한국지능시스템학회 논문지 2008, Vol. 18, No. 5, pp. 685-691

퍼지 분할을 위한 분류 경계의 추출과 패턴 분류에의 응용

Extraction of Classification Boundary for Fuzzy Partitions and Its Application to Pattern Classification

손창식[a] · 서석태[a] · 정환묵[b] · 권순학[a*]

Chang S. Son[a], Suk T. Seo[a], Hwan M. Chung[b] and Soon H. Kwon[a*]

a 영남대학교 전기공학과

b 대구가톨릭대학교 컴퓨터정보통신공학부

요 약

퍼지 규칙기반 분류 시스템에서 위한 퍼지 분할 경계들의 선택은 중요하고 어려운 문제이다. 그래서 이들을 효과적으로 결정하기 위해서 신경망, 유전자알고리즘 등과 같은 학습과정에 기반을 둔 다양한 방법들이 제안되었고, 이전 연구에서는 이들 방법에 대한 문제점을 지적하고 이를 개선하기 위하여 중첩 형태에서 퍼지 분할을 결정할 수 있는 방법에 대해서 논의하였다. 본 논문에서는 이전 연구의 방법을 3가지 형태의 분류 경계들, 즉 비중첩, 중첩, 1점 인접 형태로 확장하였다. 또한 이들을 학습에 의존하지 않고 주어진 데이터로부터 얻어진 통계적 정보만을 사용하여 결정하는 방법을 제안하고, 이를 패턴 분류 문제에 적용하여 제안된 방법의 효용성을 보인다.

키워드 : 패턴분류, 분류경계, 퍼지분할, 규칙감축

Abstract

The selection of classification boundaries in fuzzy rule-based classification systems is an important and difficult problem. So various methods based on learning processes such as neural network, genetic algorithm, and so on have been proposed for it. In a previous study, we pointed out the limitation of the methods and discussed a method for fuzzy partitioning in the overlapped region on feature space in order to overcome the time-consuming when the additional parameters for tuning fuzzy membership functions are necessary. In this paper, we propose a method to determine three types of classification boundaries(i.e., non-overlapping, overlapping, and a boundary point) on the basis of statistical information of the given dataset without learning by extending the method described in the study. Finally, we show the effectiveness of the proposed method through experimental results applied to pattern classification problems using the modified IRIS and standard IRIS datasets.

Key Words : pattern classification, classification boundaries, fuzzy partition, rule reduction

1. 서 론

퍼지규칙 기반 분류 시스템에서 분류의 성능은 주어진 특징공간상에 정의된 퍼지 분할구간들과 그 분할된 공간들로부터 생성된 규칙들의 수에 의해서 결정된다. 최근에는 이들을 주어진 수치적인 데이터들로부터 추출하고 생성하기 위한 다양한 방법들이 제안되었다[1-6]. 그러나 이들 대부분의 방법들은 패턴분류 문제에서 규칙의 수를 최적화할 수 있다는 장점을 가지지만, 초기에 퍼지 분할구간을 발견적인(heuristic)방법으로 정의하기 때문에 분류의 경계영역을 찾기 위해서 소속함수를 조정하는 추가적인 학습과정이 필요하고, 그들을 최적화하는데 다소 많은 시간이 소요된다.

참고문헌 [7]에서는 이러한 제약점들을 해결하기 위해 학습기법의 사용 없이 주어진 데이터의 통계적인 정보만을 사용하여 특징공간에서 클래스들 간의 중첩 영역에서 퍼지 분할을 결정할 수 있는 방법을 제시하였다. 본 논문에서는 참고문헌 [7]의 방법을 확장하여 퍼지 분할을 위한 추가적인 2가지 형태의 분류 경계영역 (비중첩 형태 및 1점 인접 형태)을 정의하고, 주어진 데이터로부터 얻은 통계적인 정보(최소값, 최대값, 표준편차)를 사용하여 이들 경계영역들을 추출할 수 있는 방법을 제안한다. 또한 그 추출된 경계영역으로부터 퍼지 소속함수를 생성할 수 있는 방법에 대해서도 논의한다. 본 논문의 구성은 다음과 같다. 2장에서는 퍼지 분할을 위한 3가지 형태의 분류 경계영역에 대해서 설명하고, 그 경계영역들을 주어진 데이터의 통계적인 정보만을 사용하여 추출할 수 있는 방법과 그 추출된 영역으로부터 퍼지 소속함수를 생성할 수 있는 방법에 대해서 논의한다. 3장에서는 제안된 방법의 타당성을 보이기 위해서 수정된 IRIS와 표준 IRIS 데이터에 적용한 실험결과를 바탕으로 기존의 몇몇 분류방법들과 성능을 비교 분석하고, 4장에서 결론을 맺는다.

접수일자 : 2008년 2월 12일
완료일자 : 2008년 9월 25일
*Corresponding Author

2. 퍼지 분할을 위한 분류 경계영역의 추출

본 장에서는 이전 연구, 퍼지 분할의 선택방법[7]을 확장하여 특징공간에서 패턴분류 경계영역을 추출할 수 있는 방법과 그 추출된 영역으로부터 퍼지 소속함수를 구성하는 방법에 대해서 논의한다.

2.1 분류 경계영역의 추출

만약 n개의 입력속성을 가진 데이터가 k개의 클래스로 분류된다면,

$$\begin{aligned} &A=\{a_{i1}, a_{i2}, ..., a_{in}\},\ (i=1,2,...,s)\\ &C=\{c_1, c_2, ..., c_k\},\\ &a_{ij} \in c_l,\ (j=1,2,...,n;\ l=1,2,...,k) \end{aligned} \tag{1}$$

여기서 $a_{i1}, a_{i2}, ..., a_{in}$은 n개의 입력속성을 가진 임의의 데이터를 의미하고, i는 데이터의 인덱스를 나타낸다. 또한 C는 출력 속성 즉 클래스의 집합을 나타내고, $a_{ij} \in c_l$은 i번째 입력 데이터는 k개의 클래스들 사이에서 l번째 클래스로 분류됨을 의미한다. 이때 각 입력속성의 특징공간에서 퍼지 분할의 수와 그들의 구간은 다음과 같은 단계들에 의해서 결정된다.

단계 1 각 입력속성에 대해서 s개의 입력 데이터의 통계적 정보 (최소값, 최대값, 그리고 표준편차)를 계산한다.

$$\begin{aligned} &a_{ij}=(a_{ij}^{\min}, a_{ij}^{\max}, a_{ij}^{\sigma})\\ &a_{ij}^{\min}=\min(a_{1j}, a_{2j}, ..., a_{sj})\\ &a_{ij}^{\max}=\max(a_{1j}, a_{2j}, ..., a_{sj})\\ &a_{ij}^{\sigma}=\sqrt{\left\{\sum_{i=1}^{s}(a_{ij}-m)^2\right\}/s}\\ &m=\left\{\sum_{i=1}^{s}a_{ij}\right\}/s \end{aligned} \tag{2}$$

여기서 j번째 입력속성 a_{ij}는 3개의 요소로 구성되어 있으며, $a_{ij}^{\min}$, $a_{ij}^{\max}$, a_{ij}^{σ}는 각각 j번째 속성의 최소값, 최대값, 표준편차를 의미하고, m은 그 속성의 평균값을 나타낸다.

단계 2 각 클래스에 대응하는 입력속성들의 데이터들을 추출한다.

$$\begin{aligned} &(a_{ij,m}|c_{l=1,...,k}^{m})=\begin{cases}1, & \text{if } a_{ij}\in c_{l=1,2,...,k}^{m}\\ 0, & else\end{cases}\\ &i=1,2,..,s;\ j=1,2,...,n; \end{aligned} \tag{3}$$

여기서 $(a_{ij,m}|c_{l=1,...,k}^{m})$은 k개의 클래스들 중에서 m번째 클래스에 속하는 j번째 입력속성의 데이터 $a_{ij,m}$을 추출함을 나타낸다. 또한 $a_{ij}\in c_{l=1,2,...,k}^{m}$은 j번째 입력속성에서 m번째의 클래스를 포함하는 데이터가 존재하면 1, 그렇지 않으면 0임을 나타낸다.

단계 3 식 (3)으로부터 추출된 데이터를 근거로 각 클래스의 입력속성의 내부 도메인 구간을 추출한다.

$$\begin{aligned} &a_{ij,m}=[a_{ij,m}^{\min}, a_{ij,m}^{\max}]\\ &a_{ij,m}^{\min}=\min(a_{1j,m}, a_{2j,m}, ..., a_{sj,m})\\ &a_{ij,m}^{\max}=\max(a_{1j,m}, a_{2j,m}, ..., a_{sj,m}) \end{aligned} \tag{4}$$

$$\begin{aligned} &a_{ij,n}=[a_{ij,n}^{\min}, a_{ij,n}^{\max}]\\ &a_{ij,n}^{\min}=\min(a_{1j,n}, a_{2j,n}, ..., a_{sj,n})\\ &a_{ij,n}^{\max}=\max(a_{1j,n}, a_{2j,n}, ..., a_{sj,n}) \end{aligned}$$

여기서 $a_{ij,m}=[a_{ij,m}^{\min}, a_{ij,m}^{\max}]$와 $a_{ij,n}=[a_{ij,n}^{\min}, a_{ij,n}^{\max}]$은 각각 j번째 입력속성에 대해서 m번째와 n번째 클래스들의 내부 도메인 구간을 의미한다. 다시 말해서, 이들 값들은 그 입력속성 내에서 표현되어지는 클래스들의 하한과 상한경계를 나타낸다.

단계 4 식 (4)에서 얻은 각 입력속성의 내부 도메인 구간을 근거로 분류 경계영역을 추출한다. 또한 이들 경계영역은 다음과 같이 3가지 형태로 분류될 수 있다.

(1) m과 n번째 클래스 구간이 서로 중첩되는 분류 경계영역을 가지는 경우[7]

$$\begin{aligned} &BR_1=[a_{ij,n}^{\min}, a_{ij,m}^{\max}]\\ &NR_1=[a_{ij,m}^{\min}, a_{ij,n}^{\min}], [a_{ij,m}^{\max}, a_{ij,n}^{\max}] \end{aligned}, \text{ if } a_{ij,m}^{\min}<a_{ij,n}^{\min}<a_{ij,m}^{\max}<a_{ij,n}^{\max} \tag{5}$$

$$\begin{aligned} &BR_2=[a_{ij,m}^{\min}, a_{ij,n}^{\max}]\\ &NR_2=[a_{ij,n}^{\min}, a_{ij,m}^{\min}], [a_{ij,n}^{\max}, a_{ij,m}^{\max}] \end{aligned}, \text{ if } a_{ij,n}^{\min}<a_{ij,m}^{\min}<a_{ij,n}^{\max}<a_{ij,m}^{\max} \tag{6}$$

$$\begin{aligned} &BR_3=[a_{ij,m}^{\min}, a_{ij,m}^{\max}]\\ &NR_3=[a_{ij,n}^{\min}, a_{ij,m}^{\min}], [a_{ij,m}^{\max}, a_{ij,n}^{\max}] \end{aligned}, \text{ if } a_{ij,n}^{\min}<a_{ij,m}^{\min}<a_{ij,m}^{\max}<a_{ij,n}^{\max} \tag{7}$$

$$\begin{aligned} &BR_4=[a_{ij,n}^{\min}, a_{ij,n}^{\max}]\\ &NR_4=[a_{ij,m}^{\min}, a_{ij,n}^{\min}], [a_{ij,n}^{\max}, a_{ij,m}^{\max}] \end{aligned}, \text{ if } a_{ij,m}^{\min}<a_{ij,n}^{\min}<a_{ij,n}^{\max}<a_{ij,m}^{\max} \tag{8}$$

여기서 $BR_i(i=1,...,4)$은 m번째 클래스 구간 $[a_{ij,m}^{\min}, a_{ij,m}^{\max}]$과 n번째 클래스 구간 $[a_{ij,n}^{\min}, a_{ij,n}^{\max}]$이 서로 중첩될 때의 분류 경계영역이고, 반면에 $NR_i(i=1,...,4)$은 비중첩된 경우의 영역을 나타낸다. 그림 1은 j번째 속성에서 클래스들의 내부 도메인 구간들 사이에서 중첩된 경계영역 (즉 식 (8)의 예) 을 나타낸 것이다.

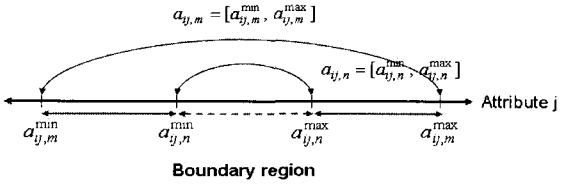

그림 1. 중첩된 경계영역
Fig. 1. Overlapped boundary region

그림 1에서 점선으로 표시된 영역은 서로 다른 2개의 클래스 구간이 중첩된 영역을 의미하고, 실선으로 표시된 영역은 비중첩된 영역을 나타낸다.

(2) m과 n번째 클래스 구간들 사이에서 1점 인접 형태의 분류 경계영역을 가지는 경우

$$\begin{aligned} &BR_1=a_{ij,m}^{\max} \text{ or } a_{ij,n}^{\min}\\ &NR_1=[a_{ij,m}^{\min}, a_{ij,m}^{\max}], [a_{ij,n}^{\min}, a_{ij,n}^{\max}] \end{aligned}, \text{ if } a_{ij,m}^{\min}<a_{ij,m}^{\max}=a_{ij,n}^{\min}<a_{ij,n}^{\max} \tag{9}$$

$$BR_2 = a_{ij,n}^{\max} \text{ or } a_{ij,m}^{\min}$$
$$NR_2 = [a_{ij,n}^{\min}, a_{ij,n}^{\max}], [a_{ij,m}^{\min}, a_{ij,m}^{\max}], \text{ if } a_{ij,n}^{\min} < a_{ij,n}^{\max} = a_{ij,m}^{\min} < a_{ij,m}^{\max} \quad (10)$$

여기서 $BR_i, NR_i (i=1,2)$은 한 지점에서의 분류 경계영역과 그 이외의 영역을 나타낸다. 이 분류 경계영역의 특징은 식 (9)과 (10)에서 볼 수 있듯이, 클래스의 내부 도메인 구간의 하한경계 혹은 상한경계를 기준으로 서로 다른 클래스로 구분된다는 점이다. 그림 2는 클래스의 내부 도메인 구간들에 의해서 결정된 1점 인접 형태에서의 분류 경계영역을 나타낸 것이다.

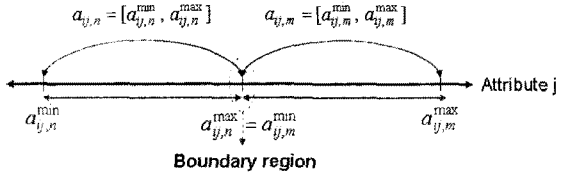

그림 2. 한 지점에서의 경계영역
Fig. 2. Boundary region with a boundary point

그림 2에서 실선으로 표시된 영역은 그림 1에서처럼, 2개의 클래스 구간들 사이에서 비중첩된 영역을 나타내고, 점선으로 표시된 영역은 2개의 클래스가 명확하게 구분될 수 있는 하나의 지점을 나타낸 것이다.

(3) m과 n번째 클래스 구간이 서로 중첩되지 않는 분류 경계영역을 가지는 경우

$$NR_1 = [a_{ij,m}^{\min}, a_{ij,m}^{\max}], [a_{ij,n}^{\min}, a_{ij,n}^{\max}], \text{ if } a_{ij,m}^{\min} < a_{ij,m}^{\max} < a_{ij,n}^{\min} < a_{ij,n}^{\max}$$
$$NR_2 = [a_{ij,n}^{\min}, a_{ij,n}^{\max}], [a_{ij,m}^{\min}, a_{ij,m}^{\max}], \text{ if } a_{ij,n}^{\min} < a_{ij,n}^{\max} < a_{ij,m}^{\min} < a_{ij,m}^{\max} \quad (11)$$

여기서 $NR_i, (i=1,2)$는 주어진 클래스 구간들이 서로 중첩되지 않음을 나타내고, 이것에 대한 분류 경계영역은 그림 3과 같다.

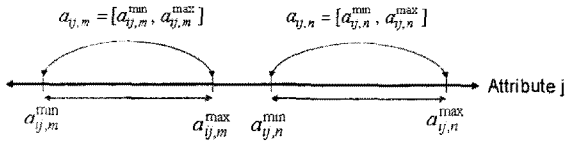

그림 3. 비중첩된 경계영역
Fig. 3. Non-overlapped boundary region

그림 3에서 실선으로 표시된 부분은 서로 다른 클래스들의 구간이 비중첩됨을 보여준다.

2.2 분류 경계영역으로부터 퍼지 소속함수의 생성

2.1절에서 기술된 3가지 형태의 경계영역들을 근거로 퍼지 소속함수의 수와 폭을 결정할 수 있는 방법에 대해서 설명한다.

(1) 중첩된 분류 경계영역을 가지는 경우[7]

$$p_j = (p_j^{BR} + p_j^{NR}), \; p_j \in [\alpha_j^{\min}, \alpha_j^{\max}]$$
$$(\alpha_j^{\min} = a_{ij}^{\min} - a_{ij}^{\sigma}, \; \alpha_j^{\max} = a_{ij}^{\max} + a_{ij}^{\sigma}) \quad (12)$$
$$p_j^{BR} = round\left(\frac{a_{ij}^{\max} - a_{ij}^{\min}}{a_{ij}^{\sigma}}\right), \; p_j^{BR} \in [p_{BR}^{\min}, p_{BR}^{\max}]$$
$$p_j^{NR} = 2,$$
$$d_j = \frac{(p_{BR}^{\max} - p_{BR}^{\min})}{p_j^{BR} - 1}$$

여기서 p_j는 j번째 입력속성에서 전체 퍼지 소속함수의 수를 의미하고, 그들의 특징공간은 구간 $[\alpha_i^{\min}, \alpha_i^{\max}]$을 범위로 가진다. p_j^{BR}와 p_j^{NR}은 각각 중첩영역과 비중첩영역에서의 퍼지 소속함수의 수를 나타낸다. 여기서 $p_j^{NR}=2$는 그림 1에서 볼 수 있듯이, 중첩된 경계영역을 제외한 영역에서는 선형적으로 분리가 가능하기 때문에 각각 1개씩의 퍼지 소속함수를 가짐을 의미한다. 또한 d_j는 중첩된 분류경계 영역 (즉, p_j^{BR})에서 각 퍼지 소속함수들의 *half-width*를 나타내고, $p_{BR}^{\min}$와 $p_{BR}^{\max}$는 중첩영역에서의 하한과 상한경계를 나타낸다.

(2) 1점 인접 형태의 분류 경계영역을 가지는 경우

$$p_j = round\left(\frac{(a_{ij}^{\max} - a_{ij}^{\min})}{a_{ij}^{\sigma}}\right), \; p_j \in [\alpha_j^{\min}, \alpha_j^{\max}]$$
$$d_j = \frac{(a_{ij}^{\max} - a_{ij}^{\min})}{p_j - 1} \quad (13)$$

여기서 p_j는 j번째 입력속성에서 퍼지 소속함수들의 수를 나타낸다. 만약 서로 다른 2개의 클래스 구간들을 비교하였을 때, 1점 인접의 경계기점을 가진다면 (그림 2 참조), 그 경계점을 기준으로 퍼지 소속함수를 생성한다.

(3) 비중첩된 분류 경계영역을 가지는 경우

퍼지 소속함수의 수와 폭의 결정은 하나의 분류 경계점을 가진 경우와 동일한 방법으로 생성한다. 그러나 하나의 분류 경계점을 가진 경우와의 차이점은 그림 3에서처럼 m번째 클래스 구간의 상한경계와 n번째 클래스 구간의 하한경계의 평균지점 (즉 $(a_{ij,m}^{\max} + a_{ij,n}^{\min})/2$)을 기준으로 생성하는 점이다.

2.3 규칙생성

2.2절에서 획득한 분류 경계영역들을 근거로 퍼지 if-then을 생성하기 위한 과정은 다음과 같이 2부분으로 구성된다: 1) 주어진 수치적인 입력 패턴들로부터 규칙생성 2) 생성된 규칙들 간의 커플링 문제의 최소화. 규칙생성과정 (참고문헌[8]에서 사용된 방법)은 주어진 수치적인 입력 패턴들에 대응된 규칙들 가운데 규칙의 이행정도(*DOF: degree of fulfillment*)가 최대인 규칙들을 추출하여 생성하였다. 예를 들어, 퍼지 IF-THEN 규칙이 다음과 같다고 가정하자.

$$Rule\ R^i: IF\ x_{i1}\ is\ A_{i1}\ \text{and} \ldots \text{and}\ x_{in}\ is\ A_{in}$$
$$THEN\ C_j \quad (i=1,2,\ldots,N)$$

여기서 $x_{i1},...,x_{in}$는 n개의 입력 변수를 의미하고, $A_{i1},...,A_{in}$과 $C_j(j=1,2,...,k)$는 각각 입력부 소속함수와 출력부 클래스를 나타낸다. 또한 N은 퍼지 if-then 규칙의 수이고, k는 클래스의 수를 나타낸다. 이때 수치적인 입력 패턴으로부터의 규칙생성은 다음과 같다.

$$r_{Class\ j}(i) = \sum_{X_p \in Class\ j} \mu_i(X_p), \tag{14}$$

$$\mu_i(X_p) = \min(\mu_{i1}(x_{p1}),...,\mu_{in}(x_{pn})) \tag{15}$$

$$r_{Class\ \hat{j}}(i) = \max_{1 \le j \le k} r_{Class\ j}(i) \tag{16}$$

여기서 $\mu_{in}(\bullet)$은 n개의 수치적인 입력패턴에 대한 소속함수 값이고, $r_{Class\ \hat{j}}(i)$는 대응된 규칙들 가운데 규칙의 이행정도가 최대인 규칙만을 선택함을 나타낸다.

식 (17)-(20)은 각 규칙의 이행정도와 빈도수(즉 각 입력패턴에 의해서 활성화된 규칙의 카운팅 횟수)를 근거로 생성된 규칙들 간의 커플링 문제를 최소화하기 위한 방법을 나타낸다[9].

(1) 커플링 규칙이 발생하지 않는 경우

추출된 규칙들의 빈도수와 규칙의 이행정도에 관계없이, 각 클래스에 대응하는 규칙은 해당 클래스의 규칙으로 선택한다.

(2) 커플링 규칙이 발생하는 경우

$$f_{C_m} = \sum_{i=1}^{N} r_{Class\ m}(i) \tag{17}$$

$$f_{C_n} = \sum_{i=1}^{N} r_{Class\ n}(i) \tag{18}$$

여기서 f_{C_m}, f_{C_n}는 각각 전체 N개의 규칙 중에서 m와 n번째 클래스에 대한 규칙의 빈도수를 나타낸다.

$$r^*_{Class} = \begin{cases} C_m, & \text{if } f_{C_m} > f_{C_n} \\ C_n, & otherwise \end{cases} \tag{19}$$

여기서 $f_{C_m} > f_{C_n}$는 각각 m번째 규칙의 빈도수가 n번째 규칙의 빈도수보다 큰 경우 m번째 클래스의 규칙으로 선택함을 의미하고, 그렇지 않으면 n번째 클래스의 규칙으로 선택함을 나타낸다.

$$r^*_{Class} = \begin{cases} C_m, & \text{if } r_{Class\ \hat{m}}(i) > r_{Class\ \hat{n}}(i) \\ C_n, & otherwise \end{cases}, \text{ for } f_{C_m} = f_{C_n} \tag{20}$$

식 (20)은 m번째 규칙의 빈도수와 n번째 규칙의 빈도수가 같다면, 최종 규칙은 m번째와 n번째 규칙의 이행정도를 비교하여 보다 큰 이행정도를 가지는 규칙을 해당 클래스의 규칙으로 선택함을 나타낸다.

2.4 규칙감축

2.3절의 규칙생성과정에 의해서 생성된 규칙은 여러 불필요한 중복속성을 포함하기 때문에 제안된 방법에서는 Skowron[10]에 의해서 제안된 러프집합의 식별가능행렬을 이용하여 중복속성을 제거하였다.

$$(c_{ij}) = \{a \in A : a(x_i) \neq a(x_j)\}, \text{ for } i,j = 1,2,...,n \tag{21}$$

여기서 c_{ij}는 속성 x_i, x_j을 구별하게 하는 모든 속성들의 집합 (즉, $n \times n$ 식별가능행렬)을 의미한다. 따라서 식별가능 행렬에서 코어는 주어진 속성들 중에서 단일 원소로 이루어진 속성들의 집합이므로 다음과 같이 정의될 수 있다.

$$core(A) = \{a \in A : c_{ij} = (a), \text{ for } some\ i,j\} \tag{22}$$

식 (22)으로부터 리덕트를 계산하기 위하여 다음과 같이 식별가능 함수 $f(A)$을 계산할 수 있다.

$$f(A) = \prod_{(x,y) \in U^2} \{\sum \delta(x,y) : (x,y) \in U^2 \text{and } \delta(x,y) \neq 0\} \tag{23}$$

식 (23)에 의해서 계산된 리덕트로부터 각 속성 x에 대한 최종 식별가능 함수 $f^x(A)$을 다시 정의할 수 있다.

$$f^x(A) = \prod_{y \in U} \{\delta(x,y) : y \in U \text{and } \delta(x,y) \neq 0\} \tag{24}$$

식 (21)-(24)로부터 생성된 규칙에는 하나 이상의 리덕트를 포함하기 때문에, 조합 가능한 다양한 규칙 패턴들이 존재한다. 그러므로 제안된 방법에서는 보다 적은 수의 규칙을 획득하기 위해서 하나의 리덕트만을 가진 규칙 패턴들을 추출함으로써 최종 규칙을 구성하였다.

3. 실험결과 및 검토

3.1 수정된 IRIS 데이터의 패턴 분류

실험에서는 제안된 방법의 타당성을 보이기 위해서 UCI Machine Learning Repository[11]의 IRIS 데이터를 사용하였다. 또한 2장에서 논의된 3가지 형태의 분류 경계를 가진 패턴분류 실험을 위해 IRIS 데이터에서 18개의 입력패턴들을 제외한 수정된 IRIS 데이터를 사용하였다.

표 1은 수정된 IRIS 데이터를 생성하기 위해서 표준 IRIS 데이터의 150개 입력패턴들 중에서 제거된 18개 패턴들의 인덱스 번호와 전체 데이터 수를 보여준다.

표 1. 실험 데이터
Table 1. Experimental data

	수정된 IRIS
제거된 인덱스 번호	51, 53, 55, 57, 59, 64, 71, 73, 74, 77, 78, 84, 87, 92, 120, 130, 134, 135
총 데이터	132개 (50개, 36개, 46개)
입력 속성	4개
클래스	3개

표 2는 수정된 IRIS 데이터에 대한 각 속성의 통계적인 정보를 나타낸다. 표 2에서 Min과 Max는 주어진 132개의 데이터에 대해서, 2.1절의 단계 1 - 3을 통해 생성된 각 속성의 최소값과 최대값을 나타내고, SD는 각 속성의 표준편차를 나타낸다.

표2에서 속성 SL과 SW의 클래스 구간들은 3개의 중첩된 경계영역을, 속성 PL은 2개의 비중첩 경계영역들과 1점 인접 형태의 경계영역을 가진다. 또한 속성 PW는 3개의 비중첩 경계영역들을 포함하고 있음을 볼 수 있다. 표 3은 표 2의 통계적 정보와 단계 4를 통해 추출된 각 속성의 분류

경계영역을 보여준다.

표 2. 수정된 IRIS 데이터의 통계적 정보
Table 2. Statistical information on the modified IRIS dataset

클래스 / 속성	Setosa		Versicolor		Virginica		SD
	Min	Max	Min	Max	Min	Max	
SL	4.3	5.8	4.9	6.7	4.9	7.9	0.8413
SW	2.3	4.4	2.0	3.4	2.5	3.8	0.4468
PL	1.0	1.9	3.0	4.5	4.5	6.9	1.8224
PW	0.1	0.6	1.0	1.6	1.7	2.5	0.8042

SL: Sepal Length; SW: Sepal Width; PL: Petal Length; PW: Petal Width

표 3. 분류 경계영역
Table 3. Boundary regions for classification

속성	경계영역	비경계영역
SL	[4.9, 6.7]	[3.4587, 4.9000], [6.7000, 8.7413]
SW	[2.3, 3.8]	[1.5509, 2.3000], [3.8000, 4.8491]
PL	[2.45, 4.5]	[-0.8223, 2.4500], [4.5000, 8.7233]
PW	[0.8, 1.65]	[-0.7032, 0.8000], [1.6500, 3.3032]

표 3에서 속성 SL의 경계영역은 각 클래스 구간들의 중첩영역들을 모두 포함하는 영역을 의미한다. 즉, 경계영역은 3개의 중첩된 경계영역 ([4.9, 5.8], [4.9, 6.7], 그리고 [4.9, 5.8])을 포함하는 전체영역에서 결정됨을 볼 수 있다: $(Seto. \cap Versi.) \cup (Versi. \cap Virgi.) \cup (Seto. \cap Virgi.)$. 비중첩 영역에서의 경계영역은 식 (12)로부터 추출되었다.

제안된 방법의 타당성을 보이기 위해서 기존의 패턴 분류방법들과 제안된 방법의 10-fold CV(cross validation)에 대한 평균 분류 정확도를 비교하였다. 다음은 10-fold CV 실험을 위해 사용된 기존 분류 방법들의 학습조건을 보여준다.

1) Ishibuchi[2]

i) Population size: 100, ii) Number of evaluations: 10,000, iii) Number of individuals: 20, iv) Crossover probability: 1.0, v) Mutation probability: 0.1, vi) 'Don't care' label probability: 0.9.

2) Ishibuchi[3]

i) Weight for the number of classified patterns: 10, ii) Weight for the size of the rule set: 1.0, iii) Population size: 10, iv) Probability to include a rule in the initial populations: 0.5, v) Mutation probability: 0.01, vi) Number of total generations: 1,000.

3) Gonzalez[4]

i) Population size: 100, ii) Number of iterations: 500, iii) Mutation probability: 0.01, iv) Use rule weight: Yes.

4) Garcia[12]

i) Hidden layers: 2, ii) Hidden nodes: 15, iii) Transfer function: HTan, iv) η: 0.15, v) α: 0.1, vi) λ: 0.0, vii) Cycles: 10,000, viii) Ensemble method: BEM.

5) Quinlan[13]

i) Pruned: Yes, ii) Confidence: 0.25, iii) Instances Per Leaf: 2.

표 4는 위의 학습조건으로 수행된 10-fold CV의 평균 분류 정확도를 보여준다.

표 4. 분류 결과 (기존의 방법)
Table 4. Classification results (conventional methods)

패턴 분류방법	퍼지분할 수	평균 정확도(%)
Ishibuchi[2]	K=3	96.21
	K=4	90.22
	K=5	100
	K=6	99.23
Ishibuchi[3]	K=3	96.27
	K=4	98.46
	K=5	98.46
	K=6	99.23
Gonzalez[4]	K=3	99.23
	K=4	94.01
	K=5	99.29
	K=6	97.69
Garcia[12]	-	96.21
Quinlan[13]	-	98.52

표 4에서 "평균 정확도"는 10-fold에 대한 분류 실험 후 계산된 평균 분류정확도를 보여준다. 표 5는 제안된 방법에서의 10-fold CV 실험 후 계산된 평균 분류정확도를 보여준다.

표 5. 분류 결과 (제안된 방법)
Table 5. Classification results (the proposed method)

패턴 분류방법	퍼지분할 수	평균 정확도(%)
제안된 방법	Fold 1-8, 10 (6, 7, 5, 5)	98.52
	Fold 9 (6, 8, 5, 5)	

표 5에서 퍼지 분할의 수, 즉 '6, 7, 5, 5'는 10개의 fold 중에서 1-8, 10에서 각 속성의 퍼지 분할 수를 의미하고, '6, 8, 5, 5'는 9번째 fold에서 사용된 퍼지 분할의 수를 나타낸다. 여기서 각 속성의 경계영역과 퍼지분할의 수는 각 fold의 실험데이터(testing data)를 평가하기 위해 2장에서 논의된 것처럼 훈련데이터(training data)의 통계적인 정보만을 사용하여 구성하였다. 표 4와 5의 결과로부터 Ishibuchi[2]와 [3]은 각각 K=5, 6 그리고 K=6일 때 제안된 방법에 비해 보다 좋은 분류 정확성을 보였고, Gonzalez[4]는 K=3, 5일 때 좋은 분류 성능을 보였다. 또한 제안된 방법은 Quinlan[13]의 C4.5 방법과는 동일한 분류 성능을 보였고, Garcia-pedrajas[12]의 nonlinear boosting projection 방법에 비해서는 2.31% 정도 향상됨을 볼 수 있었다. 그러나 이들 방법들 중에서 Ishibuchi[2], [3], 그리고 Gonzalez[4]의 방법은 최적의 분류 경계면을 찾기 위해 퍼지 소속함수들을 조정하기 위한 추가적인 학습과정들이 필요하지만, 제안된

방법은 주어진 데이터의 통계적 정보만을 고려하여 이들을 결정하기 때문에 보다 단순한 방법으로 퍼지 분류기를 설계할 수 있음을 볼 수 있다.

그림 4는 제안된 방법과 기존의 분류 방법들 간의 각 fold에서 실험 데이터의 분류 정확도를 보여준다.

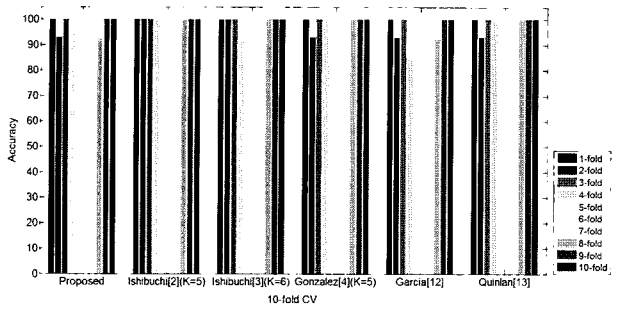

그림 4. 분류 정확도 (수정된 IRIS 데이터)
Fig. 4. Classification accuracies (Modified IRIS data)

그림 4에서 Ishibuchi[2]와 Gonzalez[4]에서는 K=5일 때 각 fold의 분류 정확도는 10-fold CV 실험에서 가장 좋은 평균 분류정확도를 보여주고, Ishibuchi[3]에서는 K=6일 때의 각 fold의 분류 정확도를 나타낸다.

3.2 일반적인 IRIS 데이터의 패턴 분류

본 절에서는 좀 더 객관적인 평가를 위해 150개의 입력 패턴들을 모두 고려한 표준 IRIS 데이터를 사용하였을 때의 10-fold CV 실험을 하였다. 표 6은 표 4에서 사용된 학습조건을 근거로 실험하였을 때의 기존 분류방법들의 평균 분류 정확도를 보여준다.

표 6. 분류 결과 (기존의 방법)
Table 6. Classification results (conventional methods)

패턴 분류방법	퍼지 분할 수	평균 정확도(%)
Ishibuchi[2]	K=3	96.67
	K=4	88.30
	K=5	94.00
	K=6	94.67
Ishibuchi[3]	K=3	92.67
	K=4	94.67
	K=5	95.33
	K=6	95.33
Gonzalez[4]	K=3	95.33
	K=4	94.67
	K=5	94.67
	K=6	96.00
Garcia[12]	-	68.00
Quinlan[13]	-	94.00

표준 IRIS 데이터의 10-fold CV 실험에서 각 입력속성에 대한 경계영역과 퍼지 분할의 수는 이전 실험에서와 같은 방법으로 결정되었다.

표 7. 분류 결과 (제안된 방법)
Table 7. Classification results (The proposed method)

패턴 분류방법	퍼지 분할 수	평균 정확도(%)
제안된 방법	Fold 3,5,7,8,10 (6, 7, 5, 5)	96.67
	Fold 1,2,4,6,9 (6, 8, 5, 5)	

표 7에서 '6, 7, 5, 5'와 '6, 8, 5, 5'는 10-fold CV 실험에서 사용된 각 fold의 퍼지분할 수를 나타낸다. 그림 5는 이전 실험에서 보여준 것처럼 제안된 방법과 기존의 분류 방법들 간의 각 fold에서 분류 정확도를 나타낸다.

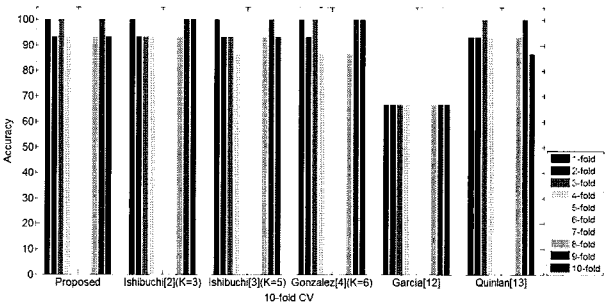

그림 5. 분류 정확도 (표준 IRIS 데이터)
Fig. 5. Classification accuracies (Standard IRIS data)

표 6과 7의 실험 결과들로부터 제안된 방법은 가장 좋은 분류 성능을 제공하는 Ishibuchi[2]의 퍼텐분류 방법과 동일한 성능을 제공함을 볼 수 있다. 또한 위의 실험결과로부터 3가지 분류 경계영역을 고려한 제안된 방법이 패턴분류 문제에서 효과적으로 적용될 수 있음을 보여준다.

4. 결 론

본 논문에서는 퍼지 분할을 위한 3가지 분류 경계영역들을 추출하기 위해서 추가적인 학습과정을 사용하지 않고 주어진 수치적인 데이터로부터 획득한 통계적인 정보만을 사용하여 결정할 수 있는 방법을 제안하였다. 또한 3가지 형태의 분류 경계영역을 고려하여 퍼지 소속함수를 생성할 수 있는 방법에 대해서도 논의하였다. 제안된 방법의 타당성을 보이기 위해서 실험에서는 수정된 IRIS 데이터와 표준 IRIS 데이터를 사용하였을 때 기존의 여러 분류방법들과의 분류 성능을 비교하였고, 그 결과로부터 제안된 방법이 기존의 학습과정을 고려한 여러 분류방법들에 비해 향상된 분류성능을 제공할 수 있음을 보였다.

참 고 문 헌

[1] J-S. R. Jang, "ANFIS : Adaptive network based fuzzy inference systems," *IEEE Transactions on SMC*, vol. 23, no. 3, pp. 665-695, 1993.
[2] H. Ishibuchi, T. Nakashima, and T. Murata, "Performance of fuzzy classifier systems for multidimensional pattern classification problems,"

IEEE Transactions on SMC, Part B: Cybernetics, vol. 29, no. 5, pp. 601-618, 1999.
[3] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 3, pp. 260-269, 1995.
[4] A. Gonzalez and R. Perez, "Selection of relevant features in a fuzzy genetic learning algorithm," *IEEE Transactions on SMC - Part B: Cybernetics*, vol. 31, no. 3, pp. 417-425, 2001.
[5] D. Nauck, U. Nauck, and R. Kruse, "Generating classification rules with the neuro-fuzzy system NEFCLASS," *In Proc. the biennial conference of NAFIPS, Berkeley*, pp. 19-22, 1996.
[6] H. Ishibuchi, T. Murata, and I. B. Turksen, "Single objective and two-objective genetic algorithms for selecting linguistic rules for pattern classification problems," *Fuzzy sets and systems*, vol. 89, no. 2, pp. 135-150, 1997.
[7] 손창식, 정환묵, 권순학, "퍼지 규칙기반 분류시스템에서 퍼지 분할의 선택방법," *한국지능시스템학회 논문지*, 제18권, 3호, pp. 360-366, 2008.
[8] H. Ishibuchi, T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 506-515, 2001.
[9] 손창식, 정환묵, 서석태, 권순학, "규칙의 커플링문제를 최소화하기 위한 퍼지-러프 분류방법," *한국퍼지 및 지능시스템 학회 논문지*, 제17권, 4호, pp. 460-465, 2007.
[10] A. Skowron and C. M. Rauszer, "The discernibility matrices and functions in information systems," *Institute of computer sciences report 1/91, Technical University of Warsaw*, pp. 1-41, 1991.
[11] UCI Repository of Machine Learning Databases, *Depatment of Information and Computer Science, University of California, Irvine, CA, Available*: http://mlearn.ics.uci.edu/MLRepository.html.
[12] N. Garcia-Pedrajas, C. Garcia-Osorio, and C. Fyfe, "Nonlinear boosting projections for ensemble construction," *Journal of Machine Learning Research*, vol. 8, pp. 1-33, 2007.
[13] J. R. Quinlan, C4.5: Programs for machine learning, *Morgan Kaufman*, 1993.

저 자 소 개

손창식(Chang S. Son)

제18권 제3호 (2008년 6월호) 참조
E-mail : fuzzyrisk@paran.com

서석태(Suk T. Seo)

제18권 제3호 (2008년 6월호) 참조
E-mail : kenneth78@ynu.ac.kr

정환묵(Hwan M. Chung)

제18권 제3호 (2008년 6월호) 참조
E-mail : hmchung@cu.ac.kr

권순학(Soon H. Kwon)

제18권 제3호 (2008년 6월호) 참조
E-mail : shkwon@yu.ac.kr