

효율적 패턴 인식을 위한 퍼지 규칙의 자동 생성

Automatic Construction of Fuzzy Rules for Efficient Pattern Recognition

이 양 원*, 최 형 일*

요 약

본 논문에서는 주어진 문제 영역상의 자료를 특성에 따라 분류하고, 자동적으로 퍼지 규칙을 생성할 수 있는 방법을 제안한다. 제안된 방법은 기존의 방법에 비하여 효율적으로 퍼지 공간을 분할하고, 분할된 퍼지 공간의 부분적인 합병을 통하여 퍼지 규칙의 수를 최적화한다. 또한, 본 논문에서는 생성된 퍼지 규칙들이 정형적인 형태를 유지하도록 하며, 이용상의 적응력을 높이기 위하여 누적-히스토그램을 이용하여 귀속 함수를 정의하는 방법을 제안한다.

ABSTRACT

This paper discusses how to automatically extract fuzzy rules from given data. The fuzzy space which contain given data are first subdivided into a set of hypercubes, each of which contains the homogeneous data belonging to the same class, and then a fuzzy rule is defined based on the constructed hypercube. In order to dynamically adjust the size of a hypercube, the fuzzy space is to be splitted based on a center vector and then the splitted subspaces are to be merged through the adjacency relation. The membership functions, which are to be embedded in a fuzzy rule, are to be formed through analyzing the cumulative histogram of given data along each axis of the constructed hypercube.

I. 서 론

오늘날 컴퓨터와 통신 기술의 급속한 발전은 산업사회를 정보화 사회로 탈바꿈시키고 있으며, 여러 분야에서 사용되던 모델 및 접근 방법의 원천적인 변화를 요구하고 있다. 특히, 패턴 인식 분야에서 퍼지 이론의 도입은 인간의 인식기능에 버금가는 컴퓨터 시스템의 구현을 가능하게 하고 있다. 퍼지 이론은 애매모호한 상황을 표현하기 위하여 퍼지 집합, 퍼지 논리, 퍼지 숫자등의 개념을 사용하며, 이러한 개념들에 대한 수학적 모델은 매우 잘 정립되어 있다. 이러한 퍼지 이론은 개발 초기부터 패턴 인식 분야에 강한 영향을 주었으며, 특히 다음과 같은 두 가지 측면에서 패턴 인식을 위한 매우 유용한 도구로서 인정받고 있다[3]. 첫째는 인식 단계에서의 퍼지 이론의 도입이다. 즉, 인간의 인지 행위를 모델링하기 위한 도구로서 퍼지 이론을 사용하는 것이다. 둘째는 학습 단계에서의 퍼지 이론의 도입이다. 즉, 복잡한 패턴 공간을 자연스런 그룹들의 집합으로 분할하기 위하여 퍼지 이론을 사용하는 것이다.

* 숭실대학교 전자계산학과

퍼지 이론은 패턴 인식 분야에서 중요한 역할을 담당하고 있으나, 이를 실제 응용 분야에 적용하기 위하여는 퍼지 규칙을 생성하는 방법이 강구되어야 한다. 현존하는 대부분의 퍼지 패턴 인식 시스템은 인간 전문가에 의하여 수작업으로 퍼지 규칙을 정의하고 이를 이용하고 있다. 따라서 퍼지 패턴 인식을 위한 연구의 많은 부분이 퍼지 규칙을 자동으로 생성하는 효율적인 방법의 확립에 초점을 두고 있는 실정이다[1, 2, 4, 7, 9, 10]. 본 논문에서는 패턴 인식 과정에 사용되는 퍼지 규칙을 자동으로 생성하는 방법을 제안한다. 즉, 각 퍼지항(fuzzy term)에 대한 귀속 함수(membership function)를 정의하는 방법과 정의된 퍼지항들을 규칙의 전제부와 결론부에 포함시켜 퍼지 규칙을 생성하는 방법에 대하여 기술한다. 각 퍼지항의 귀속 함수를 유도하기 위하여 본 논문에서는 군집화 기법(clustering technique)을 이용한다. 군집화는 주어진 자료들을 분석하여 동일한 특성을 갖는 자료들의 군집을 형성하는 작업이다. 즉, 주어진 자료 공간을 군집들의 집합으로 분할하는 작업이다. 이러한 군집화 기법은 기본적으로 탐색 기법이기에 때문에 그 기법도 다양할 수밖에 없다. 본 논문에서는 참고 문헌 [7, 9, 10]에서 제안된 방법을 개선하여 퍼지 공간을 분할하는 방법을 제안한다.

본 논문의 목적은 복잡한 패턴 인식 시스템의 퍼지 모델링을 위한 알고리즘적인 접근 방법의 모색에 있다. 이를 위하여 제 2 절에서는 주어진 퍼지 공간(자료 공간)을 퍼지 규칙으로 표현되기에 적합한 부공간들의 집합으로 분할하는 방법에 대하여 기술하고, 제안된 방법과 최근 문헌에 소개된 방법들을 비교 분석한다. 제 3 절에서는 제안된 방법에 의하여 분할된 각 부공간에 적합한 퍼지항을 정의하고, 정의된 퍼지항들을 이용하여 퍼지 규칙을 생성하는 방법에 대하여 기술한다. 제 4 절에서는 본 논문에서 제안한 방법을 실제 자료에 적용한 실험 결과에 대하여 기술하고, 제 5 절에서는 결론 및 향후 개선되어야 할 문제점에 대하여 기술한다.

II. 퍼지 공간의 분할

퍼지 규칙을 생성하기 위하여는 퍼지 공간상에 분포되는 자료들을 분석하여 공통된 특성을 갖는 집단으로 분류하고, 분류된 자료들을 기초로하여 퍼지 집합을 형성하는 작업을 수행하여야 한다. 이를 위한 일반적인 방법들은 다음의 세가지 유형으로 분류될 수 있다[5]. 첫째는 수작업에 의하여 퍼지 공간을 분할하고 분할된 공간에 기초하여 퍼지 규칙을 생성하는 방법이다. 즉, 인간 전문가의 경험적 지식에 기초하여 퍼지 규칙의 생성이 이루어진다. 이는 가장 많이 사용되는 방법으로서 인간 전문가의 판단이 정확할 때에는 매우 유용한 방법이지만, 모델링하고자 하는 시스템이 매우 복잡한 경우 또는 규칙 베이스의 갱신을 자주 요하는 경우에는 그 때마다 인간 전문가에 의하여 수작업으로 자료의 정제화를 수행하여야 하는 매우 번거로운 방법이다.

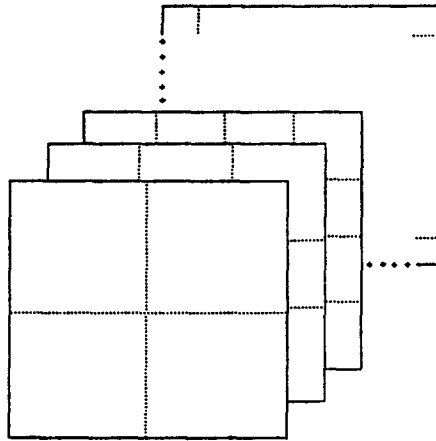
두번째 방법은 퍼지 공간에 분포된 자료들을 신경 회로망(neural network)의 학습 과정에 적용하여 퍼지 규칙을 생성하는 방법이다. 즉, 신경 회로망의 자기 조정 및 학습 기능을 퍼지 규칙의 생성 및 조율(tuning)에 도입하는 방법이다. 이는 퍼지 이론의 속성과 신경 회로망 이론의 속성을 비교할 때 매우 설득력이 있는 방법이지만, 학습 과정에서 매우 많은 계산 시간이 소요되는 단점이 있다.

세번째 방법은 퍼지 공간에 분포된 자료들을 퍼지 군집화(fuzzy clustering) 등과 같은 적절한 방법에 의하여 분할하고, 분할된 각 군집에 포함된 자료들의 특성을 분석하여 퍼지 규칙을 생성하는 방법이다. 본 논문에서 제안하는 방법은 세번째 유형의 방법으로서 주어진 퍼지 공간을 가상입방체(hypercube)들의 집합으로 분할하고, 분할된 각 가상입방체에 분포된 자료들의 누적-히스토그램(histogram)에 기초하여 퍼지 집합 및 퍼지 규칙을 생성한다.

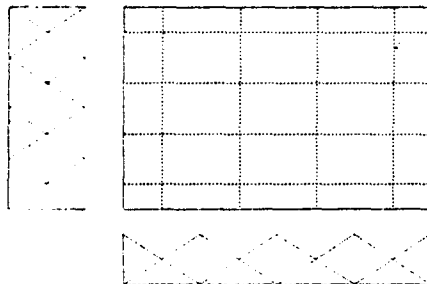
퍼지 공간을 체계적으로 분할하기 위하여는 분할 기준과 분할 방법이 정의되어야 한다. 즉, 주어진 퍼지 공간을 부공간들의 집합으로 분할할 것인가 또는 현 상태에서 분할을 종료할 것인가 하는 문제와 분할을 수행한다면 어떠한 형태의 부공간들을 생성할 것인가 하는 문제를 해결하여야 한다. 일반적으로 사용되는 분할 기준은 주어진 퍼지 공간에 포함된 모든 자료들이 하나의 집단(class)에 속하는 자료들인 경우 또는 그 자료의 수가 미리 정한 임계값보다 작은 경우에 분할을 종료하는 방식이다. 즉, 분할된 각 부공간에 포함된 자료들은 균일한 특성을 만족하여야 한다는 원칙을 분할 기준으로 채택한다. 분할된 각 부공간에 대하여 하나의 퍼지 규칙이 할당되므로 이러한 분할 기준은 매우 설득력이 있는 기준으로 평가된다. 본 논문에서도 이러한 분할 기준을 사용하여 퍼지 공간의 분할을 수행한다.

두번째 고려 사항인 분할 방식은 방법의 효율성(eficiency)과 최적성(optimality)에 기초하여 결정되어야 한다. 즉, 분할 방식은 다양한 유형의 자료들에 대하여 적용될 수 있으며 구현 과정이 용이하여야 한다. 동시에, 분할된 각 부공간은 퍼지 규칙으로 표현되기에 적합하여야 하며 부공간들의 수는 최소가 되어야 한다. 가장 보편적으로 사용되는 분할 방식은 퍼지 군집화에 의한 분할이다. 이 방법은 부공간의 형태에 대하여 제약을 받지 않으며 사전에 정의된 거리 척도(distance metric)을 이용하여 귀속 함수를 용이하게 도출할 수 있다는 장점이 있지만, 분할된 각 부공간을 대표하는 퍼지 규칙을 유도하는 작업이 복잡한 절차를 거쳐야하는 단점이 있다.

퍼지 공간을 계층적으로(hierarchically) 분할하기 위한 시도로서 주어진 n -차원 퍼지 공간을 동일한 크기의 K^n 개의 부공간으로 분할하는 방식이 참고 문헌 [9, 10]에서 제안되었다. K 는 각 차원을 분할하는 구간의 수를 말한다. 주어진 퍼지 공간에 가장 적합한 K 를 구하기 위하여 작은 값을 갖는 K 에서 시작하여 K 의 값을 점진적으로 증가하며 분할된 부공간에 포함된 자료들의 분포를 분석하는 방법을 사용하였다. 이러한 방법은 분할된 부공간들이 모두 동일한 크기를 가지므로 각 부공간을 대표하는 퍼지 집합 및 퍼지 규칙을 생성하기가 용이하나, 생성된 퍼지 집합 및 퍼지 규칙의 유용성은 상대적으로 낮게되는 문제점이 있다. (그림 2-1)은 2차원 퍼지 공간을 분할된 각 부공간에 타 집단의 자료가 포함되지 않을 때 까지 동일한 크기의 4개, 9개, 16개의 부공간으로 분할하는 과정을 보여준다. 분할된 부공간들은 모두 동일한 크기를 가지므로, 각 부공간은 자료의 분포 특성을 반영하지 못하게 되며 각 부공간을 대표하는 귀속 함수의 형태는 (그림 2-2)에서와 같이 동일한 형태로 정의된다.

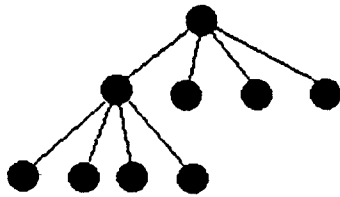


(그림 2-1) 동일한 크기의 부공간들에 의한 퍼지 공간의 분할

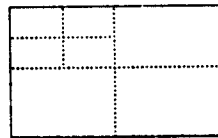


(그림 2-2) 동일한 크기의 부공간들에 대한 귀속 함수

참고 문헌 [7]에서는 참고 문헌 [9, 10]에서 제안된 분할 방식을 개선한 방법을 제안하였다. 즉, 주어진 n -차원 퍼지 공간을 동일한 크기의 2^n 개의 부공간으로 분할하고, 분할된 각 부공간에 대한 분할을 동일한 방법으로 반복하는 분할 방식이다. 각 부공간에 대한 분할이 N 번 반복되는 경우에 생성되는 부공간의 총수는, 즉 퍼지 규칙의 총수는 $2^N \cdot 2^n$ 개가 된다. 이 방법은 퍼지 공간의 분할을 재귀적으로(recursively) 수행하므로 분할을 체계적으로 수행할 수 있다는 장점이 있다. 그러나, 하나의 공간을 분할하여 만들어지는 2^n 개의 부공간들은 모두 동일한 크기를 갖게되어 자료들의 분포 특성이 반영되지 못하는 단점이 있다. (그림 2-3)은 2차원 퍼지 공간에서 재귀적으로 분할이 수행되는 과정을 보이는 분할 트리와 재귀적인 분할에 의하여 생성된 부공간들을 보여준다. 즉, 분할 기준을 만족하는 평면에 대하여 반복적으로 동일한 크기의 사분면(quadrant)으로 분할하는 방식으로서, 이러한 분할이 수행되는 과정을 나타내는 분할 트리는 사진 트리(quadtrees)의 형태를 취한다.



(a) 2차원 퍼지 공간에 대한 분할 트리



(b) 분할된 2차원 평면

(그림 2-3) 동일한 크기의 부공간을 갖는 재귀적 분할 방식

본 논문에서는 n -차원 퍼지 공간의 분할을 재귀적으로 수행하고, 동시에 분할된 각 부공간의 형태가 그 부공간에 포함된 자료들의 분포 특성을 반영하는 (알고리즘)과 같은 분할 방식을 제안한다. 즉, 분할 기준을 만족하는 n -차원 퍼지 공간을 동일한 크기의 2^n 개의 부공간으로 분할하지 않고, 그 공간에 분포된 좌표들의 중심 벡터 $C = (C_1, C_2, \dots, C_n)$ 를 기준으로하여 2^n 개의 부공간으로 분할한다.

< 알고리즘 2-1 >

```

/* 주어진 자료들에 대하여 퍼지 공간의 각 축상의 최소 및 최대값 획득
   ==> 초기 공간 설정(최소 가상 입방체(Minimal Hypercube) */
 $X^0(i)_{MIN} = MIN(x(i)_1, x(i)_2, \dots, x(i)_M); \quad X^0(i)_{MAX} = MAX(x(i)_1, x(i)_2, \dots, x(i)_M)$ 

/* 작업 종료 검토 */
label1:
  if (number of data in the subspace < Th)
    then goto label2 :
  if (all data in the subspace belong to same class)
    then goto label2 :
  else goto label3 :

label2:
  if (it is a last subspace)
    then stop :
  else
  {
    choose next subspace to be processed :
    goto label1 :
  }
  
```

```

}
/* 주어진 부공간에 대한 중심 벡터 C = (C1, C2, ..., Cn) 추출 */
label3:

      
$$C_i = \frac{\sum_j x(i)_j}{\text{total number of data in the subspace}}$$

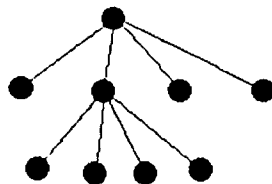

/* 주어진 부공간에 대한 분할
   => 중심 벡터를 기준으로 2^n개의 부공간 생성 */

X^{k+1,1}(i)_{MIN} = X^k(i)_{MIN};   X^{k+1,1}(i)_{MAX} = C_i;
X^{k+1,2}(i)_{MIN} = C_i;         X^{k+1,2}(i)_{MAX} = X^k(i)_{MAX};

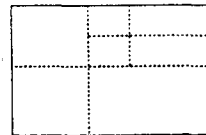
goto label1 :
    
```

(알고리즘 2-1)에서 x_j 는 주어진 부공간에 포함된 j-번째 자료로서, 각 자료는 n-차원 벡터로서 표현된다. 즉, $x(i)_j$ 는 x_j 의 i-번째 차원의 좌표값을 나타낸다. $X^0(i)_{MIN}$ 과 $X^0(i)_{MAX}$ 은 초기 공간을 나타내는 가상입방체의 i-번째 차원의 최소값과 최대값을 나타낸다. $X^k(i)_{MIN}$ 과 $X^k(i)_{MAX}$ 은 k-번째 가상입방체의 i-번째 차원의 최소값과 최대값을 나타내고, $X^{k+1,1}(i)_{MIN}$ 과 $X^{k+1,1}(i)_{MAX}$ 은 k-번째 가상입방체를 중심 벡터를 기준으로하여 분할할 때 i-번째 차원의 첫번째 구간을 나타내고 $X^{k+1,2}(i)_{MIN}$ 과 $X^{k+1,2}(i)_{MAX}$ 은 두번째 구간을 나타낸다.

(그림 2-4)는 2차원 평면으로 정의되는 퍼지 공간을 (알고리즘 2-1)을 이용하여 중심 벡터를 기준으로 분할한 예를 보여준다. (그림 2-3)과 (그림 2-4)를 비교하면, (그림 2-3)은 (그림 2-4)의 특별한 경우에 해당됨을 알 수 있다. 따라서, 참고 문헌 [7]에서 정의된 분할 방식은 본 논문에서 제안하는 방식의 특별한 경우로 간주할 수 있다. 이처럼 중심 벡터를 기준으로 분할을 수행하는 이유는 다음과 같은 원칙을 전제로 한다. "각 집단(class)에 속하는 자료들은 비교적 균일한 특성을 만족하며 퍼지 공간에서 적절한 형태의 군집을 형성한다." 따라서, 주어진 퍼지 공간이 분할 기준을 만족한다면, 그 공간에 포함된 자료들은 적어도 두개 이상의 집단에 속하고, 각 집단은 적절한 중심 벡터 주위에 밀집되어 분포된다. 만약, 퍼지 공간에 포함된 자료들이 두개의 집단에 속하는 자료들인 경우에는 이 자료들의 중심 벡터가 두 집단을 분리하는 경계 조건으로 매우 적합할 것이다.



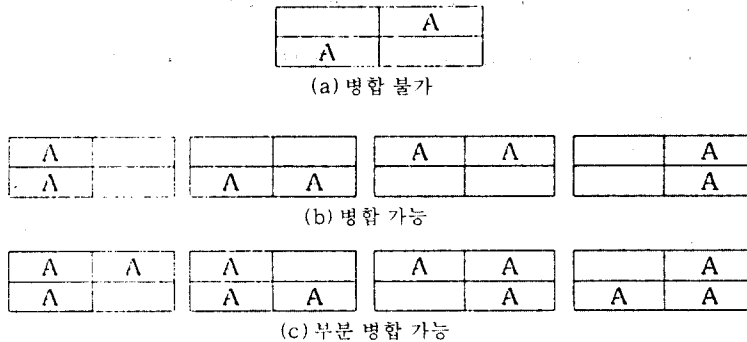
(a) 2차원 퍼지 공간에 대한 분할 트리



(b) 분할된 2차원 평면

(그림 2-4) 가변 크기의 부공간을 갖는 분할 방식

반면에, 세개 이상의 집단에 속하는 자료들이 하나의 퍼지 공간에 포함되는 경우에는 이 자료들의 중심 벡터는 집단간의 경계를 정의하는 조건으로서 적합하지 못할 수 있다. 또한, 하나의 n -차원 퍼지 공간은 무조건 2^n 개의 부공간으로 분할되므로 하나의 집단이 복수 개의 부공간들로 분산될 수 있다. 즉, 분할된 여타 공간들이 합해질 수 있는 가능성을 고려하지 않고, 타집단의 자료가 동일한 부공간에 포함되지 않도록 분할을 수행한 결과를 초래한다. 예를 들어, 평면으로 정의되는 2차원 퍼지 공간에서 분할된 각 부공간은 (그림 2-5)의 (a), (b), 그리고 (c)의 형태로 생성될 수 있다. (그림 2-5)에서 레이블 'A'를 갖는 부공간들은 동일한 집단에 속하는 자료들을 포함한다. 이 중에서 (b)와 (c)의 경우를 보면, 최종 분할된 각 부공간이 이웃하는 부공간과 병합될 수 있는 가능성을 지니게 된다. 이러한 문제점을 보상하기 위하여 본 논문에서 제안하는 방식은 분할된 부공간들에 대하여 (알고리즘 2-2)에 의한 병합 작업(merge)을 수행한다. 즉, 분할된 2^n 개의 부공간들 중에서 면을 접하고 있는 부공간들을 조사하여 이들이 동일한 집단에 속하는 자료들을 포함하는 경우에는 이들을 병합한다. (그림 2-5)는 2차원 평면으로 정의되는 퍼지 공간에서 병합 작업이 불필요한 경우 (a), 병합 작업이 절대적으로 요구되는 경우 (b), 인접한 두 개의 부공간에 대한 부분적인 병합 작업이 요구되는 경우 (c)의 예를 보여준다. 즉, 분할된 각 부공간에 인접한 두 개의 부공간들이 병합 후보로 고려될 수 있다. 일반적으로 n -차원 퍼지 공간인 경우에는 하나의 부공간에 병합될 수 있는 후보 공간의 수는 n 개가 된다.



(그림 2-5) 인접한 부공간들이 동일한 집단에 속하는 예

병합 작업을 수행하는 (알고리즘 2-2)는 분할 트리에서의 각 부공간의 위치에 따라 다음과 같은 세 단계로 구성된다. 첫단계는 병합 대상이 되는 두 부공간의 부모 노드가 같은 경우이다. 즉, 동일 구역내에서 인접한 부공간들에 대한 병합을 수행하는 단계이다. 두번째 단계는 병합 대상이 되는 두 부공간의 부모 노드는 서로 다르지만, 부공간들이 분할 트리의 같은 레벨에 이웃하여 위치하는 경우이고, 세번째 경우는 병합 대상이 되는 부공간들이 분할 트리의 서로 다른 레벨에 위치하는 경우이다.

(알고리즘 2-2)에서 label(sub-k)은 부공간-k의 레이블을 돌려주는 함수이고, adjacent(sub-k,sub-l)은 부공간-k와 부공간-l의 인접성을 검사하는 부울 함수이며, merge(sub-k,sub-l)은 부공간-k와 부공간-l에 대한 병합 작업을 수행하는 함수이다. (그림 2-6-a)는 2차원 퍼지 공간상에서 (알고리즘 2-2)의 첫단계에 대한 예를 보여준다. 즉, 병합되어질 부공간 s_1 과 부공간 s_2 는 동일한 부모 노드를 갖으며 서로 인접하고 있다. (그림 2-6-b)는 (알고리즘 2-2)의 두번째 단계의 예로서, 병합 대상이 되는 두 부공간의 부모 노드는 서로 다르지만, 부공간들이 분할 트리의 같은 레벨에 이웃하여 위치하는 경우이다. (그림 2-6-c)는 (알고리즘 2-2)의 세번째 단계의 예로서, 부공간 s_1 은 분할 트리의 레벨 3에 위치하고 s_2 는 분할 트리의 레벨 2에 위치한다.

(알고리즘 2-2)의 첫번째 단계의 병합은 병합하고자 하는 각 부공간을 정의하는 N 개의 축 중에서 $N-1$ 개의 축은 공통되는 범위를 갖으므로, 공통된 범위를 갖지 않는 나머지 한 축의 범위를 합집합 연산하므로서 병합된 공

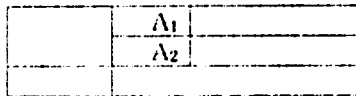
```

        < 알고리즘 2-2 >

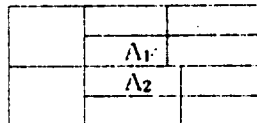
/* 동일한 부모 노드를 갖는 부공간들의 병합 */
STEP 1:
for (each pair of subspaces sub-k and sub-l with same parent node)
{
    if { (label(sub-k) == label(sub-l)) && adjacent(sub-k,sub-l) }
    then merge(sub-k,sub-l)
}

/* 분할 트리의 같은 레벨에 위치하는 부공간들의 병합 */
STEP 2:
for (each pair of sub-k and sub-l at the same level of tree, but with
    different parent nodes)
{
    if { (label(sub-k) == label(sub-l)) && adjacent(sub-k,sub-l) }
    then merge(sub-k,sub-l)
}

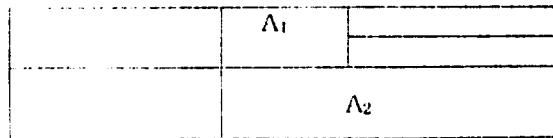
/* 분할 트리의 다른 레벨에 위치하는 부공간들의 병합 */
STEP 3:
for (each pair of sub-k and sub-l at the different level of tree)
{
    if { (label(sub-k) == label(sub-l)) && adjacent(sub-k,sub-l) }
    then merge(sub-k,sub-l)
}
    
```



(a) 첫번째 단계의 병합 예



(b) 두번째 단계의 병합 예



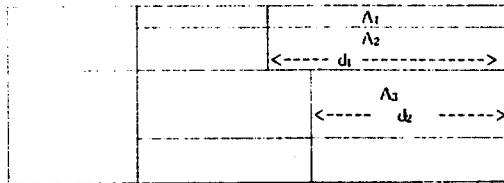
(c) 세번째 단계의 병합 예

(그림 2-6) (알고리즘 2-2)의 각 단계별 병합 처리 예

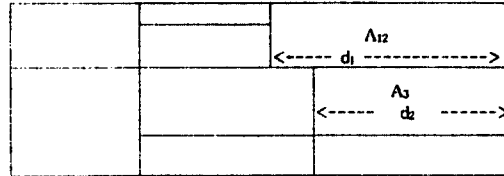
간이 하나의 가상입방체(hypercube)로 정의될 수 있다. 그러나, 두번째 및 세번째 단계의 병합에서는 각 부공간을 정의하는 N 개의 축 중에서 복수 개의 축들이 공통된 범위를 갖지 않을 수 있으므로, 병합된 공간이 가상입방체의 형태를 취하기 위하여는 병합된 공간에 대한 영역 조정 작업이 수행되어야 한다. 병합 작업이 수행되는 과정을 설명하기 위하여 (그림 2-7)을 생각하자. (그림 2-7-a)는 분할 작업을 수행하여 생성된 부공간들을 보여준다.

(그림 2-7-a)에서 부공간 A_1 , A_2 , 그리고 A_3 는 동일한 집단에 속하는 자료들을 포함한다고 가정하자. 이들에 대한 병합 작업은 두 단계로 나누어 진행된다. 먼저 일단계 병합 작업에서 A_1 과 A_2 에 대한 병합이 이루어진다.

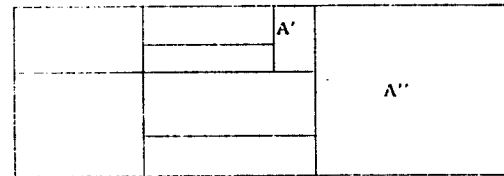
(그림 2-7-b)는 이들을 병합하여 생성된 부공간 A_{12} 를 보여준다. 이제 이단계 병합 작업에서 A_{12} 와 A_3 를 병합하는 문제를 생각하자. A_{12} 를 정의하는 수평 축의 범위 d_1 과 A_3 를 정의하는 수평 축의 범위 d_2 가 일치하지 않는다. 따라서, 영역 조정 작업을 수행하여 (그림 2-7-c)에서와 같이 부공간 A' 과 부공간 A'' 을 최종적인 영역으로 결정한다. 영역 조정 작업은 중복 되는 범위 d_1 과 d_2 를 비교하여 크기가 작은 d_2 를 기준으로 하여 부공간 A'' 을 우선적으로 형성한 후, 병합에서 제외된 나머지 영역을 A' 으로 한다. 이때 병합에서 제외된 나머지 영역 A' 에 포함된 자료의 수가 미리 정한 임계값 이하인 경우에는 이 영역은 무시된다. 영역 조정 과정에서 크기가 작은 범위를 기준으로 하여 병합 작업을 수행하는 이유는 병합된 부공간이 다른 부공간을 포함하는 경우를 배제하기 위함이다.



(a) 병합이 가능한 부공간들의 배치도



(b) 일단계 병합 작업 수행 후의 배치도



(c) 이단계 병합 작업 수행 후의 배치도

(그림 2-7) 두 단계에 걸쳐서 처리되는 병합 작업의 예

(알고리즘 2-2-1)은 위에서 기술된 영역 조정 과정을 알고리즘 형태로 정리한 결과이다. 이 알고리즘은 (알고리즘 2-2)에서 사용되는 함수 $merge()$ 에 의하여 호출된다. (알고리즘 2-2-1)에서 $range(arg-1, arg-2)$ 는 부공간 $arg-2$ 를 정의하는 가상입방체의 $arg-1$ 차원의 범위를 나타내는 함수이다.


```

< 알고리즘 2-2-1 >

iadj := index of axis on which sub-k and sub-l are adjacent
inonadj := index of axis on which sub-k and sub-l are not adjacent
msub-j := j-th subspace formed by merging sub-k and sub-l, l

N = 1;
range(iadj, msub-N) = ∩ [range(iadj, sub-k), range(iadj, sub-l)];
range(inonadj, msub-N) = ∪ [range(inonadj, sub-k), range(inonadj, sub-l)];

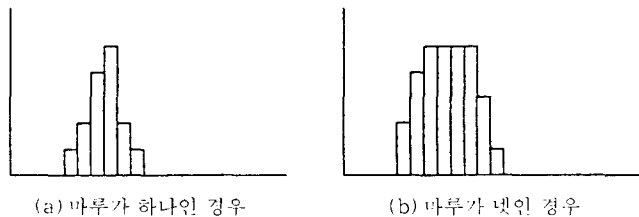
while ( ( ∪j=k,l sub-j ) ≠ ( ∪j=1N msub-j ) )
{
    N = N + 1;
    msub-N := hyperspere covered by ( ∪j=k,l sub-j ) - ( ∪j=1N msub-j )
}
    
```

Ⅲ. 퍼지 규칙의 생성

본 절에서는 분할된 각 부공간을 대표하는 퍼지 규칙을 생성하는 과정을 기술한다. 일반적으로 퍼지 규칙은 한 개의 입력이 하나의 출력을 결정하는 경우(SISO: Single Input Single output), 복수 개의 입력이 하나의 출력을 결정하는 경우(MISO: Multiple Input Single Output), 복수 개의 입력이 복수의 출력을 결정하는 경우(MIMO: Multiple Input Multiple Output) 등의 세가지 유형으로 분류될 수 있다. 본 논문에서 정의하는 퍼지 규칙은 (식 3-1)에서와 같이 복수 개의 입력이 하나의 출력을 결정하는 경우로 한정한다. 따라서, 퍼지 공간이 M 개의 부공간들의 집합으로 분할되는 경우, 시스템의 행위는 (식 3-1)과 같은 M 개의 퍼지 규칙들(M 퍼지 함축)의 집합에 의하여 정의된다.

$$\begin{aligned}
 R_j: & \text{IF } (h_1 \text{ is } A_1^{r_j} \text{ AND } h_2 \text{ is } A_2^{r_j} \text{ AND } \dots \text{ AND } h_k \text{ is } A_k^{r_j}) \\
 & \text{THEN } y \text{ is } B^{r_j}, \text{ where } r_j \in \{1, 2, 3, 4, 5\}
 \end{aligned}
 \tag{3-1}$$

(식 3-1)에서 R_j 는 j 번째 부공간을 대표하는 퍼지 규칙을 나타내고, h_i 는 이 부공간의 i 번째 차원에 대한 입력값을 나타내고, $A_i^{r_j}$ 는 이 부공간의 i 번째 차원에서 정의되는 퍼지 집합을 나타낸다. 본 논문에서는 각 퍼지 집합에 대한 귀속함수의 형태는 삼각형 또는 사다리꼴로 제한한다. 이제 각 부공간에 포함된 자료들의 분포 특성을 분석하여 귀속 함수를 정의하는 과정을 생각하자. 귀속 함수의 결정은 (알고리즘 3-1)을 이용하여 수행된다. 즉, 부공간에 포함된 자료들을 각 축에 대하여 누적시킨 후, 누적된 자료들에 대한 히스토그램을 작성한다. 작성된 히스토



(그림 3-1) 각 축에 투영된 자료에 대한 평활화된 히스토그램

그램을 평활화시킨 후, 평활화된 히스토그램의 형태를 분석하여 삼각형 또는 사다리꼴의 귀속 함수를 정의한다. 즉, 평활화된 히스토그램이 하나의 마루(peak)를 갖는 경우에는 삼각형의 귀속 함수를 정의하고 (그림 3-1-a), 마루의 갯수가 복수이면 사다리꼴의 귀속 함수를 정의한다 (그림 3-1-b).

귀속 함수의 유형이 결정되면 이제 각 유형을 정의하는 사선의 시작점과 끝점을 결정하므로 귀속 함수에 대한 정의가 완료된다. 본 논문에서는 오른쪽 사선의 시작점은 가장 오른쪽 마루의 좌표로 하고, 왼쪽 사선의 시작점은 가장 왼쪽 마루의 좌표로 한다. 사선의 끝점은 이웃하는 부공간이 같은 집단에 속하는 자료를 포함하는지를 판단하여 결정한다. 즉, 이웃하는 부공간이 같은 집단에 속하는 자료를 포함하는 경우에는 그 부공간의 누적 히스토그램을 작성하고, 작성된 히스토그램의 가장 왼쪽 마루의 좌표와 가장 오른쪽 마루의 좌표를 사선의 왼쪽 끝점 및 오른쪽 끝점으로 한다.

일반적으로 많은 시스템들 [7, 9, 10]은 획일적인 귀속 함수를 사용한다. 그러나, 본 논문에서는 자료의 분포 특성을 분석하여 귀속 함수의 유형을 결정하고, 귀속 함수의 형태 역시 자료의 분포 특성 및 인접 부공간의 특성을 분석하여 결정한다. 따라서, 귀속 함수의 결정 과정이 좀 복잡한 단점은 있으며, 결정된 귀속 함수의 문제에 대한 적응성은 매우 높을 것으로 판단된다.

```

< 알고리즘 ( 3 - 1 ) >

/* 누적-히스토그램 구성 */
STEP 1.
for (i = XMIN; i <= XMAX; i++)
    H[F[i]]++;

/* 평활화 작업 */
STEP 2.
for (i = XMIN; i <= XMAX; i++)
    S(i) = ( H(F(i-1)) + H(F(i)) + H(F(i+1)) ) / 3;

/* 귀속 함수형 결정 */
STEP 3.
if (number of peaks == 1)
{ /* membership function is triangular type */
    if (there exists adjacent subspace with same label)
    {
        left side is from one's peak to neighbor's peak;
        right side is from one's peak to neighbor's peak;
    }
    else
    {
        left side is from one's peak to neighbor's center;
        right side is from one's peak to neighbor's center;
    }
}
else /* membership type is trapezoidal type */
{
    if (there does not exist adjacent subspace with same label)
    {
        left side is from one's leftmost peak to neighbor's center;
        right side is from one's rightmost peak to neighbor's center;
    }
    else
    {
        left side is from one's leftmost peak to neighbor's
        leftmost peak;
        right side is from one's rightmost peak to neighbor's
        rightmost peak;
    }
}
}

```

완전한 퍼지 규칙을 생성하기 위해서는 앞에서 정의된 귀속함수에 대한 퍼지 집합을 언어학적으로 표현하여야 한다. 전문가에 의한 언어학적 표현은 실 환경의 주변 여건에 적합한 다양한 단어로 나타낼 수 있다. 즉, 퍼지 규칙상에 명시되는 단어는 폭넓고 깊은 의미를 갖는 어휘로 표현될 수 있다. 그러나, 전문가가 아닌 기계에 의한 표현은 다양한 어휘를 사용하기가 현재로는 불가능하다. 따라서, 퍼지 규칙의 자동 생성이라는 취지에서는 한정된 단어를 모든 환경에 적용하여야 하는 제한성을 지닌다. 본 논문에서 사용하고자 하는 단어는 “매우 낮다(Very Low)”, “낮다(Low)”, “조금 낮다(High Low)”, “낮은 중간(Low Medium)”, “정 중앙(Medium)”, “높은 중간(High Medium)”, “조금 높다(Low High)”, “높다(High)”, “매우 높다(Very High)” 등으로 제한한다.

산출된 귀속함수에 대응된 단어로 퍼지 규칙을 언어학적으로 표현하면 ‘if h_1 is Low’, ‘if h_2 is High’, ‘if h_3 is Medium’ 등과 같이 표현된다. 궁극적으로 주어진 한 가상입방체에 대한 퍼지 규칙이 생성되려면, 퍼지 규칙의 전제부는 해당 공간에 속하는 특징들에 관한 퍼지항들을 논리적 곱연산(logical AND)에 의하여 나열하면 된다.

if h_1 is Small then
 if h_2 is Large then

 if h_n is Medium then
 if h_1 is Small AND h_2 is Large AND
 AND h_n is Medium then

전제부가 생성된 후에는 각 가상입방체에 대한 퍼지 규칙이 (식 3-1)과 같은 형태를 갖추기 위하여 생성된 전제부에 대응되는 결론부의 도출이 요구된다. 본 논문에서는 전제부의 각 퍼지항을 정의하는 축(axis)의 범위에 포함되는 자료수에 기초한 가중치를 이용하여 규칙의 결론부를 생성한다. 규칙의 결론부를 생성하는 과정은 먼저 부공간의 각 축의 범위에 포함되는 자료의 수에 기초하여 각 퍼지항에 대한 가중치를 산출하고, 산출된 각 퍼지항에 대한 가중치를 종합하여 결론부를 생성한다. 각 퍼지항에 대한 가중치는 (식 3-2)에 의하여 정의된다. 즉, 퍼지항을 정의하는 축의 범위에 포함된 자료 갯수를 그 부공간이 대표하는 집단의 자료의 총 갯수로 나눈 백분율로 한다. 따라서, 각 퍼지항에 대한 가중치는 ‘0’과 ‘1’ 사이의 값을 가지며, 동일한 집단을 나타내는 부공간들의 동일한 축에서 정의되는 퍼지항들에 할당된 가중치들을 총계하면 ‘1’이 된다. (식 3-2)에서 $N_k(h_l(A_l^j))$ 는 분할된 l 번째 부공간내에서 퍼지항 A_l^j 이 정의되는 축의 범위 h_l 에 포함되는 자료의 수를 나타내고, $W(h_l(A_l^j))$ 는 퍼지항 A_l^j 에 할당된 가중치를 나타낸다.

$$W(h_l(A_l^j)) = \frac{\sum_{k=1}^m N_k(h_l(A_l^j))}{\sum_{l=1}^n (\sum_{k=1}^m N_k(h_l(A_l^j)))} \quad (3-2)$$

$$\sum_{l=1}^n W(h_l(A_l^j)) = 1 \quad (l = 1, \dots, n; k = 1, \dots, m)$$

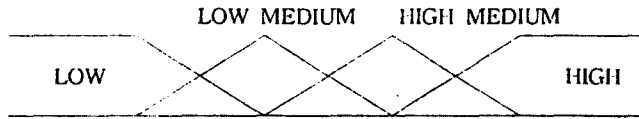
(식 3-2)에 의하여 각 퍼지항에 대한 가중치가 산출되면, 산출된 각 퍼지항에 대한 가중치를 (식 3-3)에 의하여 종합하여 퍼지 규칙의 결론부를 생성한다.

$$\begin{aligned} W(Y_{r_1}) &= W(h_1(A_1^1)) \times W(h_2(A_2^1)) \times \dots \times W(h_n(A_n^1)) \\ W(Y_{r_2}) &= W(h_1(A_1^2)) \times W(h_2(A_2^2)) \times \dots \times W(h_n(A_n^2)) \\ &\dots\dots\dots \\ W(Y_{r_m}) &= W(h_1(A_1^m)) \times W(h_2(A_2^m)) \times \dots \times W(h_n(A_n^m)) \end{aligned} \quad (3-3)$$

(식 3-3)에서 Y_{r_i} 는 i -번째 규칙의 결론부에서 도출하는 변수를 나타내고, $W(Y_{r_i})$ 는 그 변수에 대한 가중치를 나타낸다. 패턴 인식의 측면에서는 규칙의 결론부를 단순히 $W(h_{r_i}(A_{r_i}^{r_i}))$ 로 정의하여 사용할 수 있다. 즉, 퍼지 규칙의 형태를 (식 3-4)와 같이 정의할 수 있다.

$$R_j : \text{IF } (h_1 \text{ is } A_1^{r_1} \text{ AND } h_2 \text{ is } A_2^{r_2} \text{ AND } \dots \text{ AND } h_k \text{ is } A_k^{r_k}) \\ \text{THEN } y \text{ is } W(h_{r_i}(A_{r_i}^{r_i})), \text{ where } r_i \in \{1, 2, 3, 4, 5\} \quad (3-4)$$

그러나, 퍼지 규칙의 형태를 (식 3-1)과 같은 형태로 정의하기 위하여는 앞에서 정의된 가중치를 이용하여 결론부에 대한 언어학적인 퍼지함을 도출하여야 한다. 결론부의 퍼지함에 대한 귀속 함수를 정의하기 위하여는 전제부의 퍼지함을 정의하기 위하여 사용되었던 자료 분포에 대한 누적 히스토그램을 사용할 수 없다. 따라서, 본 논문에서는 결론부의 퍼지함에 대한 귀속 함수의 유형은 삼각 함수 형태로 규격화하고, 각 귀속 함수의 범위를 나타



(그림 3-2) 결론부의 귀속함수 형태

< 알고리즘 3 - 2 >

/* 각 전제부의 각각의 귀속함수에 대한 가중치 산정 */

$$W(h_l(A_l^{r_l})) = \frac{\sum_{k=1}^m N_k(h_l(A_l^{r_l}))}{\sum_{l=1}^n (\sum_{k=1}^m N_k(h_l(A_l^{r_l})))}$$

$$\sum_{l=1}^n W(h_l(A_l^{r_l})) = 1$$

($l = 1, \dots, n ; j = 1, \dots, l ; k = 1, \dots, m$)

/* 각 전제부에 대한 가중치 계산 */

$$W(Y_{r_n}) = W(h_1(A_1^{r_1})) \times W(h_2(A_2^{r_2})) \times \dots \times W(h_n(A_n^{r_n}))$$

$$W(Y_{r_{n-1}}) = W(h_1(A_1^{r_1})) \times W(h_2(A_2^{r_2})) \times \dots \times W(h_{n-1}(A_{n-1}^{r_{n-1}}))$$

$$W(Y_{r_m}) = W(h_1(A_1^{r_1})) \times W(h_2(A_2^{r_2})) \times \dots \times W(h_m(A_m^{r_m}))$$

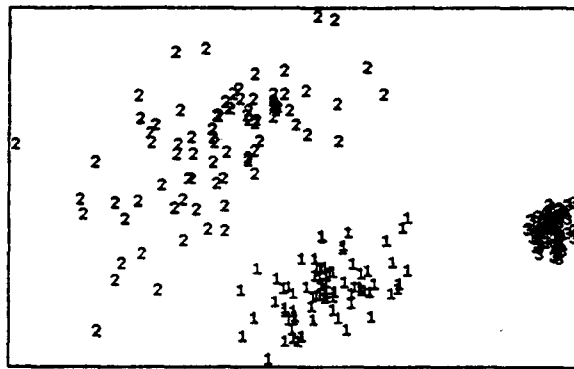
/* 산출된 가중치 값의 큰 순서로 표현할 언어 단어 매칭 */

```
for ( i=1; i < n; i++)
{
    K = MAX( W(Y_1), W(Y_2), W(Y_3), ..., W(Y_n) );
    if( K == 0 ) stop;
    for( j=1; j < n; j++)
        if( K == W(Y_j) ) W(Y_j) = 0;
    print table(K);
}
```

내는 언어적 표현은 각 집단을 정의하기 위하여 산출된 규칙의 수와 각 규칙의 결론부에 대한 가중치에 기초하여 정의된다. 예를 들어, 집단 A에 대한 두 개의 퍼지 규칙이 만들어지고, 각 규칙의 결론부에 대한 가중치 $W(Y_1)$, $W(Y_2)$ 이 산출된 경우를 가정하자. 만약, 이들 가중치의 값이 상이하고 $W(Y_1)$ 이 $W(Y_2)$ 보다 작은 값을 갖는다면, 규칙-1의 결론부 Y_1 에 대한 퍼지항은 'Low Medium'으로 표현되고, 규칙-2의 결론부 Y_2 에 대한 퍼지항은 'High Medium'으로 표현되고, 이들 퍼지항에 대한 귀속 함수는 (그림 3-2)와 같은 형태를 갖게된다. 이상에 기술된 퍼지 규칙의 결론부를 생성하는 과정을 알고리즘의 형태로 정리하는 (알고리즘 3-2)와 같이 표현될 수 있다.

IV. 실험

본 논문에서 제안된 방법의 효율성을 검증하는 실험은 IBM 호환 기종인 PC 486급에서 전자펜(electronic pen)을 통하여 입력된 서명 자료(signature)를 실험 자료로 하여 구현되었다. (그림 4-1)은 전자펜을 통하여 입력된 3인의 서명 자료에서 두 가지 특징을 추출한 후, 추출된 특징값들을 X 및 Y축상에 나타낸 것이다. 시각적인 효과를 위하여 각 특징값은 150배로 확대되었으며, 화면의 크기는 3배로 확대하여 출력하였다. 일부 특징값들은 매우 유사하여 같은 위치에 중복되어 출력되었다.

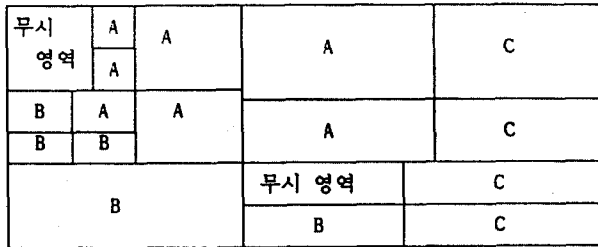


(그림 4-1) 실험에 사용된 3인의 서명 자료들의 퍼지 공간

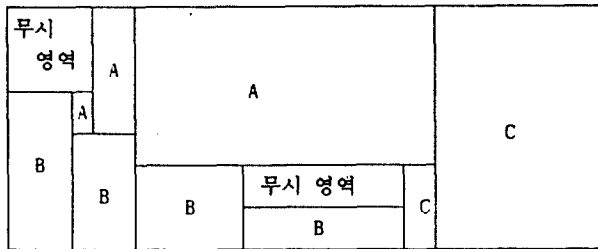
3인의 서명 자료들을 포함하는 퍼지 공간은 첫 단계에서 $(-1.424164, 0.314359)$ 를 중심으로 사분할된다. 이 단계에서 영역-1은 타 집단의 자료가 포함되지 않는 영역으로 확인되었다. 따라서 공간 분할의 두번째 단계에서는 영역-1을 제외한 나머지 세 영역(영역-2, 영역-3, 영역-4)이 각각 사분할되었다. 영역-2는 $(-1.801575, 0.733084)$ 을 중심으로 분할되어 영역-23과 영역-24가 고유 집단에 속하는 자료들만을 포함하는 영역으로 판정되었고, 세번째 단계에서 유사한 방법으로 영역-21과 영역-22가 사분할되었다. 영역-21에 대한 분할은 하위 레벨에서 각기 고유 집단의 자료를 포함하는 영역으로 종결되었다. 영역-22의 하위 레벨인 영역-223과 영역-224는 고유 집단에 속하는 자료를 포함하므로 더 이상 분할되지 않으며, 영역-221과 영역-222 역시 고유 집단에 속하는 자료들만을 포함하지만 이들 영역에 포함된 자료의 수가 미리 정의된 임계치 이하이므로 이들은 무시 영역으로 처리되었다. 즉, 영역-221과 영역-222는 퍼지 규칙이 퍼지 규칙이 생성되지 않는 영역으로 처리된다. 다른 영역들에 대하여도 유사한 방법으로 분할을 반복하면 최종적으로 (그림 4-2)와 같은 부공간들이 생성된다.

분할된 부공간들을 병합하는 작업은 퍼지 규칙의 수를 줄이기 위하여 절대적으로 필요한 조치이다. (그림 4-2)의 부공간들에 대하여 본 논문에서 제시한 병합 알고리즘을 적용하여 영역 재조정을 거친 결과는 (그림 4-3)과 같은 형태로 나타난다.

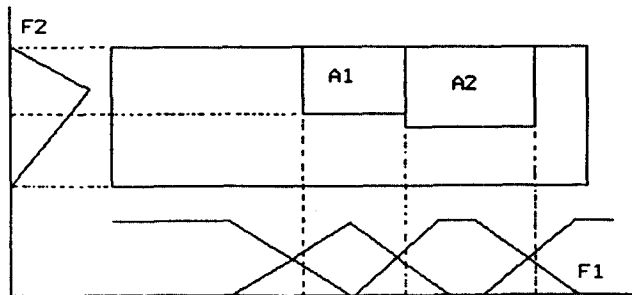
(그림 4-2)에 나타난 부공간들에 대하여 퍼지 규칙을 생성하면 생성되는 퍼지 규칙의 총 수는 16개가 되지만, 병합 작업에 의하여 영역이 재조정된 (그림 4-3)에 대하여 퍼지 규칙을 생성하면 생성되는 퍼지 규칙의 총 수는 9개로 7개 가량이 줄어든 것을 알 수 있다. 이제 생성된 각 부공간을 대표하는 퍼지 규칙을 정의하기 위하여 먼저 부공간에 포함된 자료들을 각 축에 대하여 누적시킨 후, 누적된 자료들에 대한 히스토그램을 작성한다. 작성된 히스토그램을 평활화시킨 후, 평활화된 히스토그램의 형태를 분석하여 삼각형 또는 사다리꼴의 귀속 함수를 정의한다. 예를 들어, (그림 4-3)에서 레이블 A로 표시된 부공간들의 각 축에 대한 귀속 함수는 (그림 4-4)와 같이 정의된다.



(그림 4-2) 분할된 부공간들의 배치도



(그림 4-3) 병합된 부공간들의 배치도

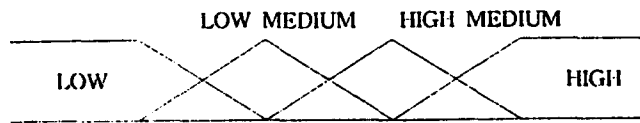


(그림 4-4) 자료 분포 특성에 기초한 귀속 함수

귀속 함수가 결정되면 결정된 귀속 함수를 이용하여 퍼지 규칙의 전제부를 형성한다. 전제부가 형성된 후에는 형성된 전제부의 각 퍼지항에 대한 가중치를 산출하고, 산출된 각 퍼지항에 대한 가중치를 종합하여 결론부를 생

성한다. 예를 들어, 집단 A에 대한 퍼지 규칙-1에서는, 즉 영역 A1에 대한 퍼지 규칙에서는 F_1 축에서 정의되는 퍼지항 'Low Medium'의 가중치는 14% 그리고 F_2 축에서 정의되는 퍼지항 'High Medium'의 가중치는 86%로 산출되었다. 또한, 집단 A에 대한 퍼지 규칙-2에서는, 즉 영역 A2에 대한 퍼지 규칙에서는 퍼지항 F_1 축에서 정의되는 퍼지항 'High Medium'의 가중치는 86% 그리고 F_2 축에서 정의되는 퍼지항 'Low Medium'의 가중치는 100%로 산출되었다. 따라서, 규칙-1의 결론부에 대한 가중치 $W(Y_1)$ 은 0.14×0.86 으로 12%가 되고, 규칙-2의 결론부에 대한 가중치 $W(Y_2)$ 는 0.86×100 으로 86%가 된다.

이제 집단 A을 대표하는 규칙-1과 규칙-2의 결론부를 퍼지항으로 표현하기 위하여 산출된 가중치를 이용한다. 위의 예에서는 2개의 규칙이 집단 A를 대표하고 각 규칙의 결론부에 대한 가중치가 상이한 값을 가지므로 $W(Y_1)$ 의 퍼지항에 대한 언어적 표현은 'Low Medium' 그리고 $W(Y_2)$ 의 퍼지항에 대한 언어적 표현은 'High Medium'으로 결정하고 이들에 대한 귀속 함수의 형태는 (그림 4-5)와 같이 정의된다. 본 논문에서 사용한 실험 자료들에 대하여 제안된 방법을 적용하여 최종적으로 생성된 규칙들을 정리하면 (표 4-1)로 표시된다.



(그림 4-5) 결론부의 귀속함수 형태

(표 4-1) 실험 데이터에 의하여 생성된 퍼지 규칙

A Class	
규칙1.	IF F_1 is Low Medium and F_2 is High Medium THEN Y is Low Medium
규칙2.	IF [F_1 is High Medium] and [F_2 is High Medium or F_2 is Low Medium] THEN Y is High Medium
B Class	
규칙1.	IF [F_1 is Low] and [F_2 is Low High or High Medium or Low Medium or Low] THEN Y is Medium
규칙2.	IF [F_1 is Low Medium] and [F_2 is High Medium or Low Medium or Low] THEN Y is Low High
규칙3.	IF [F_1 is High Medium] and [F_2 is Low Medium or Low] THEN Y is Low Medium
규칙4.	IF F_1 is Low High and F_2 is Low THEN Y is Low
C Class	
규칙1.	IF F_1 is Medium THEN Y is High

V. 결 론

본 논문은 패턴 인식 분야에 퍼지 이론을 접목함에 있어서 문제가 되고 있는 퍼지 규칙의 자동 생성에 관한 문제를 연구하였다. 퍼지 규칙의 자동 생성을 위한 선결 사항으로 퍼지 공간을 분할하는 문제를 해결하였다. 본 논

문에서 제안한 퍼지 공간의 분할 알고리즘은 기존의 방법과 비교하여 자료의 분포 특성을 더욱 반영하고 있으며, 생성되는 부공간의 수를 줄이므로써 퍼지 규칙의 수를 줄이는 효과를 갖는다.

또한, 귀속 함수형의 규정에 있어서도 자료와 영역의 크기에 밀접한 관계를 갖는 변형적이고 신뢰성있는 방법을 제안하였다. 즉, 귀속 함수의 유형은 부공간내에 분포된 자료의 누적-히스토그램과 평활화 처리를 한 결과로 결정된다. 귀속 함수의 결정에 의하여 퍼지 규칙의 전체부가 형성되고, 형성된 퍼지 규칙의 합성과 가중치에 의한 결론부의 귀속함수를 유도하여 각 가상입방체에 대한 퍼지 규칙이 최종적으로 생성된다.

퍼지 패턴 인식 시스템에서 퍼지 규칙의 생성 과정은 필연적인 작업으로서, 대부분의 시스템은 전문가의 경험적인 지식에 의존하여 작성하고 있는 실정이다. 따라서, 본 논문의 연구는 체계적으로 실험 자료에 기초하여 퍼지 규칙을 자동으로 생성함으로써 패턴 인식 분야에 매우 큰 파급 효과를 가져오리라 기대된다. 특히, 시스템의 성능과 효율성간의 상호관계를 고려하여 퍼지 규칙의 수를 조정할 수 있는 방안이 제기됨으로 이는 실용적인 측면에서 매우 큰 의의를 갖는다고 여겨진다. 본 논문에서 제기한 방법은 다양한 패턴 인식 시스템에 적용할 수 있으리라 기대되며, 특히 영상패턴 및 음성 패턴과 같이 애매모호한 특성을 본질적으로 내포하고 있는 패턴 인식 분야에는 더욱 더 효율 가치가 높을 것으로 기대된다.

참 고 문 헌

1. Hideyuki Takagi & Isao Hayashi, "NN-Driven Fuzzy Reasoning," Inter. J. Approximate Reasoning 5, pp191-212, 1991
2. Shigeo Abe and Ming-Shong Lan, "A Classifier Using Rules Extracted Directly from Numerical Data," 2nd IEEE Inter. Conf. on Fuzzy Systems, Vol II, pp1191-pp1198, 1993, Mar. 28-April 1
3. W. Pedrycz, "Fuzzy Sets in Pattern Recognition: Methodology and Methods," PR Vol 23. , No 1/2, pp121-146, 1990 .
4. Takeshi Yamakawa & Masuo Furukawa, "A Design Algorithm of Membership Functions for A Fuzzy Neuron using Example-Based Learning," IEEE Inter. Conf. on Fuzzy Systems 1992, pp75-82, 1992, March 8-12
5. Hideyuki Takagi, "Fusion Technology of Fuzzy Theory and Neural Networks-Survey and Future Directions-," Iizuka, Japen, July 20-24, pp13-26, 1990
6. T. Takagi and M. Sugeno, "Fuzzy Identification of systems and its application to modelling and control," IEEE Trans. System Man Cybernet 15, pp116-132, 1985
7. Hisao Ishibuchi & Ken Nozaki & Hideo Tanaka, "Efficient Fuzzy Partition of Pattern Space for Classification Problems," 2nd Inter. Conf. on Fuzzy Logic & Neural Networks, pp671-674
8. Hisao Ishibuchi & Ken Nozaki & Hideo Tanaka, "Selecting Fuzzy Rules by Genetic Algorithm for Classification Problems," 2nd IEEE Inter. Conf. on Fuzzy Systems, Vol II, pp1119-pp1124, 1993, Mar. 28-April 1
9. Hisao Ishibuchi & Ken Nozaki & Hideo Tanaka, "Pattern Classification by Distributed Representation of Fuzzy Rules," IEEE Inter. Conf. on Fuzzy Systems 1992, March 8-12, pp643-650, 1992
10. Hisao Ishibuchi & Ken Nozaki & Hideo Tanaka, "Distributed Representation of Fuzzy Rules and its application to Pattern Classification," Fuzzy sets and systems 52, pp21-32, 1992
11. Sam-Kit Sin and Rui J. P. deFigueiredo, "Fuzzy System Design Through Fuzzy Clustering and Optimal Predefuzzification" 2nd IEEE Inter. Conf. on Fuzzy Systems, Vol II, pp190-pp195, 1993, Mar. 28-April 1