

Design of a Statistical Package for Hidden Markov Models and it's Construction¹⁾

ChangHyuck Oh²⁾, DeokKwan Woo³⁾, YongHyung Lee⁴⁾

Abstract

An HMM statistical package with graphic user interface is constructed with C++ classes for HMM operations. An isolated word speech recognition experiment is conducted with nine English words.

1. 서론

사회과학, 자연과학, 의학 등의 다양한 분야의 자료분석을 위한 확률모형을 만들고, 주어진 모형의 분석 기법을 연구하고, 실제로 자료를 분석할 수 있는 도구를 개발하는 것 등이 통계학의 중요한 역할이라고 할 수 있다. 통계학의 이러한 역할은 음성인식이나 영상인식과 같은 분야에서도 마찬가지이며, 평균이나 분산을 구하는 기초적 통계분석은 물론 회귀분석, 분산분석, 시계열분석, 판별분석 등 다양한 통계기법이 이들 분야의 자료를 분석하기 위하여 사용되고 있다(Rabiner 외, 1993; Fukunaga, 1990). 아울러 음성자료나 영상자료와 같은 자료의 개수가 많고 종속성을 가정하는 것이 타당해 보이는 자료의 특성을 분석하기 위하여 숨은 마코브모형(hidden Markov model, HMM)이나 인공신경망 등과 같은 확률모형을 이용한 자료분석기법이 많이 활용되고 있다(Rabiner 외, 1993; Mammone, 1994).

HMM은 "불완전관측치를 가진 이중확률과정(doubly stochastic processes with incomplete observations)" (Levinson, 1985)으로써, 1960년대말과 1970년대초에 발표된 연구 결과들을 이용하여 카네기멜론대학교의 Baker(1975)와 IBM의 Jelinek(1975)가 각기 음성인식시스템을 구축하면서부터 이 분야에 사용되기 시작되었으며 현재는 대부분의 음성인식시스템은 HMM을 이용한 인식 기능을 가지고 있다(Rabiner, 1989). HMM은 음성인식 분야 뿐만 아니라 영상인식(Paulus 외, 1997), 생명공학(Karchin and Hughey, 1998; Gundy 외, 1997) 등의 분야에서도 자료분석의 도구로 사용되고 있으며 특히 유전자 분석과 같은 일을 전담하는 HMM 소프트웨어 VEIL(Henderson 외, 1997)이나 HMMPro(Net-ID, 1999)도 개발되어 사용되고 있다.

HMM 분석 모듈은 상용 음성인식 개발 도구인 HTK(The Hidden Markov Model Tool Kit)(Young, 1999), CSLU Toolkit(Cole, 1999)에 포함되어 있으며 특히 이들 두 제품은 C 언어로 작성된 프로그램 코드를 공개하고 있다. 따라서 HMM을 이용한 독자적 음성인식시스템을 구축하거나 HMM을 이용하여 음성자료의 분석을 하고자 하는 경우 이들이 제공하는 프로그램 코드를

1) This paper was accomplished with Reserach Fund provided by Korea Research Foundation, Support for Faculty Research Aborad.

2) Professor, Departmenat of Ststistics, Yeungnam University, Kyungsan, 712-749

3) Graduate, Departmenat of Ststistics, Yeungnam University

4) Graduate, Departmenat of Ststistics, Yeungnam University

사용할 수 있다. 그러나 이들 코드는 C 언어로 작성되어 있어 C 언어가 가지는 많은 장점에도 불구하고, 최근에 많이 사용되고 있는 객체언어 C++에 비해 확장성이나 재사용성, 그리고 사용의 용이성에 많은 어려움이 따른다(Pohl, 1999). 또한 음성인식을 주목적으로 개발되었으므로, HMM에서의 간단한 계산 문제 등을 위한 학습용 소프트웨어를 작성하려고 하는 경우나 음성인식 외의 다른 분야의 자료를 분석하기 위한 통계패키지를 구축하려는 경우 지나치게 복잡하여 사용이 간단하지 않다. 이들 개발 도구의 이러한 특성은 음성인식을 위한 HMM 모듈을 C 언어로 작성한 데서 기인한다고 판단되며, 객체지향 프로그래밍기법(Object Oriented Programming)을 사용하여 기능별로 객체화 혹은 클래스(class)화하는 경우에는 이러한 어려움을 극복할 수 있으리라고 판단된다.

한편 HMM 분석도구들은 UNIX 운영체제를 사용하는 워크스테이션과 같은 고가의 컴퓨터에서 개발되었다. 이는 대용량의 자료에 많은 양의 계산을 실시간으로 해결해야 하는 음성인식의 특성 때문이었다. 그러나, 최근들어 저가의 개인용 컴퓨터(PC)의 성능이 급격히 향상되어 PC에서도 음성인식과 같은 기능을 수행할 수 있게 되었으며 대부분의 통계학자들이 Microsoft Windows(MS 윈도우) 환경의 PC를 사용하고 있으므로, MS 윈도우 운영체제와 그래픽사용자환경(GUI) 입출력 방식을 기반으로 설계된 HMM 패키지의 필요성이 대두되고 있다.

이러한 점들을 고려하여 본 연구에서는 C++ 언어를 이용하여 객체형 HMM 모듈을 개발하고 MS 윈도우 운영체제에서 HMM의 여러 가지 계산 문제를 GUI 방식으로 해결하는 패키지를 설계 구현하였다. 개발된 HMM 모듈은 음성인식이나 영상인식, 혹은 생명공학과 같은 여러 분야의 자료 분석도구를 위한 엔진(engine)으로 사용되어 질 수 있을 것이다.

따라오는 2절에서는 HMM의 개략적 소개를 하며, 3절에서는 HMM 분석시스템의 설계와 구현에 관한 내용을 다루며, 4절에서는 구현된 분석시스템의 응용 예를 보인다.

2. 숨은 마코브모형의 개략적 소개

숨은 마코브모형에서는 두 개의 확률과정이 존재한다. 관측할 수 없는 확률과정은 마코브모형으로 가정한다. 관측할 수 있는 확률과정은 관측할 수 없는 마코브모형의 상태에 의존하는 것으로 가정한다.

숨은 마코브모형을 적용할 수 있는 예로서는 음성모형을 들 수 있다. 즉, 말하는 사람이 표현하고자 하는 음의 열, 예컨대 'ㄱ', 'ㅏ', 'ㄴ' ... 등은 마코브모형을 따른다고 가정하고 관측할 수 있는 것은 발성하는 각 음의 영향을 받는 소리 신호 열이라고 가정하는 것이다. 또 다른 예로서는 문자인식을 위한 모형으로 진행되는 자모의 열은 마코브모형을 따른다고 가정하고, 광학기구로 받아들이는 것, 즉 관측할 수 있는 것은 각 자모의 영향을 받는 흑점과 백점들을 포함하는 일정한 크기의 사각형과 같은 그림의 열로 가정하는 모형이다. 컴퓨터의 입장에서 보면 진행되는 자모의 열은 관측될 수 없는 것이다. 이 절에서는 시스템의 구현에 필요한 알고리즘을 중심으로 HMM을 소개한다. 자세한 이론적 배경과 내용, 음성인식에서의 응용 예 등은 Levinson(1985), Rabiner(1989), Huang 외(1990) 혹은 Juang 과 Rabiner(1991)을 참조하기 바란다.

시각 $t=1, 2, \dots, T$ 에서 시스템의 상태가 N 개의 상태 $\{1, 2, \dots, N\}$ 중 하나가 되는 N -상태 마코브연쇄를 생각하자. 시각 t 에서 상태를 나타내는 확률변수는 Q_t , 변이확률행렬은 $A=(a_{ij})$, $a_{ij} = \Pr(Q_t=j|Q_{t-1}=i)$, $1 \leq i, j \leq N$, $t=1, 2, \dots$ 라고 하자. 시각 $t=1$ 에서는 시스템의 상태는

초기 상태확률 $\pi_i = \Pr(Q_1 = i)$ 로 정해진다고 가정한다. 이 때, 주어진 마코브연쇄에 의하여 상태열 $\mathbf{q} = (q_1, q_2, \dots, q_T)$ 가 생성될 확률은 다음 식으로 주어진다.

$$\Pr(\mathbf{Q} = \mathbf{q} | A, \pi) = \pi_{q_1} a_{q_1 q_2} \cdots a_{q_{T-1} q_T}.$$

다음은 마코브연쇄에서 상태열 \mathbf{q} 를 직접 관측할 수 없다고 가정하자. 대신에 시스템의 상태 $q_t \in \{1, 2, \dots, N\}$ 에 의존하는 확률변수 O_t 가 관측된다고 하자. 각 가능한 상태 i 에서 O_t 의 관측 확률 분포는 $B = \{b_i(o_t)\}_{i=1}^N$ 으로 주어진다고 가정하자. 단, $b_i(o_t) = \Pr(O_t = o_t | Q_t = i)$. 이 때, 관측열 $\mathbf{o} = (o_1, o_2, \dots, o_T)$ 의 확률은 다음 식으로 주어진다.

$$\Pr(\mathbf{O} = \mathbf{o} | \pi, A, B) = \sum_{\mathbf{q}} \pi_{q_1} \prod_{t=1}^T a_{q_{t-1} q_t} b_{q_t}(o_t). \tag{2.1}$$

숨은 마코브모형은 초기 확률분포 π , 변이확률행렬 A , 조건부 관측확률분포 B 를 이용하여 $\lambda = (\pi, A, B)$ 로 나타내기로 한다. 여기에서는 λ 를 그냥 모형이라고 부르기로 하자.

숨은 마코브모형에서는 다음의 세가지 문제가 주요한 관심의 대상이다. 첫째, 주어진 모형 λ 에 대해서 확률 $\Pr(\mathbf{O} = \mathbf{o} | \lambda)$ 을 효과적으로 계산하는 방법, 둘째, 주어진 초기모형 λ 에 대하여 관측치 \mathbf{o} 를 이용하여 개정된 모형 $\bar{\lambda}$ 을 구하는 문제, 셋째 모형 λ 와 관측열 \mathbf{o} 가 주어졌을 때, 관측열을 생성케한 상태열 \mathbf{q} 를 추정하는 문제이다. 음성인식의 영역에서는, 이들 세가지 문제는 각각 평가문제, 추정문제, 해독문제라고 불리어진다.

관측열의 확률을 계산할 때는 효율성이 주요한 문제가 된다. 확률 $\Pr(\mathbf{O} = \mathbf{o} | \lambda)$ 를 식 (2.1)로부터 직접 계산하면 필요한 연산의 수가 $2 \cdot T \cdot N^{T+1}$ 가 되어 음성자료와 같은 대용량 자료의 경우 컴퓨터에서의 실행 속도가 문제가 된다. 그러나 전진절차 혹은 후진 절차를 이용하면 연산의 수가 획기적으로 줄어들어 $T \cdot N^2$ 에 비례하게되므로 음성인식 등을 위한 시스템에서의 사용이 가능해진다. 다음은 전진변수 $\alpha_t(i) = \Pr(O_1 = o_1, \dots, O_t = o_t, q_t = i | \lambda)$ 를 이용한 전진절차이다.

1) 초기화 :

$$\alpha_1(i) = \pi_i b_i(o_1), \quad 1 \leq i \leq N.$$

2) 반복 : $t=2, \dots, T$ 에 대하여

$$\alpha_t(j) = \left[\sum_{i=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(o_t), \quad 1 \leq j \leq N.$$

3) 종료 :

$$\Pr(\mathbf{O} = \mathbf{o} | \lambda) = \sum_{i=1}^N \alpha_T(i).$$

관측열 $\mathbf{O} = \mathbf{o}$ 가 주어졌을 때 모수 추정에 사용되는 방법 중 하나는 최우추정량을 구하는 Baum-Welch 알고리즘이다. 이 알고리즘은 Dempster 등(1977)의 EM 알고리즘과 일치하며 국소 최대점을 찾아 준다. 먼저 $t=1, 2, \dots$ 에 대하여 다음을 정의한다.

$$\xi_t(i, j) = P(Q_t = i, Q_{t+1} = j | \mathbf{O} = \mathbf{o}, \lambda), \quad \gamma_t(i) = \sum_{j=1}^N \xi_t(i, j). \tag{2.2}$$

초기확률, 변이확률행렬, 조건부관측확률을 다음과 같이 추정한다.

$$\bar{\pi}_i = \gamma_1(i), \quad \bar{a}_{ij} = \sum_{t=1}^{T-1} \xi_t(i, j) / \sum_{t=1}^{T-1} \gamma_t(i), \quad \bar{b}_j(k) = \sum_{t=1}^T \gamma_t(j) / \sum_{t=1}^T \gamma_t(j). \quad (2.3)$$

s.t. $O_t = v_k$

식 (2.2)와 (2.3)의 방법으로 초기 모형 $\lambda = (\pi, A, B)$ 를 개정하여 얻은 모형을 $\bar{\lambda} = (\bar{\pi}, \bar{A}, \bar{B})$ 라고 할 때, 일정한 수렴조건을 만족하면 중지하여 최종 개정모형을 얻고 그렇지 않은 경우 $\bar{\lambda}$ 를 λ 로 대치한 후 같은 과정을 반복하게된다.

주어진 관측열 o 에 대하여 상태열을 추정하는 방법 중 최우추정량을 찾는 Viterbi 알고리즘은

$$\delta_t(i) = \max_{q_1, \dots, q_{t-1}} \Pr(Q_1 = q_1, \dots, Q_{t-1} = q_{t-1}, Q_t = i, O_1 = o_1, \dots, O_t = o_t | \lambda)$$

에 대하여 다음과 같이 주어진다.

1) 초기화 :

$$\delta_1(i) = \pi_i b_i(o_1), \quad \phi_0(i) = 0, \quad 1 \leq i \leq N.$$

2) 반복 : 에 $t=2, \dots, T$ 대하여

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t), \quad \phi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 1 \leq j \leq N.$$

3) 종료 :

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)], \quad q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)].$$

4) 경로(상태열)추적 : $t=T-1, T-2, \dots, 1$ 에 대하여

$$q_t^* = \phi_{t+1}(q_{t+1}^*).$$

한편 관측열에 대하여 연속성을 가정할 때에도 위의 세가지 문제를 해결하는 알고리즘은 동일하다. 다만 이산형에서 사용한 확률값 대신 연속확률분포의 밀도를 사용한다. 음성인식의 경우, 연속형을 위한 분포로는 혼합정규분포가 흔히 사용된다.

3. HMM 패키지의 설계 구현

일반적으로 패키지의 설계 구현을 위한 개발환경을 위해서는 하드웨어, 운영체제, 개발언어, 컴파일러 등이 고려되어야 한다. 이러한 개발환경의 설정에 따르는 문제들은 다음과 같다.

하드웨어 환경 : 10년전 만 하더라도 HMM을 이용한 음성인식시스템을 만들기 위해서는 고속의 중앙연산장치와 대용량의 기억용량이 필요하였다. 따라서 음성인식연구나 시스템 구축은 워크스테이션 급 이상의 컴퓨터에서 이루어졌었다. 그러나 최근에 들어서는 하드웨어의 성능이 급격히 개선되어, 개인용 컴퓨터에서도 실시간으로 HMM 음성인식 기능을 수행할 수 있게 되었고 대부분의 컴퓨터 사용자가 PC를 사용하고 있으므로 개인용 컴퓨터를 기반으로 하는 시스템을 만들기로 하였다.

운영체제 : 개인용 컴퓨터에 사용되는 운영체제는 Microsoft Windows(MS 윈도우) 계열과, 공개 운영체제인 Linux 등이 있다. 이 중에서 MS 윈도우는 개인용 컴퓨터의 일반적인 운영체제로

보급되어 있어 대부분의 컴퓨터 사용자가 사용하고 있으므로 MS 윈도우를 기반으로 하는 시스템을 개발하기로 하였다.

개발언어 : UNIX 환경의 HMM 시스템은 거의 모두 C 언어를 기반으로 하고 있다. 이는 음성 인식시스템은 실시간으로 작동해야하므로, 이를 뒷받침해 줄 수 있는 언어가 필요하기 때문이다. 본 시스템을 개발하기 위한 언어로 C 언어를 1차적 후보로 선정하였으나 객체지향 프로그래밍 기법이 가지는 코드의 우수한 재활용성, 시스템의 확장성, 모듈성 등의 장점을 살리기 위해 객체지향적 언어인 C++와 JAVA를 후보로 선정하였다. 그러나 JAVA는 기억장소 할당과 해제 등의 기능이 없고 컴파일 언어가 아닌 인터프리터 언어여서 실행 속도가 C++에 비해 느려 제외하였다(Lemay 외, 1996).

컴파일러 : 시스템 구축을 위해서 Microsoft사의 Visual C++(Ver. 6.0) 컴파일러를 이용하였다. Visual C++에서 제공되는 MFC 라이브러리는 다양한 윈도우 입출력 기능을 지원하므로 GUI 방식의 입출력환경을 구현하기에 적당한 것으로 판단되었다(전병선, 1999).

사용자 환경 : 통계패키지는 S-Plus나 SAS와 같은 명령어 기반의 사용자 입출력 환경을 지원하는 것도 있고, 최근 판의 SPSS처럼 그래픽 사용자 입출력 환경(GUI, graphic user interface)을 지원하는 것도 있다. 또한 MATLAB처럼 명령어와 그래픽환경을 병용하는 경우도 있다. 본 시스템에서는 사용자의 편리성을 위하여 GUI 입출력 환경을 기반으로 하기로 하였다. 다만 시스템의 확장의 경우에 대비하여 명령어 기반의 환경을 설정하여 두었다. 추후로 MATLAB 등과 같은 명령어 체계를 구축하여 사용할 수 있도록 하였다.

시스템에는 HMM의 모수 자료 파일을 입력하는 기능, 관측자료를 입력하는 기능, 주어진 관측열의 확률 계산 기능, 주어진 관측열에 대하여 마코브 상태열 추정 기능, 모형 재추정 기능이 포함되었다. 또한 음성인식 실험을 하기 위하여 음성신호를 13차원의 LPC(Rabiner, 1989) 자료로 변환하는 기능, 모형을 훈련하는 기능, 훈련된 모형을 이용하여 마이크를 통하여 고립어를 음성인식하는 기능을 포함하였다.

HMM의 여러가지 기능은 C++의 class(클래스)를 이용하여 객체화하였으며 HMM 모수 파일을 입력하는 클래스 CooHMMSpaceFiles, 관측열 파일을 입력하는 클래스 CooHMMObsFiles, HMM의 세가지 문제를 해결하는 클래스 CooHMMOpera와 보조 클래스로 구성하였다. 특히 클래스 CooHMMOpera는 두 개의 클래스 CooHMMSpaceFiles와 CooHMMOpera에서 public 상속하였다. 이들 클래스를 정의한 header 파일과 기능은 다음과 같다.

```
ooHMMSpaceDefin.h, HMM 모수 구조 정의
ooHMMSpaceFiles.h, HMM 파일 입력 기능의 클래스 CooHMMSpaceFiles 선언
ooHMMObsDefin.h, 관측열 구조 정의
ooHMMObsFiles.h, 관측 파일 입력 기능의 CooHMMObsFiles 선언
ooHMMOpera.h, HMM 관련 연산 기능의 CooHMMOpera 선언
ooConstants.h, 상수열 정의
ooMemAlloc.h, 기억장소 할당 해제 기능
ooVectMat.h, 벡터정의
ooLPC.h, LPC 계산기능의 클래스 CooLPC 선언
```

클래스 CooHMMOpera에서는 HMM의 주요한 세가지 문제를 해결하는 평가문제(전진절차 및

후진절차 확률계산), 모형재추정, 최적경로추정 함수를 포함하고 있다. 적용되는 HMM은 이산형 HMM, 관측열에 대하여 혼합정규분포를 가정하는 연속형 HMM이다. 혼합정규분포의 원소인 각 정규분포는 분산공분산행렬이 대각행렬인 경우만 구현하였으며 일반적인 분산공분산행렬의 경우에 대하여는 추후 구현할 예정이다. 다음은 클래스 CooHMMOpera에 포함된 HMM의 세가지 문제를 해결하는 함수와 관련 알고리즘이다. 각 함수에서 입력변수는 모형의 번호(hIndex)와 관측열의 번호(oIndex)이다.

```

확률계산(전진절차 · 후진절차) :
char *Forward_ProbObsSeq(int hIndex, int oIndex);
char *Backward_ProbObsSeq(int hIndex, int oIndex);
    
```

```

경로추정(Viterbi) :
char *FindBestPath(int hIndex, int oIndex);
    
```

```

모형재추정(Baum-Welch, 이산형 · 연속형) :
char *ModelEst(int hIndex, int oIndex);
char *ModelEst(int hIndex, int oIndex, int iq);
    
```

그림 3.1은 모수파일과 관측파일을 읽고 HMM의 세가지 문제에 대한 계산을 하기 위한 대화상자이다. HMM 모수와 관측열은 파일로부터 읽어 들인 후 모수 번호와 관측열 번호를 지정하고 관측열의 확률, 상태열 추정, 모형의 재추정을 실행케 하는 단추들이 배치되어 있다.

두번째 대화상자는 HMM 초기 모수와 훈련용 관측열 파일을 읽어 들인 후 모수를 재추정 하는

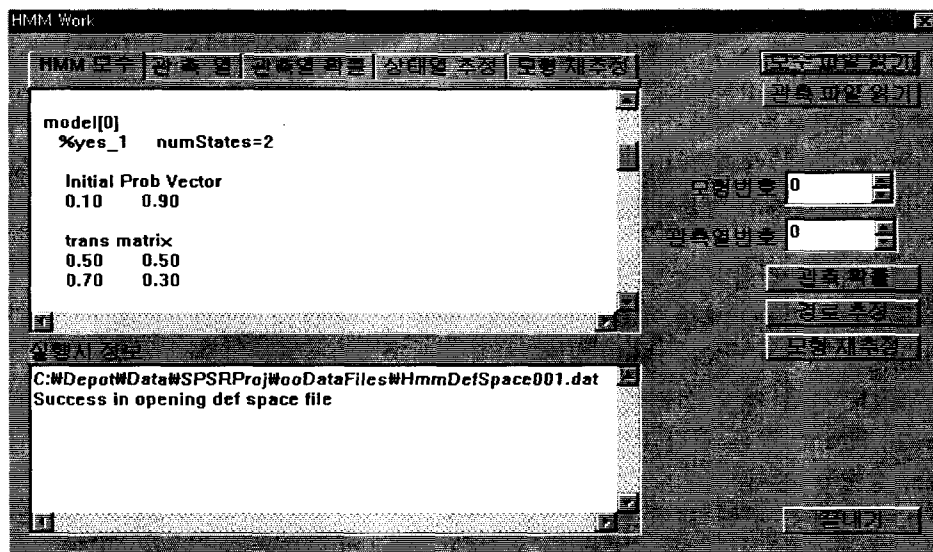


그림 3.1: 대화상자

일을 수행한다. 음성인식에서 훈련용 관측열이란 주어진 관측열을 생성케한 모형을 알고 있는 것이다. 일반적으로 해당 모형에 대하여 이름표를 붙여 두므로 훈련용 관측열은 이름표가 있다. 음

성인식의 경우에는 이름표는 음소나, 단어, 혹은 문장 등이 될 수 있다. 훈련의 경우, 각 모형에 대하여 여러 개의 관측열이 존재하므로, 이들 모두를 이용하여 모형을 재추정하도록 하고 있다. 그리고 훈련된 각 모형에 대하여 어떤 주어진 관측열이 나올 확률을 계산하는 기능을 부가하였다. 음성인식의 경우, 주어진 관측열에 대하여 가장 큰 확률을 가지는 모형이 인식하고자 하는 단어의 모형이라고 해석한다.

세번째 대화상자는 음성인식을 위한 HMM 모수 파일을 읽어 들인 후 마이크를 통하여 음성인식을 해보도록 하는 기능이다.

4. HMM 패키지의 응용

예제 1. 확률계산에의 이용

HMM의 세가지 문제에 대한 수치 결과를 알아보기 위해서는 첫번째 대화상자를 사용하면 된다. 이 때 계산에 필요한 값들은 파일로 부터 읽어들이는 데 입력 파일은 HMM 모수 파일과 관측열 파일이다. 모수파일은 HMM 모수 공간의 특성을 정의하는 파일과, 각각의 HMM 모수 값을 포함하는 파일로, 관측열 파일은 관측열 공간을 정의하는 파일과 각각의 관측열의 값을 포함하는 파일로 구성되어 있다. 모수 파일과 관측열 파일에서 값을 읽어 들이고 난 후에는 단추들을 눌러 확률의 계산, 상태추정, 모형재추정을 하면 된다. 이 때 여러 개의 HMM과 여러 개의 관측열이 존재할 수 있으므로 이들의 번호를 지정하면 된다. 그림 3.1은 읽어 들인 초기모형의 예이며, 그림 4.1은 관측열

2 1 0 1 2 0 0

을 이용하여 Baum-Welch 알고리즘으로 주어진 모형을 재추정한 결과이다.

예제 2. 고립어 음성인식

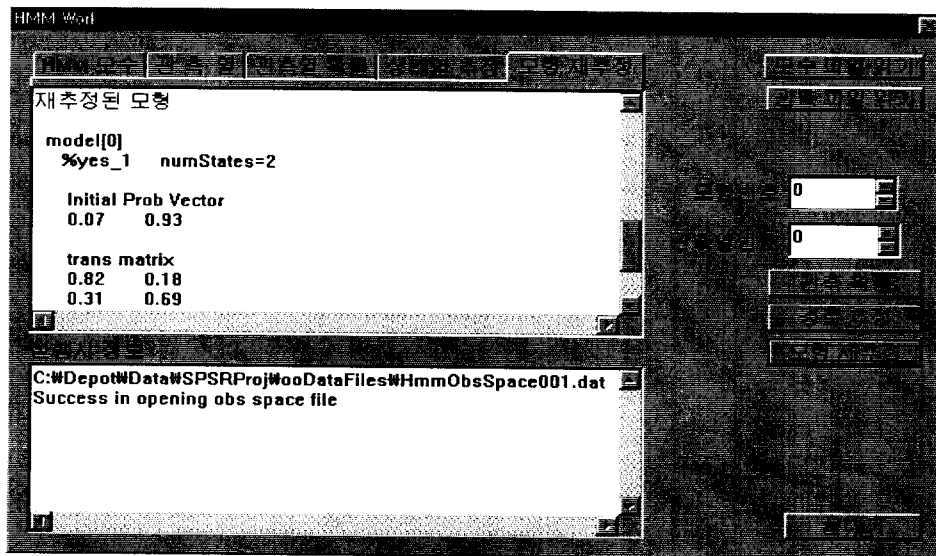


그림 4.1: 대화상자

본 연구에서 개발된 HMM 시스템을 고립어 음성인식에 적용하는 방법을 소개한다. 여기에서는 초등학교 영어교과서의 기본 단어 중에서 9개의 단어를 인식실험하기로 하였다. 그 이유는 훈련용 음성 자료의 수집이 상대적으로 수월하다는 데 있다. 훈련용 음성자료수집은 초등학교 영어 학습용 CD 오성식 생활영어(1993), 박영일 생활영어, Bionic English(1997), ABC영어회화(1998), 계몽 초등영어사전(1996), 단어암기게임(1997) CD에 포함된 wave 파일을 수집하였다. 특히 훈련자료의 균일성을 확보하기 위하여 각 단어의 여러 wave 파일을 청취한 후 여성의 비슷한 발음을 골랐다. 다음은 훈련에 사용된 영어 단어와 각 단어에 대한 훈련용 wave 파일의 개수이다.

apple(7), baby(5), ball(8), car(12), father(12), fish(12), green(11), table(15), tree(15)

사용된 단어의 선택 기준은 훈련용 wave 파일을 충분히 확보할 수 있는지였다. 한편 HMM의 모수의 초기 값을 정하기 위해서는 각 단어에 대하여 훈련용으로 선정된 wave 파일을 하나씩 골라 해당 음소라고 판단되는 곳을 청취 후 절단하고 이들의 13 차원의 LPC 벡터를 구한 다음 각 원소의 히스토그램을 그려 혼합정규분포의 평균과 분산을 적당하다고 판단되는 값으로 정하였다. 그림 4.2는 단어 car의 음성파형을 음소 별로 분리한 그림이다. 모형을 훈련시킨 후 훈련용 파일

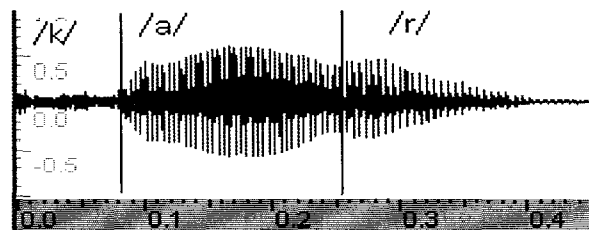


그림 4.2: 음성파형 'car'의 음소분리

을 이용하여 인식실험을 한 결과 정답률은 약 61% 이었다. 여기서 제시한 고립어인식 실험방법은 단어의 수가 100개가 되어도 동일한 방법으로 실험을 할 수 있다.

5. 논의

현재의 패키지는 HMM 시스템을 구현함에 있어 HMM의 세가지 문제를 해결하는 모듈 즉 클래스의 개발에 초점을 맞추었다. 그러나 객체지향언어인 C++의 특성을 이용하면 다양한 기능을 하는 모듈을 추가하여 시스템의 기능을 향상시키는 것이 수월하리라 판단된다. 또한 Microsoft사의 MFC를 이용함으로써 윈도우 환경하에서 다양한 GUI 입출력 기능을 추가하는 데 큰 어려움이 없으리라 판단된다. 또한 본 패키지는 HMM에서 계산문제에 따르는 중간과정을 보여줌으로써 계산보조도구로도 사용할 수 있고 학습보조도구로도 사용할 수 있다. 또한 본 HMM 시스템의 원시 프로그램은 공개되므로 이를 이용하여 필요한 소프트웨어를 작성할 수 있으리라 판단된다. 음성인식이나 영상인식 등의 분야에서 HMM 엔진을 개발하려는 경우 저자에게 연락하면 원시프로그램을 사용할 수 있을 것이다.

참고문헌

- [1] 계몽초등영어사전 (1996). 계몽사, 서울.
- [2] 광영일 생활영어 Vol. 1. 대신정보통신주식회사, 서울.
- [3] 단어암기게임 (1997). 웅진미디어, 서울.
- [4] 오성식 생활영어 (1993). SOS, Part 1, 동아출판사, 서울.
- [5] 전병선 (1999). *Microsoft Visual C++ 6.0 MFC Programming*, 삼양출판사.
- [6] ABC영어회화 (1998). 글로리소프트, 서울.
- [7] Bionic English (1997). 태경테크, 서울.
- [8] Baker, J. K. (1975). The DRAGON System - An overview, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5, 179-190.
- [9] Cole, R. (1999). Tools for research and education in speech science, *Proceedings of the International Conference of Phonetic Sciences*, San Francisco, CA.
- [10] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society, Ser. B*, Vol. 39, 1-38.
- [11] Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*, Academic Press, NY.
- [12] Grundy, W. N., Bailey, T. L., Elkan, C. P., and Baker, M. E. (1997). Hidden Markov model analysis of motifs in steroid dehydrogenases and their homologs, *Biochemical and Biophysical Research Communications*, Vol. 231, 760-766.
- [13] Henderson, J., Salzberg, S., and Fasman, K. (1997). Finding genes in DNA with a hidden Markov model, *Journal of Computational Biology*, Vol. 4, 127-141.
- [14] Huang, X. D., Ariki, Y., and Jack, M. A. (1990). *Hidden Markov Models for Speech Recognition*, Edinburgh University Press, Edinburgh.
- [15] Jelinek, F. (1975). A fast sequential decoding algorithm using a stack, *IBM Journal of Research and Development*, Vol. 13, 675-685.
- [16] Juang, B. H. and Rabiner, L. R. (1991). Hidden Markov models for speech recognition, *Technometrics*, Vol. 33, 251-272.
- [17] Karchin, R. and Hughey, R. (1998). Weighting hidden Markov models for maximum discrimination, *Bioinformatics*, Vol. 14, 772-782.
- [18] Kelly, A. and Pohl, I. (1997). *A Book on C*, 3rd Ed., Addison-Wesley, MA.
- [19] Lemay, L. and Perkins, C. L. (1996). *Teach Yourself JAVA in 21 days*, Sams.net Publishing, IN.
- [20] Levinson, S. E. (1985). Structural methods in automatic speech recognition, *Proceedings of the IEEE*, Vol. 73, 1625-1650.
- [21] Mammone, R. (1994). *Artificial Neural Networks for Speech and Vision*, Ed. R. Mammone, Chapman and Hall, NY.
- [22] Net-ID (1999). <http://www.netid.com>.
- [23] Paulus, D. W. R. and Hornegger, J. (1997). *Pattern Recognition of Images and Speech in C++*, Verlag Vieweg, Wiesbaden.

- [24] Pohl, I. (1999). *C++ for C Programmers*, 3rd Ed., Addison-Wesley, MA.
- [25] Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, Vol. 77, 257-285.
- [26] Rabiner, L. R. and Juang, B. H. (1993). *Fundamentals of Speech Recognition*, Prentice Hall, NJ.
- [27] Young, S. (1999). *The HTK Book*, Entropic Ltd., Washington DC.