

급성복통과 관련된 지능형 질환 진단시스템을 위한 퍼지 규칙 생성과 이의 최적화

현 우 석*

요 약

본 논문에서는 급성복통과 관련된 지능형 질환 진단시스템에서 지식베이스의 최적화에 대해서 논한다. 급성복통과 관련된 지능형 질환 진단시스템의 지식베이스는 퍼지 규칙과 퍼지 멤버쉽 함수들로 구성되는데, 본 연구에서는 효율적으로 퍼지 규칙을 생성하는 알고리즘을 적용한 개선된 급성복통과 관련된 지능형 질환 진단 시스템(A-IDS-DAAP)을 제안한다. 제안하는 시스템은 기존의 IDS-DAAP, IDS-DAAP-NN과 비교해 볼 때, 진단의 정확성을 높이면서 수행속도를 향상시켰다.

Fuzzy Rule Generation and Optimization for the Intelligent Diagnosis System of Diseases associated with Acute Abdominal Pain Based on Fuzzy Relational Products

Woo-Seok Hyun*

ABSTRACT

This paper describes knowledge base optimization of an intelligent diagnosis system based on fuzzy relational products(IDS-DAAP) for the diseases with acute abdominal pain. The knowledge base of IDS-DAAP is composed of the fuzzy rules and the fuzzy membership functions. The author here proposes an advanced intelligent diagnosis system (A-IDS-DAAP) in which the fuzzy rule generation algorithm is applied. Comparing with previous IDS-DAAP and IDS-DAAP-NN, a modified approach with A-IDS-DAAP shows that it improves the diagnosis rate and reduces the time to diagnose.

키워드 : 지능시스템(Intelligent System), 급성복통(Acute Abdominal Pain), 퍼지 규칙 생성(Fuzzy Rule Generation)

1. 서 론

계속적으로 변하는 데이터를 가지는 전문영역(domain)에서 진단을 내리는 과정은 어렵지만 다양한 문제들을 해결하기 위해서는 꼭 필요한 일이다. 특히 전구 증상 없이 갑자기 복통이 발생하게 되는 급성복통(acute abdominal pain)은 소화기 질환, 생식 및 비뇨기계 질환 등 여러 질환에서 환자들이 계속해서 변하는 증상들을 호소하게 되는데 이를 진단하는 것은 어려운 일이다. 퍼지관계곱(fuzzy relational product)[1-4] 기반 급성복통과 관련된 지능형 질환 진단시스템(IDS-DAAP : Intelligent Diagnosis System for Diseases associated with Acute Abdominal Pain based on fuzzy relational product)[5]은 Bandler와 Kohout가 제안한 퍼지관계곱을 이용하였는데, 이것은 환자와 증상에 관한 애매한 정도와

증상과 질병에 관한 불확실성 등을 고려하여 환자와 질병과의 관계를 구하여 적절한 진단을 내리는데 매우 효과적인 방법으로서 진단에 필요한 지식을 생성 규칙으로 표현한 규칙기반 시스템이다. 그런데 이 시스템에서는 지식베이스(knowledge base)를 최적화하는 것에 대해서는 충분히 고려하지 못한 채 질환을 진단하여 진단 성공률이 다소 낮아질 수 있으며, 시스템의 수행시간이 증가되는 문제점을 지닐 수도 있다.

지능시스템의 실현을 위해 중요한 것은 획득한 지식을 효율적이고 효과적으로 표현하는 지식표현이며, 지능시스템의 성패는 지식표현의 우열에 좌우된다고 해도 과언이 아니다. 일반적으로 지능시스템에서는 지식베이스에 지식들을 표현하게 되는데 지식베이스를 어느 정도 효과적으로 최적화하느냐에 따라 시스템의 성능에 영향을 미치게 된다. 특히 지능시스템에서 규칙들을 자동적으로 생성해 내는 것은 인공지능 분야에서 중요하게 인식되고 있다. 이 분야의 국외 연구로는 신경 회로망을 사용한 결함 진단(fault diagnosis)에서 자동적인 규칙 생성에 관한 연구[6], 고급 수준 컴퓨터

* 본 연구는 2002학년도 한국성서대학교 교내 학술연구비 지원으로 수행되었습니다.

† 종신회원 : 한국성서대학교 정보과학부 교수

논문접수 : 2004년 7월 15일, 심사완료 : 2004년 10월 16일

비전에서 퍼지 규칙 생성 방법에 관한 연구[7], 퍼지 다층 지각과 추론, 규칙 생성에 대한 연구[8]와 2차원 격자 구조에서 퍼지 규칙 생성에 관한 연구[9] 등이 있으며 다중 퍼지 신경 회로망(dynamic fuzzy neural network)을 사용한 예 패턴에서 퍼지 규칙 생성 방법[10], FCM(Fuzzy C-Means) 클러스터링 방법을 사용한 퍼지 규칙 생성에서 빠른 클러스터링에 관한 연구[11], 자동차 조립 공장에서 End-of-Line test를 위한 퍼지 규칙 생성에 관한 연구[12] 등이 있다. 또한 국내의 퍼지 규칙 생성에 관한 연구로는 풀 온도 제어 시스템에서의 지식베이스 최적화에 관한 연구[13]와 자동 규칙 생성에 기반한 주식 매매 시스템에 관한 연구[14] 등이 있다.

본 연구에서는 위에서 언급된 문제점을 개선하고자 급성 복통과 관련된 지능형 질환 진단시스템에서 지식베이스의 최적화에 대해서 논의한다. 급성복통과 관련된 지능형 질환 진단시스템의 지식베이스는 퍼지 규칙(fuzzy rule)과 퍼지 멤버쉽 함수(fuzzy membership function)들로 구성되는데, 본 연구에서는 시스템의 성능을 향상시키기 위하여 퍼지 규칙을 생성하는 알고리즘(algorithm)을 적용한 개선된 급성 복통과 관련된 지능형 질환 진단 시스템(A-IDS-DAAP : Advanced Intelligent Diagnosis System for Diseases associated with Acute Abdominal Pain based on fuzzy relational product)을 제안한다. 제안하는 시스템은 기존의 IDS-DAAP과 IDS-DAAP-NN[6]과 비교해 볼 때, 진단의 정확성을 높이면서 수행속도를 향상시켰다.

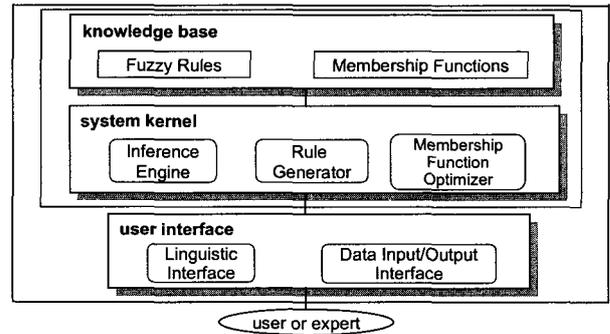
2. 개선된 급성복통과 관련된 지능형 질환 진단 시스템(A-IDS-DAAP)

개선된 급성복통과 관련된 지능형 질환 진단 시스템(AIDS-DAAP)에서는 기존의 퍼지관계곱 기반 급성복통과 관련된 지능형 질환 시스템(IDS-DAAP)[5]에서 지식베이스를 최적화하여 시스템의 성능을 향상시키기 위하여 퍼지 규칙 생성 알고리즘[12]을 적용하였다.

2.1 A-IDS-DAAP의 구조

개선된 급성복통과 관련된 지능형 질환 진단 시스템(A-IDS-DAAP : Advanced Intelligent Diagnosis System for Diseases associated with Acute Abdominal Pain based on fuzzy relational product)의 구조는 (그림 1)과 같다. 중앙에 있는 시스템 커널(system kernel)은 추론 엔진(inference engine), 규칙 생성기(rule generator) 그리고 멤버쉽 함수 최적화기(membership function optimizer)와 같은 시스템 구성 요소로 되어 있다. 상측에 있는 지식 베이스(knowledge base)는 퍼지 규칙들(fuzzy rules)과 멤버쉽 함수들(membership functions)로 구성되어 있다. 본 시스템의 급성복통 진단을 위한 퍼지 진단 지식은 처음에는 전문가인 의사들로부터 수집되어 생성되지만 퍼지 규칙 생성 알고리즘을 사용하여 지식베이스의 퍼지 규칙들을 최적화하게 된다. 하측에 있는 사용자 인터페이스(user interface)는 언어적 인터페이스(linguistic in-

terface)와 데이터 입출력 인터페이스(data input/output interface)로 구성되어 있다. 언어적 인터페이스는 의사인 전문가와 지식베이스 사이에서 인터페이스를 제공하기 위한 것이며, 데이터 입출력 인터페이스는 사용자와 시스템 커널 사이에서 인터페이스를 제공하기 위한 것이다. 본 논문에서는 지식베이스에서 퍼지 규칙 생성의 관점에서 연구를 수행한다.



(그림 1) A-IDS-DAAP의 구조

2.2 퍼지 규칙 생성

퍼지 규칙 생성(fuzzy rule generation)으로 인한 지식베이스 최적화는 시스템이 급성복통과 관련된 전문 지식을 습득하게 되는 학습 단계에서 수행되어 진다. 학습 단계에서는 처음에 퍼지 규칙들을 생성시키기 위해서 멤버쉽 함수의 임계점(critical point)의 초기 집합을 사용하며, 시간이 지남에 따라서 멤버쉽 함수의 임계 파라미터(critical parameter)들이 가장 좋은 성능으로 최적화되어 진다.

급성복통과 관련된 지능형 질환 진단 시스템에서, 퍼지 규칙들은 다음과 같은 두 가지 변수들의 집합으로서 특징지어 지는데, 환자가 자각하는 다양한 증상들을 나타내는 식 (1)의 S라는 제어변수(control variable)와 시스템이 진단한 환자의 질환을 나타낸 d라는 해결변수(solution variable)가 있다.

$$S = \{s_1, s_2, \dots, s_n\} \tag{1}$$

각각의 변수들은 퍼지 용어 집합들과 함께 사용된다. 증상을 나타내는 제어변수 S는 always, sometimes, medium, never 등의 빈도 혹은 인식세기를 나타내는 퍼지 용어들을 포함하며 식 (2)와 같다. 또한 환자의 질환을 나타내는 해결변수 d는 always, sometimes, never, unknown 등의 질환과의 관련 정도를 나타내는 퍼지 용어 집합들로 구성되며 식 (3)과 같다.

$$\sum_i = \{a_i^1, \dots, a_i^{p_i}\} \tag{2}$$

$$T = \{t_1, \dots, t_q\} \tag{3}$$

급성복통과 관련된 지능형 질환 진단시스템에서는 식 (4)와 같은 규칙의 형식을 고려한다.

$$\text{IF } (s_k^1 \text{ is } a_i^k 1) \text{ AND } (s_k^2 \text{ is } a_i^k 2) \text{ AND } \dots (s_k^m \text{ is } a_i^k m), \\ \text{THEN } d \text{ is } t_j \tag{4}$$

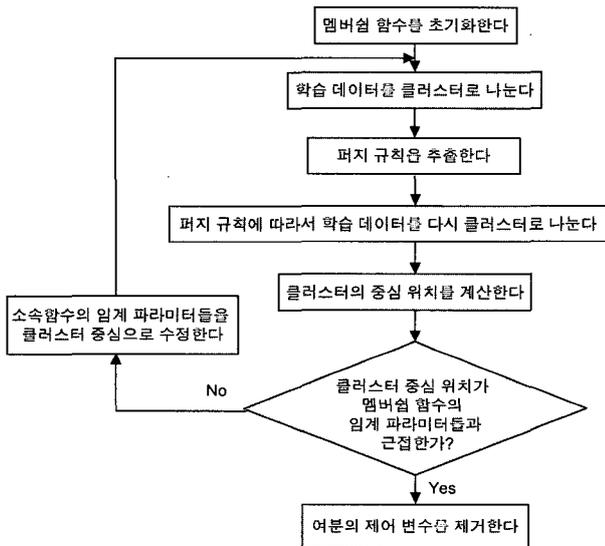
$$\text{단, } m \leq n, \{s_k^1, s_k^2, \dots, s_k^m\} \subset S, \{\alpha_k^1, \alpha_k^2, \dots, \alpha_k^m\} \subset \sum_i, \text{ and } t_j \in T$$

일반적으로, 규칙 생성기는 임계 파라미터들의 임의 초기점(initial point)에서 시작하게 되는데, 이것들은 퍼지 규칙 생성 중이나 생성 후에 최적화 되어진다. 단순화하기 위하여 임계 파라미터들은 다음과 같이 초기화되며 식 (5)와 같이 표현되는데, i번째 제어변수에 s_i 에 대한 k_i 소속 함수들이 있다고 가정했을 때, j번째 소속 함수들의 임계 파라미터를 나타내게 된다.

$$cp_{jk} = \min(S_i) + \{\max(S_i) - \min(S_i)\} \frac{j}{k_i} \quad (5)$$

단, $\min(s_i)$: 학습 데이터 집합 S에서 s_i 의 최소값
 $\max(s_i)$: 학습 데이터 집합 S에서 s_i 의 최대값
 $j = 1, 2, \dots, k_i$

퍼지 규칙 생성 알고리즘[12]은 (그림 2)와 같이 7단계로 나타낼 수 있다.



(그림 2) 퍼지 규칙 생성 알고리즘

퍼지 규칙 생성 알고리즘을 설명하기 위하여 시스템이 s_1 과 s_2 라는 2개의 제어변수와 d라는 1개의 해결 변수를 가지며, 각 제어 변수와 해결 변수는 never, sometimes, always 라는 3가지 멤버십 함수와 관련이 있다고 가정했을 때의 간단한 예를 보인다. 제어 변수 값들을 고려했을 때 학습 데이터는 $3^2 = 9$ 개의 클러스터로 구분되어지며, 각 제어 변수의 가장 큰 소속 함수를 찾게 된다. <표 1>은 현 반복(iteration)에서 얻어질 수 있는 학습 데이터 통계로서 1000개의 예가 있다고 가정한 것을 보여주고 있다.

<표 1>을 설명하기 위해서 클러스터 #1 을 예로 살펴보자. 이 클러스터는 200개의 예를 가지고 있다. 이 예들 중에서 s_1 에 대한 가장 큰 소속 함수는 sometimes이고, s_2 에 대한 가장 큰 소속 함수는 never이다. 이 클러스터에서 생성된 규칙의

진제는 “if s_1 is sometimes and s_2 is never”이다. 이 200개의 예들은 생성된 규칙을 지배하는 예들(dominated samples)이라고 부르며, 이 규칙을 지배적인 규칙(dominant rule)이라고 부른다. 하지만 이 지배적인 규칙의 결론은 아직 결정되지 않았다. 200개의 예 중에서 30개의 예에서는 d가 never이고, 140개의 예에서는 d가 sometimes이며, 30개의 예에서는 d가 always이다. 그러므로 규칙 #1의 결론은 “d is sometimes”이다. 이러한 Winner-Take-All mechanism을 사용함에 의해서 9개의 규칙들이 추출되는데 <표 2>에 나타나 있다.

<표 2>에서 보면, 생성된 모든 규칙들은 규칙들을 제거하기 위해서 합리적이고 효율적인 기준으로서 사용되는 지지예(supporting sample)들의 수에 따라서 정렬된다. 지지율(support rate)은 지지 예들의 수를 규칙을 지배하는 예(dominated sample)들로 나누어서 구해진다. 그러나 지지율 자체는 규칙을 제거하는데 사용될 수 없다. 예를 들면, 하나의 규칙이 하나의 지배적인 예를 가진다면, 그 지지율은 항상 100%이다. 다른 규칙이 200개의 지배적인 예들을 가진다면 지지율은 75%이다. 따라서 전자는 버리고 후자를 택하는 것이 더 합리적이다. <표 2>의 규칙 #5, #6, #7에서 이 클러스터에 속하는 예가 없기 때문에 결론은 null이다. 이러한 영 규칙(null rule)들은 규칙 단순화과정(reduction process)에서 매우 효율적이다. <표 2>에서 규칙 #5~#7을 고려하지 않고 규칙 #0~#2는 다음과 같이 병합될 수 있다.

○ Rule #0~#2 → “if s_2 is never, then d is sometimes”

새로운 규칙의 지지 예들의 수는 $100 + 140 + 150 = 390$ 이다. #5~#7의 영 규칙을 고려했을 때 <표 2>의 규칙 #3~#8은 다음과 같이 병합될 수 있다.

○ Rule #3~#5 → “if s_2 is sometimes, then d is never”

○ Rule #6~#8 → “if s_2 is always, then d is always”

두 규칙의 지지 예들의 수는 $120 + 100 + 0 = 220$ 과 $0 + 0 + 80 = 80$ 이다. 이런 반복(iteration)에서 추출된 최종 퍼지 규칙들은 <표 3>과 같다. 소속 함수들의 임계 파라미터들이 각 반복에서 재 계산되어 갱신되기 때문에 클러스터 분배가 변할 수도 있으며 각 반복에서 추출된 퍼지 규칙들이 달라질 수도 있다.

<표 1> 현 반복에서 얻어진 학습 데이터 통계

| 클러스터 # | the largest MBF for each control variable | | solution variable | | |
|--------|---|-----------|-------------------|-------------|----------|
| | s_1 | s_2 | d never | d sometimes | d always |
| 0 | never | never | 60 | 100 | 40 |
| 1 | sometimes | never | 30 | 140 | 30 |
| 2 | always | never | 20 | 150 | 30 |
| 3 | never | sometimes | 120 | 20 | 20 |
| 4 | sometimes | sometimes | 100 | 20 | 10 |
| 5 | always | sometimes | 0 | 0 | 0 |
| 6 | never | always | 0 | 0 | 0 |
| 7 | sometimes | always | 0 | 0 | 0 |
| 8 | always | always | 10 | 20 | 80 |

〈표 2〉 Winner-Take-All mechanism에 의해 추출된 규칙

| rule # | Antecedent | | Consequence | | |
|--------|----------------|----------------|-------------|----------------------|--------------|
| | s ₁ | s ₂ | d | # of support samples | support rate |
| 0 | never | never | sometimes | 100 | 50% |
| 1 | sometimes | never | sometimes | 140 | 70% |
| 2 | always | never | sometimes | 150 | 75% |
| 3 | never | sometimes | never | 120 | 75% |
| 4 | sometimes | sometimes | never | 100 | 77% |
| 5 | always | sometimes | anything | 0 | - |
| 6 | never | always | anything | 0 | - |
| 7 | sometimes | always | anything | 0 | - |
| 8 | always | always | always | 80 | 73% |

〈표 3〉 현재 반복에서 추출해 낸 최종 퍼지 규칙들

| rule # | Antecedent | | Consequence | |
|--------|----------------|----------------|-------------|----------------------|
| | s ₁ | s ₂ | d | # of support samples |
| 0 | anything | never | sometimes | 390 |
| 1 | anything | sometimes | never | 220 |
| 2 | anything | always | always | 80 |

퍼지 규칙들이 연결 가중치(inter connection weights) 중에서 분산적인 방식으로 기호화되는 퍼지 신경 회로망과 달리, 개선된 급성복통과 관련된 지능형 질환 진단 시스템에서 퍼지 규칙들의 내부 표현은 look-up table 방식이다. look-up table 표현은 다양성을 제공하며, 최소한의 메모리 요구량을 가지게 된다. 퍼지 신경 회로망에서는 신경 회로망이 재구축되지 않는 한 수행속도가 감소하지 않는 반면에, 개선된 급성복통과 관련된 지능형 질환 진단 시스템에서는 규칙 표에 들어 있는 규칙들과 새로이 추출된 규칙들이 병합되어 규칙 표에 나타나기 때문에 검색해야 할 규칙들의 수를 감소시켜 시스템의 수행속도를 향상시킨다. 또한 규칙 표는 연속적으로 검색되어 지고 가장 자주 활성화되는 규칙들이 표의 맨 앞쪽에 위치하게 되어 결과적으로 규칙들의 지지 예들의 수에 의해 규칙 표가 정렬됨에 따라서 시스템의 성능이 더욱 향상될 수 있을 것으로 기대된다.

위에서 설명된 예는 규칙 생성 알고리즘의 각 반복에서 사용되어 지는 클러스터링과 규칙 추출 알고리즘[12]으로 요약될 수 있는데 다음과 같이 11단계로 나타낼 수 있다.

- 1단계 : 의사들에 의해서 정해질 수 있는 우선 규칙(prior rule)으로서 규칙 표를 초기화 하거나 혹은 규칙 표를 비워둔다.
- 2단계 : 각각의 제어 변수에 대해서 가장 큰 멤버십 함수를 찾음에 의해서 학습 데이터를 클러스터로 구분 짓는다.
- 3단계 : 하나의 클러스터를 연속적으로 선택한다. 만약 클러스터가 비어있지 않는다면, Winner-Take-All mechanism을 기반으로 하나의 클러스터로부

터 규칙을 추출해 낸다. 그리고 규칙을 “지지 예들의 수(# of supporting sample)”를 가지고 레이블을 붙인다.

- 4단계 : 하나 이상의 해결 변수(solution variable)가 있다면, 각 해결 변수에 대한 전제를 추출하기 위해 3단계를 반복한다. 또한 각 해결 변수에 대해서 지지 예들의 수를 구한다. 최종적인 지지 예들의 수는 가장 적은 것이 된다.
- 5단계 : 클러스터에 예가 없다면, 영 규칙이 추출되고 지지 예들의 수는 0이 된다.
- 6단계 : 규칙 표에 있는 우선 규칙들과 새롭게 추출된 규칙을 비교한다. 새롭게 추출된 규칙이 우선 규칙과 충돌한다면, 새롭게 추출된 규칙을 버리고 10단계로 간다. 만약 새롭게 추출된 규칙이 우선 규칙들 중의 하나와 같다면, 10단계로 간다.
- 7단계 : 새롭게 추출된 규칙과 규칙 표에 이미 있던 규칙들을 병합(merge)하도록 한다. 병합이 발생한다면, 여분의 제어 변수들을 새롭게 추출된 규칙의 전제로부터 제거하고 새 규칙의 지지 예들의 수는 모든 병합된 규칙들의 지지 예들의 수로 합을 구한다.
- 8단계 : 하나 이상의 여분의 제어 변수가 있는 경우에는 규칙들이 더 이상 병합될 수 없을 때까지 7단계를 반복한다.
- 9단계 : 병합한 후에 새롭게 추출된 규칙은 규칙 표에 추가된다.
- 10단계 : 남아 있는 클러스터가 없다면, 11단계로 간다. 그렇지 않다면, 3단계로 간다.
- 11단계 : 모든 영 규칙을 지운다. 바람직하다면, 지지 예들의 수가 적은 규칙들은 제거될 수 있다.

다른 클래스에 속하는 학습 예들이 같은 클러스터에 포함될 수도 있기 때문에, 알고리즘에서 재 클러스터링(re-clustering)과 클러스터 중심들의 위치 계산이 필수적이다. 그러므로 단지 하나의 클래스로부터 추출된 규칙은 유효하다고 보기 힘들다. 이것은 fuzzy c-means 알고리즘에서 만나게 되는 문제와 유사하다. fuzzy c-means 알고리즘에서 데이터 포인트들은 항상 근접한 클러스터들에 속하게 되는데, 다른 클래스에 속하는 데이터 포인트들은 클러스터 중심들의 위치를 계산하는 동안 구별되지 않는다. 그러므로 이 문제는 fuzzy c-means 알고리즘에서는 해결될 수 없다. 본 논문에서는 이 문제를 해결하기 위해서 추가적인 재 클러스터링 과정[12]을 사용하였다.

추가적인 재 클러스터링 과정은 하나의 예와 하나의 클러스터 사이에 거리 측정을 사용하게 된다. 일반화를 잃지 않으면서 퍼지 규칙은 식 (6)과 같은 형식을 고려한다.

$$IF (S_1 \text{ is } a_1 \text{ and } S_2 \text{ is } a_2 \text{ and } \dots \text{ and } S_n \text{ is } a_n), \\ THEN (d \text{ is } a) \tag{6}$$

제어 변수 s_i 의 클래스 a_i 에 대하여 클래스 s_i 의 멤버십 함수가 가장 큰 값을 가지게 되는 구간(range) (S_{icN} , S_{icA})가 존재한다. 그러므로 해결 변수의 클래스 a_i 는 이 규칙에 의해서 입력 공간상에서 n 차원의 정육면체에 대응하게 된다. n 차원의 정육면체는 (S_{icN} , S_{icA}) 라는 구간의 집합에 의해서 정의되는데, 여기에서 $i=1, 2, \dots, n$ 이다. 정육면체 안의 모든 예들을 클러스터라고 부르며, s 번째 예와 하나의 클러스터간의 거리는 식 (7)과 같다.

$$D = \sum_{i=1}^n \epsilon_{s_i}(s) \tag{7}$$

단,

$$\epsilon_{s_i}(s) = \begin{cases} (s_{icN} - s_i^{(s)})^2, & \text{if } (s_i^{(s)} < S_{icN}) \\ 0, & \text{if } (S_{icN} < s_i^{(s)} < S_{icA}) \\ (s_i^{(s)} - s_{icA})^2, & \text{if } (s_{icA} < s_i^{(s)}) \end{cases}$$

모든 학습 예들이 재 클러스터 된 후에 각 클러스터의 중심의 위치가 계산되어 지는데, 만약 임계 파라미터들이 클러스터 중심에 충분히 근접하지 않는다면, 멤버십 함수들의 임계 파라미터들이 클러스터 중심의 위치로 갱신된다.

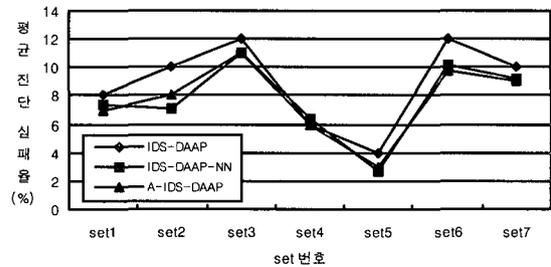
2.3 멤버십 함수의 최적화

퍼지 논리에서 각 변수는 각 퍼지 용어와 대응되는 퍼지 멤버십 함수의 집합과 관련되어 있으며, 일반적으로 퍼지 변수의 멤버십 함수들은 멤버십 함수들의 특성을 명확하게 나타내는 임계 파라미터들의 집합에 의해서 정의될 수 있다. 실험 데이터와 퍼지 규칙들이 주어졌을 때, 임계 파라미터들에 다른 값을 부여하게 되면 추론 엔진의 결과가 다르게 나타날 뿐만 아니라 규칙 생성기는 다른 퍼지 규칙들을 생성하게 된다. 퍼지 멤버십 함수 최적화는 기본이 되는 퍼지 집합들의 모양을 변화시킴에 의해서 적절한 제어를 하여 시스템이 진단을 내리도록 하는데, 본 논문에서는 다양한 선형, 삼각형, Sigmoid, Gaussian 멤버십 함수 중에서 삼각형 멤버십 함수를 선택하여 사용하였다.

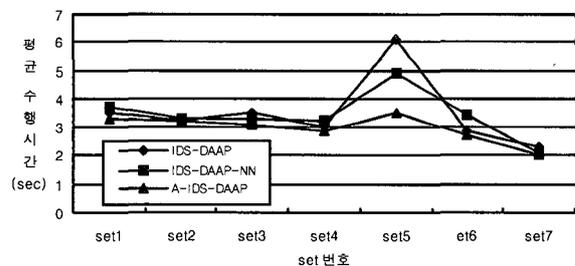
3. 구현 및 평가

본 논문의 개선된 급성복통과 관련된 지능형 질환 진단 시스템(A-IDS-DAAP)은 Windows 환경 하에서 Visual BASIC 6.0을 사용하였으며, 메모리가 128M인 Pentium IV Personal Computer상에서 시뮬레이션 시스템을 구현하였다. 제안하는 A-IDS-DAAP에서는 구토, 복벽의 통증, 만성 복통, 상복부 통증, 하복부 통증, 소화 불량 등 18개의 제어 변수들과 담낭염, 간질환, 대장 질환, 위염 등 진단 가능한 7개의 질환 각각을 나타내는 1개의 해결 변수 그리고 각각의 퍼지 변수들은 always, almost always, very often(very strong), often(strong), sometimes(medium), seldom(weak), very seldom(very weak), almost never, never, unknown 등의 10개의 퍼지 용어들과 관련이 있다[5]. 여기서 입력 변수와 출력 변수의 범위는 0에서 1 사이의 퍼지값으로 표현된다. 데이

터는 G 병원으로부터 획득한 350명의 실제 데이터를 18개의 제어 변수로부터 생성될 수 있는 질환명을 나타내는 7가지 test set으로 나누어서 진단을 수행하였다. 실험에 사용된 실제 데이터는 환자가 느끼는 증상들을 G 병원의 전문가인 의사들이 수치적 데이터로 변환하여 준 것을 퍼지 시스템의 입력 데이터로 사용하였다. G 병원의 전문가인 의사들로부터 초기에 수집한 200개의 퍼지 규칙들과 각 시스템 변수에 대한 10개의 멤버십 함수만을 가지고 진단을 수행한 IDS-DAAP와 IDS-DAAP-NN(Intelligent Diagnosis System for Diseases associated with Acute Abdominal Pain based on fuzzy relational product using fuzzy rule generation by Neural Network)[6] 그리고 test set 1을 학습 데이터로서 퍼지 규칙 생성 알고리즘[12]을 적용하여 진단을 수행한 A-IDS-DAAP에 대하여 평균 진단 실패율과 평균 수행속도를 비교하였는데 (그림 3), (그림 4)와 같다.



(그림 3) 시스템에 따른 평균 진단 실패율 비교



(그림 4) 시스템에 따른 평균 수행시간 비교

여기서 평균 진단 실패율은 맞는 진단을 시스템에서 틀리다고 진단한 환자 수의 합을 질환이 진단된 환자의 총 수로 나누는 것의 퍼센트로 구하며, 평균 수행 시간은 총 환자에 대하여 시스템이 한 명의 환자의 질환을 진단하는데 걸리는 시간의 합을 환자의 총 수로 나누어서 구하게 된다. 제안하는 A-IDS-DAAP에서는 기존의 IDS-DAAP-NN과 비교했을 때 평균 진단 실패율은 큰 차를 보이지 않았지만, 기존의 IDS-DAAP와 비교했을 때는 평균 진단 실패율이 감소되었다. 또한 기존의 IDS-DAAP-NN과 IDS-DAAP보다 제안하는 A-IDS-DAAP에서 수행속도가 향상되었다.

4. 결론 및 향후과제

제안하는 A-IDS-DAAP는 기존의 IDS-DAAP와 비교해 볼 때 평균 진단실패율이 감소하여 정확성을 더욱 높이게

되었다. 이는 기존의 IDS-DAAP는 초기에 전문가인 의사들로부터 수집한 지식과 멤버쉽 함수만을 가지고 진단을 하였으나, 제안하는 A-IDS-DAAP에서는 초기에 의사들로부터 수집한 지식베이스의 규칙들과 멤버쉽 함수에 대해 퍼지 규칙 생성 알고리즘을 적용하여 지식베이스를 진단하기에 적합하게 최적화하였기 때문이다. 기존의 IDS-DAAP-NN과 비교해 볼 때 평균 진단 실패율은 큰 차이를 보이지 않았는데, 이것은 방법은 다르지만 퍼지 규칙 생성 알고리즘을 사용한 시스템이 사용하지 않은 시스템에 비해 평균 진단 실패율을 감소시켜 주기 때문이다. 또한 시스템의 수행시간을 감소시키게 되었다. 퍼지 신경망에서는 신경망이 재구축되지 않는 한 수행속도가 감소하지 않는 반면에, 제안하는 A-IDS-DAAP에서는 퍼지 규칙들의 내부 표현이 look-up table 방식으로 표현되어 최소한의 메모리 요구량을 가지게 했을 뿐만 아니라, 규칙 표에 들어 있는 규칙들과 새로이 추출된 규칙들을 병합하여 규칙 표에서 검색해야 할 규칙들의 수를 감소시켰기 때문에 시스템의 수행시간이 감소되어 수행속도를 향상시키게 되었다. 또한 규칙 표는 연속적으로 검색되어 지고 가장 자주 활성화되는 규칙들이 표의 맨 앞쪽에 위치하게 되어 결과적으로 규칙들의 지지 예들의 수에 의해 규칙 표가 정렬됨에 따라서 시스템의 성능이 더욱 향상될 수 있을 것으로 기대된다.

향후 연구과제로는 다음과 같은 항목들을 고려하여 이루어져야 한다. 첫째, 급성복통 이외에도 여러 가지 유형의 질환에 관련되는 증상을 잘 처리할 수 있는 새로운 CAMD (Computer Aided Medical Diagnosis) 모델에 관한 연구가 필요하다. 둘째, 퍼지 멤버쉽 함수 최적화를 자동화해 주는 알고리즘에 대한 연구가 요구된다. 셋째, 급성복통과 관련된 질환의 진단 뿐 아니라 치료법도 제시해 주는 시스템의 확장에 대한 연구가 필요하다.

참 고 문 헌

[1] L. J., Khout, E. Keravnou and W. Bandler, "Automatic documentary information retrieval by means of fuzzy relational products," In Gaines, B. R., Zadeh, L. A. and Zimmermann, H. J., editors Fuzzy Sets in Decision Analysis, North-Holland, Amsterdam, pp.308-404, 1984.

[2] W. Bandler and L. J. Kohout, "Fuzzy Relational Products as a Tool for Analysis and Synthesis of the Behaviour of Complex natural and Artificial System," in : S. K. Wang and P. P. Chang, eds., Fuzzy Sets : Theory and Application to Analysis and Information Systems, Plenum Press, New York, pp.341-367, 1980.

[3] W. Bandler and L. J. Kohout, "Semantics of Implication Operators and Fuzzy Relational Products," Intl. Journal of Man-Machine Studies, Vol.12, pp.89-116, 1980.

[4] W. Bandler and L. J. Kohout, "Fuzzy Power Sets and Fuzzy Implication Operator," Fuzzy Set and Systems 4, pp.13-30, 1980.

[5] 현우석, "퍼지관계곱 기반 급성복통과 관련된 지능형 질환 진단시스템의 설계 및 구현", 정보처리학회논문지B, 제10-B권 제2호, pp.197-204, 2003.

[6] M. Ayoubi, "Neuro-fuzzy structure for rule generation and application in the fault diagnosis of technical processes," Proc. of American Control Conference, Seattle, pp.2757-2761, 1995.

[7] F. C.-H. Rhee and R. Krishnapuram, "Fuzzy rule generation methods for high-level computer vision," Fuzzy Sets and Systems, Vol.60, pp.245-258, 1993.

[8] S. Mirta and S. K. Pal, "Fuzzy multi-layer perceptron, inference and rule generation," IEEE Trans., Neural Networks, Vol.6, pp.51-63, 1995.

[9] E. Tazaki and N. Inoue, "A generation method for fuzzy rules using neural networks with Planar Lattice architecture," Proc. of IEEE Int. Conf. Neural Networks, pp. 1743-1748, 1994.

[10] Shiqian Wu, Meng Joo Er and Yang Gao, "A Fast Approach for Automatic Generation of Fuzzy Rules by Generalized Dynamic Fuzzy Neural Networks," IEEE Transactions on Fuzzy Systems, Vol.9, No.4, 2001.

[11] T. W. Cheng, D. B. Goldgof and L. O. Hall, "Fast clustering with application to fuzzy rule generation," Proc. of IEEE Int. Conf. Fuzzy Syst., pp.2289-2295, 1995.

[12] Yi Lu and Tie Qi Chen, "Fast Rule Generation and Membership Function Optimization for a Fuzzy Diagnosis System," The Tenth International conference on Industrial & Engineering Applications of Artificial Intelligence Expert Systems, June, 1997.

[13] 김성학, "Genetic 알고리즘을 이용한 풀 온도 제어 시스템의 지식베이스 최적화", 정보처리학회논문지, 제1권 제3호, pp. 319-326, 1994.

[14] 이종우, 김유섭, 김성동, 이재원, 채진석, "패턴 매칭과 자동 규칙 생성에 기반한 2단계 주식 트레이딩 시스템", 정보처리학회논문지B, 제10-B권 제3호, pp.257-264, 2003.



현 우 석

e-mail : wshyun@bible.ac.kr

1987년 이화여자대학교 전자계산학과 (이학사)

1993년 이화여자대학교 교육대학원 컴퓨터교육전공(교육학 석사)

2001년 국립경상대학교 대학원 컴퓨터 과학과(공학 박사)

1987년~1993년 한국외환은행 전산실

2001년~2002년 경희대학교 전자정보학부 컴퓨터공학전공 강의 전임강사

2002년~2004년 한국성서대학교 정보과학부 전임강사

2004년~현재 한국성서대학교 정보과학부 조교수

관심분야 : 인공지능, 지식기반 시스템, 퍼지 시스템, 정보검색 시스템, 의료정보시스템, bioinformatics, 정보 보안 등