State의 분리에 의한 HMM 자동 구성 방법

임 헌규*○ 김진형**
*한국전자통신연구소
**한국과학기술원

Automatic HMM Configuration by State Splitting

Hun Gyoo Lim* Jin Hyung Kim**
*ETRI **KAIST

요 약

본 논문에서는 임의의 훈련 데이타 집합으로 부터 이에 적절한 하나의 HMM(Hidden Markov Model)을 생성하는 절차를 제시하고자 한다. 지금까지는 HMM의 State 수와 이들의 연결 관계를 훈련 데이타를 분석하여 프로그래머가 직접 지정해 주고 관련 파라메타 값들을 잘 알려진 알고리즘에 의해 구하는 절차를 거쳤다. 그러나 프로그래머가 적절한 모델 구성을 찾기가 쉽지않으며, 정해진 모델이 해당 훈련 데이타를 적절히 표현한다고 할 수 없다. 본 연구에서는 한 Class의 훈련 데이타로 부터 연속적인 State의 분리를 통하여 적절한 HMM을 구성하는 방법을 제시하고자 한다.

1. 모델 구성의 문제점

온라인 한글 인식 시스템을 Hidden-Markov Model을 이용하여 개발하고자 할 때 우선적으로 진행되어야 하는 과정은 각 자소에 대한 훈련 데이타를 소집하고, 이들 각 자소에 대한 Hidden Markov 모델을 생성하는 일이다. 현재까지는 HMM을 구성하는 State의 수 및 이들 사이의 연결을 프로그래머가 직접 훈련 데이타를 분석하여 결정하였다.

그러나 이러한 방법에는 두가지의 문제점이 존재한다. 첫번째로 샘플 데이타에 대한 모델을 몇개로 나눌 것인가에 대한 결정에 따라 샘플 데이타가 몇개의 그룹으로 분리되어 각 그룹별로 HMM 구성이 이루어져야 한다. 그러나 이러한 분리는 훈련 데이타의 수집때에 분류된 그룹별로 데이타의 수집이 있어야 하며, 많은 시행 착오를 거쳐야 한다.
두번째 문제점은 분류된 각 그룹에 대한 훈련 데이타를 해당 모델이 적절하게 표현할 수 있는가에 대한 것이다. 전체적으로는 기본적인 모델의 Architecture가 정해질 수 있으나 각 데이타 그룹마다의 간단한 Sequence 상의 차이를 프로그래머가 쉽게 찾을 수 없다. 각 훈련 데이타 그룹마다 Architecture를 지정해 주기가 쉽지 않으며, 따라서 보통은 같은 Architecture를 이용하여 훈련 데이타 그룹에 대한 각 모델의 파라메타 값을 훈련시키게 된다.

본 논문에서는 이러한 과정을 Automatic하게 수행할 수 있는 한가지 방법을 제안하고자 한다. 즉 초기에 최소한의 State와 연결에 의해 구성된 모델의 파라메타 값을 훈련 데이타에 의해 구하고, 이 모델에 대한 파라메타값들을 분석하고,훈련 데이타에 맞도록 State를 분리하여 모델 Architecture를 재구성하는 과정을 반복하여 하나의 HMM Architecture를 찾는 절차에 대해 기술하기로 한다.

2. 인식 Score의 변화

한글 자소의 HMM 모델에서 각 모델들의 일반적인 연결 형태인 Left-to-Right 연결의 경우만 고려하였을 때, 한 자소 모델에 대한 인식 Score의 State 수에 따른 변화와 파라메타 분포 변화의 예를 그림 1에 나타내었다. State 0 (초기 State)와 Final State에서의 Self-Transition은 제한하였다. State의 수가 3개인 경우(즉 Self Transition이 한 State에서만 존재)에 자소 '기역'에 대한 출력 심볼의 확률 분포가 크게 두개의 그룹으로 구분된다. 각 그룹의 상대적인 심볼 생성 시기는 평균 길이가 7인 상태에서 각각 2.58과 4.54를 나타내고 있다. 이것은 두 그룹의 출력 심볼들 사이의 순서 관계가 있음을 나타내고 있는 것이다.

State수를 하나 증가 시켜본 결과 두개의 그룹은 두개의 State로 분산되어 각 State에서 하나의 그룹만을 형성하게 되었다. 그러나 훈련데이타가 두개 이상의 서로다른 Code Sequence를 갖는 샘플 데이타 집합으로 구성된 경우에는 State 수를 계속 증가시켜도 심볼 그룹이 분산되지 않는다. 즉 이러한 경우에는 두 심볼 그룹사이에 Parallel한 관계가 존재함을 나타내는 것이라 할 수 있다.

또한 그림 1에서 보여지는 훈련 데이타에 대한 Score는 State 수가 증가함에 따라 계속적인 증가를 보인 반면, 훈련에 사용되지 않은 데이타에 대한 Score는 State수가 5인 경우는 오히려 감소하는 경향을 보이고 있다. 이러한 현상은 기타 자소 모델에서도 동일하게 나타나고 있다.

3. 훈련데이타의 모델 구성

본절에서는 주어진 데이타 Set에 대한 모델을 찾아내기 위한 State의 분리 기준 및 방법, 전체적인 절차에 대해 기술하기로 한다.

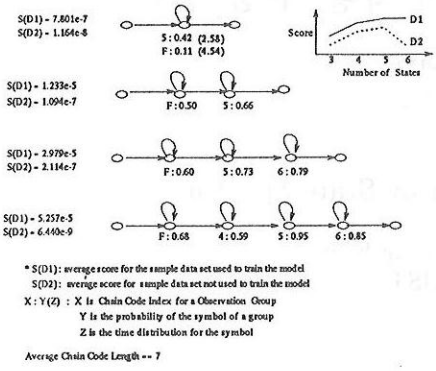

그림 1: State수와 파라메타 분포의 변화-I

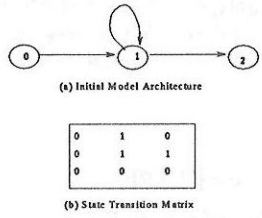

그림 2: HMM 생성을 위한 초기 Architecture

3.1 초기 모델

초기 모델은 그림 2에서와 같이 3개의 State로 구성되어 있으며, 초기 State와 Final State에서는 Self-Transition이 없는, Left-to-Right 형태의 Architecture를 갖는다. 초기 State에서 Self-Transition을 없도록 한 것은 첫번째 State에서의 State 분리를 방지하기 위한 것이다.
정해진 한 모델의 파라메타 값은 주어진 훈련 데이타를 이용해 구한 후 이 파라메타 값의 분석을 통하여 모델의 재구성 방향을 결정하게 된다.

3.2 출력 심볼의 Duration

임의의 State i에서의 각 출력 심볼들에 대한 평균 Duration 값은 샘플 데이타의 갯수를 K, 한 샘플 데이타 k의 Input Code중 State i에서 Self-Transition시에 심볼 o를 출력하는 Duration을 d_{kio}, 샘플 k의 Code 길이를 l_k, State i를 거치며, 심볼 o를 출력하는 샘플 데이타의 수를 S_{io}, 훈련 데이타의 평균 코드 길이를 L이라 할 때, State i에서의 각 심볼 o에 대한 평균 Duration D_{io}는

$$D_{io} = \frac{\sum_{k=1}^{K} d_{kio}\frac{L}{l_k}}{S_{io}}$$

이 된다. D_{io}의 값은 심볼 o에 대한 State i에서의 평균 Self-Transition 횟수를 나타내므로, 만일 한 State의 Self-Transition에 출력 심볼 그룹이 하나 존재하면 대표 출력 심볼에 대한 시간 분포 값을 중심으로 출력 심볼이 해당 State에서 Consume 될수 있음을 뜻한다.

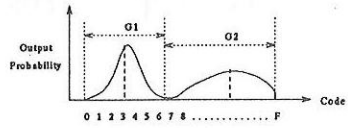

그림 3: Transition의 출력 심볼 그룹

3.3 출력 심볼의 시간 분포

또 하나의 중요한 정보인 각 State에서의 출력 심볼 각각에 대한 시간 분포 값은 해당 State에서 Self Transition이 이루어질때 임의의 출력 심볼, o가 생성되는 시기를 의미한다. 임의의 State i에서 Self-Transition시에 심볼 o를 출력하게되는 평균 코드길이에 대한 상대적인 시점, T_{io}는, 아래의 식과 같이 표현될 수 있다.

$$T_{io} = \frac{\sum_{k=1}^{K}\sum_{t=0}^{l_k}((\frac{I_k(i,o,t)}{l_k})L)}{F_{io}}$$

- K : 훈련 데이타의 샘플 수
- l_k : k번째 샘플 Data의 Code 길이
- L : 모든 샘플 데이타에 대한 평균 Code 길이
- $I(i,o,t)$: 임의의 샘플 데이타의 t 번째 Input 심볼 o가 State i에서 Self-Transition하며 Consume될때는 t, 그렇지 않으면 0이 되는함수
- F_{io} : 훈련데이타 전체에 대해 State i에서의 Self-Transition시에 심볼 o를 출력한 총횟수

즉 T_{io}는 State i에서의 Self-Transition시에 심볼 o가 Consume되는 상대적인 위치(전체 샘플 코드에서의)를 나타낸다.

3.4 출력 심볼의 그룹

한 State에서의 Self-Transition에 대한 출력 심볼들의 확률 분포에 따라 출력 심볼들은 1개 이상의 그룹을 형성하게 된다. 출력 심볼들에 대한 확률 분포가 그림 3과 같은 경우 출력 심볼의 그룹은 극점에 의해 두개의 그룹으로 분리될 수 있다.

각 그룹은 해당 State에서의 Self-Transition시에 관찰될 수 있는 심볼의 분포를 나타내며, 각 그룹의 대표 심볼이 consume될 수 있음을 나타낸다. 임의의 심볼 그룹 G의 대표 심볼 O_G에 대한 Code Index I_{O_G}는

$$I_{O_G} = \sum_{o_G}^{All\ of\ o_G} \frac{P(o_G) * I_{o_G}}{\sum_{o_G} P(o_G)}$$

- o_G : 그룹 G에 속한 임의의 심볼
- $P(o_G)$: 출력 심볼 o_G에 대한 Probability
- I_{o_G}: 심볼 o_G에 대한 Code Index

에 의해 Indexing되는 심볼이 된다. 즉 임의의 그룹 G를 대표하는 심볼 O_G는 해당 State에서의 예상 심볼이라 할 수 있다. 한 심볼 그룹에 대한 입력 Code Sequence 상에서의 상대적인 위치는 위에서 구해진 그룹을 대표하는 심볼 O_G의 상대적인 위치 값이라 할 수 있다. 즉 한 심볼 그룹 G에 대한 시간상의 상대적인 위치 T_{iG}는

$$T_{iG} = T_{iO_G}$$

가되며, 이 그룹을 대표하는 시간 분포라 할수 있다. 이 그룹에 대한 State i에서의 Duration은 앞에서 구한 D_{io}로부터 구해진다. 즉 임의의 State에서의 한 그룹 G에 대한 Duration은,

$$D_{iG} = \sum_{o_G}^{All\ of\ o_G} D_{io_G}$$

이된다.

3.5 심볼 그룹들 사이의 관계

임의의 Transition에 포함된 심볼 그룹 사이에는 Sequencial한 관계(한그룹 다음에 또다른 그룹의 심볼이 뒤따르는) 혹은 Parallel한 관계(두 그룹중의 한 그룹의 Symbol Sequence가 선택되는 경우)로 나눌수 있다.

임의의 두 심볼 그룹 G_1과 G_2 사이의 관계 구분은 각 그룹을 대표하는 심볼의 시간 분포와 Duration에 의해 결정된다. 임의의 두 그룹 G_1과 G_2에 대해, 각각의 대표 심볼에 대한 시간 분포를 T_{G_1}과 T_{G_2}라 하고, 평균 Duration을 각각 D_{G_1}와 D_{G_2}라 하면, 두 그룹의 심볼 출력 시기(데이타의 평균 길이에 대한 상대적인 위치)의 차이 $\Delta T = |T_{G_1} - T_{G_2}|$의 값과, 두 그룹의 Duration Boundary의 합인

$$B_{G_1G_2} = (\frac{D_{G_1}}{2} + \frac{D_{G_2}}{2}) \times r$$

에 따라 두 그룹의 관계가 결정된다. 위의 식에서 r의 값은 백분율을 나타내는 파라메타로 각 그룹의 Boundary를 결정하는 변수이다. 즉 두 그룹 G_1과 G_2 사이의 관계는

- Sequencial 관계 : $\Delta T > B_{G_1G_2}$
- Parallel 관계 : $\Delta T < B_{G_1G_2}$

와 같이 결정된다.

3.6 State의 분리

State의 분리는 임의의 State의 Self-Transition에서 두개 이상의 출력 심볼 그룹이 존재하는 경우와 어떠한 State에서도 두개 이상의 그룹이 존재하지 않는 경우로 구분된다. 첫번째와 같이 한 State에서의 Self-Transition에서 두개 이상의 출력 심볼 그룹이 존재하는 경우는 해당 State를 두 State로 분리해야 하는데, Serial 분리와 Parallel 분리의 선택은 다음의 규칙에 따른다.

- 두개 이상의 Self-Transition에서 두개 이상의 출력 심볼 그룹이 존재하는 경우는 초기 State에 가까운 State를 분리
- 한 Self-Transition에서 3개 이상의 출력 심볼 그룹이 존재하는 경우는 시간 분포가 가장 큰 그룹과 가장 작은 그룹의 관계에 따라 선택
- 두 그룹사이의 관계가 Parallel한 관계이면 Parallel 분리
- 두 그룹사이의 관계가 Sequencial한 관계이면 Serial 분리

두번째의 경우로 두개 이상의 출력 심볼 그룹을 갖는 Self-Transition이 모델내에 존재하지 않는 경우는 모든 State에서의 심볼 그룹의 분산 정도가 가장 심한 State에 대해 Serial 분리를 실행 한다. 한 State에 대한 분리 방법 및 State의 연결을 그림 4에

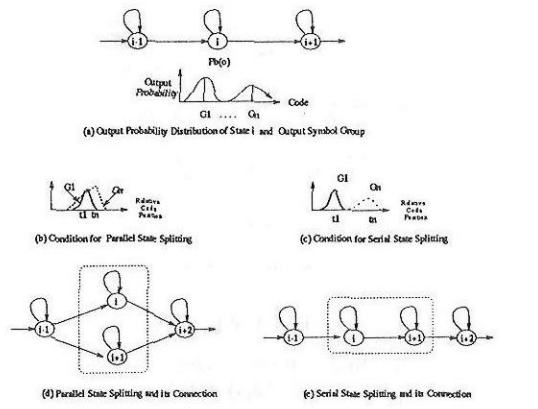

그림 4: State의 분리

나타내었다.

3.7 모델의 재구성

한 State에 대한 분리 방법이 결정되면 Transition Network에 대한 재구성을 해야 한다. 그림 4에서 볼수 있듯이, Serail 분리의 경우는 그림 4-(e)와 같이 네트워크가 구성되며, Parallel 분리의 경우는 (d)와 같이 네트워크가 재구성되며, State의 수도 전 모델보다 하나가 증가된다.

위와 같은 과정의 반복은 우선 State의 Parallel 분리가 없을 때까지 계속되며, 더이상의 Parallel 분리가 없는 경우는 각 State의 Self-Transition의 출력 심볼 그룹의 분산이 가장 큰 경우를 선택하여 Serial 분리를 계속한다. 이러한 Serial 분리는 훈련에 사용되지 않은 데이타에 대한 전 단계의 모델 Score 값이 재구성된 모델의 Score보다 좋은 경우는 재구성 과정을 종료하고, 전단계의 모델 Architecture를 선택하여 최종 모델의 파라메타 값을 구한다.

4. 실험 결과

앞에서 제안된 방법에 따라 몇가지 자소의 훈련 데이타 집합에 대한 실험을 실시한 결과 서로 다른 Code Sequence를 갖는 데이타가 섞여있지 않은 경우는 단순히 Serial 연결에 의한 모델이 구성되었다. 이것은 데이타의 분류가 HMM의 Sequencial 입력 패턴에 대한 적응에 일치하는 것으로 볼 수 있다. 그림 5에 두가지 훈련 데이타 집합(한글 자소 '기역+니은'과 '디귿')에 대한 재구성 과정의 예를 나타내었다.

제안된 방법이 Multiple Sequence를 갖는 훈련 데이타 집합에 대해 HMM을 구성할 수 있는가를 실험하기 위하여 위의 실험에서 분류된 데이타들중 2 종류 혹은 3종류를 섞어 새로운 데이타 집합을 만들어 실험을 하였다. 그 결과 여러개의 State가 Multiple Transition을 갖는 모델이 구성 되었으나, 일부 복잡한 데이타 집합에 대한 모델은 원하는 결과를 내지 못했다. 제안된 방법에 의해 생성되는 모델이 예상했던 것과 다르게 나타난 경우는 Serial 분리와 Parallel

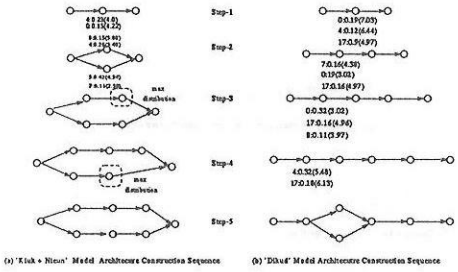

그림 5: HMM 구성 과정의 예

분리의 기준이 되는 State Duration 값에 대한 파라메타 r가 적절하지 못했기 때문이다. 특히 이러한 현상은 초기 모델의 파라메타 값 분석시에 초기 State에서 모든 심볼들의 출력이 있는 관계로 같은 부류의 심볼이 Sequence 상에 두번 이상 나오는 경우에는 심볼의 시간 분포 정보가 잘못 계산되어 Parallel 분리를 실시하게 되는 경우가 발생한다.

일반적으로 Sequencial한 입력 패턴에 대한 인식 능력이 좋은 HMM에 있어서 Multiple Path를 갖는 State가 모델내에 존재하는 경우에는 같은 Label의 서로 다른 두개의 모델로 구분할 수도 있다. 이러한 경우에는 하나의 훈련 데이타를 두개의 훈련 데이타로 분리해야 하는데, Multiple Path가 존재하는 State에서 각 Path로 가는 샘플 데이타를 별도로 분류함으로써 새로운 두개의 훈련 데이타 집합을 얻을 수 있다.

5. 결론

본 논문에서는 임의의 한 샘플 데이타 집합이 주어졌을때 이에 적절한 HMM의 Architecture를 Automatic하게 생성하는 한 방법을 제시하였다. 제시된 방법에 의하면 각각의 자소 모델에 대한 적절한 HMM Architecture의 구성을 Automatic하게 수행할 수 있으며, 따라서 프로그래머가 훈련데이타를 분석하여 데이타를 모델의 Architecture를 결정하는 번거로움을 제거할 수 있게 된다.
그러나 서로 다른 여러 Code Sequence가 혼합된 경우는 적절한 모델 생성이 어려웠으며, Parallel 분리가 잘못된 경우에 이를 취소할 수 있는 방법이 없어서, 한번 잘못된 분리는 교정이 불가능하였다. 향후 연구는 제시된 방법에 따라 모델을 생성한 후 전체적인 인식 Score의 향상이 어느정도 있었는가와, State의 결합에 대한 방법을 찾는 것이다.

참고 문헌

[1] L.R. Rabiner, B.H. Juang, *An Introduction to Hidden Markov Models*, IEEE ASSP Magazine, January 1986.

[2] K.F. Lee, H.W. Hon, *Speaker-Independent Phone Recognition Using Hidden Markov Models*, IEEE ASSP Magazine, November 1989.

[3] J. Takami, S. Sagayama, *A Successive State Splitting Algorithm for Effiecient Allophone Modeling*, IEEE, 1986.

[4] Kai-Fu Lee, *Automatic Speech Recognition : The Development of the SPHINX System*, Kluwer Academic Publishers, Boston 1989.

[5] Paolo D'Orta, Marco Ferretti, and Stefano Scarci, *Phoneme Classification for Real Time Speach Recognition of Italian*, CH2396-0/87/0000 0081 IEEE, 1987.