 돌릴 수 있다. 즉 8 방향 코드 모델

에 대해서는 훈련 데이타 양이 대체로 충분했다고 볼 수 있으나, 16 방향 코드 모델은 텍스트, 곧 테스트 글자 set의 변화에 민감하므로 훈련 데이타 양이 아직 충분한 정도는 아니라고 설명할 수 있을 것이다.

표 1:8 방향 코딩 모델과 16 방향 코딩 모델;한개/다섯개 후보에 대한 인식률 (%).

모 델	Data set H	Data set K
Λ_8	84.24 / 94.56	84.99 / 94.84
Λ_{16}	87.82 / 94.67	83.27 / 92.94

[출력 분포 평활화]

윗 실험 결과에 따라 데이타 부족 여부를 확인하는 방법은 훈련 샘플을 많이 보충해서 모델을 훈련시켜 보는 것이다. 그러나 많은 훈련 데이타를 얻는다는 것은 결코 값싼 작업이 아니다. 그대신 *forward-backward* 알고리즘에 의한 훈련이 끝난후 모든 HMM의 출력 확률 분포 파라미터를 perturbation smoothing을 적용한다. 이것은 단순한 blind smoothing 으로서, HMM의 각 출력 분포를 흔든다는 개념에 의한 것인데, 임의 심볼 x의 확률을 이웃한 심볼의 확률값과의 weighted sum

$$\hat{b}_{ij}(x) = \sum_{y \in NGB(x) \bigcup \{x\}} \lambda_{|x-y|} b_{ij}(y)$$

으로 대치하는 방식이다. 여기서 $NGB(x)$는 적절한 distance measure에 의한 x 의 이웃 심볼의 집합을 나타내며, $\lambda_{|x-y|}$ 는

$$\sum_{y \in NGB(x) \bigcup \{x\}} \lambda_{|x-y|} = 1$$

를 만족하는 가중치이다. 체인 코드라면 체인 코드값의 차이 또는 해당 각도의 차를 distance measure로 쓸 수 있을 것이다. 그리고 이 때의 이웃 $NGB(x)$는 시계 방향 및 반시계 방향으로 일정한 거리(또는 각도) 이내의 코드로 정의한다. 실험에는 $NGB(x)$의 크기가 2, 즉 양쪽으로 이웃한 코드 하나씩, 두개의 코드를 포함케 하여, 가중치를 변화 시켜가면서 평가를 하였다. 가중치 비가 1:n:1 이라면 코드 x의 확률 값을 $n+2$ 등분하여 x의 두 이웃 코드에 $\frac{1}{n+2}$ 씩 나누어 주고 자기는 $\frac{n}{n+2}$을 갖는다는 뜻이다. 그림 3.4는 1:1:1의 가중치로 perturbation smoothing한 결과 심볼 출력 확률 분포의 예를 보여준다.

베이스라인 모델의 확률 분포를 평활화가 모델의 성능에 미친 효과는 그림 3.4에 나타내었

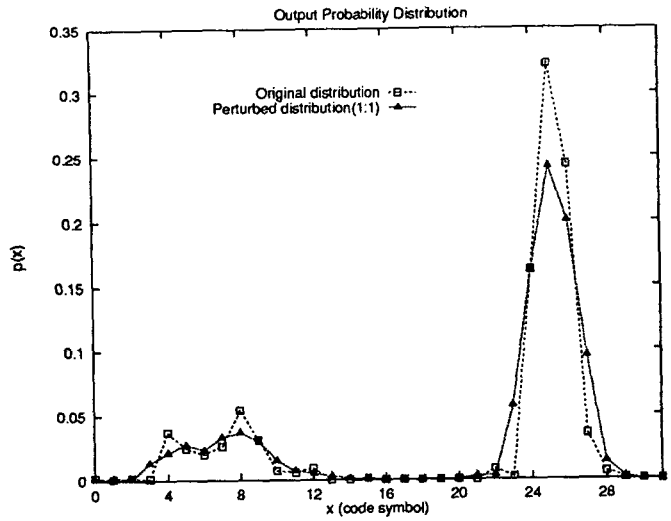

그림 4: HMM 출력 분포를 1: n (=1) : 1 의 비율로 perturbation smoothing 한 후의 확률 분포 예.

다. 가장 오른쪽의 값이 평활화 안된 베이스라인 모델이고, 왼쪽으로 갈 수록 perturbation 정도가 커진다. 8 방향, 16 방향 모델 전체에 적용한 경우와 연결획 모델에만 적용한 — 연결획 모델별 훈련 데이타 수의 편차가 매우 심하고 또 연결획의 역할을 간접적으로 평가하기 위한 것 — 두가지 경우를 볼 수 있는데, 두 경우 모두 perturbation을 많이할 수록 성능이 향상되었다. 이점은 데이타 부족을 확인해 주었다고 할 수 있다. 그리고 8방향 모델의 경우 완만한 증가를 보이다가 perturbation 정도가 1:5:1 되는 지점을 분수령으로 해서 급락하는 효과를 보이는 반면, 16 방향에서는 일관되게 거의 끝까지 계속 인식률이 증가한다. 이점 또한 16 방향 모델에 대해서는 데이타가 부족하여 훈련 샘플에 지나치게 편향 훈련된 것으로 설명할 수 있다.

한편, 테스트 data set K의 텍스트를 다른 필기자가 쓴 샘플을 추가하여 훈련한 결과, 모델의 인식률은 8 방향 모델과 16 방향 모델 각각이 88.55%, 87.43%로 향상되었다. 여기에 다시 perturbation smoothing을 적용한 결과를 그림 3.4에 보였는데, 여전히 비슷한 추세로 인식률의 상승을 보여준다. 추가한 샘플의 양은 베이스라인 모델 훈련 데이타 양의 1/5 정도이다. 위 결과와 비교해 볼 때 단순하지만 perturbation smoothing의 효과는 입증되었다고 하겠다.

그리고 마지막으로, 평활화된 모델의 출력 분포를 다시 한번 perturbation smoothing 하는 것은 이웃, $NGB(x)$의 크기를 늘이는 효과를 가져온다. 이렇게 중복 평활화된 모델의 성능 분포는 그림 3.4과 같다. 이중 perturbation으로 더이상 현저한 성능 개선이 나타나지는 않는

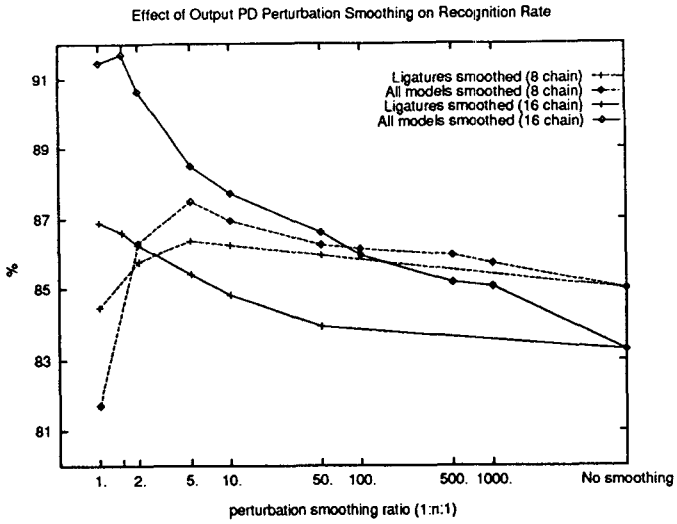

그림 5: 모델 훈련 후의 perturbation smoothing 정도에 따른 인식률; 8 방향, 16 방향 각 모델을 두가지 — 모델 전부 또는 연결획 모델만 - 경우로 구분하여 평활화한 것.

다. 그리고 계속 더 평활화를 하면 오히려 성능이 떨어지게된다. 위 결과를 종합해 볼 때 엄밀히 말해서 perturbation smoothing의 효과가 실제로 훈련 샘플의 부족에 기인한 것인지, 아니면 평활화의 결과 (최적화 기준의 문제점 때문에) HMM 간에 존재하던 모델링 능력, 분별력의 차이를 완화 시킨 덕택인지에 대한 의문은 그대로 남아 있다고 할 수 있다.

3.5 자소 분할

Viterbi 알고리즘에 의한 네트워크 탐색은 주어진 입력에 대해서 최적 경로를 얻을 수있다. 역추적 단계에서 최종 노드에서 시작하여 거꾸로 시작 노드에 이르기까지 경로를 거슬러 간다. 공간적으로는 네트워크의 아크를 거슬러 이동하는 것이고, 시간적으로는 체인 코드열의 길이 만큼의 과거로 거슬러 올라 가는 것이다. 만약 최종 노드가 하나 이상 있으면 그중 확률이 제일 높은 노드가 주어진 입력에 대한 최종 노드로 된다. 역추적 경로를 따라 가다가 공간적으로 자소 모델의 경계를 만나면 그 시점에 대응되는 입력 코드열에서의 위치가 곧 자소의 경계가 된다. 입력 시퀀스에 사상된 이러한 점들의 집합은 최적 원리 [Bell57] 정의에 따라 최적 자소 분할을 구성한다. 이렇게 얻어진 자소 경계의 예는 그림 3.5과 같다. 예의 그림중 일부는 또박또박 쓴 필기와 마찬가지로 자소간 경계가 뚜렷해서 기존의 방법으로도 간단히 자소 분할이 가능하다. 그러나 흘려쓴 일부 글씨에서는 전례가 없이 뛰어난 분할 결과를 볼 수 있다.

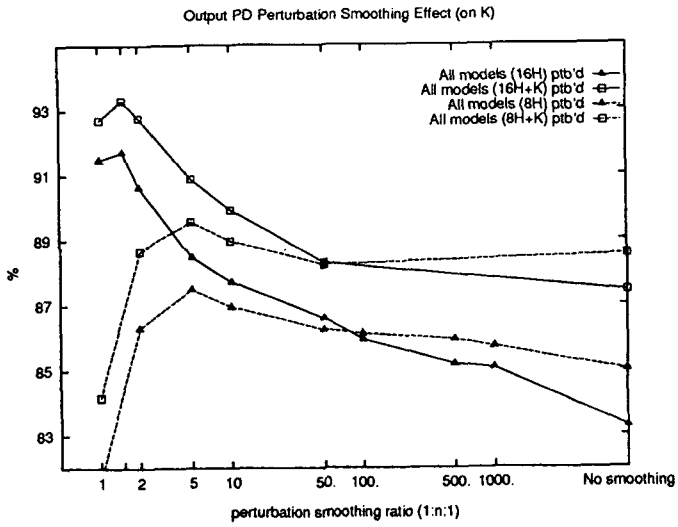

그림 6: 훈련 샘플 보충과 perturbation smoothing 효과.

4 결 론

은닉 마르코프 모델이라는 모델링 도구를 사용한 온라인 한글 필기 모델을 중심으로 통계적 인식 모델에 담긴 기본 이론과 계산 방법에 대하여 소개를 하였다. 단한 구조의 모델과 체인 코드 만으로 테스트한 결과 예상 밖의 결과를 얻었다. 흘려쓴 필기 모델링의 핵심이라 할 수 있는 연결획 모델을 도입한 효과는 더욱 그러하였다. 인식과 동시에 얻게되는 자소 분할을 분석해본 결과에서 이러한 확신을 얻었다.

HMM의 역할이 돋보였다고 하지만 HMM 자체의 기본 가정에 따라 피할 수 없는 약점이 있는 것도 사실이다. 또 모델 최적화 과정 방법 자체는 인식 문제를 본 것이 아니라 독립된, 단순히 하나의 모델에 대한 최우 파라미터를 추정할 뿐이다. 그래서 일반적으로 overgeneralization, 훈련 샘플이 과부족하면 overspecialization을 초래할 소지가 많다.

참고 문헌

[성91] 성태진, 방승양, "문자 조합 규칙 학습에 의한 한글 온라인 필기 인식기의 설계," 한국정보과학회 추계 학술 발표 논문집, pp. 223-226, 1991. 10.

[신92] 신봉기, "통계적 방법에 의한 온라인 한글 필기 인식," 제4회 한글 및 한국에 정보처리 학술발표 논문집, pp. 533-542, 1992. 10.

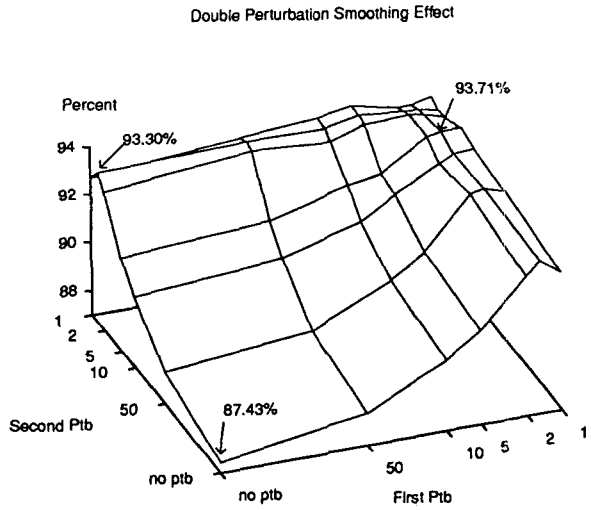

그림 7: 중복 평활화의 효과; 훈련 샘플 보충한 모델의 평활화 예.

[이93] 이성환, 박희선, "한글 인식의 사례 연구: 최근 5년 동안의 연구 결과를 중심으로," 제1회 문자 인식 워크샵 발표 논문집, pp3-46, 1993. 5.

[Duda73] R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis," New York, John Wiley & Sons, 1973.

[Vite67] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically optimum decoding algorithm," *IEEE Trans. Information Theory*, v.IT-13, pp.260-269, 1967.

[Baker75] J. K. Baker, "The dragon system — An overview," *IEEE Trans. Acoust., Speech, Signal Processing*, v.ASSP-23, pp.24-29, Feb. 1975.

[Baum70] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization techinique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Stat.*, v.41, n.1, pp.164-171, 1970.

[Rabi89] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Apllications in Speech Recognition," *Proc. of the IEEE*, v.77 ,n.2, pp.257-286, 1989.

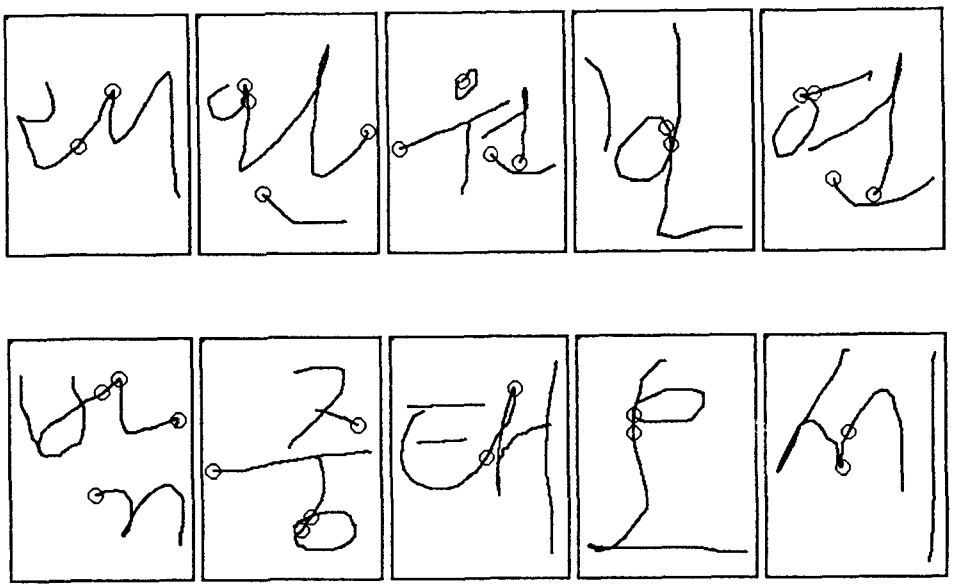

그림 8: 인식 결과 자소 분할의 예.

[Ward88] J. R. Ward and T. Kuklinski, "A Model for Variability Effects in Handprinting with Implications for the Design of Handwriting Character Recognition Systems," *IEEE Trans. Systems, Man, Cybernetics*, v.18, n.3, May/June 1988.

[Tapp90] C. C.Tappert, C. Y.Suen and T. Wakahara, "The State of the Art in On-Line Handwriting Recognition," *IEEE Trans. on PAMI*, v.12, n.8, pp.787–808, Aug. 1990.

[Brow87] P. Brown, "The Acoustic Modeling Problem in Automatic Speech Recognition," PhD thesis, Comp. Sci. Dep., Carnegie Mellon Univ., May 1987.

[Ephr89] Y. Ephraim, A.Dembo, and L.R. Rabiner, "A minimum discrimination information approach for hidden Markov modeling," *IEEE Trans. Inform. Theory*, v.IT-35, n.5, pp.1001-1013, Sept. 1989.

[Lee89] K. F. Lee and H. W. Hon, "Speaker-Independent Phone Recognition Using Hidden Markov Models," *IEEE Trans. on ASSP*, v.37, n.11, pp.1641–1648, Nov. 1989.

[Lee90a] K. F. Lee, "An overview of the SPHINX Speech Recognition System, *IEEE Trans. on ASSP*, v.38, n.1, pp.35–45, Jan. 1990.

[Lee90b] K. F. Lee, "Context-Dependent Phonetic HMMs for Speaker-Indenpendent Continuous Speech Recognition," *IEEE Trans. on ASSP*, v.38, n.4, pp.599–609, Apr. 1990.

[Giac92] E. P. Giachin, C. H. Lee, L. R. Rabiner, A. E. Rosenberg, R. H. Pieraccini, "Ont the use of inter-word context-dependent units for word juncture modeling," *Comp. Speech and Lang.*, v.6, pp.197–213, 1992.

[Pavl77] T. Pavlidis, "Structural Pattern Recognition," New York, Springer-Verlag, 1977.

[Busg87] M. A. Bush and G. E. Kopec, "Network-based connected digit recognition," *IEEE Trans. on ASSP*, v.35, pp.1401–1413, Oct. 1987.

[Bell90] J. R. Bellegarda and D. Nahamoo, "Tied Mixture Continuous Parameter Modeling for Speech Recognition," *IEEE Trans. on ASSP*, v.38, n.12, Dec. 1990.

[Vint71] T. K. Vintsyuk, "Element-wise recognition of Continuous Speech Composed of Words from a Specified Dictionary," *Kibernetika*, v.7, pp.133–143, Mar.–Apr. 1971.

[Sako71] H. Sakoe and S. Chiba, "A dynamic programming approach to continuous speech recognition," in *Proc. 7th Int. Congress Acoust.*, Budapest, Hungary, 1971.

[Sako79] H. Sakoe, "Two-level DP-matching - A dynamic programming-based pattern matching algorithm for connected word recognition," *IEEE Trans. on ASSP*, v.27, pp.588–595, Dec. 1979.

[Myer81] C. S. Myers and L. R. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," *IEEE Trans. on ASSP*, v.29, n.2, pp.284–297, Apr. 1981.

[Ney84] H. Ney, "The use of a one-stage dynamic programming algorithm for connected word recognition," *IEEE Trans. on ASSP*, v.32, n.2, pp.263–271, Apr. 1984.

[Bell57] R. Bellman, *Dynamic Programming*, Princeton, NJ: Princeton UniversityPress, 1957.