최 종 연 구 보 고 서

화자검증 에 관한 연구

수탁기관 충남대학교

한 국 전 자 통 신 연 구 원

제 출 문

한국전자통신연구원장 귀하

본 보고서를 화자검증에 관한 연구의 최종연구보고서로 제출합니다.

2000년 7월 31일

수 탁 기 관 : 충남대학교
수탁기관장 : 총장 윤 형원 (인)
연구책임자 : 이 원돈 (인)
참여연구원 :
김 원겸, 변 수정, 최 범석,
오 재경, 이동재

요 약 문

1. 제 목

화자검증에 관한 연구

2. 연구의 목적 및 중요성

1) 연구의 목적

본 연구의 목적은 화자의 확인을 필요로 하는 모든 분야에서 기존의 보안 시스템의 결점을 보완하고 사용자의 거부감을 배제하며
자연스럽고 편리한 화자 확인을 위한 해결 방법으로 음성을 이용한 방법을 제안하는데 있다.

2) 연구의 중요성

기술적 측면에서 음성인식은 소규모 명령어인식 및 화자검증의 경우 기술 난이도가 비교적 쉽다고 생각하고 있으나, 전화선의 채널특성 및 배경잡음 처리 등 인식률에 저해되는 요인으로 인해 실용화의 성공을 어렵게 하고 있으며, 외국의 경우 몇몇 실용화를 하였다해도 그 기술은 일반에게 비공개적이다.
또한 외국의 상품화된 사례를 볼 때 아직까지 그 기술은 물론 제품의 성숙도에 있어 미흡한 부분이 많아. 기술의 경쟁 또는 우위의 가능성을 보여주고 있다.
채널잡음 및 환경잡음에 대한처리 기술은 음성관련 다양한 제품과 직결되는 기술로서 인식은 물론 합성의 분야에서도 그 파급효과가 지대하고 신경회로망 연구를 통하여 인간의 음성인식 기능과 관련된 뇌의 동작특성을 좀 더 잘 이해할 수 있게 함으로써 인공지능 등의 관련 분야에서 학문적 발전에 기여 할 수 있다.

경제, 산업적 측면에서 본 연구는 텔레뱅킹, 현금 출납기 등에서의 사용자 인증은 필요성이 증대하여, 우리나라의 겨우도 대부분의
은행에서 이를 필요로 하고 있어, 막대한 수입 대체효과를 가져올 수 있으며, 나아가 외국시장을 부분적으로 점하더라도 커다한 경
제적 효과를 가져 올 수 있다.

또한 음성인식의 실용화는 전자 민원서비스, 원격지 장애자교육, 놀이방, 원거리 원격조정, 각종안내 및 예약시스템 등에 직접 활용할 수 있어 막대한 산업적 부가가치를 창출할 수 있다.
뿐만 아니라 사회, 문화적 측면에서 전용자만이 갖는 유일한 특징을 활용 타인의 모방이 불가능하여 분실, 도난 등의 사회적 문제를 해결할 수 있다.
음성에 의한 서비스는 자연스럽고 신뢰성이 있으며, 기존의 명령 시스템이 가지는 거부감을 급격히 줄일 수 있다.

3. 연구의 내용 및 범위

본 연구는 신경회로망 기법을 이용한 실용적인 화자인증 시스템 구축을 위한 여러 가지 기반 기술과 신뢰성을 보장할 수 있는 시스템 구축을 위한 연구를 내용으로 하고 있다.

먼저 신경회로망의 특성인 학습, 병렬 수행 능력, 적응성 등을 음성인식 분야에 적용하여 기존의 방법이 갖는 단점을 보완했다. 특징 표현 및 추출 기법, 음성 신호의 분할 기법, 화자인증 문제의 특성에 적합한 구조의 신경회로망 모델 연구, 환경 변화에 대한 적응 능력과 노이즈 제거 방법을 위한 Sectptral Subtraction을 음성인식 시스템에 적용하였다.

또한 시스템의 성능과 적응성 향상을 위하여 모듈구조의 신경망을 사용하는 인식모형을 연구하며 이러한 내용을 적용한 시제품의 구현을 통해, 제시된 이론의 타당성과 유용성을 실험적으로 평가한다.

4. 연 구 결 과

1) 위탁 과제 최종 보고서, 신경회로망을 이용한 화자인식 시스템

2) 개발된 화자 검증 시스템의 사양 / 성능

종류	적용 환경	비 고
전화기의 종류	일반 유선 전화기, ISDN 전화	
송수신 채널 특성	Analog Channel, ISDN등의 Digital Channel	
주변환경잡음	잡음이 적은 환경	

<Discriminative HMM 학습>

종 류	목 표 수 준	비고
단어 인식율	90% 인식	70~140개 단어

<Kernel based PCA를 이용한 DTW>

종 류	목 표 수 준	비고
Kernel PCA	83 % 인식	58명의 화자
Kernel PCA from Three Utterances	81 % 인식	58명의 화자

<Spectral Subtraction>

종 류	목 표 수 준	비고
단어 인식율	90 % 인식	Learning data, 50개
단어 인식율	80 % 인식	Testing data, 40개

5. 활용에 대한 건의

Neural Network을 이용한 화자 인증 기술은 음성인식 ARS, 음성을 이용한 은행업무, Word Spotting등에 여러 부문에 적용되어 사용될 수 있으며, 또한 Neural Network을 이용하는 분야에 모델링 기법 및 learning 기법 등을 적용할 수 있다.

화자검증에 대한 연구결과로 화자검증을 위한 특징벡터 추출이나 단어분할 기법, noise 처리 기법 등의 부가 기술도 음성을 이용한 모든 응용의 중요한 부분으로 음성 관련 연구에 기반 기술로 사용될 수 있다.

6. 기 대 효 과

당해 기술의 시장성은 매년 급상승을 하는 추세로 기술의 사용 범위는 텔레뱅킹 시스템, PBX Network, Cellular Telephone
Network, Desktop PC, Telephone Calling Card, Multiple doors, Building and sites등 여러 부문에 걸쳐 방대하게 사용되고 있다.

당해 기술의 개발은 음성 또는 음성을 이용한 사용자 인증을 사용하는 모든 기술에 적용될 수 있는 기술이며, 현재 기술 선진국에서는 이 분야에 대한 많은 연구가 이루어지고 있다. 특히 유럽의 경우 CAVE나 OVID와 같이 EU에서 중점 지원하는 프로젝트를 수행하여, 이 기술에 대한 우위성 확보를 위해 노력하고 있다.

따라서 당해 기술 개발은 음성인식 및 음성 인증 분야에 대한 기술적 종속을 막을 수 있으며, 우수한 기술 개발로 인하여 음성분야에 대한 기술적 파급효과가 크다고 할 수 있다.

Summary

The goal of this research is to suggest a way of using voice to make up the weak point of current security systems and downsize the load of users by affording natural and convenient speaker verification.

In the aspect of technology, people usually think that it is not so much difficult to implement voice recognition system technically in case of several words. However, both the characteristic of a telephone line and the background noise prevent high quality of speaker verification. Even of this kind of system is practically being used in some countries, the technology itself is in the veil. Besides, when we measure the quality of the implemented system of foreign country, it is far from the expectation of ordinary users but that fact also means we can compete with foreign country and even gain the high position on the technology, Processing of channel noise and environment noise is the key technology for the diverse products of voice recognition. Therefore, it can spread great influence through not only speech recognition fields but also voice synthesis. It can also contribute on the artificial intelligence field by making it easy to understand the characteristic of brain through neural network study.

In the aspect of economy and industry, the requirement of this technology is increasing especially for the tele-banking, ATM and speaker verification in bank systems, This requirement on our research will bring great advantage against importing technology from foreign country and even give a chance to occupy foreign market on this field.

Additionally, the practical use of voice recognition will create great industrial added value by being used in electrical civil petition service, remote handicapped education, play room, remote control, various guidance and reservation service. In the aspect of social, cultural view, by using the unique characteristics of each person, this technology will provide a solution for the social problem like illegal copy, loss and robbery, The service by voice is natural and trustful and it can reduce awkwardness that ordinary system Possessed.

This research contains the contents on basic technology to build a practical speaker verification system using neural networks and on research will focus on making up the weak point of current existing method by applying the characteristics of neural networks like learning ability, parallel processing ability and adaptability. And this research will advance to characteristic presentation and extracting method, technique of dividing voice signal, research on now neural network model to solve speaker verification problem. adaptability for changing environment such as discriminative HMM. DTW using kernel-based PCA, spectral subtraction. To make improvement of system performance and adaptability, we study recognition model using module structured neural network and evaluate appropriateness and usefulness experimentally by implementing a prototype.

Speaker verification technology using neural network can be used through many kinds of fields like voice recognition ARS, bank service using voice, word spotting and etc, Moreover, modeling methods and learning methods can be applied to the fields which use neural network. Above this, it can be used in door lock system of as ID key. As the added result of the research on speaker verification, the technique for characteristic vector extraction or word spotting and noise processing also can be used as basic technology in the research dealing voice.

The need for the technology of this research is increasing every year and its range of usage is wide from tele-banking system. PBX network. cellular telephone network. desktop PC. telephone calling card and multiple doors to buildings and sites, This technology can be widely applied to every application field using voice or voice recognition. Furthermore, a lot of researches on this technology are under progress in the developed countries. Especially, in the case of Europe, they are trying to hold a dominant position on this technology by performing project like CAVE or OVID which are mainly supported by EU. Therefore, investment on this technology can prevent technical subordination on the field of speech recognition and speaker verification and its advantage coming from developing this technology can spread through all the related fields.

목 차

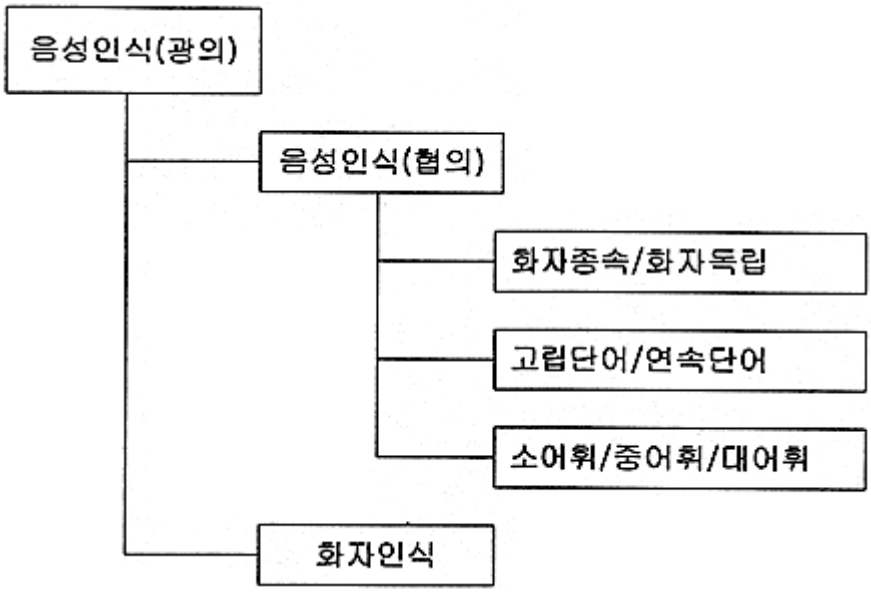
제 1 장 서론

사람들 사이에 정보를 전달하는 가장 보편적이고 일반적인 방법은 대화를 통하여 정보를 전달하는 것이다. 음성인식은 인간의 정보 전달 방식의 하나인 음성을 이용하여 화자가 전달하고자 하는 정보를 추출하여 그 내용을 결정하는 과정이다.

컴퓨터가 개발된 이후, 이것을 쉽게 사용하고자 소프트웨어의 발달이 병행되었고, 사람과 컴퓨터 사이의 communication을 쉽게
할 수 는 인터페이스의 발전이 꾸준히 요구되어 왔다. 컴퓨터 인터페이스로는 키보드, 마우스, 디지타이저, 터치 스크린 등 많은 종류가 개발되어 있지만. 인간의 언어를 컴퓨터가 이해할 수 있어 이것을 인터페이스 도구로 사용한다면 다른 어떤 것보다 자연스럽고
간편할 것이다. 또한, 음성인식은 음성입력이 타이핑이나 마우스 클릭 보다 쉽고, 빠르며, 말하는 사람이 다른 작업을 하면서도 음성으로 입력작업을 할 수 있다는 장점이 있다. 그러나 사람에 따라 차이가 있고 시간적으로 길이가 크게 다르며, 음성인식의 경우, 주위의 잡음에 민감하다는 단점을 가지고 있다.

음성인식의 대상은 인간이 발성하는 음성이다. 화자는, 듣는 사람에게 전하고자 하는 내용을 문장 형식으로 구성하고, 다음에 발성기관을 움직여서, 실제로 관측되는 음성신호를 생성한다. 음성인식은 연속적으로 생성된 음성신호를 관측하여, 음소와 음절의 이산적인 언어기호를 변환하는 것을 주목적으로 한다.

[그림 1.1] 에 나타낸 것 같이 광의의 음성인식은 내용을 인식하는 협의의 음성인식과 화자를 판정하는 화자인식으로 나누어진다. 또 협의의 음성인식은, 인식의 객체에 따라 고립단어인식, 연속단어인식으로 나눈다. 또 협의의 음성인식은, 인식의 객체에 따라 고립단어인식, 연속단어인식으로 나눈다. 고립단어 인식은 비교적 적은 어휘의 단어에 대한 인식이 목표가 되지만, 연속단어인식은 많은 양의 어휘가 필요하고 전체적인 문장의 의미를 인식하는 것이 목표가 된다. 그리고, 화자의 의존성에 따라 화자종속, 화자독립 음성인식으로 나눈다. 화자종속 음성인식은 화장에 따라 기준 패턴이 변하지만, 화자독립은 화자에 제한을 받지 않고 인식이 가능해야 한다.

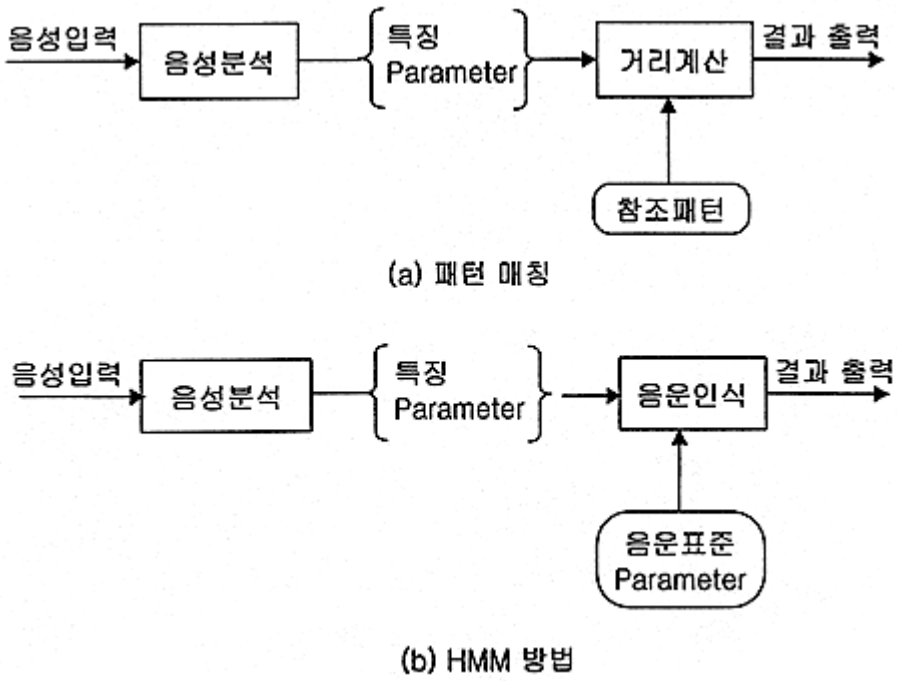


[그림 1.1] 음성인식의 기술적 분류

화자독립음성인식은 화자에 따라 음성의 특성이 다르기 때문에 화자종속 음성인식 보다 이식이 어렵다. 화자 독립 음성인식을 해결하는 방법으로는 여러 사람의 음성을 Clustering하여 N개의 인식기준을 만들거나, 새로운 화자의 음성인식에 앞서 적절한 음성으로부터 그 화자의 특성을 파악하여 인식시스템에 적응시키는 방법이 있다. 그리고 인식하려는 대상 어휘수 등의 차이에 따라 소어휘 음성인식, 중어휘 음성인식, 대어휘 음성인식으로 나누어진다.

음성을 인식하는 방법으로는 [그림 1.2] 에 나타낸 바와 같이, 크게 음성 패턴과 기준 패턴의 차이를 비교하는 (a) DTW(Dynamic
Time Warping)방법과 확률분포로 이루어진 모델에서 음성 패턴이 나올 확률로 인식하는 (b) HMM (Hidden Markov Model)방법이 있다.

패턴 매칭에 기초한 방법은 입력을 처리하여 얻은 표준패턴을 이용하여 입력된 음성 data와의 유사성을 계산하여 가장 유사도가 큰 것을 발음되었다고 생각하는 것이다. 여기서 문제점은 같은 단어라도 발음할 때마다 시간길이가 다르다는 것이다. 따라서 시간축 상에서의 비선형 패턴 매칭이 필요하다. 시간 축을 정규화 하는 매칭은 표준 패턴을 특징 파라메터의 시계열로 준비해 놓고, 입력패턴과의 비선형 대응을 조사함으로써 발성에 의한 시간축 상의 음운의 변동을 흡수하는 것이다.



[그림 1.2] 음성인식 방법

이러한 방법으로 DTW가 이용된다.

HMM은 Markov Chain의 개념을 확장하여 패턴 $X=(x_1,x_2,.....,x_T)$를 인식하는데 적합하게 구성한 모델로서 T개의 특징 벡터로
이루어진 패턴 X가 관찰될 확률이 HMM에서 구해진다. HMM은 여러 개의 State로 구성되어 있고, 자신을 포함한 각 State간에 전이확률(Transition Probability),로 구성되어 있어 세 가지 확률 parameter로 표현된다. HMM의 학습(training)은 HMM의 기본적인 문제로서 parameter 공간에서 해당 음성의 확률을 최대로 하는 값을 찾아가는 것이다. 학습방법으로는 Baum-Welch 알고리즘이나,
segmental K-means알고리즘이 사용된다. 두 알고리즘은 ML(Maximum Likelihood) estimation과 MAP (Maximum a
Posteriori) estimation을 이용할 수 있다.

$$\lambda_{ML} = \arg\max_{\lambda} f(X|\lambda) \qquad \text{(식1.1)}$$

λ는 HMM의 parameter를 의미하고 f는 evaluation 함수를 나타낸다. ML estimation은 학습 데이터(training data)에 대하여 최대의 확률이 나올 수 있는 parameter를 찾아가는 방법이다.

$$\lambda_{MAP} = \arg\max_{\lambda} g(\lambda|X) = \arg\max_{\lambda} f(X|\lambda)g(\lambda) \qquad \text{(식 1.2)}$$

MAP estimation은 λ의 분포에 대한 적당한 가정(g)이 주어질 수 있고, g에 대한 정보가 이용될 수 있는 특징이 있다. 드문드문한 학습 데이터로부터 λ 를 구하는데 MAP가 ML보다 정확하게 구하는 장점이 있다. 일반적으로 ML, MAP estimation은 EM(Expectation-Maximization)알고리즘을 사용하여 얻는다.

연속적인 HMM에서는 특징벡터의 관찰 확률분포가 Gaussian 분포로 표현되고, 그 분포의 평균과 공분산이 HMM parameter로 사용된다. Gaussian 분포는 특징벡터를 clustering하여 얻은 class로부터 평균과 공분산을 구하여 표현한다. 충분한 양의 학습 데이터가 존재하며 ML estimation은 특징벡터의 올바른 확률분포를 찾아낼 수 있지만, 충분하지 않은 양이라면 확신할 수 없는 일이다. 그래서, 올바른 확률분포를 찾기 위한 다른 방법으로 MM(Maximum Mutual Information) estimation이 있다. MMI estimation은 음성 인식하려는 단어를 a‘ 라고 하면, 학습 데이터가 a의 HMM에서 나오는 확률과 a’ 의 HMM에서 나오는 확률로 구성되는 목적함수를 최대화하는 방법이다.

마지막으로, HMM 모델을 기반으로 하여 음성인식의 성능을 향상하기 위한 효과적인 Noise 제거 방법을 제안하고 시스템을 구현해 CMS와의 성능차이를 실험하였다.

본 과제에서는 제안된 Discriminative HMM 학습 모델과 Kernel-based PCA를 이용한 DTW 방법을 이용하여 음성인식과 화자 검증 실험을 수행하였고. 잡음 제거를 위한 Spectral Subtraction을 음성인식 시스템을 적용하여 CMS(Cepstral Mean Subtraction)와 성능을 비교하였다.

제 2 장 특징벡터 추출과 음성인식 학습 모델

특징벡터 추출은 사람에 의해 발성된 음성으로부터 음성의 특성을 잘 반영하는 특징을 결정하는 것이다. 음성 파형은 아날로그 신호이지만 디지털화 과정을 통해 디지털 신호로 변환된다. 이 디지털 신호는 여러 가지 변환 과정을 거쳐 특징벡터를 구성한다.

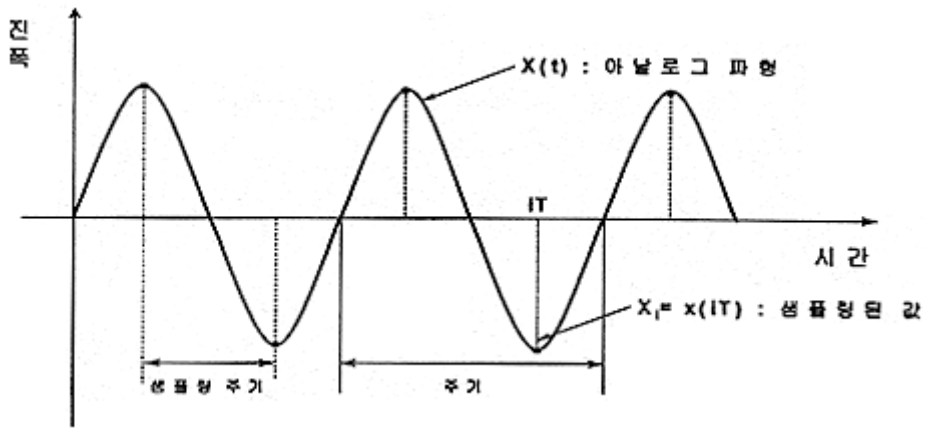
본 장에서는 특징벡터로의 변환을 위한 여러 가지 신호처리에 대해 알아보고 음성인식에 많이 사용되고 있는 모델에 대해 알아본다.

제1절 디지털화(digitization)

아날로그 신호인 음성을 사용하기 위해서는 먼저 디지털 신호로 변환하는 디지털화(digitization) 과정을 거치게 된다. 디지털화 과정은 일반적으로 샘플링(Sampling), 양자화(quantization)로 나뉘어 진다. 샘플링은 연속적인 아날로그 신호의 값을 이산적인 시점에서 값을 취한다. 이러한 과정은 쉐논의 sampling 이론- 어떤 대역폭을 갖는 아날로그 신호에 대해 샘플링 주기를 그 대역폭의 반으로 하면 원래 신호를 완벽히 복원시킬 수 있다. -에 근거하고 있다. 이 과정을 거쳐 이산적인 값으로 취해진 신호는 미리 정해진 유한개의 값으로 표현되는데 이러한 과정을 양자화라 한다.

예를 들어 전화선에서 사용하는 신호는 8KHz의 샘플링 비율(sampling rate)에 8bit의 양자화(quantization)을 사용한다. 즉, 신호의 대역폭은 샘플링 비율의 반인 4KHz이고 2^8 개의 이산적인 값으로 표현된다.

다음 [그림 2.1] 은 시간영역에서의 샘플링 과정을 보여주는 그림이고 1/T =2W [Hz] 를 나이키스트(Nyquist) 주파수라고 한다.
(T : 주기, W :대역폭)



[그림 2.1] 시간영역에서의 샘플링

제2절 선형 예측 분석법

선형 예측 분석법(linear prediction analysis)은 음성 분석 및 합성, 압축 등 음성처리 여러 분야에 광범위하게 사용되는 기법이다. 선형 예측 분석은 계산량이 적고 적은 수의 파라미터만으로 음성 혹은 음성 스펙트럼의 특성을 정확하게 표현할 수 있다.

시간 t에서의 음성 샘플을 X_t 라고 하면 선형 예측 분석법은 현재의 음성 샘플을 이전 pro의 샘플로부터 예측한다. 이때 예측값과 실제값 간의 차이를 e_t 라고 하면 다음 (식 2,1)이 성립한다.

$$X_t = -\sum_{i=1}^{p} \alpha_i x_{t-i} + e_t \qquad \text{(식 2.1)}$$

(식 2.1)에서 α_i 를 선형 예측 계수(linear prediction coefficient : LPC), 혹은 LPC 계수라 하고 e_t를 잔차신호(residual)라 한다. 시간 t에서의 음성 샘플의 예측값을 x ,라고 하면 아래 (식 2.2)로 예측값을 표현할 수 있다.

$$x_t = -\sum_{i=1}^{p} \alpha_i x_{t-i}, \quad x_t - x_t = e_t \qquad \text{(식 2.2)}$$

(식 2.2)의 예측값을 Z변환 시키면 다음 (식 2.3)을 얻을 수 있다.

$$X(z) = F(z)X(z), \quad F(z) = -\sum_{i=1}^{p} \alpha_i z^{-i} \qquad \text{(식 2.3)}$$

(식 2.2)의 오차를 나타내는 식을 Z변환하고(식 2.3)을 결합하면 A(z)에 의해 표현되는 (식 2.4)이 된다.

$$X(z) = \frac{1}{A(z)} E(z), \quad A(z) = 1 + \sum_{i=1}^{p} \alpha_i z^{-i} \qquad \text{(식 2.4)}$$

(식 2.4)을 보면 잔차신호를 1/A(z)에 통과시키면 원음을 얻을 수 있으므로 1/A(z)를 선형 예측 필터(linear prediction filter)라고 하며 원음을 A(z)에 통과시키면 잔차신호를 얻을 수 있으므로 A(z)를 역필터(inverse filter)라고 한다.

일반적으로 선형 분석법이라고 하면 A(z)의 계수인 α_i를 계산하는 방법을 말한다. t_0 ~ t_1까지 존재하는 음성 신호에 대한 선형 예측
계수α_i를 계산하는 경우 이 구간에 대한 예측 에러의 합β 은 다음과 같이 표현할 수 있다.

$$\beta = \sum_{t=t_0}^{t_1} e_t^2 = \sum_{i=0}^{p} \sum_{j=0}^{p} \alpha_i c_{ij} \alpha_j \qquad (\text{단}, c_{ij} = \sum_{t=t_0}^{t_1} x_{t-i} x_{t-j}, \alpha_0 = 1) \qquad \text{(식 2.5)}$$

이때, (식 2.5)의 β 를 최소화하기 위해 $\alpha_i \cdots \alpha_p$ 로 편미분한 결과를 0으로 하면 p개의 연립 방정식을 얻을 수 있고 이 연립 방정식의 해가 β 를 최소화하는 $\alpha_i \cdots \alpha_p$의 값이 된다.

제3절 LSP(line spectrum pair) 계수와 FFT 켑스트럼

LPC 계수는 스펙트럼 포락(spectrum envelope)의 모양을 나타내는 정보로서 음성 신호를 표현하기 위한 중요한 수단으로 사용된다. 스펙트럼 포락을 나타내기 위한 다른 방법으로는 PARCOR(partial autocorrelation) 계수, LSP(line spectrum pair) 계수 등이 있다. 이 중에서도 LSP 계수는 각 계수들의 크기가 순서대로 배열되는 특징을 갖고 있을 뿐만 아니라 변형에 강하기 때문에 음성 압축분야에서 스펙트럼 포락을 양자화 하는데 널리 사용되고 있다.

켑스트럼은 로그 크기 스펙트럼(logarithmic amplitude spectrum)의 역 푸리에 변환으로 정의된다. 켑스트럼은 주파수 영역의 함수를 역변환한 것이기 때문에 시간영역의 함수라고 할 수 있다. 켑스트럼의 가장 큰 특징은 음성이 갖는 정보를 스펙트럼 포락 정보와
세부 구조 정보를 구분한다는 것이다.
음성 생성 모델에서 음성 신호 x(t)는 다음 (식 2.6)과 같이 음원 g(t)와 스펙트럼 포락 함수 h(t)의 컨볼루션(convolution) 형태로 표현된다.

$$x(t) = \int_0^t g(\tau)h(t-\tau)d\tau, \qquad \log|X(\omega)| = \log|G(\omega)| + \log|H(\omega)| \qquad \text{(식 2.6)}$$

(식 2.6)의 뒤에 나오는 부분은 주파수 영역으로 변환한 후 양변에 절대값 함수와 로그를 차례로 적용한 것이다. 켑스트럼 $c(\tau)$는 이것의 양변을 역 푸리에 변환한 것으로 (식 2.7)와 같다.

$$c(\tau) = F^{-1}\log|X(\omega)| = F^{-1}\log|G(\omega)| + F^{-1}\log|H(\omega)| \qquad \text{(식 2.7)}$$

위의 식에서 F^{-1}은 역 푸리에 변환을 의미하고 오른쪽 두 항의 합으로 표현된다. 이때 첫 항은 음원 정보, 즉 스펙트럼의 세부 구조를 의미하고 두 번째 항은 스펙트럼 포락을 의미한다.(식 2.7)로부터 얻어진 켑스트럼을 푸리에 변환에 의해서 얻어진다는 의미로
FFT 켑스트럼이고 부른다.

제4절 음성인식 학습 모델

음성인식의 학습 모델은 사용 용도나 데이터의 특성 등에 따라 여러 가지 모델로 나눌 수 있고 요즘은 이러한 모델들의 장점을 결합된 혼합 모델이 많이 사용되고 있다. 그 모델들로는 시간상에서의 패턴을 비교하는 동적 시간 정합법과 고도의 병렬 계산 능력과 적응성을 가진 신경망 모델을 이용한 방법, 확률 모델로써 지금 까지 가장 좋은 성능을 나타내고 있는 은닉 마코프 모델(Hidden
Marcov Model, 이하 HMM)등이 있다. 각 모델을 살펴보기에 앞서 패턴의 차원을 줄이기 위한 벡터 양자화의 방법인 K-means 알고리즘과, LBG 알고리즘에 대해 먼저 알아본다.

1. 벡터 양자화

벡터 양자화는 음성인식, 음성압축, 화상처리 등에서 많이 사용되고 데이터의 차이가 너무 크거나 그 범위가 매우 큰 경우 대표 패턴이 저장된 코드북으로부터 이에 대응되는 양자화 값으로 차원의 수를 줄이거나 범위를 줄이는 방법이다.

K-means 알고리즘은 크게 두 단계로 나누어 볼 수 있다. 첫째는 각 부공간과 코드워드가 결정되어 있는 상태에서 입력벡터가 속하는 가장 가까운 부공간을 찾는 것이다. 두 번째는 모든 학습 벡터에 대해 각각 속하는 부공간이 결정된 후 새로운 코드워드를 계산하는 것이다. 그러나 K-means 알고리즘은 항상 최적의 해를 보장하지 않기 때문에 반복적인 실험을 통해 전체 평균 왜곡이 적은 코드북을 사용하는 방법이 사용되기도 한다.

K-means 알고리즘
단계 1 : 초기화 초기 코드 벡터의 집합 {$z_i(0)$, $1 \le i \le K$} 단계 2 : 분류 입력 자료들을 주어진 코드 벡터와 거리를 계산하여 최소 거리를 갖는 코드 워드로 분류한다. 이를 수식으로 나타내면 다음과 같다. $x \in C_i(t), \text{if } f \quad d[x, z_i(t)] \square d[x, z_j(t)], \quad \text{for all } i \neq j$ 단계 3 : 코드벡터 갱신 각 코드벡터로 분류된 학습자료들에 대해서 평균을 구하고 이를 새로운 코드벡터로 교체한다. t = t+1, $z_i(t)$ = $C_i(t)$의 평균, $1 \le i \le K$ 단계 4 : 종료조건 만일 전체 왜곡 D(t)와 D(t-1)의 차이가 허용 범위보다 작은 경우 종료하고 아니면 단계2부터 과정을 반복한다.

LBG 알고리즘은 학습자료를 2,4,...,2^m개의 부공간으로 점차 분할하여 원하는 코드북의 크기가 될 때까지 반복한다. LBG 알고리즘에서 단계 3과 4는 K-means 알고리즘과 동일하고 코드북의 크기를 두 배로 분할하는 방법은 다양한 경험적인 방법이 사용된다.

LBG 알고리즘
단계 1 : 초기화 코드북의 크기 L=1로 하고 전체 학습 자료의 중심 벡터를 구한다. 단계 2 : 분할 L=2L로 하여 이전 코드북 코기의 두 배가 되도록 분할한다. 단계 3 : 분류 학습 자료들을 가장 가까운 코드워드에 속하는 부공간으로 분류한다. 단계 4 : 코드복 갱신 각 부공간의 코드워드를 분류된 학습 자료를 이용하여 갱신한다. 단계 5 : 종료조건 1 현재의 전체 왜곡과 이전 전체 왜곡과의 차이가 허용 범위보다 작은 경우 종료하고 아니면 단계 3부터 과정을 되풀이한다. 단계 6 : 종료조건 2 만약 L이 원하는 코드북 보다 작으면 단계 2로 가고 그렇지 않으면 끝낸다.

2. 동적 시간 정합법(Dynamic Time Warping, DTW)

기존에 사용되던 선형 신축(linear scaling)에 의한 두 패턴 길이의 정규화는 음성이 시간에 따라 신축 정도가 비선형적이고 모음과 자음의 신축정도가 달라 여러 가지 문제점을 가지고 있었다. 이러한 문제를 비교적 빠르고 효과적으로 해결할 수 있는 방법이 동적 프로그래밍(dynamic programming)에 기반한 시간축의 비선형 신축에 의한 정합법이다. DTW는 길이가 다른 두 개의 데이터에서 최적의 정합 경로를 찾아 두 데이터를 비교할 수 있는 방법이다. DTW를 이용할 경우에는 서로 유사한 프레임들을 중첩시켜 전체
유사도를 최소화하도록 만든다.

여러 개의 데이터를 중첩할 때는 먼저 두 개의 패턴에서 DTW를 이용하여 첫번째 패턴의 프레임의 열과 가장 잘 정합하는 두 번째의 프레임의 열을 찾는다. 이렇게 하여 짝지워진 프레임들은 각각 중첩되어 새로운 하나의 표준 패턴을 만든다.

$$D(T_x T_y) = \min_{\varphi_x \varphi_y} \sum_{\square=1}^{T} d(\varphi_x(\square), \varphi_y(\square)) m(\square) \qquad \text{(식 2.8)}$$

DTW 알고리즘
단계 1 : 초기화 DA(1, 1) = d(1. 1)m(1) 단계 2 : Recursion $1 \leqq i_x \leqq Tx, \quad 1 \leqq i_y \leqq Ty$인 i_x, i_y에 대해서 $D(i_x, i_y) = \min [D(i'_x, i'_y) + \zeta(i'_x, i'_y), (i_x, i_y))]$

3. 신경회로망(Neural Network)

신경회로망은 사람의 인간 두뇌의 생물학적 작용을 모방한 것으로 기존의 von Neuman 방식의 컴퓨터의 한계를 극복하고자 영상인식, 음성인식, 적응제어 등의 무제에 적용되고 있다. 신경망은 모델이 단순하지만 상당히 많은 수의 비선형 처리 소자로 이루어져 대규모의 병렬성, 정보가 분산 저장되어 일부가 손상되어도 전체 시스템에 주는 영향이 적은 안전성(fault tolerance) 그리고, 학습하면서 점차 환경 적응을 할 수 있는 장점이 있다.

신경회로망은 연결방식에 따라 크게 완전 연결구조(fully connected)와 부분 연결구조(partial connected)로 분류할 수 있고 연결 방향에 따라 전방향 연결구조 (forward connected)와 역방향(backward connected)으로 구분된다.
학습방법에는 한 노드가 인접 노드의 활성화에 기여한다면 이들 두 노드를 연결하는 연결선의 강도는 증가되어야 한다는 hebbian
rule에 의한 방법과 실제 출력 된 값과 제지된 값간의 차이를 줄이도록 노드간의 연결 강도를 조절하고 상위 오차를 아래층으로 전파시켜 하위층의 오류를 교정하는 delta rule에 의한 방법이 있다.

신경회로망은 원하는 입력자료와 출력자료를 이용해서 그들간의 연관 구조를 학습함으로써 별도의 프로그래밍이 필요하지 않다. 그리고 기존의 순차적인 처리방식과는 달리 신경회로망 자체의 구조적인 특성으로 인해 병렬 수행이 가능하므로 많은 자료의 효율적으로 처리하는 특성을 갖는다. 또한, 학습 자료가 변할 때마다 새로운 식별함수를 만들거나 저장된 규칙(rule)을 수정하는 대신 단순히 layer의 노드를 연결하는 연결선의 가중치를 변경함으로써 새로운 자료에 대한 적응을 수행한다.

그러나, 이러한 장점에도 불구하고 학습에 사용되는 학습률과 관계된 값들의 사전 추정이 불가능하고 적절한 학습자료의 선정이 어렵고 최적화를 위한 초기 분포의 산정이 여전히 문제로 남아있다. 음성인식에 사용되는 신경회로망은 사용용도에 따라 잡음제거 및
특징 추출 등의 전처리 용도, 분류기 혹은 인식기 용도, 후처리 용도로 사용된다.

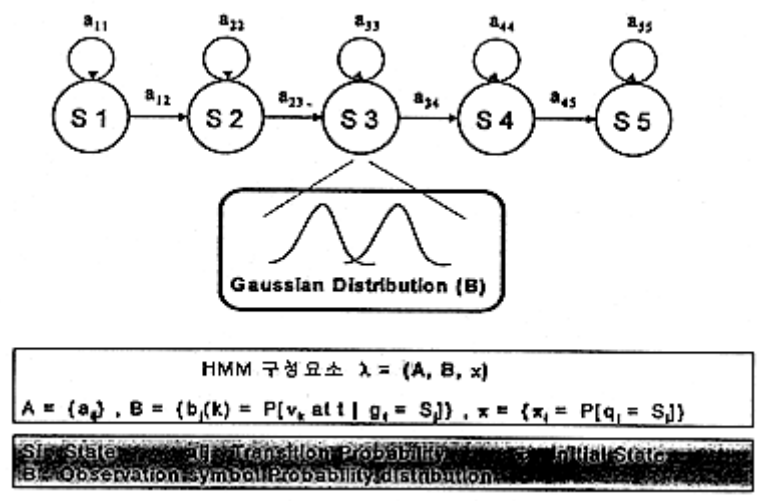
전처리에서는 신경회로망의 적응(adaptive) 특성, 함수 근사 특성 등을 적용함으로써 음성에 존재하는 잡음을 제거하고 음성으로부터 특징벡터을 추출한다. 분류기(classifier) 또는 인식기(recognizer)로 사용되는 것은 시간 정렬과 패턴 매칭을 위한 방법으로 사용된다. 입력으로 들어오는 특징벡터들을 공산상의 패턴으로 간주하여 음성을 분류하는 동적구조 신경회로망으로 분류한다. 정적구조 신경회로망은 적은 어휘수의 단어인식이나 숫자음의 경우 좋은 성능을 보이며 다층구조 신경회로망, SOFM(Self Organizing
Feature Map), LVQ(Learning Vector Quantization)등이 있다.

동적구조 신경회로망은 음성신호를 정적 패턴의 시간적인 변화까지 고려하여 시-공(spatial-temporal) 패턴으로 보고 입력 패턴의 시간적인 궤적을 학습한다. 여기에 속하는 것으로는 순환(recurrent)구조 신경회로망, temporal flow model, BPS(Back
Propagation for Sequence) 등이 있다. 후처리나 언어처리를 위해 또한 신경회로망이 사용되는데 이것은 어휘간의 분별력 향상이나 범주 예측, 지식 표현을 위해 사용된다.

여러 가지 장점에도 불구하고 음성인식에 신경회로망을 적용하는데는 많은 문제점이 있다. 음성의 불규칙성, 시간정보의 표현, 처리시간 및 계층적인 결합문제 등이 아직도 해결해야 할 과제로 남아있다.

4. HMM 모델

최근 음성인식에 가장 많이 사용되며 기본적인 모델로 자리잡은 것이 은닉 마코프 모델(Hidden Markov Model), 이하 HMM)이다.
HMM은 확률 분포에 기반한 모델로 길이가 일정치 않은 시계열의 완전-불완전 데이터(complete-incomplete data)을 이용한 방법이다. HMM중 음성인식에서 대표적인 LR(left-to-Right) model 의 기본적인 구성요소는 다음 [그림 1] 과 같다.



[그림 2.1] HMM의 구성요소

HMM은 실제적인 관측을 통해서 변화되는 통계적인 특징을 확률적으로 모델링하기 위해 마코프 과정을 이용한다. 각 상태는 은닉되어 있고 다른 관측 가능한 확률적인 과정들의 집합을 통해서만 관측될 수 있다. 이 모델의 각 은닉상태 꼬는 각 상태간의 천이는 이산 또는 확률 밀도함수로 표현되는 출력 확률 분포 집합과 연관된다.

그리고, 은닉 상태열과 관측 가능한 확률 과정 사이의 장막은 출력 확률에 의해서 표현된다. 일반적으로 HMM은 $\lambda=(A,B,\pi)$로 표현되며 상태 수 N과 이산 기호의 수 L, 그리고 매트릭스 형태의 A, B, π인 확률분포가 필요하다. 또한 초기상태 집합 S_I와 종료상태 집합 S_F가 필요하며 각각의 상태 수를 나타내는 N_I와 N_F로 나타내며 일반적으로 1로 설정한다.

HMM이 정의되면 다음과 같은 세가지 중요한 문제가 해결되어야 한다.

평가문제 (Evaluation Probiem)	관측열 $O=O_1,O_2, \cdots,O_T$와 모델 λ=(A. B. π)가 주어지면 모델에 의해 생성되는 관측열에 대한 확률 $P(O \mid \lambda)$를 어떻게 계산할 것인가의 문제이다. 이 문제는 여러 개의 경쟁 모델들과 관측열이 주어지면 분류나 인식 문제의 목적에 맞게 관측과 가장 잘 정합하는 모델을 선택하는 관점에서도 살펴 볼 수 있다.
예측문제 (Estimation Probiem)	주어진 관측열 O에서 $P(O \mid \lambda)$를 최대화하는 모델 변수 λ=(A. B. π)를 조정하는 문제이다. 따라서 이 문제에서 는 관측들이 어떻게 출현되는가를 가장 잘 표현하도록 모델 변수를 최적화시키는 문제이다.
해석문제 (Decoding Problem)	관측열 O가 주어지면 최적화 기준에 따라 가장 가까운 상태열 $S=s_1,s_2,\ldots,s_T$를 구하는 문제이다. 이 문제는 모델의 은닉된 부분을 복원시키는 과정이다.

[표 2.1] HMM의 세 가지 문제

(1) 전향-후향 알고리즘(forward-backward algorithm)

관측에 대한 확률을 가장 직접적으로 계산하는 방법은 길이 T(관측의 경우 수) 에 대한 모든 가능한 상태열을 나열함으로써 계산되어질 수 있다. 이러한 가정하에 모든 고정된 상태열 $S=S_1,S_2,\cdots,S_T$ 에 대한 관측열 O의 확률값은 $P(O \mid S,\lambda)$로 표현되며 다음과 같은 식으로 나타낼 수 있다.

$$P(OS,\lambda) = b_{s1}(O_1)b_{s2}(O_2)\cdots b_{sT}(O_T) \qquad \text{(식 2.9)}$$

이러한 상태열 S에 대한 확률값은 다음과 같이 표현된다.

$$P(S|\lambda) = \alpha_{s_0 s_1}\alpha_{s_1 s_2}\alpha_{s_2 s_3}\cdots\alpha_{s_{T-1}s_T} \qquad \text{(식 2.10)}$$

위의 두 식을 이용하여 O와 S가 동시에 일어날 확률을 구하면 다음과 같다.

$$P(O|\lambda) = \sum_{all\ s} P(O|S,\lambda)P(S|\lambda)$$

$$= \sum_{all\ S} \prod_{t=1}^{T} a_{s_{t-1}s_t} b_{s_t}(O_t) \qquad \text{(식 2.11)}$$

주어진 상태가 N개로 제한되면 생성될 관측열의 길이에 관계없이 가능한 상태열은 이들 N개의 상태로 제한된다. 이러한 특성에 기반을 두고 유도된 알고리즘이 전향-후향 알고리즘(forward-backward algorithm)이다. 전향 알고리즘에서의 전향 변수는 다음과 같이 정의된다.

$$\alpha_t(i) = P(O_1, O_2, \cdots, O_t, s_t = i|\lambda) \qquad \text{(식 2.12)}$$

위 식은 모델 λ가 주어지면 시간 t에 도달하는 상태 i에서 t 시간까지의 부분 관측에 대한 확률을 나타낸다. 이 확률값은 다음과 같이 귀납적으로 계산되어 질 수 있다.

전향 알고리즘
단계 1 : 모든 상태 i에 대해 $a_1(i) = \pi_i b_i(O_1)$를 계산한다. (만약 i∈S_1이면, $\pi_i=1/N_1$ 이며 그렇지 않은 경우 $\pi_i=0$ 이 된다.)
단계 2 : 시간 t=2,..., T인 경우에 상태 j에 대해 α()를 계산한다. $$a_t(j) = \left[\sum_i a_{t-1}(i) a_{ij} \right] b_j(O_t)$$
단계 3 : 최종 확률은 다음에 의해 계산된다. $$P(O\|\lambda) = \sum_{i \in S_F} a_{T(i)}$$

위의 전향 반복에서 단계1은 모든 상태들에 대해 초기 확률값으로 전향 확률값을 초기화한다. [표 2-5] 에서 단계2의 식은 시간 t의 상태 j가 시간 t-1에서의 모든 가능한 상태 i에서부터 도달할 수 있음을 보여준다. 여기에서 $a_{t-1}(i)$는 관측열이 $O_1, O_1, \cdots, O_{t-1}$이며, 마지막 상태가 i를 나타내는 병합 확률이다. 따라서 $\alpha_{t-1}(i)a_{ij}$는 병합 사건 $O_1, O_1, \cdots, O_{t-1}$이 관측되고 사간 t-1에서 상태 i를 통하여 시간 t에 상태 j가 도달함을 나타낸다. 이 값은 모든 가능한 상태 i에 대해 합함으로써 모든 가능한 이전 상태에서 시간 t에 상태 j가 도달하는 확률을 구하게 된다. 상태 j에서의 관측기호 O_t를 나타내는 출력 확률 $b_j(O_t)$를 곱함으로써 최종적으로 $\alpha_t(j)$를 구하며 이 값은 시간 t에 상태 j에서 새로운 관측열 $O_1, O_1, \cdots, O_{t-1}, O_t$이 출현할 확률을 나타낸다.

단계3의 최종 상태에서의 전향 변수 $\alpha_t(i)$를 더하여 원하고자 하는 최종 확률 $P(O|\lambda)$를 구한다.

비슷한 방법으로 모델 λ 에서 주어진 시간 t에서 상태 irk 주어지면 시간 t+ 1에서부터 최종 관측 T까지의 부분 관측에 대한 확률을 나타내는 후향 변수 $\beta_t(i)$는 다음과 같이 정의될 수 있다.

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \cdots, O_T | s_t = i, \lambda) \qquad \text{(식 2.13)}$$

후향 알고리즘
단계 1 :$i \in S_F$인 모든 상태에 대해 $\beta_T(i)$ = 1/N_F. 그렇지 않으면 $\beta_T(i)$ =0 단계 2 : t =T-1, T-2,...,1인 시간축에 대해 상태 j에서의 β()를 계산 $$\beta_t(j) = \sum_i ab_j(O_{t+1})\beta_{t+1}(i)]$$ 단계 3 : 다음 식에 의하여 최종 확률을 구한다. $$P(O \mid \lambda) = \sum_{i \in S_1} \pi_i b_i(O_1)\beta_1(i)$$

후향 변수는 전향 변수 $a()$와 비슷하게 귀납적으로 계산될 수 있다. 단계1은 모든 최종 상태들에 대해 $\beta_T(i)$를 임의적으로 1/N_F로 정의한다. 단계2는 시간 t에서 상태 j에 머물면서 관측열의 나머지 부분을 고려하는 과정으로 상태 j에서 시간 t+1의 가능한 모든 상태로의 천이와 그에 대응되는 상태의 관측 기호 O_{t+1}가 나올 확률을 고려한 것이다. 이상에서 언급한 바와 같이 전향/후향 알고리즘은 평가 문제로서의 $P(O|\lambda)$를 계산하는데 사용된다. 또한 이 방법은 모델의 매개변수 문제를 해결하는데 사용될 수 있다.

(2) Viterbi 알고리즘

HMM의 은닉된 부분, 즉 상태열은 밝혀질 수 없으나 다른 의미 있는 방법들로 해석될 수 있다. 복원된 상태열은 모델의 구조에 대해 학습할 수 있도록 하며, 각각의 제한된 상태에서 일반적인 통계나 행동 등을 예측할 수 있다.

주어진 관측열이 주어지면 대응되는 최적의 상태열을 찾는 방법은 여러 가지가 있을 수 있다. 한가지 가능한 방법은 최대의 $P(O,S|\lambda)$를 가지는 최적 경로를 찾는 방법이다. 이러한 방법은 Viterbi 알고리즘이라 하며 천이 경로가 무시되는 동적 시간 정합 방법과 유사하다. Viterbi 알고리즘은 평가문제에서도 사용될 수 있다.

Viterbi 알고리즘은 모든 가능한 상태열 S에 대해 $P(O,S|\lambda)$를 합함으로써 $P(O|\lambda)$를 계산하는 전향-후향 알고리즘과는 달리 모든 S에 대해 $P(O,S|\lambda)$의 값이 최대인 경로만을 고려한다. 따라서 Viterbi 알고리즘은 전향-후향 알고리즘의 특수한 형태로 해석할 수 있다. 또한 Viterbi 알고리즘은 계산 과정에서 상태열을 구할 수 있어 많은 음성인식 알고리즘에서 사용되고 있다.

Viterbi 알고리즘
단계 1 : 초기화. 모든 상태 i에 대해 초기 경로 거리를 구하고 저장 경로를 초기화한다. $\delta_1 = \pi_i b_i(O_i) \qquad \Psi_1(i) = 0$ 단계 2 : 재귀함수. 시간 2≤t≤T에서 모든 상태 j에 대해 $\delta_t(j) = \max_i[\delta_{t-1}(i)a_{ij}]b_j(O_t)$ $\Psi_t(j) = \arg\max_i[\delta_{t-1}(i)a_{ij}]$ 단계 3 : 종료(*는 최적 결과를 나타냄)$s^*_T = \arg\max_{s \in S_F}[\delta_T(s)]$ $P^* = \max_{s \in S_F}[\delta_T(s)]$ $s^*_T = \arg\max_{s \in S_F}[\delta_T(s)]$ 단계 4 : 경로 복구. t=T-1,...,1에 대해서 다음을 구한다. $S^*_T = \Psi_{t+1}(S^*_{t+1})$

(3) Baum-Welch 재예측 알고리즘(Baum-Welch re-estimation algorithm)

HMM에서 가장 어려운 문제는 주어진 모델에서 관측열의 확률을 최대화하도록 모델 변수(A,B,π)를 조정하는 것이다. 이 문제를 최대 우도(maximum likelihood) 모델 방법을 이용하여 분석적으로 풀 수 있는 방법은 없으며 최적화시키기 위해서는 반복적인 학습이나 기울기 방법(gradient algorithm)을 사용하여야 한다. HMM 기반의 음성인식에서 사용되는 반복 알고리즘은 Baum-Welch 알고리즘으로 알려져 있다. 이 방법은 혼합 밀도(mixture density) 문제를 위한 EM(Expectation Maximization) 알고리즘 기법과 같은
최적화 방법을 사용한다. EM 알고리즘은 완전 데이터(complete data)로부터 로그 우도(log likelihood)에 대한 기대값을 반복적으로 최대화함으로써 불완전 데이터(incomplete data)로부터의 로그우도를 최대화하도록 한다. 이와 같은 성질 때문에 EM 알고리즘은 최대 우도 모델 방법이라고 부르지만 계산 방법에 있어서도 일반적이 최대 우도 방법보다 덜 복잡하다.

재예측 알고리즘의 기본적인 과정은 먼저 관측 기호열이 주어지면 모델의 매개변수를 계산한다. 모델에 대한 변수가 얻어지면 전향-후향 알고리즘을 이용하여 주어진 관측열에 대해서 생성되는 확률값을 얻고 그 확률값에 기초하여 원래의 모델 변수들을 예측한다.

전향-후향 알고리즘을 이용하여 관측열에서 규정된 상태 i에서 j까지의 후 천이확률 (posterior probability of transition) ɣij 는 다음과 같다.

$$r_t(i,j) = P(s_t = i, s_{t+1} = j|O,\lambda)$$
$$= \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} \qquad \text{(식 2.14)}$$

일반적으로 a_{ij}의 물리적인 의미는 상태 i에서 j로의 천이 확률을 나타낸다. 따라서 a_{ij}에 대응하는 $\bar{a}_{ij}$는 다음과 같이 나타낼 수 있다.

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1}\gamma_t(i,j)}{\sum_{t==1}^{T-1}\gamma_t(i)} \qquad \bar{b}_j(K) = \frac{\sum_{O_t = V_k \gamma_t}(i)}{\sum_{t=1}^{T}\gamma_t(j)} \qquad \text{(식 2.15)}$$

비슷하게 $b_j(k)$의 물리적인 의미는 상태 j에서 발생하는 관측 기호 V_K에 대한 확률을 나타낸다. 이 값은 상태 j에서 전체 관측 기호의 빈도수에 대한 관측 기호 V_K의 발생 빈도에 의해서 계산되어 진다.

마지막으로 초기 상태 확률에 대한 예측은

$$\overline{\pi}_i = \gamma_1(i) \qquad \text{(식 2.16)}$$

로서 구해진다.

제 3 장 Discriminative HMM 학습

본 장의 실험 목표는 HMM의 학습을 향상시켜 최적의 parameter를 찾아서 음성인식률을 높이고자 하는 것이다. 최적의
Gaussian 분포를 찾기 위하여 신경회로망을 이용한 학습 과정을 제안한다. 학습에 경쟁이 되는 데이터를 이용하여 인식률을 높이는 방법을 사용하여 오인식을 줄이도록 한다. 학습 데이터와 경쟁이 되는 학습 데이터를 가지고 신경회로망을 이용한 분류를 하여
새로운 Gaussian 분포를 구한다. 새로운 Gaussian 분포는 경쟁이 되는 학습 데이터들의 관찰될 확률을 낮춤으로써 인식률을 향상시킨다.

제1절 개 요

HMM $\lambda=(A, B, \pi)$에서 B는 Gaussian 분포를 나타내는 parameter인데 특징 벡터들을 클러스터링 한 후, 각 class로부터 평균 벡터, covariance matrix, 가중치(weight)를 얻게 된다. B로 표현되는 Gaussian 분포는 특징 벡터가 각 state에서 관찰될 확률을 구하는데 사용된다. 보통 state는 여러 개의 Gaussian 분포가 합성되어 있는데 HMM의 parameter 중에서 B의 값이 중요한 자리를
차지하고 있다.

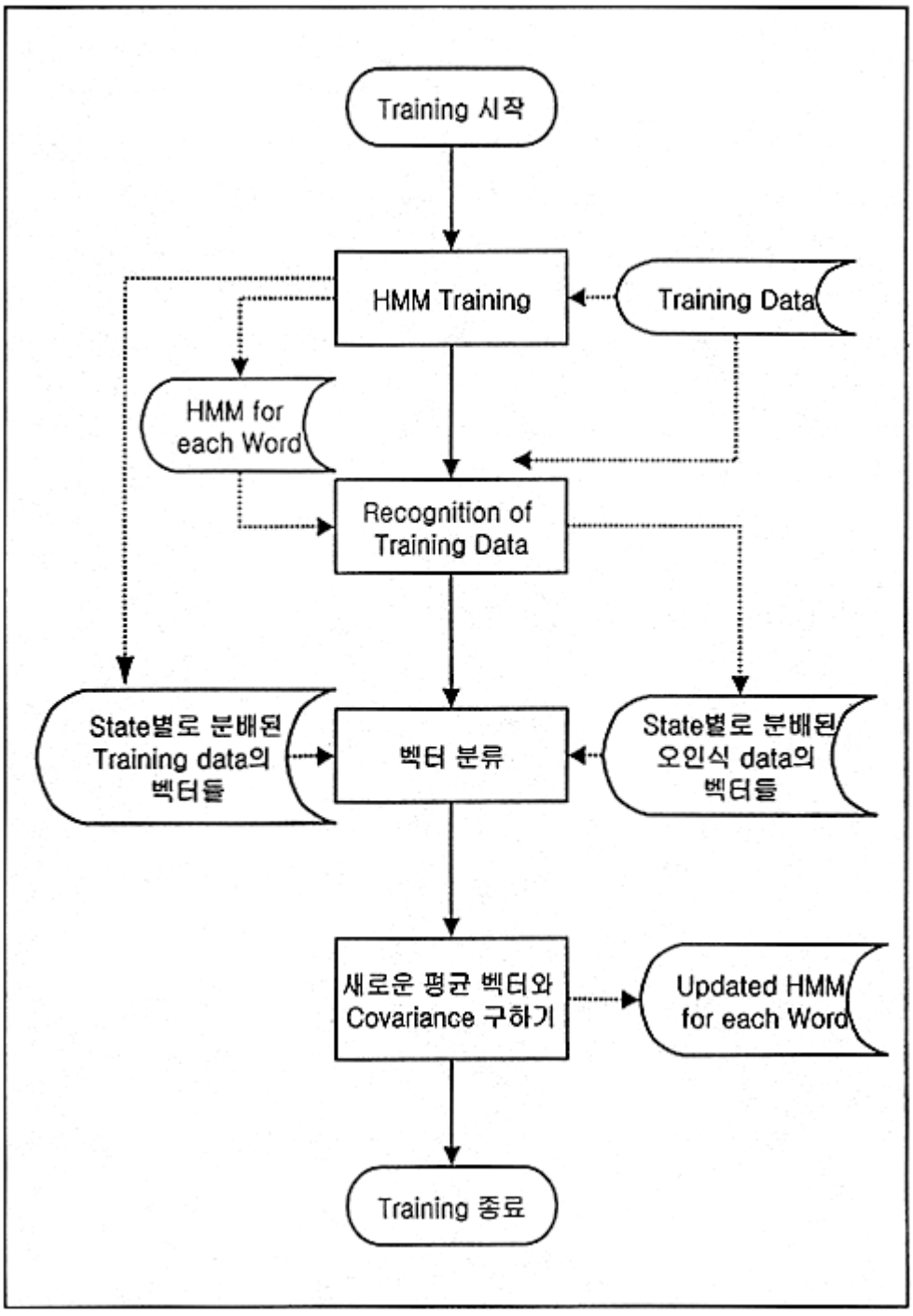
고립단어인식을 할 경우에 비슷한 음운을 갖는 단어끼리 오인식이 잘 되는 결과를 볼 수 있다. 두 단어는 발음의 유사성으로 인하여 서로 오인식이 잘 일어난다. 오인식된 단어는 자신의 HMM에서 나올 확률보다 다른 HMM에서 나올 확률이 더 크게 나왔기 때문에 오인식된 것이다. 확률이 크게 나온 이유는 오인식된 결과의 state에서 관찰될 확률이 높기 때문이고, 관찰될 확률이 높은 이유는 Gaussian 분포가 최적이 아니기 때문이다.
학습에 경쟁이 되는 데이터를 이용하여 인식률을 높이는 방법을 사용하여 오인식을 줄이도록 한다. 학습데이터와 경쟁이 되는 학습 데이터를 가지고 분류(classification)를 하여 새로운 Gaussian 분포를 구한다. 새로운 Gaussian 분포는 경쟁이 되는 특징 벡터들의 관찰될 확률이 낮아짐으로 인식률이 향상된다.

학습과정에서 학습에 경쟁이 되는 데이터는 오인식된 데이터를 선택한다. 예를 들어 '당행'에 해당하는 학습데이터를 가지고 HMM을 구하는 경우를 생각해 보자. '당행'으로 오인식되는 '타행' 학습 데이터는 경쟁이 되는 학습 데이터이다. 음성의 학습 데이터는 $X=(x_1,x_2,\cdots\cdots,x_T)$로 되어 있어서 특징 벡터를 state 별로 분배하는 과정이 필요하다. State 별로 분배하는 과정은 Viterbi 알고리즘으로 구한 optimal state sequence에 따라서 수행한다. 학습 데이터와 경쟁이 되는 데이터를 optimal state sequence에 따라 특징 벡터를 분배한다. State 별로 모여진 두 종류의 특징 벡터들을 분류하는 과정이 필요하다. 분류기법으로는 통계학에서 널리 쓰이는 Fisher의 판별 함수를 이용한다. 분류를 통하여 얻은 학습 class로부터 Gaussian 분포를 구한다. 결국, 새로운 Gaussian 분포에서 경쟁이 되는 학습 데이터의 특징 벡터가 관찰될 확률이 줄어든다.

제2절 Discriminative HMM 학습 모델

1. 알고리즘

이 절에서는 성능을 높이기 위한 HMM 학습 과정을 설명하고 있다. Fisher의 판별함수를 이용한 HMM 학습 과정이 [그림 3.1]에 나와 있다.



[그림 3.1] 제안된 HMM 학습과정

제안된 HMM 학습 과정에 대해 살펴보면, 우선 모든 단어에 대한 HMM 모델이 만들어진 후에 이것을 참조로 하여 학습 데이터를 입력하여 인식 결과를 살펴본다. 오인식된 데이터는 최대 확률이 나온 모델에서의 optimal sequence에 따라 특징 벡터들을 재분배한다. 모든 학습 데이터에 대한 인식과 벡터 분배가 끝난 후에는 모델 각각의 state에 분배된 벡터와 경쟁이 되는 벡터들을 가지고
classification 을 한다. Classification을 통하여 Gaussian mixture에 변화가 생긴다.

다음은 알고리즘의 각 프로시저에서 처리하는 일을 구체적으로 설명한다.
㉮ HMM 학습
인식의 대상이 되는 단어들의 학습 데이터로부터 단어마다 HMM (Hidden Markov Model)을 구한다. 각 단어마다 HMM의 State의 수,
각 State내의 cluster 수, transition probability, State내의 각각의 Cluster에 할당되어 있는 벡터의 수, observation probability를 파일로 저장한다. 일반적으로 음성인식에서는 initial state probability가 처음 state에서만 1이고, 나머지 state는 0으로 하기 때문에 파일 저장에서 생략한다. 학습 데이터를 optimal sequence에 따라 state 별로 모아서 파일에 저장한다.

㉯ 학습 데이터 인식
모든 단어의 학습 데이터에 대한 인식 결과를 구한다. $P(X|\lambda)$의 값은 Viterbi 알고리즘으로 구하는데, 여기서 오인식된 data는 오인식된 결과의 HMM에서 구해진 state sequence에 따라 특징 벡터를 state별로 저장한다.

㉰ 벡터 분류

State별로 모인 학습 데이터와 오인식된 data의 특징 벡터를 분류한다. 분류 방법으로 Fisher의 선형 판별함수와 entropy 함수를 이용한다. Fisher의 판별함수는 통계 분야에서 쓰이는 방법으로 우수함이 알려져 있고, entropy 함수는 threshold를 구하는데 사용된다.

이 때, 경쟁 데이터로 사용되는 데이터의 수가 학습 데이터보다 월등히 적기 때문에 벡터 분류를 한 결과 각 state에서 생성되는
cluster의 수가 3개 보다 적은 경우가 발생한다. 각 state내의 클러스터의 수가 3개보다 적게 되면, 평균과 covariance를 구해 인식에 사용할 경우 한 state의 covariance가 너무 크게 되어 거의 모든 벡터가 한 state로 수렴해 오히려 인식 결과가 떨어지는 문제가 있다. 이러한 경우를 방지하기 위하여 벡터 분류를 마친 후, 생성된 클러스터의 수가 3개 보다 적으면 다시 segmental K-means 알고리즘을 이용하여 state내의 클러스터의 수가 세 개 이상이 되도록 크게 분류된 클러스터를 한 번 더 분류한다.

㉲ 새로운 평균 벡터와 covariance 구하기
벡터 분류를 한 결과로 만들어진 class로부터 변경된 Gaussian 분포를 구한다. 각 단어의 HMM을 새롭게 구해진 평균 벡터와 공분산으로 변경하여 파일에 저장한다.

다음은 각 저장 파일에 대한 설명이다.
㉮ 학습 데이터
음성 단어는 시간에 따라 순서가 있는 특징 벡터들의 모임이다. 학습 데이터는 같은 단어별로 여러 사람이 발음한 여러 개의 음성을 벡터의 수와 특징 벡터의 형태로 저장되어 있다.

㉯ HMM for each word
HMM은 transition probability(A), observation probability(B), intial probability(π)로 구성된다. 즉, λ=(A, B, π)와 같이 표기하는데 π를 제외한 A, B를 단어마다 저장하고 있다. 또한 벡터 분류를 이용하기 때문에 각 state 마다 클러스터의 수가 다를 수 있기 때문에 각 단어마다 state의 수와 각 state마다 포함된 클러스터의 수, 그리고 각 클러스터 내에 포함된 벡터의 수를 저장하고 있다.

㉰ State 별로 분배된 학습 데이터의 벡터들
Segmental K-means 알고리즘에 의한 training이 끝난 직후의 optimal sequence에 따라 벡터들을 state 별로 저장한다. 각 단어마다 벡터들이 분배되어 저장되어 있으며, Fisher의 판별함수에서 positive data로 사용된다.

㉱ State별로 분배된 오인식 데이터의 백터들
음성 인식 과정에서 오인식된 음성 데이터를 오인식된 결과의 HMM에서 나온 optimal sequence에 따라 분배하여 저장한다. 각 단어마다 벡터들이 분배되어 저장되어 있으며, Fisher의 판별함수에서 negative data로 사용되며, 평균과 covariance를 조정하는데 사용된다.

2. Fisher의 판별하수와 entropy 함수를 이용한 분류(classification)

HMM의 각 state에서는 정확한 벡터와 부정확한 벡터들을 가지고 분류(classification)를 하게 되고 최종적으로 분류된 클래스로부터 Gaussian 분포에 변화가 있게 된다. Gaussian 분포의 변화로 오인식된 패턴은 그 모델에서 나올 확률을 떨어뜨리고, 그 모델은 일반적으로 모델에 속한 단어들에 대한 확률이 향상되는 방향으로 변화가 있게 될 것이다. Classification은 Fisher의
discriminant function과 entropy function을 사용하여 수행하였다. 다음은 이들 함수에 대한 설명이다. [18]

Fisher의 판별함수는 n차원 공간에 분포하고 있는 패턴(X)을 1차원 공간 (W^TX)에 투사(mapping)하고 각 class들이 갖는 투사점들의 분포에 따라 각 class의 패턴 공간을 분류하는 방법을 사용하고 있다.

Fisher는 분류면을 결정하기 위해 우선 최적의 투사면, 즉 적절한 기울기를 갖는 투사면을 고려한다. 이때 기울기는 여러 class 사이를 가장 적절하게 구분할 수 있는 최적의 방향을 의미한다. 이 최적의 기울기를 갖는 투사면을 weight 벡터로 한다. 여기서 최적이라 함은 패턴을 투사면에 투사했을 때 (식 3.1)에서 (식 3,2)과 같이 어느 class를 갖는 투사점의 집합과 다른 class의 투사점들의
집합과의 상호거리를 가장 멀리 떨어지도록 하며, 같은 class 내의 벡터들 사이의 거리는 가깝도록 하는 것을 말한다.

따라서 $V^{-1}B$를 최대로 하는 투사면이 weight가 되는 것이다. 이러한 여러 개의 class를 갖는 패턴을 위한 Fisher의 판별식이 (16)에 나타나 있다.

$$B = \sum (\overline{X_i} - \overline{X})(\overline{X_i} - \overline{X})^T$$

$$V = \sum_i \sum_j (rX_{ij} - \overline{X_i})(X_{ij} - \overline{X_i})^T \qquad \text{(식 3.1)}$$

$$\frac{W^T B W}{W^T V W} = \frac{W^T \left[\sum (\overline{X_i} - \overline{X})(\overline{X_i} - \overline{X})^T \right] W}{W^T \left[\sum_i \sum_j (X_{ij} - \overline{X_i})(X_{ij} - \overline{X_i})^T \right] W} \qquad \text{(식 3.2)}$$

$V^{-1}B$를 최대로 하는 값을 위해 확장된 Caushy-Schwartz 부등식을 사용하는데 이 식의 형식에 맞도록 $a = V^{1/2}W$ 라 하면 (식 3,2)은 $\frac{a^T V^{-1/2} B V^{-1/2} a}{a^T a}$ 이 된다.

$\underset{a \neq 0}{MAX} \frac{\alpha^T Y \alpha}{\alpha^T \alpha} = \lambda_1, a = e_1$ 일 때 Y는 최대값을 갖는다.

우선 $V^{-1/2} B V^{-1/2}$의 고유값들 중 λ_1, e_1을 구한다. 위 부등식에 의해 $a = e_1$ 일 때 $V^{-1}B$는 최대값을 갖게 된다. 따라서

$e_1 = V^{1/2}W$ 가 되어 weight(W)= $V^{-1/2} e_1$으로 구한다. 이 값을 쉽게 구하기 위해 고유값 벡터를 재구성하는 spectral
decomposition을 적용한다.

[Spectral Decomposition]

$P = [e_1, e_2, e_3, ..., e_k]$ 인 $K^* K$ 벡터

$\Lambda = \lambda_1, \lambda_2, \lambda_3, \cdots, \lambda_k$의 요소를 갖는 K*K 단위 벡터

$$V^{-1} = \sum_{i=1}^{k} \frac{1}{\lambda_i} e_i e_i^T = P \Lambda^{-1} P^T$$

$$V^{-\frac{1}{2}} = \sum_{i=1}^{k} \frac{1}{\lambda_i^{1/2}} e_i e_i^T = P \Lambda^{1/2} P^T \qquad \text{(식 3.3)}$$

W를 구하고 난 다음에 입력 패턴 각각을 투사면 $(W^T X)$에 투사하고 투사 방향에서 분리면을 결정하는 threshold 값을 계산한다.

이 값은 불확실성의 정도(measure of uncertainty)를 나타내는 (식 3.4)과 같은 엔트로피 식을 사용하여 구한다.

$H(C\delta^d = PL^* \cdot H(P_1) + PR^* \cdot H(P_2)$ (식 3.4)

여기서 d 는 투사면상의 투사점들을 순서대로 정렬했을 때, 각 투사점들 사이의 평균, $d_k = (W^T X_{k-1} + W^T X_k)/2, k = 1,2,3,\cdots,$(패턴의 개수-1)을 말한다.

$$PL^* = \frac{n_1}{n_1 + n_2} \qquad PR^* = \frac{n_2}{n_1 + n_2}$$ (식 3.5)

$n_1(n_2) : d$ 를 기준으로 왼쪽(오른쪽) 영역에 속한 패턴의 개수

$$H(P_i) = -\sum P_{ij} \log_2 P_{ij}$$ (식 3.6)

i=1, 2 j=1,…, 부류의 수

X_j =각 영역(왼쪽, 오른쪽)에서 각 class j의 패턴개수

$P_{ij} = X_i/n_i$

투사면에 투사된 각 입력 패턴의 점들은 특징 분포를 갖게 되는데 이들 중 작은 값부터 순서에 따라 $d_1(W^T X_1)$부터

$d_{n-1}(W^T X_{n-1})$까지 d 를 기준으로 입력 패턴을 왼쪽과 오른쪽으로 나누고 (식 3.4)을 이용해 최소의 엔트로피 값을 갖는 위치를

찾아 이 위치 값을 threshold 값으로 한다.

Fisher의 판별함수를 이용하여 입력된 벡터들을 분류하게 되면 분류된 클래스들이 Gaussian 분포를 가진다는 보장을 할 수 없지만 본 논문에서는 Fisher의 판별함수를 사용하는 목적이 학습 데이터의 분포를 경쟁이 되는 데이터를 이용하여 세밀하게 분류하고자 하는 것이었기 때문에 분류된 클래스들은 Gaussian 분포를 가진다고 가정하였다.

제3절 실험 및 결과

이 절에서는 HMM의 성능을 개선시킨 학습 과정을 은행의 폰뱅킹 서비스 단어에 적용한 실험 결과를 나타내고 분석해 본다. 이 장음 음성 데이터와 특징 벡터에 대해 기술하고, 실험 과정과 실험 결과를 보여 준다. 그리고, 실험 결과를 통한 학습에 미치는 의미와 개선 방향을 제시한다.

1. Speech data와 HMM 구조

이 실험에서는 남자가 발음한 서비스 단어를 음성인식 데이터로 사용하였다. 연령은 대체적으로 20대에 분포되어 있으며, 서비스 단어는 전화선을 통하여 자동적으로 녹음할 수 있는 시스템에서 채집하였으며, 음성신호는 8khz로 샘플링하고 16bit로 정량화하여 녹음하였다. 각 서비스에 대한 단어를 하나씩 나누어 저장하는 작업(segmentation)은 대략적인 위치는 사람이 수동으로 작업하고, 세밀한 부분은 자동으로 start point와 end point를 나누도록 하였다. 특징 벡터로는 cepstrum이 사용되었고, cepstrum은 12차원
cepstral coefficient, 12 차원의 delta cepstral coefficient, normalized power, delta power로 구성된 26차원 벡터가 사용되었다. 음성 패턴은 짧은 서비스의 경우 20~30개 가량의 frame으로 구성되어 있고, 단어길이가 긴 서비스의 경우 70~90개 가량의
frame으로 구성되어 있다.

초기 HMM의 state수는 서비스 단어의 길이에 따라 4개에서 9개 사이로 구성되어 있고, 각 state는 세 개의 Gaussian 분포로 구성되어 있다. 제안된 학습과정에 의하면 state의 수는 변하지 않으나, 각 state의 Gaussian 분포는 가변적이다. 초기 HMM 학습 방법으로는 segmental K-means 알고리즘이 사용되었고, 인식을 위한 확률 계산은 Viterbi 알고리즘을 사용하여 계산하였다.

2. 실험 과정

음성 인식의 학습 데이터는 남자가 발음한 서비스 단어를 사용하였다. 각 서비스는 약 70개에서 140개의 단어로 구성되어 있다. 테스트 데이터는 두 종류를 사용하였다. 초기 학습 데이터에 대한 인식률을 비교하기 위하여 학습 데이터를 그대로 테스트 데이터로 사용하여 기존의 HMM에서의 인식률과 제안된 방법을 사용한 경우의 인식률을 비교하였으며, 지역과 연령 분포가 고르도록 데이터를 채집하여 학습에 사용되지 않은 데이터를 사용한 경우의 인식률을 비교하였다. 테스트 데이터는 채집이 어려워 세 가지 서비스에 대해서만 성능 분석을 하였다.

서비스 단어는 예금(serv1.ftrs), 잔액(serv2.ftrs), 거래명세(serv3.ftrs), 조회(serv4.ftrs), BC 카드(serv5.ftrs), 사용내역(serv6.
ftrs), 타행(serv7.ftrs), 계좌(serv8.ftrs), 당행(serv9.ftrs), 송금(serv10.ftrs)의 10개 단어로 구분되어 저장되어 있다. 테스트 데이터는 잔액, 조회, 계좌 단어에 대해서만 채집하여 사용하였다.

학습 과정은 우선 segmental K-means 알고리즘을 이용하여 HMM을 구하고, 이 과정에서 학습에 사용된 데이터를 classification에 사용하기 위하여 각 state별로 파일에 저장하였다. 테스트 데이터에 대한 인식을 통하여 오인식된 데이터를 추출하여, 각 state별로 파일에 저장하였다. 테스트 데이터에 대한 인식을 통하여 오인식된 데이터를 추출하여, 각 state별로 classification에서 사용될 경쟁 데이터로 모아 두었다.

Fisher의 판별함수를 이용하여 분류(classification)를 한 다음 Gaussian 분포를 갱신하고, 테스트 데이터에 대한 인식률을 조사하였다. 두 종류의 특징 벡터를 분류하는 방법은 Gaussian 분포의 개수를 제한하는 방법이 있고, 구해진 Gaussian 분포의 수를 그대로 사용하는 방법 등이 있다. 실제적인 구현에서는 최대로 생성될 수 있는 Gaussian 분포의 수를 5개로 제한하였으며, 2개 이하로 발생하는 state는 segmental K-means 알고리즘을 이용하여 최저 3개 이상이 되도록 하였다. 결과적으로 Gaussian 분포는 최저 3개 이상 최대 5개까지로 제한되었다.

Fisher의 판별함수에서 생성되는 클래스의 수를 5개 이상으로 하게 되면 각 state의 클래스마다 분포하는 벡터의 수가 적어 normal 분포를 나타내기 어렵기 때문에 적당한 normal 분포를 나타내도록 하기 위하여 한 state내의 클래스의 수를 5개 이하로 제한하였다. 또한 한 state내에 클래스의 수가 3개보다 적게되면 각 클래스의 분산이 너무 커지므로 인식할 경우 거의 모든 벡터들이 한 state에 떨어지는 확률이 높아지므로 인식률이 크게 떨어지게 된다.

3. 실험 결과

실험은 두 종류로 나누어서 진행하였다. 첫째, 학습에 사용된 데이터를 사용하여 기존의 HMM과 제안한 방법에서의 인식된 결과를 그림 13과 그림 14에서 보여주고 있다. 표 1은 기존의 HMM을 사용하여 학습 데이터를 인식한 결과이고, 표 2는 제안한 방법을 이용하여 학습 데이터를 인식한 결과이다.

표 3.1 기존의 HMM을 사용하여 학습데이터를 인식한 결과

	인식된 서비스										인식률
	Serv1	Serv2	Serv3	Serv4	Serv5	Serv6	Serv7	Serv8	Serv9	Serv10	
Serv1	75	0	0	0	0	0	0	0	0	0	100.0%
Serv2	0	75	0	0	0	0	0	0	0	0	100.0%
Serv3	0	0	73	0	0	0	0	0	0	0	100.0%
Serv4	3	4	0	160	4	3	1	1	0	0	90.9%
Serv5	0	0	0	0	74	0	0	0	0	0	100.0%
Serv6	0	0	1	0	0	73	0	0	0	0	98.6%
Serv7	0	0	0	0	0	0	70	0	2	0	97.2%
Serv8	0	0	0	0	7	0	0	138	0	0	95.2%
Serv9	0	0	0	0	0	0	1	0	70	0	98.6%
Serv10	0	0	0	0	0	0	0	0	0	132	100.0%

표 3.2 제안한 학습 과정을 사용하여 학습데이터를 인식한 결과

	인식된 서비스										인식률
	Serv1	Serv2	Serv3	Serv4	Serv5	Serv6	Serv7	Serv8	Serv9	Serv10	
Serv1	75	0	0	0	0	0	0	0	0	0	100.0%
Serv2	0	75	0	0	0	0	0	0	0	0	100.0%
Serv3	5	0	73	0	0	0	0	0	0	0	100.0%
Serv4	0	5	0	161	1	2	1	1	0	0	91.5%
Serv5	0	0	0	0	74	0	0	0	0	0	100.0%
Serv6	0	0	1	0	0	73	0	0	0	0	98.6%
Serv7	0	0	0	0	0	0	71	0	1	0	98.6%
Serv8	4	0	0	0	0	0	0	141	0	0	97.2%
Serv9	0	0	0	0	0	0	1	0	70	0	98.6%
Serv10	0	0	0	0	0	0	0	0	0	132	100.0%

표 3.1, 표 3.2는 학습 데이터에 대한 인식 결과를 단어별로 개수를 적어 놓은 것이다. 세로축은 학습 데이터를 나타내고, 가로축은 인식된 결과를 나타낸다.

위의 표 3.1, 표 3.2에서 알 수 있듯이 학습 데이터를 이용하여 인식률을 비교해 본 결과 제안한 학습 과정이 인식률을 높이는 효과가 있음을 볼 수 있다.

표 3.3은 기존의 HMM을 사용하여 테스트 데이터를 인식한 결과이고, 표 4는 제안한 방법을 이용하여 테스트 데이터를 인식한 결과이다. 세 개의 서비스를 테스트 데이터로 사용하였다.

표 3.3, 표 3.4에서 보여 주듯이 테스트 데이터에 대한 인식 결과에서도 기존의 HMM을 이용한 학습보다 제안한 방법으로 학습한 경우 인식률이 향상된 것을 알 수 있다.

표 3.3 기존의 HMM을 이용하여 테스트 데이터를 인식한 결과

	인식된 서비스										인식률
	Serv1	Serv2	Serv3	Serv4	Serv5	Serv6	Serv7	Serv8	Serv9	Serv10	
Serv2	0	84	1	0	6	2	4	0	6	0	81.6%
Serv4	5	7	2	184	2	0	3	2	0	0	89.7%
Serv8	2	0	1	0	30	0	0	68	0	0	67.3%

표 3.4 제안한 방법을 사용하여 테스트 데이터를 인식한 결과

	인식된 서비스										인식률
	Serv1	Serv2	Serv3	Serv4	Serv5	Serv6	Serv7	Serv8	Serv9	Serv10	
Serv2	0	84	1	0	6	2	4	0	6	0	81.6%
Serv4	5	5	1	188	2	0	2	2	0	0	91.7%
Serv8	6	0	0	0	25	0	0	70	0	0	69.3%

제 4 장 Kernel-based PCA를 이용한 DTW

제1절 개 요

Principal component analysis(PCA)는 고차원의 data set으로부터 structure를 추출하는 강력한 기술이다. 이것은
eigenvalue problem을 풀거나, principal component를 추정하는 iterative 알고리즘을 사용함으로써 수행된다. PCA는 우리의
데이터를 묘사하는 좌표계의 orthogonal transformation이다. 그 데이터를 나타냄으로써 얻어지는 새로운 좌표값을 principal
components라 부른다. Principal components의 적은 수로 데이터의 대부분 구조를 설명할 수 있는데, 이것을 그 데이터의
factors 또는 latent variables라고 한다.

우리는 input space의 principal components보다 input variables에 nonlinearly 관계된 variables 또는 features의 principal components를 다룰 것이다. 이러한 variables은 input variables중에서 임의의 좀 더 높은 차원의 상관관계에서 얻어진다.

Input space에서 kernel function을 이용해서 feature space에서 dot products를 계산한다. Dot products라는 용어로 표현될 수
있는 어떠한 알고리즘이 주어졌을 때, 이 kernel method는 그것의 nonlinear version을 만들 수 있게 한다.

다음절에서, 일반적인 PCA 알고리즘에 대해 검토한다. 그것을 nonlinear 경우에도 일반화가 가능하게 하기 위해서, 배타적으로 dot
products를 사용하는 방법으로 그것을 형식화한다. 또한, feature space에서 dot product를 계산하기 위한 커널메소드를 논의하고, nonlinear PCA에 대한 제안된 kernel-based 알고리즘을 제시한다.

제2절 Kernel-based PCA

1. PCA in Feature Spaces

centered observations인 χ_k $k=1,....,M$, $\chi_k \in R^N$, $\sum_{k=1}^{N} \chi_k = 0$ 의 집합이 주어졌을 때, PCA는 covariance matrix를 diagonalize한다.

$$C = \frac{1}{M}\sum_{j=1}^{M} \chi_j \chi_j^T \qquad \text{(식 4.1)}$$

이것을 풀기 위해서는, eigenvalues $\lambda \geq 0$ 이고 $v \in R^N \setminus \{0\}$에 대해 eigenvalue equation을 풀어야 한다.

$\lambda_v = C_v$ (식 4.2)

$C_v = \frac{1}{M}\sum_{j=1}^{M} (x_j \cdot v) x_j$ 이기 때문에, $\lambda \neq 0$에 대해 모든 solutions v는 $x_1,..., x_M$의 span위에 놓여 있어야 한다. 그러한 경우에 식 4.2는 다음과 동일하다.

$\lambda (x_k \cdot v) = (x_k \cdot Cv)$ 모든 k=1, ..., M에 대해 (식 4.3)

이 section의 마지막부분에서, 우리는 또 다른 dot product space F (F는 가능한 nonlinear map에 의해 input space에 연관되어 있다.)에서의 동일한 계산을 설명한다.

$\Phi : R^N \rightarrow F, x^1 \rightarrow X$ (식 4.4)

feature space로써 참조되는 F는 임의의 크고 가능한 무한한 차원을 가질 수 있다는 것을 명심해야 한다. 여기와 다음에 나오는 대문자는 F의 elements로써 사용되었고, 소문자는 R의 elements로써 쓰였다.

앞으로 돌아가서, 우리는 centered data, $\sum_{k=1}^{M} \Phi(X_k) = 0$라고 가정하자. F에서 covariance matrix를 사용해서,

$$\hat{C} = \frac{1}{M} \sum_{j=1}^{M} \Phi(x_j)\Phi(x_j)^{\top} \qquad \text{(식 4.5)}$$

(만약, F가 무한한 차원이면, 우리는 $\Phi(x_j)\Phi(x_j)^{\top}$를 X∈F를 $\Phi(x_j)(\Phi(x_j) \cdot X)$로 매핑하는 linear operator로써 생각한다.) 우리는 다음의 식을 만족하는 λ≥0 인 eigenvalues와, V∈F ₩ {0} 인 eigenvectors를 찾아야 한다.

$$\lambda V = \hat{C}V \qquad \text{(식 4.6)}$$

마찬가지로, λ≠0에 대해 모든 solutions V는 $\Phi(x_1), ..., \Phi(x_M)$의 span 위에 놓여있다. 이것은 두 가지 유용한 결과를 가진다. 첫째, 우리는 식들의 집합을 생각할 수 있다.

$$\lambda(\Phi x_k \cdot V) = (\Phi x_k \cdot \hat{C}V) \text{ 모든 k=1, ..., M에 대해} \qquad \text{(식 4.7)}$$

그리고, 둘째로 다음을 식에서의 $\alpha_i (i = 1, ..., M)$ 계수들이 존재한다.

$$V = \sum_{i=1}^{M} \alpha_i \Phi(x_i) \qquad \text{(식 4.8)}$$

식 4.7과 식 4.8을 조합하면, 다음의 식을 얻을 수 있다.

$$\lambda \sum_{i=1}^{M} \alpha_i (\Phi(x_k) \cdot \Phi(x_i)) = \frac{1}{M} \sum_{i=1}^{M} \alpha_i (\Phi(x_k) \cdot \sum_{j=1}^{M} \Phi(x_j))(\Phi(x_j) \cdot \Phi(x_i))$$

모든 k=1, ..., M에 대해 (식 4.9)

M×M matrix K를 다음과 같이 정의되고,

$$K_{ij} := (\Phi(x_i) \cdot \Phi(x_j)), \quad \text{(식 4.10)}$$

다음과 같이 다시 쓸 수 있다.

$$M\lambda K\alpha = K^2\alpha, \quad \text{(식 4.11)}$$

여기서 α는 entries $\alpha_1, \ldots, \alpha_M$의 column vector를 표시한다. (식 4.11)의 해를 찾기 위해, nonzero eigenvalues에 대해 다음의 eigenvalue 문제를 풀어야 한다.

$$M\lambda\alpha = K\alpha \quad \text{(식 4.12)}$$

$\lambda_1 \leqq \lambda_2 \ldots \leqq_M$를 K의 eigenvalues라고 표시하고, α^1, ..., α^M는 처음의로 nonzero eigenvalue인 λ_p(Φ≠0라고 가정하고)를 갖는 대응되는 eigenvectors의 완전한 집합을 표시한다. 우리는 F에서 대응되는 vectors를 normalized함으로써, α^1, ..., α^M를 normalize한다. 즉,

$(V^k \cdot V^k) = 1$ k=p, ... , M 인 모든 K에 대해 (식 4.13)

(식 4.8)과 (식 4.12)의 성질에 의해, 이것은 α^p, ... , α^M를 위한 normalization 조건으로 바꾼다.

$$1 = \sum_{i,j=1}^{M} \alpha_i^k \alpha_j^k (\Phi(x_i) \cdot \Phi(x_j)) = \sum_{i,j=1}^{M} \alpha_i^k \alpha_j^k K_{i,j}$$
$$= (a^K \cdot Ka^k) = \lambda_k (a^k \cdot a^k) \qquad \text{(식 4.14)}$$

Principal component extraction을 위해, 우리는 F (k=p, ..., M)에서 eigenvectors V^k 위로의 projections을 계산해야 한다. F에서의 이미지 Φ(x)인 테스트 포인트를 x라고 하면,

$$(V^k \cdot \Phi(x)) = \sum_{i=1}^{M} \alpha_i^k (\Phi(x_i) \cdot \Phi(x)) \qquad \text{(식 4.15)}$$

는 Φ에 대응되는 nonlinear principal components로 매핑 될 수 있다.

요약하면, 다음의 과정들은 principal components를 계산하기 위해 필요하다. (1) matrix K를 계산, (2) 그것의 eigenvectors를 계산하고 F에서의 그것들을 normalize, (3) 하나의 테스트 포인트를 eigenvectors위로의 projection을 계산한다.

간단히 하기 위해, 우리는 data가 centered 돼 있다고 가정한다. 이것은 input space에서 얻기가 쉽지만, F에서 명시적으로 $\Phi(x_i)$의 평균을 계산할 수가 없기 때문에, F 에서는 좀 더 어렵다. 하지만, 그것을 하기 위한 방법이 있고, 이것은 kernel-based PCA를 위한 약간의 수정된 식으로 이끈다.

Φ map의 역할을 살펴보기 전에, 다음의 관찰은 중요하다. Φ가 가능한 고차원의 space F로의 임의의 nonlinear map이 될 수 있어야 한다. 그러한 경우에, Φ에 의해 map 되는 input vectors의 dot products를 계산할 때, 엄청나게 많은 계산량이 필요하게 된다. 다음 section에 소개된, 이 문제에 대한 solution은 매핑된 patterns에서 dot products만 계산하면 되고, 매핑된 patterns은 필요하지 않는다는 사실 위에 기초한다.

2. Computing Dot Products in Features Spaces

$(\Phi(x) \cdot \Phi(y))$ 의 dot products를 계산하기 위해, 우리는 kernel 표현을 사용한다.

$$k(x,y) = (\Phi(x) \cdot \Phi(x)) \qquad \text{(식 4.16)}$$

이것은 Φmap을 수행하지 않고, F에서 dot products의 값을 계산할 수 있게한다. 이 메소드는 Vapnik과 Chervonenkis의 Generalized Portrait hyperplane classifier를 nonlinear support vector machines으로 확장하기 위해 Boser에 의해 사용되었다. 이것을 위해, 다음의 decision functions를 갖는 dot products의 모든 가능성에 대신, 이전의 선택된 kernel functions k로 대체하였다.

$$f(x) = sgn\left(\sum_{i=1}^{l} v_i k(x, x_i) + b\right) \qquad \text{(식 4.17)}$$

Aizerman et al. 는 F를 linearizatin space라고 명명하고, input space에서의 elements로 F의 elements에서 dot product를 표현하기 위한 classification method로 사용하였다. 만약 F가 고차원이라면, 우리는 효과적으로 계산될 수 있는 k를 위한 closed-form을 찾기를 원할 것이다. Aizerman at al.는 mapping Φ를 직접적으로 F로 대응하지 않는 미리 정해진 k의 가능성을 고려하였다. 그러면 특정 k는 적당한 Φ로 매핑된 patterns에서의 dot product로 대응 될 수 있다. polynomial approximation에서 Poggio에 의해 증명된 결과의 일반화인 유용한 예는 다음과 같다.

$$(x \cdot y)^d = \left(\sum_{j=1}^{N} x_j \cdot y_j\right)^d$$

$$= \sum_{j_1,\ldots,j_d=1}^{N} x_{j_1},\ldots,x_{j_d} \cdot y_{j_1},\ldots,y_{j_d} = (C_d(x) \cdot C_d(y)), \qquad \text{(식 4.18)}$$

여기서, C_d는 x를 vector $C_d(x)$로 매핑하는데, 이것의 entries는 x entries의 d차 products이다. 예를 들어, 만약 $x=(x_1, x_2)$ 이면,
$C_2(x) = (x_1^2, x_2^2, x_1x_2, x_2, x_2x_{1)}$ 이거나, dot product의 동일한 값을 만든다.

$$\Phi_2(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \qquad \text{(식 4.19)}$$

이번 예에서,
$(x \cdot y)^2 = (x_1^2, x_2^2, \sqrt{2}x_1x_2)(y_1^2, y_2^2, \sqrt{2}y_1y_2)\top = (\Phi_2(x) \cdot \Phi_2(x))$
는 쉽게 증명될 수 있다. 일반적으로, 함수

$$k(x,y) = (x \cdot y)^d \qquad \text{(식 4.20)}$$

는 input 좌표 계의 d차 단항식의 space에서 dot products와 대응된다. 만약 x가 pixel 값을 갖는 entries로 이미지를 표현한다면, dot products만으로 모든 것을 할 수 있다는 가정하에, 우리는 매핑된 pattern $\Phi_d(x)$를 사용하지 않고 쉽게 어떠한 d pixels의 products에 의해 span된 공간에서도 계산을 할 수 있다.

function k가 어떤 space F에서의 dot product로 대응된다는 일반적인 질문은 Boser et al.(1992)와 Vapnik(1995)에 의해 논의되었다. functional analysis에 대한 Mercer의 이론은 k가 positive integral operator의 연속한 kernel이라면, k가 dot product로써
움직이는 space안으로의 mapping이 존재한다는 것을 내포한다. (식 4.20)외에 radial basis funcions인

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right), \qquad \text{(식 4.21)}$$

와 sigmoid kernels,

$$k(x,y) = \tanh\left(x(x \cdot y)\square\Theta\right), \qquad \text{(식 4.22)}$$

는 support vector machines에서 사용된다. 이러한 다른 kernels는 매우 유사한 정확도를 갖는 support vector algorithm을 가진 polynomial classifiers, radial basis function classifiers, neural networks를 구성하는데 쓰인다. 게다가, 이들 모두는 그
supprot vectors인 training patterns의 작은 수의 subset으로부터 자신의 decision function을 만든다.

우리의 문제에 대한 (식 4.16)의 적용은 간단하다. 우리는 $(\Phi(x) \cdot \Phi(x))$의 모든 가능성의 위해 선택된 kernel function k(x,y)를 사용하면 된다. 그러면 선택된 k는 내부적으로 maping Φ와 feature space F를 결정한다.

제3절 실험 및 결과

위에서 제안한 모델의 성능을 평가하기 위해, 화자 검증을 수행하였다. 실험 과정은 화자가 자신의 암호를 세 번 정도 발음하고, 검증
과정을 수행하기 위해 녹음했던 암호를 두 번 정도 테스트하고, 녹음하지 않았던 단어를 두 번 정도 테스트한다. 자신의 암호를 발음했을 때는 DTW로 distance를 측정했을 때, 작은 값이 나오고 다른 단어를 발음했을 때는 큰 값이 나오게 된다. accept와 reject를 판단하기 위해서 threshold를 자기 자신의 데이터에 대한 DTW 결과에 대해 1.0부터 2.5배까지 0.1씩 증가하면서 측정하였다.

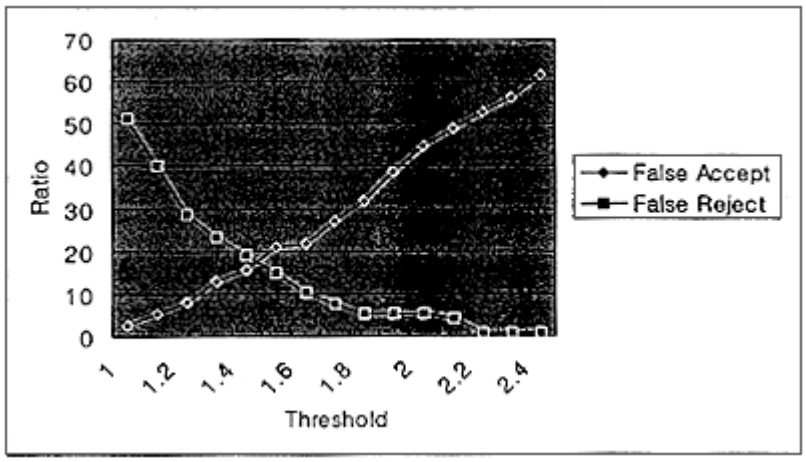
실험은 두 가지 방법으로 행해졌는데, 첫 번째 방법(Kernel PCA)은 한 화자가 3번 발음한 것을 평균하여 kernel-based PCA로 새로운 space에 projection시킨 것에 대해, 자신의 데이터로 distance 구한 것을 threshold로 잡았다. 테스트하기 위해 발음한 데이터를 kernel-based PCA로 새로운 space에 projection한 것을 DTW로 distance를 구함으로써 recognition을 하였다. 두 번째 방법(Kernel PCA from Three Utterances)은 한 화자가 3번 발음한 것을 평균 내지 않고, 각각에 대해 서로 다른 2 번의 발음 한 것에 대해 distance를 비교하여 큰 distance를 threshold로 잡았다. 테스트하기 위해 발음한 데이터를 kernel-based PCA로 새로운
space에 projection한 것을 각각의 3개에 대해 DTW로 distance를 구하고 2번 이상 한쪽(accept 또는 reject)으로 값이 치우친
경우, 치우친 쪽으로 인증을 수행하였다.

실험에 사용된 데이터는 다음의 과정을 통해 추출되었다. 화자검증에 필요한 feature를 추출하기 위해서 입력된 음성으로부터 화자의 고유한 특성을 나타내는 feature를 추출해야 한다. 이 시스템에서는 화자의 특성을 최대한 반영하여 타인과 구별되는 특성을 선택하기 위해 발음된 음성 중에서 유성음, 특히 모음에 해당하는 부분의 frame으로부터 feature를 추출하였다. 유성음 frame에 대한 기준은 평균 SNR(signal for Noise Ratio)과 threshold의 비교를 통해 구했다. feature는 26-dimensional vector로 12-dimensional cepstrum, 12-dimensional delta-cepstrum, power, delta-power로 구성되었다.

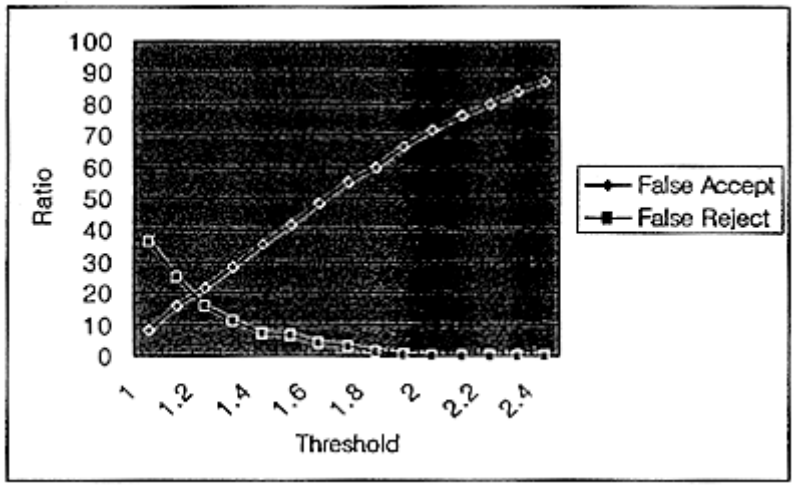
speaker verification test를 하기 위한 input data는 다음과 같이 만들었다. 58명의 화자들이 DTW를 위한 reference template를 만들기 위해, 자신의 암호를 세 번 발음하였고, 테스트를 하기위해 자신의 암호 두 번, 다른 단어 두 번 정도 발음하였다.

false acceptance error rate(FA)과 false rejection error rate(FR) mean verification error rate를 구하기 위해 계산되었다. 최적의 threshold 값은 minimum verification error rate에 의해 결정된다.

다음의 [표 4.1]에서 보는 것처럼, [Kernel PCA]의 minimum verification error rate는 17%로 [Kernel PCA from Three
Utterances]의 19%보다 2% 작게 나온 것을 볼 수 있다. 이때, 최적의 threshold는 자기 자신의 distance의 1.4배한 값이고,
correct ratio는 83%이다.



[그림 23.1] 평균한 데이터로 만든 reference template에 대한 Kernel PCA를 이용한 DTW



[그림 4.2] 개별적인 데이터로 만든 reference template에 대해 kernel PCA를 이용한 DTW

	Kernel PCA	Kernel PCA from Three Utterances
Minimum verification error rate	17%	19%
optimal threshold	자기 자신의 distance의 1.4배	자기 자신의 distance의 1.2배

[표 4.1] 실험 결과

제 5 장 Spectral Subtration

제1절 개요

스펙트럼 차감법(Spectral Subtraction)은 잡음의 효과를 최소화하는 방법으로 주변 잡음에 의해 손상된 음성 스펙트럼에서 잡음 스펙트럼의 크기 성분만을 제거하는 방법이다. 이는 주변 잡음이 음성에 산술적으로 더해지며 잡음과 음성신호 사이에 상관계수가 없다는 가정과 음성을 인지하는 청각의 특성은 음성의 주파수 성분별 위상 정보보다는 크기 정보에 더 많이 영향을 받는다는 연구에 기초한 것이다. 스펙트럼 차감법은 다음과 같은 조건을 만족하는 잡음 환경에서 사용할 수 있다.

(1) 배경잡음의 스펙트럼 형태를 미리 알고 있거나 잡음의 스펙트럼을 추정하기에 충분한 묵음 구간(약 300ms)이 주어져야 한다.
(2) 배경잡음은 최소한 부분적으로 안정(stationary)한 특성을 가져야 하며 통계적 특성이 서서히 변화하는 환경에서는 음성이 존재하는 구간과 잡음만이 존재하는 구간을 검출할 수 있는 방법이 필요하다.

잡음신호 n(k)가 음성신호 s(k)에 더해졌을 때 손상된 음성신호 x(k)는 다음과 같이 나타낼 수 있다.

$$x(k) = s(k) + n(k)$$

이 신호에 윈도우(window)를 취하여 단시간 푸리에 변환을 하면 다음과 같다.

$$X(w) = S(w) + N(w)$$

음성신호의 크기 추정치는 잡음 스펙트럼의 평균을 사용하여 다음과 같이 구해진다.

$$|S(w)| = |X(w)| - \mu(w)$$

여기에서 μ(w) = E{+ N(w)| }을 나타내지만 실제로는 음성이 없는 잡음 구간에서 수 프레임의 데이터로부터 구한 샘플 평균(sample
mean) μ(w)로 대치하여 사용한다.

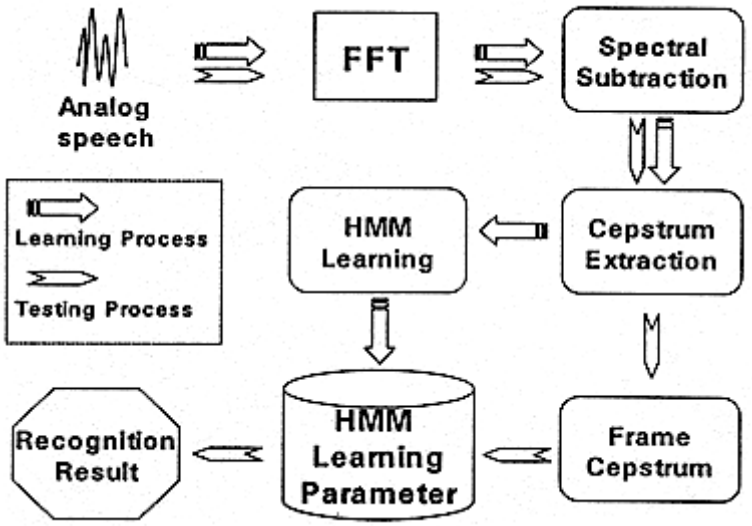
$$\mu(w) = \frac{1}{M}\sum_{i=1}^{M} |N_i(w)|$$

잡음이 제거된 음성 신호의 시간축 파형을 얻기 위해서는 위 식으로 표현된 음성의 크기 정보 이외에 음성의 위상정보가 필요하다.
그러나 주어진 상황에서 위상정보를 얻는 것이 어려우며 청각적인 측면에서의 음질향상이나 음성인식 과정에서 위상 정보가 중요한 변수가 아니라는 사실을 고려하여 음성의 위상정보 $\Theta_s(w)$를 $\Theta_x(w)$로 대치하여 사용한다. 결과적으로 개선된 음성의 주파수 영역에서의 표현은 다음과 같다.

$$S(w) = [|X(w)| - \mu(w)]\exp(-j\theta_x(w))$$

제2절 Spectral Subtraction과 음성인식 시스템

본 장에서 제안하는 spectral subtraction을 이용한 음성인식 시스템은 다음과 같은 구성을 가진다.



[그림 5.1] 음성인식 시스템의 구성도

먼저 전처리 과정에서 FFT 과정을 통해 얻어진 amplitude 값들은 start / end point detection에 사용되는 기본적인 요소로 사용된다. 일반적으로 end point detection의 기준으로 사용되는 energy 값은 전화의 low bit quantization(8 bit)으로 인해 좋은 결과를 얻지 못한다. 이러한 단점은 잡음의 정도에 따라 더욱 심하게 나타난다. 그래서 본 논문에서는 energy에 의한 방법대신 주파수 영역 상에서의 amplitude의 분산을 사용하는 방법으로 대신하였다. 이것은 주파수 영역에서의 noise와 구별되는 음성의 특성 -frequency 상에서의 amplitude가 band에 따라 비교적 뚜렷하게 구별되는 성질-을 이용한 것이다. End point가 결정되면 실제로 음성이 시작되는 프레임의 이전 3 프레임을 기준으로 spectral subtraction을 수행한다. 이후 Noise update 구간에 대한 판별은 다음 식에 의해 결정된다.

$$Noise\,Update\begin{cases} E_d^{new}, & \text{if } P_n \geq \varepsilon \\ E_d^{old}, & otherwise \end{cases}$$

Spectral Subtraction 의 기준이 되는 Noise threshold에 대한 update는 일정한 noise 구간(ε)보다 큰 구간에서만 실행된다. 일반적으로 ε은 3~5 프레임 정도의 길이를 가지며 그렇지 않은 경우에는 이전 noise threshold 값이 계속 사용된다.

위 식에서 E_d^{new}, E_d^{old}, P_n 은 각각 새로운 noise threshold, 이전 noise threshold, 현재 noise threshold 보다 작은 연속된 프레임의 수를 나타낸다.

처음 noise의 threshold는 앞서 지정한 프레임들의 평균으로 결정하고 이후 noise update 구간에서 다음과 같은 식에 의해 갱신된다.

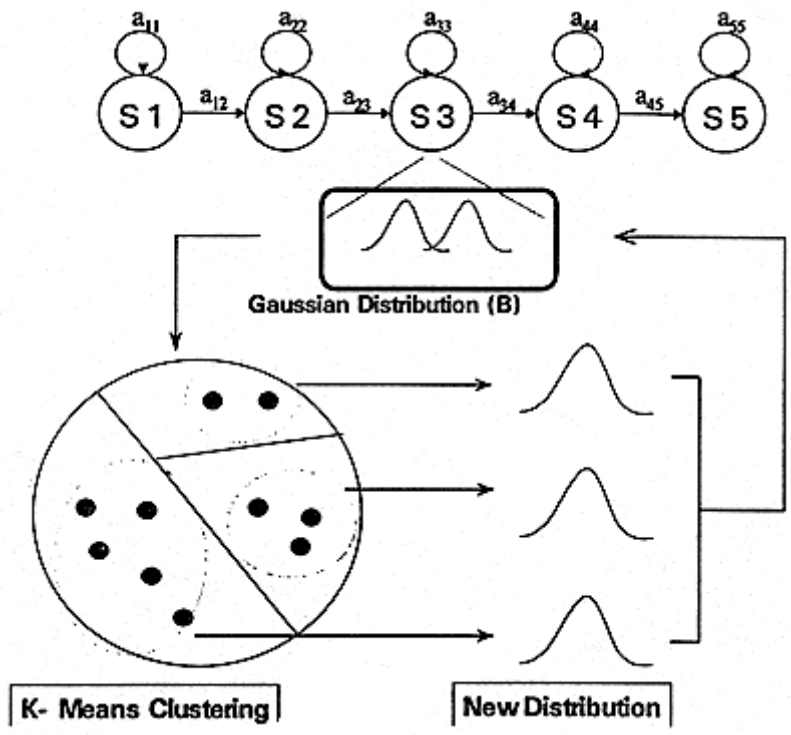
$$E_d^{new} = (1-p) \cdot E_d^{old} + p \cdot E$$

P는 새로운 noise와 이전 noise의 threshold에 대한 적용 비율을 결정하는 요소로 P의 값이 커질수록 새로운 noise 구간의 성질(E)이 많이 반영된다.

위의 식에 의해 End point까지 Noise threshold 값들이 원래 음성의 크기 (amplitude)에서 차감된다. Spectral subtraction 과정을 거친 후 추출되는 mel cepstrum은 HMM을 학습시키는 feature vector로 사용된다.

Feature vector는 12 dimensional cepstrum vector와 12 dimensional delta-cepstrum으로 구성된다.

일반적으로 많이 사용되는 power와 delta-power 값은 전화 채널에 의한 왜곡으로 모델의 학습에 있어 좋은 영향을 주지 못하는
parameter로 나타났다. 그래서 cepstrum과 delta-cepstrum으로 구성된 24 차원의 vector가 feature vector로 사용되었다. 이
프레임 단위 feature vector는 아래 그림과 같이 HMM model의 학습 후에 parameter를 결정하게 된다.



[그림 5.2] HMM 학습 과정

HMM을 구성하는 기본 요소인 상태(state)수는 단어의 음소에 비례하게 결정되고 각 상태는 3개의 cluster로 구성된다. 각 cluster의 구성은 K-Means algorithm에 의해 각 단계마다 clustering되고 viterbi 알고리즘에 의해 최적의 path를 찾고 이를 기반으로
vector를 각 상태의 cluster로 재분배한다. 이러한 clustering과 vector 재분배의 과정을 거치며 HMM의 학습은 진행되고 각각의
단계에서 전이확률(transition probability)이 전 단계와 변동이 없을 때 학습이 완료되었다고 생각하고 종료한다.

HMM을 통해 학습이 종료된 후에 결과를 이루는 파라미터는 다음과 같이 구성된다.
① 모델의 상태수와 각 상태에서의 cluster수
② 각 상태간의 전이확률(Transition Matrix)
③ 각 상태에서 각 cluster에 clustering된 vector의 수
④ 각 상태에서 각 cluster의 가중치

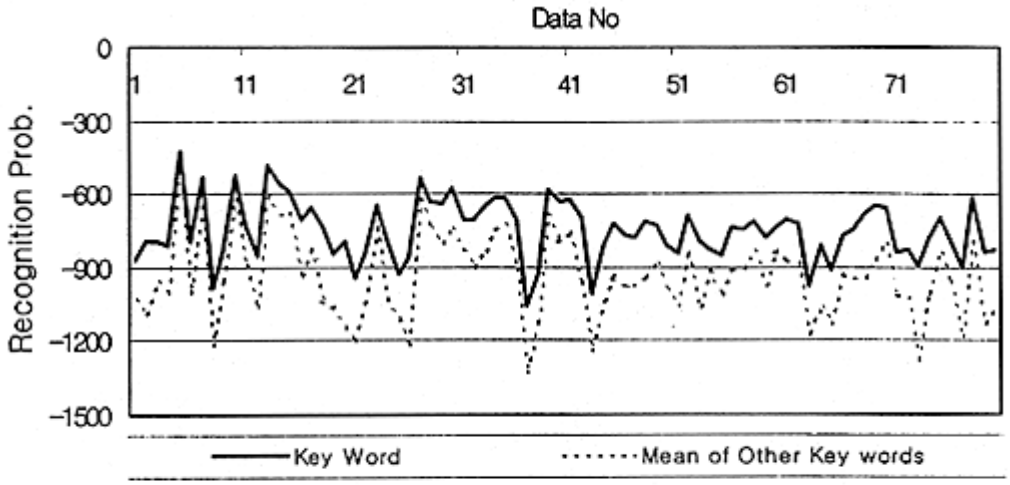
⑤ 각 상태에서 각 cluster의 mean
⑥ 각 상태에서 각 cluster의 variance

학습 종료 후에 저장되는 위의 파라미터들은 단어를 인식하는 과정에서 각 단어 모델을 대표하는 값이 되고 viterbi scoring을 통해 각 단어 모델 중에서 최고의 확률을 가지는 단어 모델을 선택함으로써 인식 결과를 표현한다.

제3절 실험 및 결과

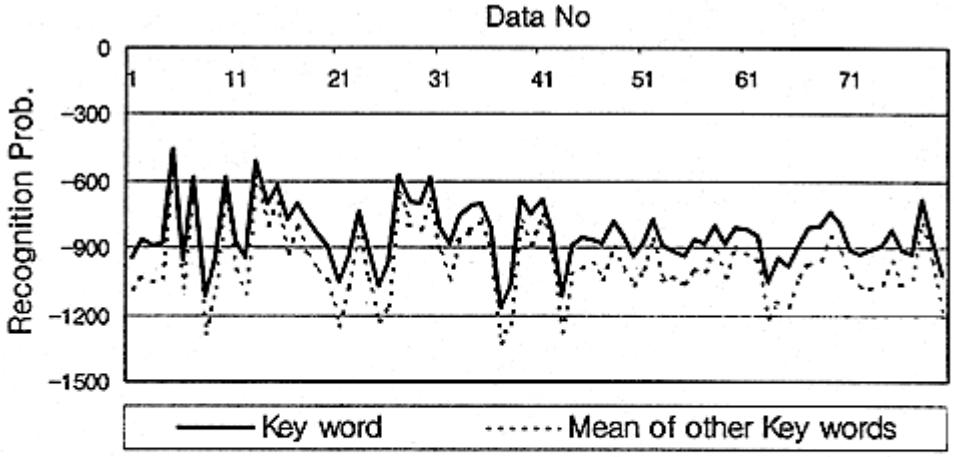
실험에 사용된 데이터는 전화상에서 Dialogic Board를 통해 녹음한 8KHz, 8bit의 wav 파일을 데이터로 사용하였다. 데이터는 텔레뱅킹에 사용되는 5개의 key word -자기앞수표, 대출금이자, 계좌, 조회, 이체 -를 사용하였고 training과 test에 사용된 데이터는 각각 80, 50명의 남성의 목소리로 하였다. 성능 비교를 위해 CMS(Cepstral Mean Subtraction)과 Speatral Subtraction에 의해 추출된 cepstrum을 feature vector로 사용하였다.

다음은 Key Word "자기앞수표“에 CMS를 적용하여 learning한 모델을 learning data 80개에 대해 test한 결과이다. 굵은 선으로 보이는 것이 ”자기앞수표“를 발음한 데이터가 계산된 확률값이고 점선으로 보이는 것이 다른 네 개의 데이터에 대한 평균 확률값이다.



	자기앞수표	대출금이자	조 회	계 좌	이 체	합 계	인식률
인식된 갯수	80	0	0	0	0	80	100%

[그림 5.3] “자기앞수표”에 CMS를 적용한 결과

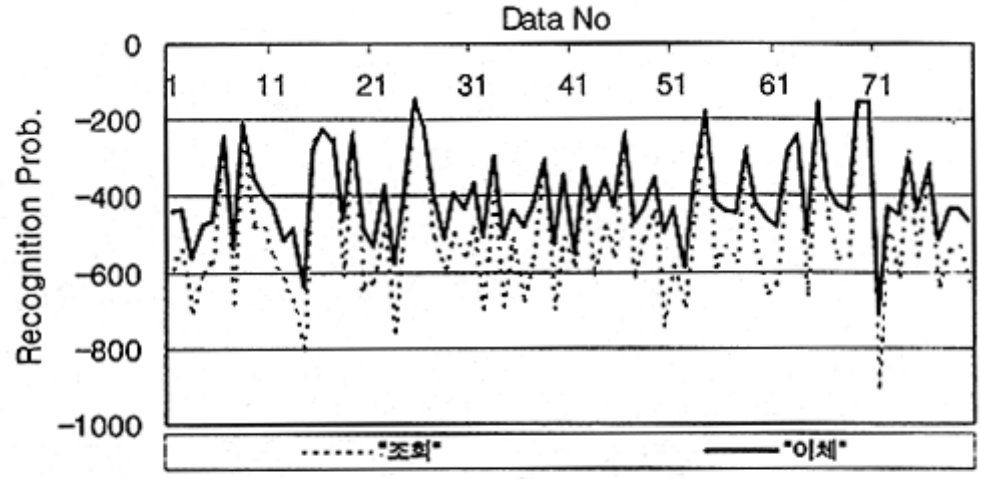


	자기앞수표	대출금이자	조 회	계 좌	이 체	합 계	인식률
인식된 갯수	80	0	0	0	0	80	100%

[그림 5.4] “자기앞수표”에 Spectral Subtraction을 적용한 결과

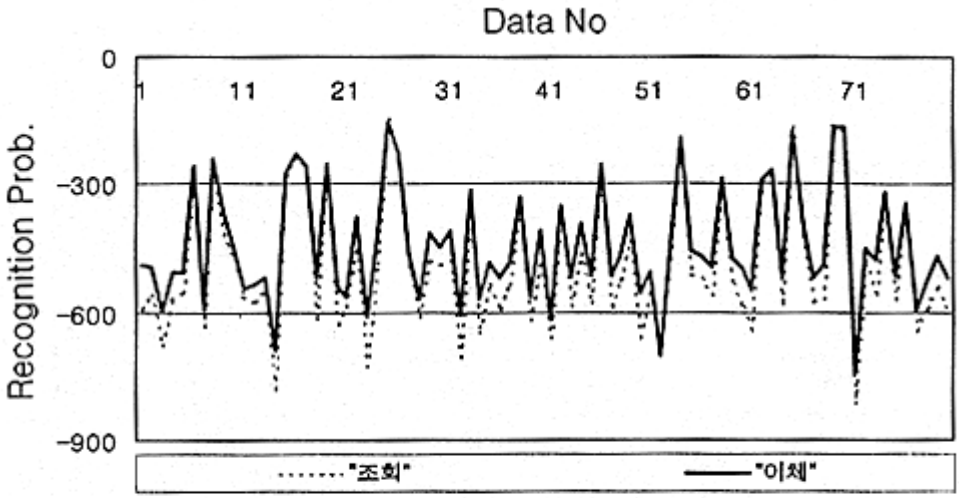
위의 두 그림에서는 CMS와 Spectral Subtraction의 차이가 그리 크지 않다. 다음은 비교적 짧은 Key word인 “이체”에 대한 같은 실험의 결과를 보여준다.

아래의 결과와 위의 결과를 비교하면 차이점을 분명히 알 수 있다. 비교적 긴 단어에 대해서는 인식률에 대해 별 차이가 없지만 상대적으로 짧은 단어일수록 인식률에 대해 뚜렷한 차이가 있음을 알 수 있다.



	자기앞 수표	대출금 이자	조 회	계 좌	이 체	합 계	인식률
인식된 갯수	0	0	9	0	71	80	88.8%

[그림 5.5] “이체” 모델에 CMS를 적용한 결과



	자기앞수표	대출금이자	조 회	계 좌	이 체	합 계	인식률
인식된 갯수	0	0	4	0	76	80	95%

[그림 5.6] “이체” 모델에 Spectral Subtraction을 적용한 결과

다음은 각각 CMS와 Spectral Subtraction을 적용한 Key word 5개 모델에 learning data 80개를 인식시킨 결과이다.

	자기앞수표	대출금이자	조회	계좌	이체	인식률
자기앞수표	80	0	0	0	0	100%
대출금이자	0	80	0	0	0	100%
조회	0	0	80	0	0	100%
계좌	1	0	1	78	0	97.5%
이체	0	0	9	0	71	88.8%

[표 5.1] CMS 적용 후 Learning data에 대한 인식결과

	자기앞수표	대출금이자	조회	계좌	이체	인식률
자기앞수표	80	0	0	0	0	100%
대출금이자	0	80	0	0	0	100%
조회	0	0	79	1	0	100%
계좌	0	0	0	79	1	98.8%
이체	0	0	4	0	76	95%

[표 5.2] Spectral Subtraction 적용 후 Learning data에 대한 인식결과

위의 두 Key word에 대한 결과를 살펴보면 비교적 길이가 긴 단어에 대해서는 인식률에 큰 차이가 없음을 알 수 있다. 이것은 일반적으로 인식대상의 길이는 인식률에 비례한다는 많은 연구결과에 기반한 결과임을 알 수 있다.

위의 결과에서 “계좌”와 “이체”에서 CMS에 비해 Spectral Subtraction이 좀 더 향상된 성능을 나타낸다. “이체”와 같은 경우에는 잘못 인식된 경우가 50%이상 줄어든걸 볼 수 있다. 위의 실험에서는 Spectral Subtraction을 한번 적용한 결과지만 이러한 과정을 반복함으로써 좀 더 향상된 결과를 얻을 수 있다.

다음은 Learning 과정에 사용되지 않은 테스트 데이터 40개에 대한 인식결과를 나타낸다.

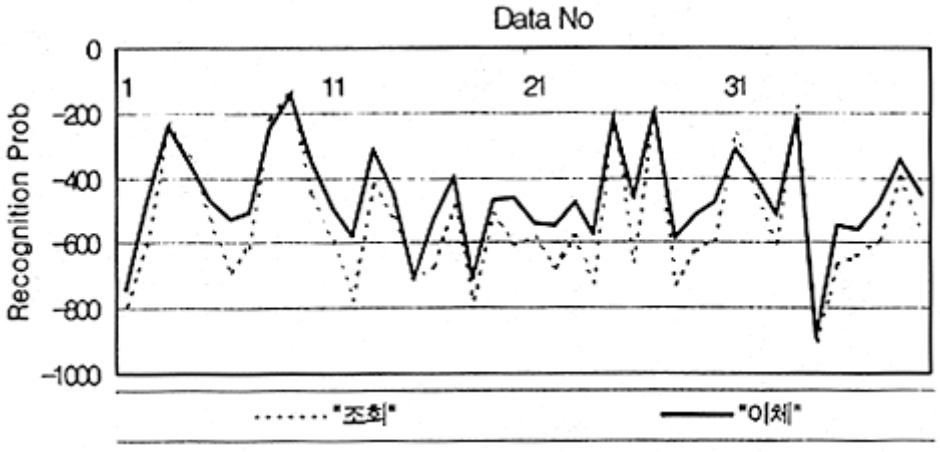
	자기앞수표	대출금이자	조회	계좌	이체	인식률
자기앞수표	40	0	0	0	0	100%
대출금이자	0	40	0	0	0	100%
조회	0	0	39	1	0	97.5%
계좌	1	0	1	38	0	95%
이체	0	1	7	3	29	72.5%

[표 5.3] CMS 적용 후 Test data에 대한 인식결과

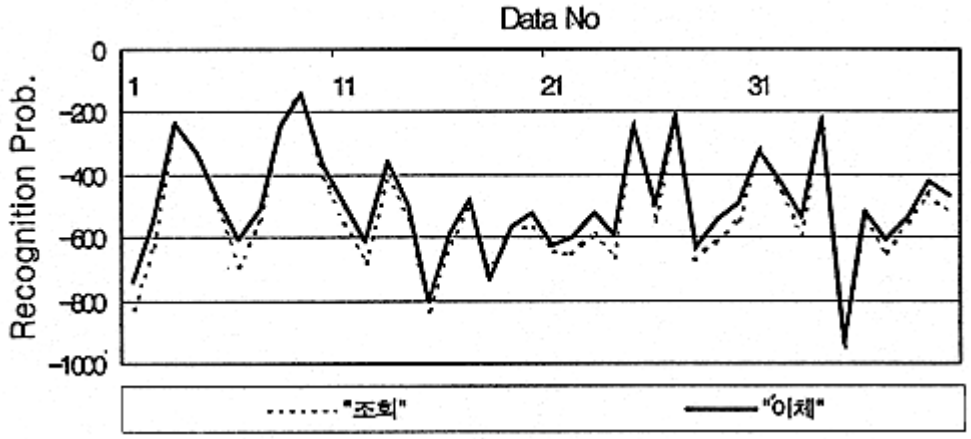
	자기앞수표	대출금이자	조회	계좌	이체	인식률
자기앞수표	40	0	0	0	0	100%
대출금이자	0	40	0	0	0	100%
조회	0	0	39	1	0	97.5%
계좌	0	0	0	49	0	100%
이체	1	0	7	0	32	80%

[표 5.4] Spectral Subtraction 적용 후 Test data에 대한 인식결과

위의 표에서 “계좌”와 “이체”에 대해서는 많은 차이가 있음을 알 수 있다. 다음은 사용된 단어 모델 중 가장 뚜렷한 결과를 보이고 있는 “이체”에 대한 인식결과이다.



[그림 5.7] “이체” 모델에 CMS후 테스트 데이터에 대한 결과



[그림 5.8} “이체” 모델에 Spectral Subtraction 후 테스트 데이터에 대한 결과

이상에서 알아본 바와 같이 Key word의 길이가 짧을수록 CMS에 비해 Spectral Subtraction을 이용한 전처리 과정이 noise 환경하의 음성인식에서 좀 더 강인하다는 결과를 보여준다.

제 6 장 결론

음성인식은 화자의 음성이 전달하는 정보를 추출하여 결정하는 과정이다. HMM(Hidden Markov Model)은 음성인식에서 많이 사용되는 방법으로 우수한 성능과 효율을 보이고 있다. HMM 학습은 각각의 패턴별로 그 패턴에 속하는 학습데이터만을 가지고 확률을 최대화하기 때문에 패턴 상호간의 정보가 전혀 고려되고 있지 않다. Discriminative 학습은 경쟁이 되는 학습 데이터를 이용하여 인식률을 높이는 방법으로 ML(Maximum Likelihood)이나 MAP(Maximum a Posteriori)보다 성능을 향상시킨다.

본 프로젝트에서는 패턴 상호간의 정보를 고려하는 HMM 학습 방법을 제시하였다. 제시한 학습 과정은 discriminant learning의 범주에 속한다. 경쟁이 되는 음성 패턴을 optimal state sequence에 따라 특징 벡터를 분배한 후, 각 state마다 그 모델의 학습 데이터로부터 나온 특징 벡터들과 classification을 하게 되면 새로운 Gaussian 분포를 구할 수 있다. Gaussian 분포는 학습 특징 벡터들로부터 긍정적인 영향을 받고, 경쟁이 되는 특징 벡터들로부터 부정적인 영향을 받는다. 모델의 Gaussian 분포에서 학습 특징 벡터들의 확률은 높아지고 경쟁이 되는 특징 벡터가 나올 확률은 낮아진다. 실험에서 관찰 확률의 분포를 조정함으로써 인식률의 향상을 볼 수 있음이 증명되었다.

HMM의 각 state에서 학습 데이터의 특징 벡터들과 경쟁이 되는 학습 데이터의 특징 벡터들과 경쟁이 되는 학습 데이터의 특징 벡터들을 분류(classification)하는데 Fisher의 판별함수와 entropy 함수를 사용하였다. Fisher의 판별 함수로 최적의 투사면을 구할 수 있고, entropy 함수로 투사점들을 분리하는 threshold 값을 구할 수 있다. 이 방법들은 벡터들을 분류하는데 좋은 성능을 보이고 있음이 잘 알려져 있다.

Fisher의 판별함수와 entropy 함수를 이용하여 분류를 할 경우 분류된 클래스들이 Gaussian 분포를 가진다고 보장할 수 없지만 경쟁이 되는 데이터를 이용함으로써 학습하고자 하는 데이터에 대한 분류를 더 세밀하게 할 수 있기 때문에 Fisher의 판별함수로 분류한 클래스들이 Gaussian분포를 가진다고 가정하였다.

학습 데이터들로부터 올바른 Gaussian 분포에 대한 정보를 알아내는 것이 HMM 학습의 큰 요소가 된다. 그러나, 불행히도 우리는
모든 데이터들을 구할 수 없으므로 관찰된 일부의 데이터들로부터 그 분포를 추정한다. 본 논문에서 제안한 HMM 학습 방법은 경쟁
이 되는 학습 데이터들로부터 Gaussian 분포에 대한 정보를 얻는데 의미가 있다.

초기 Gaussian 분포가 적절히 설정되지 않을 경우 본 프로젝트에서 제안한 학습 방법은 오히려 인식 성능을 떨어뜨리는 역효과가 나타날 수 있다. 따라서 초기 HMM 학습 과정에서 어느 정도 충분한 데이터를 확보하여 Gaussian 분포를 구한 뒤, 본 프로젝트에서 제안한 학습 방법을 사용할 것을 제안한다.

음성인식을 위한 잡음처리는 여러 방법으로 시도되고 있다. 이들 방법은 잡음 및 채널 왜곡에 강인한 음성의 특징 추출, 전처리 과정을 통한 음질 개선 방법, 그리고 모델 기반의 잡음 및 채널 왜곡 보상 방식으로 분류된다.

잡음에 강인한 특징 파라미터 추출 방법으로는 스펙트럼의 동적 특성을 이용한 방법, 선형 판별 분석에 의한 변환 · 방법, 파라미터
필터링과 잡음환경에서 특징 추정 방법 등을 들 수 있고 채널 왜곡에 강인한 특징 추출 방법으로 여러 가지 켑스트럼 평균 정규화법,
RASTA-PLP 방법 등이 효과적이라고 알려져 있다.

전처리 과정을 통한 음질개선 방법으로는 comb 필터링, 위너 필터링, 스펙트럼 사상법, 베이시안 추정법 등이 있고 본 프로젝트에서 제안한 스펙트럼 차감법도 이에 속한다. 모델 보상에 의한 잡음 처리 기법으로 HMM의 상태 의존 필터링, 지속(duration) 모델의 적응 방법 및 HMM 모델의 적응, 최소 오류 훈련 방법 이외도 잡음 마스킹과 학습 데이터의 잡음 음성화 방법 등이 있다.

전처리 과정을 통한 음질 개선 방법으로 본 프로젝트에서 사용된 스펙트럼 차감법(Spectral Subtraction)은 잡음이 주파수 영역에서 크기 성분(amplitude)의 처리에 의해 상당 부분 제거된다는 것을 여러 실험으로 확인해 보았다. 이러한 잡음처리는 깨끗한 음성을 재생할 수 있을 뿐만 아니라 인식에 있어서도 좋은 특징 파라미터를 추출할 수 있다는 사실을 실험을 통해 알아보았다. 특히 비교적 길이가 짧은 단어에 대한 모델을 구축할 때는 CMS에 비해 좀 더 향상된 결과를 보장할 수 있음을 실험을 통해 알아보았다. 또한
스펙트럼 차감법은 단일 단어 인식보다 연속단어 단위인식에 있어 가변적인 잡음을 효율적으로 제어할 수 있어서 오인식률을 상당히 줄일 수 있다. 이것은 가변적인 잡음에 대해 적절한 시기마다 잡음의 threshold를 반영시켜 잡음을 처리하기 때문이다.

본 프로젝트는 전화를 통한 텔레뱅킹 시스템에서 음성인식을 위한 시스템의 구축과 가변 잡음에 대한 효과적인 처리에 기반한 실험이다. 인식하고자 하는 단어에 대해 여러 형태로 모델을 구축할 수 있지만 100단어 미만의 소용량 인식을 위해서는 단어 단위의 모델링이 상대적으로 좋은 인식률을 보였다. 그리고 단어 단위의 모델링은 가변적인 잡음에 대한 잡음 update 구간에 대한 휴지기간을
측정하기에도 수월하기 때문에 좀 더 나은 성능을 보일 수 있었다.

참고문헌

[1] A. J. Viterbi (1967), "Error bounds for conventional codes and as an asymptotically optimal decoding algorithm", IEEE
Trans. Information Theory, IT-13, pp 260-269.

[2] A. M. Noll (1964), "Short-time spectrum and 'cepstrum' Techniques for Vocal-pitch detection", J.Acoust.Soc.Amer.,
36,2, pp 296-302.

[3] Bernhard Scholkopf, Alexander Smola and Klaus-robert muller (1998), "Nonlinear Component Analysis as a Kernel
Eigenvalue Problem", Neural Computation Vol. 10, pp. 1299-1319.

[4] C.S. Liu, C. H. Lee, B. H. Juang and A. E. Rosenberg (1994), "Speaker recognition based on minimum error
discriminative training", ICASSP-94, vol. 1, pp. 325-328.

[5] D. B. Paul (1989), "The Lincoln Robust Continuous Speech Recognizer", Proc.ICASSP89, Glasgow, Scotland,pp 449-452.

[6] E.A.Martin, R.P.Lippmann, and D.B.Paul (1988), "Dynamic Adaptation of Hidden Marcov Models for robust Isolated
Word Recognition", proc. ICASSP-88, pp 52-54, New York, USA.

[7] Jean-Luc Gauvain and Chin-Hui Lee (1994), "Maximum a Posteriori Estimation for multivariate Gaussian Mixture
Observation of Markov Chains", IEEE TRANSACTIONS OF SPEECH AND AUDIO PROCESSING, vol. 2, no. 2.

[8] K. K. Paliwal (1993), "Use of temporal correlation between successive frames in a hidden Markov model based
speech recognizer", ICASSP-93, vol. 2, pp. 215-218.

[9] L. E. Baum (1972), "An inequality and associate maximization technique in statistical estimation for probabilistics
function of Markov processes", Inequalities, vol. 3, pp. 1-8.

[10] L. R. Bahl, M. Padmanabhan, D. Nahamoo, P. S. Gopalakrishnan (1996), "Discriminative training of Gaussian mixture
models for large vocabulary speech recognition systems", ICASSP-96, pp. 613-616.

[11] L. R. Bahl, P. F. Brown, P. V. deSouza, R. L. Mercer (1986), "Maximum mutual information estimation of HMM
parameters for speech recognition", ICASSP-86, pp. 49-52.

[12] L. R. Rabiner (1989), "A tutorial on hidden Markov models and selected applications in speech recognition", Proc.
IEEE, vol. 77, pp. 257-286.

[13] L. R. Rabiner (1993), "Fundamentals of Speech Recognition", pp. 339-341.

[14] L.R. Rabiner and B-H Juang (1993), "Fundamentals of Speech Recognition", Murray Hill, New Yersey, USA.

[15] L.R. Rabiner, B-H Juang, S.E.Levinson, and M.M.sondhi (1985), "Recognition of Isolated Digits Using Hidden Marcov
Models with continuous Mixture Densitiies", AT&T Tech.J., Vol. 64, No. 6, pp 1211-1234.

[16] L. R. Rabiner, J. G. Wilpon and B. H. Juang (1986), "A segmental K-means training procedure for cennected word
recognition", AT&T Tech. J., vol. 64, no. 3, pp.21-40.

[17] L. R. Rabiner, J. G. Wilpon and F. K. soong (1988), "High performance connected digit recognition using hidden
Markov models", in IEEE ICASSP.

[18] M. E. Forsyth and M. A. Jack (1994), "Discriminating semi-continuous HMM for speaker verification", ICASSP-94, vol.
1, pp. 313-316.

[19] Petr Pollak, Pavel Sovka, and Jan Uhlir (1993), "Noise Suppression System for a Car", EUROSPEECH'93.

[20] Petr Pollak, Pavel Sovka, and Jan Uhlir (1995), "Cepstral Speech/Pause Detectors", IEEE NISP'95.

[21] Pavel Sovka, Petr Pollak, and Jan Kybic (1996), "Extended Spectral Subtraction", EUSIPCO'96.

[22] P. V. S. Rao, R. Raveendran (1996), "Training algorithm for a predictive classifier", ICASSP-96, pp. 609-612.

[23] Richard O. Duda and Peter E. Hart (1973), "Pattern Classification and Scene Analysis", Wiley-Interscience.

[24] R. P. Lippmann (1987), "An Introduction to Computing with Neural Nets", IEEE ASSP Mag., 4(2), pp 4-22.

[25] Sadaoki Furui, "Digital speech processing, synthesis, and recognition", MARCEL DEKKER, INC.

[26] T. R. Anderson (1993), "A comparison of auditory model for speaker independent phoneme recognition", ICASSP-93,
pp. II. 231-II. 234.

[27] V. Valtchev. J. J. Odell, P. C. Woodland, and S. J. Young (1996), "Lattice-based discriminative training for large
vocabulary speech recognition", ICASSP-96, pp. 605-508.

[28] Yves Normandin (1995), "Optimal splitting of HMM Gaussian mixture components with MMIE training", ICASSP-95,
pp. 449-452.

[29] 원동호 역 (1994), “정보와 부호이론”, ohm社.

[30] 이종찬 (1996), “패턴분류를 위한 점증적 신경회로망 모델”, 충남대 박사 논문.

부록

- An Algorithm to Find the Optimized Network Structure in an Incremental Learing

- A Call Admission Control Using Interval Arithmetic Coulomb Energy Network

- ATM 망에서 호 연결수락제어를 위한 Interval Arithmetic CEN

An Algorithm to Find the Optimized Network Structure in an Incremental Learning

Jong Chan Lee[i] ,Won Don Lee[2] and Mun-Sung Han[3]
[1] Dept. of AI, ChungWoon Univ., Hongsung, ChungNam, KOREA
jclee@aisun.cwunet.ac.kr
[2] Dept, of Computer Science, ChungNam Nat'l univ., Taejun, KOREA
wdlee@chungnam.ac.kr
[3] Speech Understanding Lab., ETRI, Taejun, KOREA
msh@etri.re.kr

Abstract. In this paper[1] we show a new learning algorithm for pattern classification. A scheme to find a solution to the
problem of incremental learning algorithm is proposed when the structure becomes too complex by noise patterns
included in the learning data set. Our approach for this problem uses a pruning method which terminates the learning
process with a predefined method which terminates the learning process with a predefined criterion. Then an interactive
model with a 3 layer feed-forward structure is derived from the incremental model by appropriate manipulation. Note that
this network is not fully connected between the upper and lower layer. To verify the effectiveness of the pruning method,
the network is retrained by EBP. We test this algorithm by comparing the number of nodes in the network with the
system performance, and the system is shown to be effective.

1. Introduction

Conventional iterative models such as EBP usually have a fixed feedforward network structure and use an algorithm to
gradually modify the weights of networks as learning proceeds. So this approach does not allow to expand the network
during training. This approach sometimes has a critical limitation, depending on the trial and error method or ad hoc
schemes to obtain an appropriate architecture for learning patterns.
Therefore another approach is devised to solve this problem by adding nodes th the network when necessary. This type
of learning is referred to as incremental learning as the network grows as training

1 This research is supported by Brain Science and Engineering Research Program in Korea

occurs. As a procedure, Lee et al.[2]
have proposed an incremental algorithm using Fisher's Linear Discriminant Function(FLDF)[1]. This model searches an
optimal projection plane based on the statistical method for pattern classification. And then, after projecting patterns on
this projection plane, this model starts a search procedure for an optimal hyperplane based on an entropy measure and
thus determines the neuron in the structure.

Lee et al.[3] introduced a neural network learning algorithm which transforms a structure of an incremental model into
that of an iterative model. This model showed that the weights and thresholds as well as the structure of the 3 layer
feedforward neural network can be found systematically by examining the instances statistically. It is well known that a
major part of the learning capability is in the architecture of its models.
In iterative models the approaches to solving this problem ars as follows. Kung et al.[5] proposed a method which is
learning with a network structure with a predefined node number. But this method had a problem in that it converges
less than the theoretical bases. Sietsma and Dow[7] devised an algorithm which assigns many nodes in prior learning
and then removes nodes by making an observation of inactive nodes in learning. Though this algorithm can be applied
to simple problems, when a problem is more complex one encounters many difficulties. Hanson and Pratt[9] proposed
an algorithm which removes hidden nodes with a constraint term in EBP function. But this algorithm has a side effect
which reduces the probability of the convergence. Hagiwara[8] proposed an algorithm which considers a proper node
number and weight value concurrently. But this algorithm needs much time to converge. Moody and Rognvaldsson[10]
proposed adding the complexity-penalty term, but it has much more complicated form and also demands much more
computational complexity. Wangchao et al.[11] introduced the sparselized pruning algorithm for a higher-order neural
network, but it is applied after all the higher-order weights are trained.
The incremental model uses an algorithm trying to produce a neural network with near-optimal architecture intelligently.
But this model has a drawback in that it can be extremely by noises included in patterns. In this paper we propose a
method to solve this problem.

2 Background

2.1 The Incremental Network Model

In this paper we present a pattern by a vector of n components and describe a pattern classifier as a mapping of the
input pattern space, a subset of n dimensional real space, to the set of classes 1,2,....,k.

In order to make the output
decisions we develop the constructs which build internal representations of a class description. For doing this, we
represent one unit as a hyperplane specified by elements (weight vector, threshold value) and then partition the
space as follows.

$$Hyperplane(P) = \{X \mid X \in R^n \text{and } W^t X = T\} \qquad (1)$$

where, X : Input pattern, R^n : Real space, W : Weight, T : Threshold
The hyperplane P separates the space H^n into two sets, P^L and P^R. Thus input X belongs to P^R or P^L by P.

$$P^R = \{X \mid X \in H^n \text{and } W^T X \geqq T\}$$
$$P^L = \{X \mid X \in H^n \text{and } W^T X < T\}$$

The network structure of the model consists of one input unit, a number of hidden units and output units as many as the
number of classes. Each unit has a weight vector and a threshold value. The input X is broadcasted to all units and for
each unit at most one path is activated. Thus, in the whole network one path at most is activated for each input vector.

2.2 The Training Process

During the training phase, a collection of classified patterns describing a desired class is presented to an incrementally
formed network of neurons. And the weight vector and threshold values of units of the network are determined by an
adaptation process. Each unit is assigned to represent a certain hyperplane which is part of the discriminant
hypersurface represented by the network. The training set is presented a number of times and at each presentation the
network is expanded by adding a number of new units. The adaptation process is carried out at each unit independently
of others.
In the adaptation process, the Fisher's linear discriminant function[2] is used in order to determine the optimal
hyperplane. Fisher's linear discriminant function provides the optimal weight for input pattern data for an arbitrary
distribution. Fisher's formula for n classes is shown is (2). The optimality is characterizes by the overall measure
representing the mutual distances between a set of projected points of a class and that of another class and is achieved
by maximizing the overall measure, B, standardized by V. We use Cauchy-Schwartz inequality for obtaining maximum
value of V^{-1}B.

$$\frac{W^T B W}{W^T V W} = \frac{W^T \left[\sum i (\overline{X_i} - \overline{X})(\overline{X_i} - \overline{X})^T \right] W}{W^T \sum i \sum j (X_{ij} - \overline{X_i})(X_{ij} - \overline{X_i})^T W} \qquad (2)$$

Let β = $V^{1/2}$W, then (2) becomes $\frac{\beta^T V^{-1/2} B V^{-1/2} \beta}{\beta^T \beta}$. This formulus attains the highest value when vector β becomes

the eigenvector e_1 which is associated with the highest eigenvalue λ of the matrix $V^{-1/2}BV^{-1/2}$. Thus weight vector(W)
is obtained as $V^{-1/2}e_1$.
The threshold value determining the position of the hyperplane is obtained based on the following entropy function.

$$H(C \mid \delta_d)) = PL^{(*)}H(p_1) + PL^{(*)}H(p_2) \quad \text{d :cursor position} \qquad (3)$$

After projection(W^TX), the projected points are divided into two parts by a dividing plane placed on a cursor position.
After entropy is measured, the plane is moved one at a time from $d_1(W^TX_1)$ to $d_{n-1}(W^TX_{n-1})$. The optimal position
is where the smallest value of entripy is found. Let $n_1(n_2)$ be the number of left(right) region, then $PL^* = n_1/(n_1+n_2)$
and $PR^* = n_2/(n_1+n_2)$. Let x_{ij} be the number of class j events in each region i(left, right). The probability p_{ij} whose

class will be j is x_{ij}/n_i. Then, $H(p_i) = -\sum p_{ij}\log_2 p_{ij}$ i=1,2 j=1,....., class number.

2.3 Translation into an iterative model structure

In this section we present a transforming procedure which converts the incremental network topology into an iterative
one.
Step 1 : The input layer consists of as many nodes as the number of variables (dimensions) in learning patterns.
Step 2 : The first layer has the same number of nodes as that of the hidden nodes except for leaf nodes in the
incremental model. The weight and the threshold berween the input layer and the first layer are fully connected and have
the same values as those of hidden nodes in the incremental model.
Step 3 : The second layer has as many nodes as the number of leaf nodes representing the region of each class in the
incremental model. Each region is made by the intersection of hyperplanes in the first layer, thus this layer is
characterized by AND : when all the inputs are active, the output is active. Weight and threshold between the first layer(j)
and the second layer(i) is determined by each path, from the input node to the leaf node in the incremental model : that is,

$\sigma_{iL} = -1$, $\sigma_{iR} = +1$

$W_{ij}(weight) = \sigma_i D$, $T_i = \sum \mid \sigma_{iD} \mid -0.5$

Here, i=0,...,(the number of discriminated region-1) and $\sigma_i D$(D=L(left space), D=R(right space)) denotes i_{th} path.

Step 4 : In the third layer there are as many nodes as the number of classes. The intersected regions from the second
layer are unioned in each class region. Weight and threshold between the second(j) and the third(i) layer is determined
by the class of each region.

If the region from i_{th} path contains class j : $R_{ij} = 1$

Otherwise : $R_{ij} = 0$

Wij(Weight) = R_{ij}

T_i(Threshold) = $\sum R_{ij} - 0.5$

For further information, please refer to [3].

3 Pruning in the incremental learning Model

3.1 Pruning algorithm for the proposed incremental model

Most of the network structure of the incremental learning algorithm have the shape of the binary trees and hence the
number of the nodes does not need to be predetermined as the structure is determined. But an incremental model has to
solve a new problem, as there can be too many nodes to be added. This is because a learning pattern set can contain
many noisy data. The algorithm above also has this problem as it iterates until there is only one class of data contained
at a divided space. As the algorithm proceeds recursively, not only the computational time but also the memory wastes
are increased. Moreover, the network performance can be degraded because of the noise effect. Therfore, an algorithm
must be devised to overtime these problems. The following formulas are added to the algorithm introduced in the
section 2.2 to solve those problems:

IF(N(●) > PruneRate) Continue the learning procedure.
ELSE Terminate the learning procedure.

where Prune Rate is a criterion for the percentage of the noise. And N(●) for the learning pattern P at node i is
determined as follows:

$$N_i(P) = \frac{(\square TE - \square MCE)}{\square MCE} \times 100.0 \qquad (4)$$

- #TE : Total number of patterns at node i.

- #MCE : The number of patterns of a class with the most number among the #TE patterns.

3.2 Selective-learning algorithm

In the section of 2.3 we introduced a method which transforms the structure of an incremental model into that of a 3 layer
iterative model. This transformed network structure is not a fully connected but partially connected one except for
between the input layer and the hidden layer. In this section we propose a method to reduce the network structure using
an iterative learning algorithm. This algorithm is based on the observation that, in the algorithm explained in the section
3.1, the pruning procedure corresponds to a node reduction, and the partial connection between layers corresponds to
a reduction of weights set size. The network structure of this model consists of nodes with the threshold function and
the sigmoid function. This structure is shown in Fig. 1. We use the EBP algorithm to train the structure. As an initial
structure, the first layer is constructed with the nodes with the same weights and threshold values determined in
incremental learning. And the bias nodes are added to the second hidden layer and the output layer.
The learning procedure of this model is described below. The first hidden layer(O) is activated as follows :

$$IF((\sum w_{ij}X_j) > T_i)\, O_i = 1$$

$$ELSE\ \ O_i = 0$$

An EBP learning is performed on the upper layer using the output from the first layer. As our model is partially connected,
the EBP is done according to the following :

$$\triangle_p w_{ij}(n+1) = C_{ji}(\eta\delta_{pj}O_{ip} + \alpha\triangle_p w_{ji}(n)) \qquad (5)$$

where C_{ji} is 1 if there is a connection between node j in the upper layer and node i in the lower layer, and 0 otherwise.

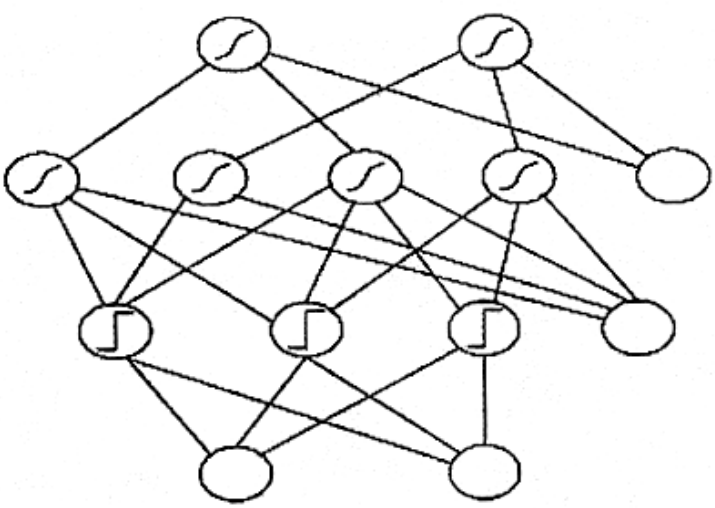


Fig. 1. The network structure of the selective learning model

4 Implementation]

To test our system, we use sleep stage scoring data sets and the speech data used by Person and Barney. And we implemented the following and measured the performance.

1. Fisher : Weight vector and threshold values are determined using the algorithm in section 2.2. Prun Rate(PR, $0 \leqq PR \leqq 100$) is varied and the performance is measured.
2. Prun : After transforming into the network topology explained in section 2.3, we train the fully connected network by EBP with the weights and the thresholds initialized from the procedure 1.
3. Sel : After transforming into the network topology explained in section 2.3, we train the partially connected network by EBP with weights and thresholds initialized from the procedure 1. The performance is measured as the number of connections varies.

PR is increased by 1% from 0 to 30%, and by 5% after 30%. The learning rate η and the momentum rate α are set to 0.2 and 0.7, respectively. From the experiment, we observe that the performance of the "Sel" upto PR equals 21% and the "Prun" upto PR equals 40% is similar or even better than that of the nodes decreases as PR increases. Fig. 3 shows that the number of connections decreases as the PR increases. As "Prun" has fully connected network structure while "Sel" has a partially connected one, the number of the connections of the "Prun" is more than that of the "Sel"

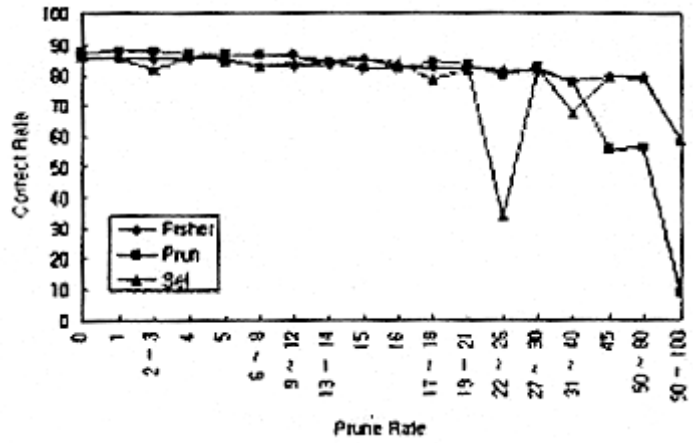


Fig. 2. The performance comparison of 3 learning methods

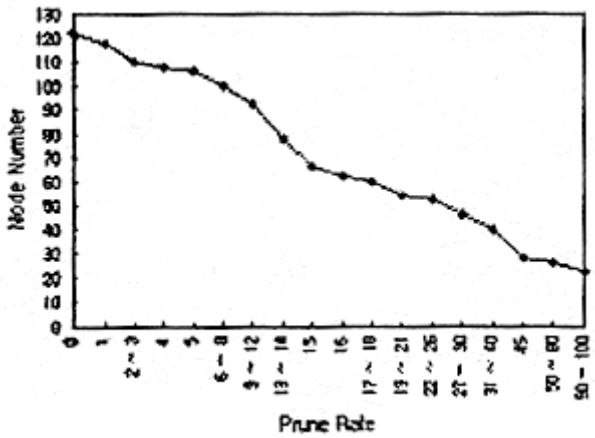


Fig. 3. The number of nodes vs. PR.

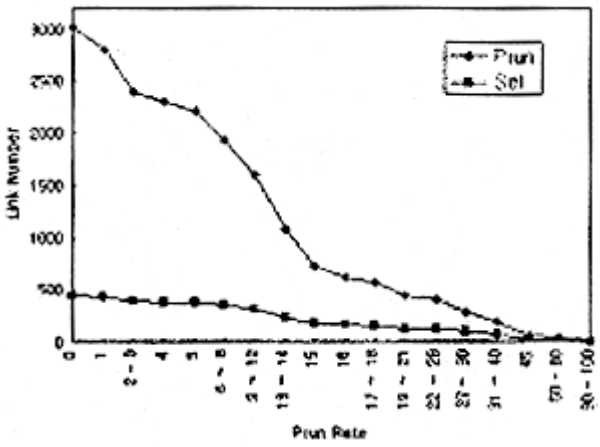


Fig. 4. The number of connection vs. PR.

5 Conclusion

In this paper a solution is proposed to solve the problem that the network structure of an incremental model can be
extended excessively when the learning pattern contains many noisy patterns. The proposed method uses a predefined
parameter, PR, to stop the recursive process in making the network structure. After this binary tree network structure is
transformed into the three layer feedforward structure, the EBP is employed to train the structure further. An appropriate
number of nodes and the corresponding weights between the nodes are determined, which is the aim of the pruning
process.

References

[1] R.A.Johnson, D.W.Wichern, "Applied Multivariate Statistical Analysis. 2ED", pp470-542, Prentice-Hall International
Editioned, 1988.
[2] J.C.Lee, Y.H.Kim, W.D.Lee, S.H.Lee, "Pattern Classifying Neural Network using Fisher's linear discriminant function",
Vol1, IJCNN June, 1992.
[3] J.C.Lee, Y.H.Kim, W.D.Lee, S.H.Lee, "A method to find the structure and weights of layered neural networks, WCNN,
pp552-555, June 1993.
[4] M.Hagiwara, "A simple and effective method for removal of hidden units and weights", Neurocomputing 6, pp207-218,

1994
[5] S.Y.Kung, J.N.Hwang, "An algebric projection analysis for optimal hidden units size and learning rates in back-propagation learning", IJCNN Vol I, pp363-370, 1988
[6] A.S.Weigend, D.E.Rumelhart, B.A.Huberman, "Generalization by weight-elimination with application to forecasting",
Advances in Neural Information Processing Systems 3, Morgan Kaufmann, pp875-882, 1991

[7] J.Seitama, R.J.F.Dow, "Neural net pruning-why and how", IJCNN Vol I, pp325-333, 1988
[8] M.Hagiwara, "Novel back propagation algorithm for reduction of hidden units and acceleration of cenvergence using
artificial selection" , IJCNN Vol I, pp625-630, 1990
[9] S.J.Hanson, L.Y.Pratt, "Comparing biases for minimal network construction with back-propagation", Advances in
Neural Information Processing Systems 1, Morgan Kaufmann, pp177-185, 1989.
[10] J.E.Moody, T.Rognvaldsson, "Smoothing regularizers for projective basis function networks", Advances in Neural
Information Processing Systems, Vol. 9, pp.585-591, 1997.
[11] L.Wangchao, W.Yongbin, L.Wenjing, Z.Jie, J.Li, "Sparselized higher-order neural network and its pruning algorithm",
IJCNN Vol. 1, pp359 -362, 1998

Chapter 12

A Call Admission Control Using Interval Arithmetic Coulomb Energy Network

Won Don Lee[1], Kyunghee Lee[2], and Hong-kee Kim[3]
[1]ChungNam National University
[2]PyongTack University
[3]Electronics and Telecommunications Research Institute

Abstract

To use ATM network resources effectively, call control in ATM networks must be able to adapt gracefully to the dynamic behavior of traffic loads and network conditions during the network operation. In this paper, we implement a call admission control with interval arithmetic Coulomb energy network , a generalization of Coulomb energy network, which can handle interval data and point data for the learning efficiency. As a modular neurla network, each single layer of interval arithmetic Coulomb energy network is trained sequentially to make a multi-layered network. Because each layer can be trained separately, the error does net propagate through the layers and the network becomes modular.
Simulation results demonstrate the advantage of modularity of Coulomb energy network and flexibility of interval arithmetic algorithm.

Keywords : Interval Arithmetic, Coulomb Energy Network, Modular Neural Networks , Asynchronous Transfer Mode(ATM), Call Admission Control, Classification

12.1 Introduction

12.1.1 Past Work

In broadband networks for intergrated services developed to operate in high-speed environments, such as Asynchronous Transfer Mode (ATM) system, the Call Admission Control(CAC) plays an increasingly important role, A Call Admission Control Using Interval Arithmetic Coulomb Energy Network from both resource and network management viewpoint. The aim of these efforts is to design CAC function to maximize the system throughput while maintaining the desired Quality of Service(QoS). However, in ATM networks, the bandwidth of a call is not clear, because all the information is segmented into small cell is not clear. because all the information is segmented into small cells, 53 octects, and the required number of cells is generated and conveyed through the network. An ATM node has a variety of cell types due to the diversity and the probabilistic complexity of source traffic caused by various sources and uperimposed traffic offered from other networks. And it needs call admission control dependent on the situation of cell types and cell numbers in the buffer of the ATM node.

Most traditional control strategies are based upon results obtained from analytical performance evaluation, such as queuing models. A major shortcoming of the currently available queuing models is that only steady-state results are tractable, and, consequently, any control algorithm tailored on the basis of such models which can ensure optimal performance control methods that dynamically regulate traffic flows according to changing network conditions require an understanding of network dynamics. Furthermore, in these models, hypotheses concerning the detailed knowledge of systems under study can seldom be justified in real life.

Recently an adaptive call admission control method using neural networks trained with a virtual output buffer method was proposed by Hiramatsu[1], where the class of Error Back Propagation(EBP) neural networks is considered. An EBP neural network is considered. An EBP neural network is a multi-layered network consisting of an input layer, an output layer, and at least on hidden layer of nonlinear processing elements called neurons. The neurons between layers are interconnected by variable connections, which are internal parameters referred to as weights. In control applications, EBP neural networks are also the most prevalent paradigms because they have the ability to learn dynamic system characteristics through nonlinear mapping. Although a multi-layered network trained by the EBP algorithm has been used to solve real-world problems in many areas such as control, prediction, and classification the EBP algorithm becomes difficult to find a parameter set which leads to convergence toward as acceptable minimum. This makes it often very difficult to get a useful solution.

There has been a considerable research effort to overcome these problems. Many of the ideas have included modularity as a basic concept. Muiet. al. proposed a locally connected adaptive modular neural network[2].

introduction

Their model employs a combination of EBP training and a Winner-Take-All layer. Bellotti et. al. have presented a modular neural system using a self organizing map and a multi-layer Perceptron which can be applied in a cosmic ray space experiment[3]. Bachmann et. al. proposed a relaxation model based on an N-dimensional coulomb potential[4]. Their system is similay to the Hopfield model[5][6] in some respects, but has arbitrary large capacity. Scofield presented a learning algorithm for N-dimensional Coulomb energy network(CEN) which is applicable to a single as well as a multi-layer networks[7].
While many learning algorithms classify patterns already exist, Scofield's learning algorithm is worthwhile to note, because it is applicable independently to each layer of a multi-layer network and does not depend on the propagation of errors through many layers. This property makes the system modular. Therefore multi-layer feed-forward networks can be split into single-layer modules and can be trained separately if the requirement of fixing the target values in the error back propagation is relaxed. In other words, as a modular neural network, each single layer of Coulomb energy network is trained sequentially to make a multi-layer network. Because each layer can be trained separately, the error does not propagate through the layers and the network becomes modular.

12.1.2 Our Approach to Call Admission Control

We propose a general form of the energy function for such networks, and show how this algorithm can be naturally combined with EBP algorithm[8]. We also propose a method that minimize the energy function on the variation net only of weight but also of temperature.
The simulation result shows that learning is done more efficiently and accurately with the proposed method. Since weight and temperature can be learned in parallel speed up in learning might be doubled if appropriate hardware support is provided.
In real-time applications such as CAC in ATM networks, the input data vectors are presented not only by point data but also by clustered data in a bounded area. It is generally known that rectangular forms such as VLI as a rule description is powerful in its expressiveness for human understanding, whereas discriminating boundaries used in this network has an asymptotically larger capacity than hyper-cubes[10]. In learning viewpoint, it is useful to codify the clustered data as an interval data vecause an interval

data can represent many point data and it can reduce the training time due to the small set of training data. Interval input data also makes it possible for us to train the neural networks from expert's knowledge representing rule-based control procedures which is used in many telecommunication systems.
An interval can be regarded as the "rules" that an expert has in his knowledge. A point datum can be seen as an experimental or observed one in the environment. Therefore, when a network is trained with intervals and point data, we can say that an expert knowledge is combined with his experience to make modified rules. Here, however, we should note that the modified knowledge is not in the form of intervals in the sense that the true interval data are formed as a product of intervals of each variable. Rather, the network itself contains the modified knowledge in its weight set and the modified knowledge can only be described with complex nonlinear equations using those weights.
The original Coulomb energy network, however, is able to deal with only the point input data and has been optimized by only varying the weights. So, it is not possible to classify some data such as linearly non-separable classification data and the interval data in a bounded area correctly. Therefore, we introduce interval arithmetic Coulomb energy network that can handle interval data and point data for the learning efficiency. As a result, interval arithmetic CEN can be regarded as a generalization of Coulomb energy network and a form of modular network to alleviate the EBP drawbacks.

12.1.3 Outline of Paper

The remainder of this paper is organized as follows. In Section 2, a modular neural network approach is introduced and it is shown that traditional supervised Coulomb energy network is a kind of modular network. A new learning algorithm for Coulomb energy network on the variation of the temperature is also discussed as an improved learning algorithm in Section 2. In Section3, a generalization of Coulomb energy network that can handle interval vector input and point vector input in the same way is discussed. In Section 4, an application to CAC, one of the most important problems to be overcome in ATM network using interval arithmetic Coulomb energy network is presented. Finally, the couclusions are given in Section 5.

12.2 Coulomb Energy Network : A Kind of Modular Neural Networks

12.2.1 A Modular Neural Network approach

There are some problems in the error back propagation algorithm which is used extensively as a neural network learning algorithm[12]. First, since it uses a descent method to find an energy minimization, it can sometimes trap into a local minimun instead of a global minimum. Second and more serious problem comes from the fact that the algorithm adjusts the synaptic weights of the hidden layers by observing input and the corresponding output values. Because weights of the hidden layers are adjusted through propagation of errors between desired output and actual output values for a given pattern, hidden layer learning is done indirectly compared with the direct output layer learning. Therefore, when we apply the EBP algorithm to a complex network, there can be a danger that hidden layers might not have inductive capability to capture input patterns effectively. The first problem might not be a serious one depending upon network structures.
But the second one is important in that it is directly linked with the problem of increasing network efficacy. The energy function in the EBP network is described as

$$E = \sum_{p} E(p) = \sum_{p} \sum_{j} (t(p,j) - o(p,j))^2 \qquad (1)$$

Where t(p,j) and o(p,j) are the target and the actual j-th bit values of the output, respectively, when input pattern p is entered to the system. The reason that EBP adjusts the hidden layer weights indirectly through propagation errors is that the target values of the output patterns are. already determined as fixed points. If we relax the requirement of the fixed target values in the output space, then it becomes possible to adjust the weights in the middle layers directly. The requirement in such a network would be the property that patterns belonging to the same class be attracted together, and those belonging to different class be repelled each other. There can be no more "hidden" layers, as we can directly adjust the weights in the middle layers.
Since this algorithm does not rely on the propagation of errors through layers, the system becomes "modular" in the sense that each single-layer

module can be trained independently. Multi-layer system can be make naturally by layering trained layers one by one. In general, module i is a single-layer network whose input comes from the output of the module (i-1), and whose output becomes the input of the module (i+ 1). Therefore, the learning of the whole system in done sequentially by first training the first module, and then training the second module until the last uppermost module is trained. Training each module separately can have advantage besides those already discussed just because it is easier to train a simple single-layer module than to train the whole multi-layer network at once. The general form of the modular neural network energy function would be :

$$E \propto \sum_{same\ class} M_1(X_i, X_j) + \sum_{diff.\ class} M_2(X_i, X_j) \qquad (2)$$

Where M_1 and M_2 are some measure functions to determine the distance between recorded output patterns X_i and X_j.

As we can see, there is a separate measure function for pairs of output patterns belonging to the same class, and another for pairs of output patterns belonging to the different class. For simplicity, we only consider the dichotomy system. The measure functions are constructed in such a way that the total energy is minimized as output patterns of same class are attracted each other and output patterns of different class are repelled each other. When we do this, the resulting vectors will be grouped together in the output space, and there may by more than or equal to one group in the space in general for exch class. There has been attempts to make such an energy function[12], and one of them is the coulomb energy network[7].

12.2.2 Traditional Supervised CEN

The energy function of the coulomb energy network is

$$\Psi = 1/(2L) \sum_{i}^{M} \sum_{j}^{M} Q_i Q_j \left| X_i - X_j \right|^{L} \qquad (3)$$

Here, $Q_i Q_j$ is '-'when patterns $X_i X_j$ belong to the same class, and '+' when patterns belong to different class. After defining specific measures, then it is easy to find learning algorithm by gradient descent method:

$$\delta\omega_{nm} = -\eta\partial\Psi/\partial\omega_{nm} \qquad (4)$$

where η is the learning rate. In the coulomb energy network, this becomes

$$\delta\omega_{nm} = (+/-)\eta \mid X(t) - X(t+1) \mid^{-(L+2)} \triangle_{nm}(p(t), p(t+1)) \qquad (5)$$

Where negative sign is for subsequent patterns of the same class, and the positive sign for patterns of different class, and $\triangle_{nm}$ is defined as

$$\triangle_{nm}(p(t), p(t+1)) \equiv (X(t) - X(t-1)) \cdot \partial/\partial\omega_{nm}(X(t) - X(t+1)) \qquad (6)$$

The CEN defines the potential energy function that makes each memory site attractive for the same class data and repulsive for the different class data. In the equation (3), the network activity vector X_i which is the result from mapping the input pattern f_i through the weight matrix ω_{nm} is defined as

$$X_i = \sum_{n=1}^{N} e_n F_n(f_i)$$

$$F_n(f_i) = 1/\left[1 + \exp(-(1/T)(\sum_{m=1}^{K} \omega_{nm} f_{mi} + \theta))\right] \qquad (7)$$

Where, e_n denotes the unit vector identifying the n^{th} node and f_i denotes the input pattern in the space R^k. ω_{nm} is the weight matrix component and θ is the bias. In the equation (7), the steepness of the logistic function is determined bythe temperature T. If T has too large value, the system trials to find the weight vector, because the learning depends on the initial weight set. So it is important to vary the temperature with the weight. And this new CEN also can be constructed in modules, because it is not affected by the error of the other layers in contrary to the EBP.

12.2.3 A Modified CEN with Varying Temperatures

Scofield proposed the learning algorithm using the CEN as a type of modular network. The CEN can construct the system in modular fashion, because it is not affected by the errors of the other layers. However, the CEN has been optimized by varying the weight. We describe the method of varying the temperature as well as the weights. And we solve the linearly non-separable classification problem that could not be solved in the supervised CEN, using the concept of the distance.

12.2.3.1 Energy Minimization of A Modified CEN

In the above section, we discussed the effects of the temperature that affects the steepness of the logistic function, sigmoid function. We show the method that varies the temperature and weight. As a general method, we calculate the gradient of energy function Ψ over weight and temperature and then, descent to minimize the energy function Ψ.

$$\delta\omega_{nm} = -\eta(\partial\Psi/\partial\omega_{nm})$$

$$\partial\Psi/\partial\omega_{nm} = -(1/2)\sum_{i=1}^{M}\sum_{j=1}^{M} Q_i Q_j [R_{ij}]^{-(L+2)} \triangle_{nm}(f_i, f_j) \qquad (8)$$

$$\delta T_n = -\eta T_n(\partial\Psi/\partial T_n)$$

$$\partial\Psi/\partial T_n = -(1/2)\sum_{i=1}^{M}\sum_{j=1}^{M} Q_i Q_j [R_{ij}]^{-(L+2)} \triangle_{nm}(f_i, f_j) \qquad (9)$$

where $Q_i Q_j$ is '−' when patterns X_i, X_j belong to the same class, and '+' when patterns belong to different class and. R_{ij} is a difference vector between vector X_i and vector X_j. $\triangle_{nm}$ and $\triangle T_n$ are defined as

$$\triangle_{nm}(f_i, f_j) \equiv R_{ij} \cdot \partial/\partial\omega_{nm} \cdot R_{ij} \qquad (10)$$

$$\triangle T_{n(f_i, f_j)} \equiv R_{ij} \cdot \partial/\partial T_n \cdot R_{ij} \qquad (11)$$

Now. we have to calculate $\triangle_{nm}(f_i,f_j)$ and $\triangle T(f_i,f_j)$. The activity vector X_i becomes the equation not only of the weights ω_{nm} but also of the temperature T_n as in the equation (7). Therefore, we see the following equations for $\triangle_{nm}$ and $\triangle T_n$:

$$\triangle T_n(f_i,f_j) = -[1/T_n^2][F_n(f_i)-F_n(f_j)][F_n(f_i)(1-F_n(f_j)S_{mi}-F_n(f_j)(1-F_n(f_j)S_{mj}] \quad (12)$$

$$\triangle_{nm}(f_i,f_j) = -[1/T][F_n(f_i)-F_n(f_j)][F_n(f_i)(1-F_n(f_j)f_{mi}-F_n(f_i)(1-F_n(f_j)f_{mj}] \quad (13)$$

where,

$$S_{mi} = \sum_{m=1}^{K} \omega_{nm} f_{mi} + \theta, \qquad S_{mi} = \sum_{m=1}^{K} \omega_{nm} f_{mj} + \theta \quad (14)$$

If we calculate the variation of temperature and weight according to the equation (8) and (9), it takes too much time in learning. So, we first select two patterns randomly and map them through weight vectors. And then, we get the variations of the temperature and the weight. We continue these steps for the defined learning count. The following is the activity vector X_i over input pattern f(t) :

$$X_i = \sum_{n=1}^{N} e_n F_n(f(t))$$

$$F_n(f(t)) = 1/\left[1+\exp(-(1/T_n)(\sum_{m=1}^{K} \omega_{nm} f_m(t)+\theta))\right] \quad (15)$$

Let f(t) and f(t+ 1) be the two randomly selected pattersn. Then the relaxation procedures, I. e., the variation of weight
and temperature becomes as follows :

$$\delta\omega_{nm} = sign \times \eta \left| X_i X_j \right|^{-(L+2)} \triangle_{nm}(f(t),f(t+1)) \quad (16)$$

$$\delta T_n = sign \times \eta T_n \left| X_i - X_j \right|^{-(L+2)} \triangle T_n(f(t), f(t+1)) \qquad (17)$$

In these equations, the sign value has the value of -1.0 if f(t) and f(t+!) belong to the same class and have the value of 1.0 if they belong to different classes.

12.2.3.2 Simulation Results

An example data which is know to be impossible to classify correctly by the traditional supervised CEN are depicted in the figure 1.
Since the basic idea of CEN is to make the system attractive for the data of the same class and repulsive for the data of the different class, the data of class 2 in figure 1 will be attracted to each other and gqthered in the center of the output space. But, the data of class l will also be ;apped onto the center of the output apace. Then, the data of class 2 collide with the data of class 1. Therefore, this problem can not be solved by the traditional supervised CEN. In this paper, we solve the problem using the concept of the distance. That is, although two selected patterns are of the same class, they reper each other if they are farther than the defined distance. In the progress of classification, we see that several groups are constructed in the same class but finally, after learning with our CEN learning algorithm, the collision between ter class 2 and the class 1 does not occur, and we can correctly classify the data

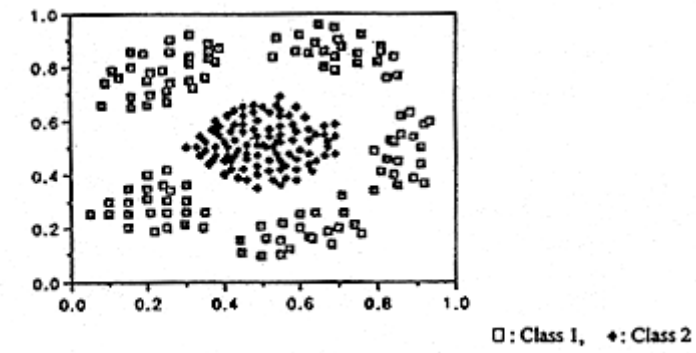


Figure 1. An example of linearly non-seperable classification data

To investigate the classification ability of our CEN we use the randomly

selected 28 input stimulus among 216 data in figure 1. The figure 2 shows a result of the output when our proposed CEN learning algorithms are applied to classify the 216 linearly non-separable data. We used the 2-2-2-1 network architecture, with the starting temperature = $1.0\pm\delta$, $\eta = 0.0005$, $\eta T_n = 0.0005$, and learning epochs = 400 in this experiment.

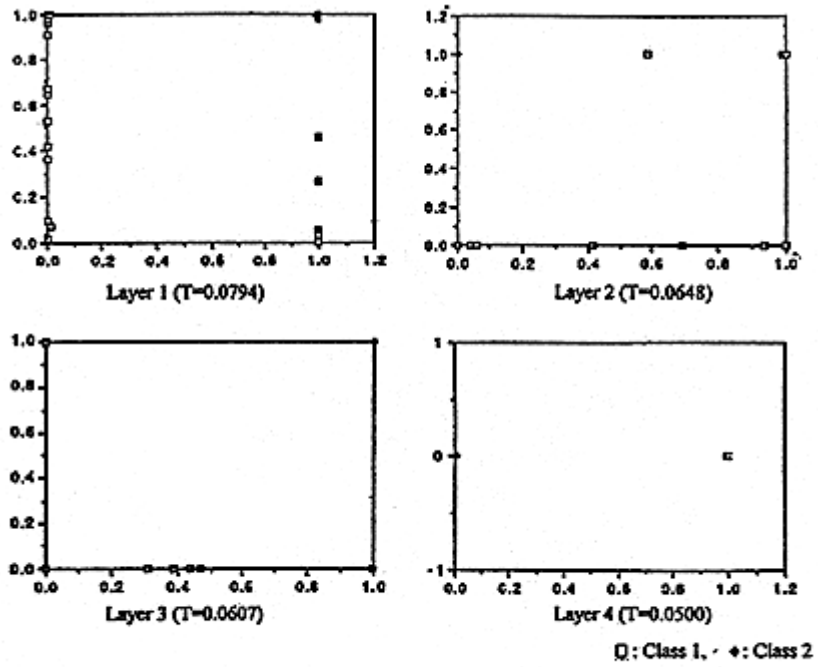


Figure 2. An example of the output by our CEN learning algorithm, a new CEN learning algorithm with the variation of temperature, when 216 linearly non-separable input data are given.

12.3 Interval Arithmetic CEN : Generalization of CEN

12.3.1 Interval Arithmetic

Conseder real valued functions of one or two independent variables. Interval arithmetic generates bounds for the set of function values that belong to all parameter values con tained in given interval. A single operation of interval arithmetic is defined by

$$A \ op \ B = \{a \ op \ b \mid a \in A, b \in B\} \qquad (18)$$

The basic arithmetic iperations on intervals used in this paper are as followes.

$$A + B = [a^l, a^u] + [b^l, b^u] = [a^l + a^u, a^u + b^u]$$

$$A - B = [a^l, a^u] - [b^l, b^u] = [a^l - a^u, a^u - b^u]$$

$$m \cdot A = m \cdot [a^l, a^u] \begin{cases} [m \cdot a^l, m \cdot a^u], & \text{if } m \geqq 0, \\ [m \cdot a^u, m \cdot a^l], & \text{if } m < 0 \end{cases} \quad (19)$$

$$\exp A = \exp [a^l, a^u] = [\exp a^l, \exp a^u]$$

$f(net) = f([net^l, net^u]) = [f(net^l), f(net^u)]$

where $f(s) = 1/1[1 + \exp(-x)]$ and the superscripts l and u denotes the lower limit and upper limit of the interval, respectively. Mormally both operands of these binary functions are intervals. The case of a scalar operand may be included by simply taking an interval with equal lower and upper bound, e.g. A = [a,a].

12.3.2 Energy Minimization of Interval Arithmetic CEN

Coulomb energy network was proposed by Scofield, making electrostatic potential energy as the equation(3), which means that the potential energy is constructed in such a way that each memoryis an attractor or a repeller of other memory sites. When we minimize the potential energy, we can obtain a collection of memory sites appropriately placed in the upper layer. And we extend Coulomb energy network to the interval arithmetic Coulomb energy network with input vector, $f_{mi} = [f_{mi}^l, f_{mi}^u]$, by defining electrostatic potential energy as follows ;

$$\Psi = 1/(2L) \sum_i^M \sum_j^M Q_i, Q_j \left(\left| X_i^l - X_j^l \right|^L + \left| X_i^u - X_j^u \right|^L \right) \quad (20)$$

where l and u in the equation mean lower and upper bounds of the interval in the memory sites, respectively. This potential energy function is good for Coulomb energy network and for interval arithmetic. And as Scofield did. we define an attractive potential energy berween memory sites of the same class, and a repulsive potential energy between memory sites of different class by $Q_i(c)$, the charge at memory site i and of class c, in such a way that:

$$sign(Q_i(c)) \neq sign(Q_i(c')) \qquad \text{for} \quad c=c'$$

$$sign(Q_i(c)) = sign(Q_i(c')) \qquad \text{for } c \neq c' \qquad (21)$$

The minimization of the potential energy forces memory sites of the same class to have the same lower and upper bounds, whereas it makes the memory sites of the different classes have different lower and upper bounds far apart from each orher. And then, according to the gradient descent, to find an optimal weight set, we do the following:

$$\delta\omega_{nm} = -\eta(\partial\Psi/\partial\omega_{nm})$$

$$\partial\Psi/\partial\omega_{nm} = -(1/2)\sum_{i=1}^{M} Q_i Q_j \left(\left| X_i^l - X_j^l \right|^{-(L+2)} \Delta_{nm}^l + \left| X_i^u - X_j^u \right|^{-(L+2)} \Delta_{nm}^u \right) \qquad (22)$$

where

$$X_i^l = \sum_{n=1}^{N} e_n F_n(\neq t_{in}^u), \quad X_i^l = \sum_{n=1}^{L} e_n F_n(net_{in}^l)$$

$$net_{in}^l = \sum_{m=1, w_{nm} \geq 0}^{Ninputs} w_{nm} \cdot f_{mi}^l + \sum_{m=1, w_{nm} < 0}^{Ninputs} w_{nm} \cdot f_{mi}^u + \theta_n$$

$$net_{in}^u = \sum_{m=1, w_{nm} \geq 0}^{Ninputs} w_{nm} \cdot f_{mi}^u + \sum_{m=1, w_{nm} < 0}^{Ninputs} w_{nm} \cdot f_{mi}^l + \theta_n$$

A Call Admission Control Using Interval Arithmetic Coulomb Energy Network

$$\triangle_{nm}^l = (F_n(net_{in}^l) - F_n(net_{jn}^l)) \cdot$$

$$\begin{cases} (F_n(net_{in}^l)(1-F_n(net_{jn}^l)) \cdot f_{mi}^l - F_n(net_{jn}^l)(1-F_n(net_{jn}^l)) \cdot f_{mi}^l), & \text{if } \omega_{nm} \geq 0, \\ (F_n(net_{in}^l)(1-F_n(net_{jn}^l)) \cdot f_{mi}^u - F_n(net_{jn}^l)(1-F_n(net_{jn}^l)) \cdot f_{mi}^u), & \text{if } \omega_{nm} < 0, \end{cases}$$

(23)

$$\triangle \omega_{nm}^u = (F_n(net_{in}^u) - F_n(net_{jn}^u)) \cdot$$

$$\begin{cases} (F_n(net_{in}^u)(1-F_n(net_{jn}^u)) \cdot f_{mi}^u - F_n(net_{jn}^u)(1-F_n(net_{jn}^u)) \cdot f_{mi}^u), & \text{if } \omega_{nm} \geq 0, \\ (F_n(net_{in}^u)(1-F_n(net_{jn}^u)) \cdot f_{mi}^{\bar{l}} F_n(net_{jn}^u)(1-F_n(net_{jn}^u)) \cdot f_{mi}^l), & \text{if } \omega_{nm} < 0, \end{cases}$$

(24)

$$F_n(\sum_{m=1}^{K} \omega_{nm} f_{mi}) = 1 \Big/ \left[1 + \exp(-(1/T)(\sum_{m=1}^{K} \omega_{nm} f_{mi} + \theta))\right] \qquad (25)$$

And e_n denotes the unit vector identifying the n^{th} node and Θ is bias which decides the trend of the node when the effent of the input is small. ω_{nm} is the weight matrix component. We calculate the gradient of energy function Ψ over temperature and then, descent to minimize the energy function Ψ.

$$\delta T_n = -\eta T_n (\partial \Psi / \partial T_n)$$

$$\partial \Psi / \partial T_n = -(\frac{1}{2}) \sum_{i=1}^{M} \sum_{j=1}^{M} (Q_i Q_j \left| X_i^l - X_j^l \right|^{-(L+2)} \quad \triangle T_n^l + \left| X_i^u - X_j^u \right|^{-(L+2)} \quad \triangle T_n^u$$

(26)

where

$$\triangle T_n^l = -[1/T_n^2][F_n(net_{in}^l) - F_n(net_{jn}^l)] \cdot [F_n(net_{jn}^l)(1 - F_n(net_i n^l)) \cdot net_{in}^l$$
$$- F_n(net_{jn}^l)(1 - F_n(net_{jn}^l)) \cdot net_{jn}^l] \qquad (27)$$

Interval Arithmetic CEN : Generalization of CEN

$$\triangle T_n^u = -[1/T_n^2][F_n(net_{in}^u) - F_n(net_{jn}^u)] \cdot [F_n(net_{in}^u)(1 - F_n(net_{in}^u)) \cdot net_{in}^u] - F_n(net_{jn}^u)(1 - F_n(net_{jn}^u)) \cdot net_{jn}^u] \quad (28)$$

In the implementation, we dropped the summation at the above equation:

$$\delta\omega_{nm} = (+/-) Q_i Q_j \cdot \eta \cdot (|X_i^l - X_j^l|^{-(L+2)} \quad \triangle_{nm}^l + |X_i^u - X_j^u|^{-(L+2)} \quad \triangle_{nm}^u) \quad (29)$$

$$\delta T_n = (+/-) Q_i Q_j \cdot \eta \cdot (|X_i^l - X_j^l|^{-(L+2)} \quad \triangle T_n^l + |X_i^u - X_j^u|^{-(L+2)} \quad \triangle T_n^u) \quad (30)$$

where the negative sign is for the pattern pairs of the same class, and positive for those of different classes.

12.3.3 Simulation Results

We present some results of experiments with computer simulation to investigate the ability of our interval arithmetic CEN to classify the interval data. In the first experiment, we use a mixture of point and interval data as a training data set. The point data can be regarded as an interval with equal lower and upper bound. In the second experiment, we use the linearly non-separable classification data encoded as interval arithmetic in training phase and some mixture of point and interval data as a text data set.
We use 2-4-4-4-1 network architecture that consists of one input layer, 3 hidden layers, and one output layer. The parameters used in the experiments are as follows: the learning rate is 0.001, the number of epochs is 1000, and starting temperature T is 1.0

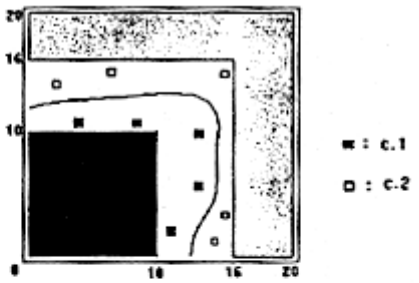
12.3.3.1 Classification of Mixture Data Including Point and Interval Data

Figure 3 shows the mixture data that is given in the interval arithmetic backpropagation [14]. The mixture data in figure 3 (a) corresponds to point and interval data in the input training data set. The point data described as closed interval data with only one point inside each interval data is also shown in figure 3(b). Figure 3(c) shows the test data and their results

of classification. As teaching input data are inserted, the system modifies the border line, which is drawn between class 1 and 2, as the area of certain class is wxpanded. Note that class 2 interval data includes "don't care" conditions in the case of c.2 such as X=[0.0, 20.0] and Y=[0.0, 20.0]. As expert knowledge is encoded with many "don't care" attributes, interval arithmetic CEN is capable of dealing with the expert rules in addition to the observed indivisual data.

12.3.3.2 Classification of Linearly Non-separable Problem Data

We also describe the result of an experiment to investigate the ability of interval arithmetic CEN with the linearly non-separable classification data, used in section 2.2, represented by interval arithmetic.



(a)

		Class 1(c.1)		Class 2(c2)	
		x-position	y-position	x-position	y-position
Training Data	Empirical Data (Point data)	[4.0, 4.0]	[11.0, 11.0]	[2.0, 2.0]	[14.0, 14.0]
		[8.0, 8.0]	[11.0, 11.0]	[6.0, 6.0]	[15.0, 15.0]
		[11.0, 11.0]	[3.0, 3.0]	[14.0, 14.0]	[2.0, 2.0]
		[13.0, 13.0]	[6.0, 6.0]	[15.0, 15.0]	[4.0, 4.0]
		[13.0, 13.0]	[10.0, 10.0]	[15.0, 15.0]	[15.0, 15.0]
	Interval Data (Expert's knowledge)	[0.0, 10.0]	[0.0, 10.0]	[16.0, 20.0]	[0.0, 20.0]
				[0.0, 20.0]	[16.0, 20.0]

(b)

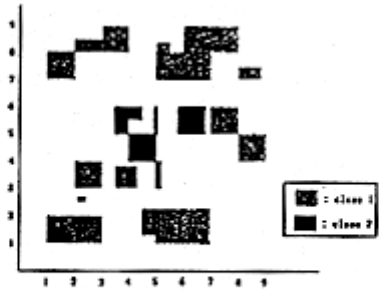
Test data		Classification Results
x-position	y-position	
[13.0, 13.0]	[13.0, 13.0]	Class 2
[3.0, 4.0]	[10.0, 11.0]	Class 1
[15.0, 16.0]	[1.0, 2.0]	Class 2
[4.0, 5.0]	[9.0, 10.0]	Class 1
[5.0, 6.0]	[14.0, 16.0]	Class 2
[10.0, 11.0]	[16.0, 19.0]	Class 2
[1.0, 3.0]	[21.0, 23.0]	Class 2
[12.0, 14.0]	[11.0, 11.0]	Class 1
[0.0, 1.0]	[10.0, 11.0]	Class 1
[0.0, 1.0]	[11.0, 12.0]	Class 1
[0.0, 1.0]	[12.0, 13.0]	Class 2
[0.0, 1.0]	[13.0, 20.0]	Class 2
[10.0, 12.0]	[3.0, 4.0]	Class 1
[14.0, 15.0]	[3.0, 4.0]	Class 2
[15.0, 16.0]	[15.0, 16.0]	Class 2

(c)

Figure 3. Example data for classification of mixture data: (a) Training data by 2 dimensional representation, (b) Training data by interval representation, and (c) Test interval data and their results of classification

Because interval arithmetic CEN is a general form of original CEN, interval arithmetic CEN also has the property of attracting the same class data while repulsing the different class data as in section 2.2. The interval arithmetic CEN, however, can handle the problem that the traditional supervised CEN can not.

Figure 4(a) shows the 2 dimensional interval data for training. Figure 4(b) show 13 test interval data, randomly selected after training the interval arithmetic CEN, and the results of classification. In figure 4(b), we can find that interval arithmetic CEN can correctly classify the linearly non-seperable classification data as expected. The difference between the modified CEN described in section 2.2 and interval arithmetic CEN is that interval arithmetic CEN has no need to have large data set to represent a certain group of data and therefore it takes less time to train the network.



(a)

Test data		Classification Results
x-position	y-position	
[5.0, 5.0]	[5.0, 5.0]	Class 2
[2.0, 3.0]	[4.0, 5.0]	Class 1
[7.0, 8.0]	[4.0, 5.0]	Class 1
[7.0, 8.0]	[5.0, 6.0]	Class 1
[5.0, 8.0]	[4.0, 5.0]	Class 2
[5.5, 5.5]	[4.0, 6.0]	Class 2
[6.0, 7.0]	[5.0, 6.0]	Class 2
[3.0, 4.0]	[5.0, 5.0]	Class 2
[1.0, 1.5]	[7.0, 7.5]	Class 1
[6.0, 7.0]	[1.5, 2.3]	Class 1
[2.0, 3.0]	[0.7, 1.5]	Class 1
[6.0, 6.0]	[5.5, 6.0]	Class 2
[3.0, 4.0]	[1.0, 2.0]	Class 1

(b)

Figure 4. An example of interval data: (a) Training data, (b) Test interval data and results

12.4 Application to CAC in ATM Network Using Interval Arithmetic CEN

12.4.1 Call Admission Control Problem

In broadband networks for integrated services developed to operate in high-speed environment, ATM technology has been recommended by the CCITT

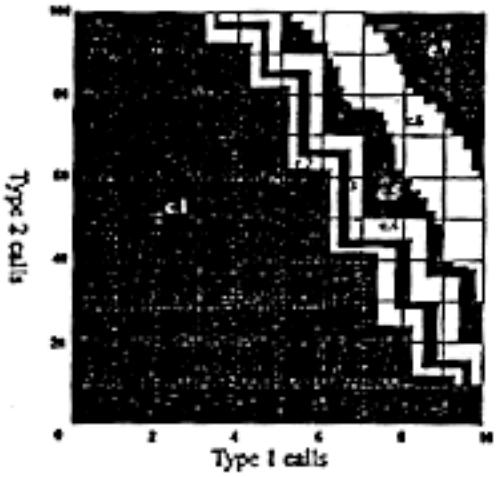
as the transport mechanism for B-ISDN traffic data. The call admission control forms an important part belonging to resource allocation problems in ATM network. When the bandwidth of a new connection call attempt exceeds the remaining capacity of a link in Synchronous Transfer Mode(STM) network, the call attempt is rejected and routed to the next link. In ATM network, however, the bandwidth of a call is not clear, since all the information is segmented into cells and the required number of cells is generated and conveyed through the network. As the number of Virtual Channels (VCs) increases in a Virtual Path(VP), the QoS at the cell level, such as the cell loss probability, deteriorates. Thus it is impossible to accept an unlimited number of VCs under QoS requirements and call admission control must decide appropriately whether to accept a new VC on the basis of its anticipated traffic characteristics, the QoS requirements of VCs, the current network load and the amount of network resource. As a result, the call admission control function plays an increasingly important role in ATM networks, from both resource management and network management viewpoint. In the context of ATM networks the rather classical problem of admission control gains a higher level of difficulty, due to the diversity and the probabilistic complexity of traffic streams. In addition the large variety of requirements for the quality of service to be fulfilled increases the complexity of the admission control problem.

12.4.2 Simulation Results

Figure 5 shows an example of the mean cell-loss rate from [1] under the condition of 150Mbps link with 100 cells buffer in ATM simulation model and the sample of results.
For example, consider that we have a value, 0.001, as a mean cell-loss rate required by the node. When 9 calls of type-1 and 90 calls of type-2 come into the node, the node should reject the calls because the call admission request point is larger than the mean cell-loss rate. The ATM node can only accept the calls within its mean cell-loss rate. Here, we implemented the call admission control using interval arithmetic CEN.
As shown in figure 5(a) we approximately classified the mean cell-loss data into seven classes of interval data. Figure
5(b) shows the number of interval data for each class used in training phase and figure 5(c) shows a part of classification results.

A Call Admission Control Using Interval Arithmetic Coulomb Energy Network

Interval arithmetic CEN is rained by 133 classified interval data of cell-lose rate. each epoch consists of 5200 iterations and the model has hour layer. In this experiment, it is difficult to learn the data which are in the border between classes, but easy in the area far from the border. Test data consists of 77 point and interval data. Two-call type point data are well classified, but 8% of interval data in the border are classified wrong. This experimental results show that interval arithmetic CEN can be used to control the call admission in ATM network effectively.



(a)

Class	Cell-loss rate	Number of interval data
c.1	$0 \sim 10^{-7}$	20
c.2	$10^{-7} \sim 10^{-6}$	13
c.3	$10^{-6} \sim 10^{-5}$	13
c.4	$10^{-5} \sim 10^{-4}$	12
c.5	$10^{-4} \sim 10^{-3}$	28
c.6	$10^{-3} \sim 10^{-2}$	32
c.7	$10^{-2} \sim 10^{-1}$	15

(b)

Test data		Classification Results
x-position	y-position	
[7.0, 7.0]	[40.0, 40.0]	c.1
[7.0, 7.0]	[50.0, 50.0]	c.4
[7.0, 7.0]	[75.0, 75.0]	c.5
[7.0, 7.0]	[90.0, 90.0]	c.6
[9.0, 9.0]	[80.0, 80.0]	c.7
[9.0, 9.0]	[30.0, 30.0]	c.3
[8.0, 8.0]	[30.0, 30.0]	c.3
[8.0, 8.0]	[57.0, 57.0]	c.5
[8.0, 8.0]	[65.0, 80.0]	c.2
[9.0, 9.0]	[25.0, 25.0]	c.2
[8.0,8.0]	[81.0, 81.0]	c.6
[9.0,9.0]	[90.0,90.0]	c.7
[6.0, 6.0]	[70.0, 90.0]	c.4
[5.0,5.0]	[0.0, 80.0]	c.1
[6.0, 6.0]	[0.0, 40.0]	c.1

(c)

Figure 5. An example of interval data monitored from an ATM CAC environment and the classification results: (a) Mean cell-loss rate vs. ATM node status, (b) Number of interval data for each class of cell-loss rate and (c) A sample of classification results

12.5 Conclusions

A general form of th energy function for modular neural networks is described and it is shown that the coulomb energy network fits the form. It is natural to see a general scheme to combine modular neural network algorithm and the EBP by relaxing the requirement of the fixed target values, and the whole feed-forward networks can be split into single-layer modules to be trained separately. This helps to control the middle layer weights directly and therefore increases the chance of making complex systems. The modified CEN learning algorithm by the gradient of energy function over not only weight but also temperature is effective in terms of taking effect immediately on the steepness of logistic function and is better than the traditional CEN in learning time and the quality of the output. Since the modified learning in temperature and weight can be done simultaneously, A Call Admission Control Using Interval Arithmetic Coulomb Energy Network

speed up in learning can be achieved with appropriate hardware support.
Unlike traditional data network, performance-driven control is vital to call control design in ATM networks. With high-speed transmission, neural control algorithm should also be effective in terms of taking effect immediately at the various network events, preferably being incorporated into hardware implementation. These ongoing design requirements represent a significant challenge to the next generation of call control by neural networks. The generalization of CEN, interval arithmetic CEN, which can handle interval dat, is suitable for call admission control with various call types, because interval arithmetic CEN can be adaptively controlled.
In the traditional AI system, there has been researches in the area of rule refinement where given expert knowledge, or a hypothesis, is modified systematically according to other hypotheses and training data set. The newly generated rules or hypothesis are again in the form of VLI formula[10], where each attribute or variable has a range values bounded by lower and upper bounds. The fact that the interval arithmetic Coulomb energy network accepts both interval and point data shows that it has the alility to refine the rules systematically, except that its newly generated reles are not in the form of VLI formula. Each interval is treated having the same weight or importance, compared not only with other rules, but also with the point data. Hence, we need to develop a scheme to modify weighted rules systematically with the point data in the future.

Acknowledgement

This research is supported by the Brain Science and Engineering Research Program in Korea.

ATM 망에서 호 연결수락제어를 위한 Interval Arithmetic CEN Interval Arithmetic CEN for ATM Call Admission Control

초록: Interval data와 point data를 함께 다룰 수 있는 일반화된 Coulomb Energy Network(CEN) 모형을 구현하고 ATM 망에서의
호 수락제어에 효과적으로 응용될 수 있음을 보여주었다.

연구 배경: 실제 ATM 방식의 광대역 통신망(B-ISDN)에서는 다양한 트래픽 소스에서 발생한 여러 종류의 셀이 중첩되어 전송되어진다. 따라서 새로운 서비스 호(call) 요구에 대한 수락 제공 여부는 셀의 유형과 노드에서의 버퍼상태 등 트래픽 상태에 따라 결정된다. 따라서 실제의 ATM 기반 초고속 망이 설치 운용됨에 따라 이러한 호 수락 제어는 망에서 제공하는 서비스의 질 (Quality of
Service) 보장 및 망 관리차원에서 중요한 역할을한다.[1]. 기존의 통상적인 제어전략은 대기모형(Queuing Model)에 기초를 둔 방식이였으며 신경회로망을 이용하여 적응적으로 제어하는경우에는 Error Back Propagation (EBP)를 이용하는 경우가 많았다[2].
비록 EBP 는 많은 제어 응용에 사용되어 개선된 결과를 보였지만 은닉층(hidden layer)의 학습이 출력층에서의 학습과 다르게 간접적으로 이루어 진다는 약점이 있다.

ATM 망에서 호 연결 수락 응용에서 다르는 트래픽 상태 데이터는 점 데이터(point data) 형태이기 보다는 구간 영역으로 표현할 수
있는 군집화(clustered) 데이터 형태이다. 이러한 구간 영역으로 표현되는 데이터(interval data)에 대한 학습이 이루어 진다면 점
데이터에 의한 학습에 비하여 학습 데이터(training data)의 수량 및 학습시간을 줄일수 있다. 본 논문에서는 Scofield 의 original
Coulomb Energy Network을 확장하여 점 데이터 및 구간 영역 데이터를 모두 다룰수 있도록 하는 Interval Arithmetic CEN을 제안한다.

Coulomb Energy Network(CEN): Scofield 에 의하여 제안된 Coulomb Energy Network은 각각의 메모리들이 주변의 다른 메모리들에 대하여 서로 같은 종류의 경우에는 끌어 당기도록 하고, 서로 다른 종류의 경우에는 서로 밀어 내도록 함으로써 electrostatic
potential energy 가 최소화 된다는 이론에서 비롯하였다[3]. 이러한 Scofield의 CEN은 EBP와 다르게 각각의 계층이 상위계층
error의 propagation에 의하여 간접적으로 학습되는 것이 아니라 각 계층마다 직접적으로 학습 되며 학습된 각 계층의 모듈화 구성에 의하여 다층의 네트워크가 자연스럽게 이루어진다는 장점이 있다.

Interval Arithmetic CEN: Interval Arithmetic CEN은 Scofield의 original CEN이 점 데이터만을 처리하였던 것에 비하여 실수 값을
갖는 구간 영역 데이터도 다룰수 있게 하는 original CEN의 일반화 모형이라고 할 수 있다. 또한 학습에 있어서도 뉴톤간의 연결가중
치(weight)에 대한 에너지의 gradient 만을 추구하였던 Scofield의 학습 알고리즘에 반하여

뉴론간의 연결 가중치 뿐만 아니라 온도
에 대한 에너지의 gradient를 계산하여 변화 시킴으로서 네트워크의 수렴속도를 개선하였고 original CEN에서 구분하지 못하는 데
이터들을 효과적으로 구분하는 성능의 개선을 확인할수 있었다.

시뮬레이션 결과 : ATM 망에서 새로운 호의 수락 여부를 결정하기 위해서는 특정 점 데이터 형태가 아닌 구간 영역 데이터로 표현이 가능한 망내의 자원의 상황 등을 학습하고 판별하기 위해 Interval Arithmetic CEN을 적용하는 실험을 하였다. ATM 시뮬레이션에 의한 ATM 호의 상황을 구분하기 위하여 4개의 계층을 갖는 네트워크를 구성하고 133 개의 셀 손실율 데이터를 학습하고 77개의 점 데이터 및 구간 영역 데이터에 대한 인식 실험을 하였다. 실험 겨로가 경계선 부근의 약 8%의 호 유형에 대해서는 잘못 인식하였으나 92%정도의 정확도로 2 개의 호 유형에 대하여 잘 인식이 되었다(그림1).

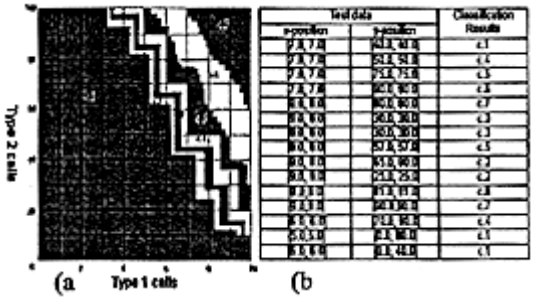


그림 1: (a)셀 손실율 데이터 예. (b)인식/분류 결과 예

[1] H.Saito, K.Shiomoto, "Dynamic Call Admission Control in ATM Network", IEEE J.Sel. Areas in Comm. 9, 982=989, 1991
[2] A.Hiramatsu , "ATM call admission control using a neural network trained with a virtual output buffer method", IEEE
Trans. on neural networks, 3611-3616, 1994
[3] C.L.Scofield, "Learning internal representation in the coulomb energy network", ICNN, I, 271-175. 1988

* 본 연구는 뇌과학 연구개발사업의 지원으로 수행되었음.