사례기반 추론을 이용한 설비 고장시기 예측

이재식† · 이영주‡
†아주대학교 경영대학 경영학부 교수 · ‡단암데이타시스템 컨설팅팀 주임

Equipment Malfunction Time Prediction using Case-based Reasoning

Jae Sik Lee, Young Ju Lee

요 약

설비에 고장이 발생하여 고객이 수리를 요청하기 전에 미리 고객을 방문하여 예방점검을 실시하는 것은 고객의 만족도를 높이고 수리기술자의 효과적인 활용을 위해서 매우 중요한 활동이다. 본 연구에서는 설비에 고장이 발생하여 수리가 이루어진 후에 그 설비의 다음 고장은 언제 발생할 것인가를 예측하기 위하여 사례기반 추론을 적용하였다.

Key words : 사례기반 추론, 고장진단.

1. 서 론

설비의 고장에 대한 애프터 써어비스는 고객의 만족도를 높이는데 큰 영향을 미치므로 많은 기업들이 그 중요성을 인식하고 있다. 그러나, 이러한 애프터 써어비스 활동의 대부분은 일단 고장이 발생하여 고객이 수리 요청을 한 후에야 수리기술자가 고객을 방문하여 이루어진다. 하지만 고객의 만족도를 좀더 높이기 위해서는 수리 요청이 오기 전에 미리 방문하여 고장 발생을 막는 소위 예방점검이 체계적으로 수행되어야 한다. 효과적인 예방점검은 설비의 고장 시기를 예측하는데서 시작한다. 본 연구에서는 설비의 고장발생 및 수리에 대한 과거 사례들을 이용하여 다음 고장 시기를 예측하고자 한다. 예측 기법으로는 사례기반 추론(CBR : Case-based Reasoning) [Kolodner, 1993]을 사용한다.

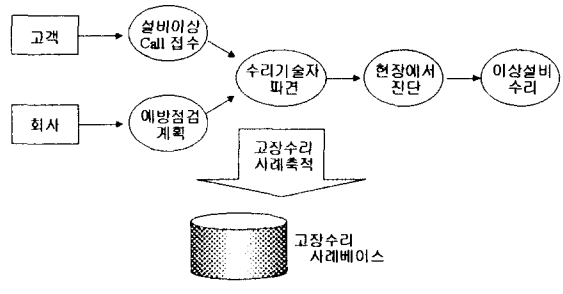
고장 진단에 인공지능 기법들을 이용한 연구는 폭넓게 수행되어 왔는데, 사례기반 추론을 이용한 기존 연구를 보면, 전자장비[Rudiger *et al.*, 1994], 비행기[Magaldi, 1994; Michel and Eric, 1996] 또는 사무용 설비[이재식과 전용준, 1995; Lee and Xon, 1996]의 고장진단 등의 분야에 적용되었다. 한편, 고장 진단 분야에 사례기반 추론과 다른 인공지능 기법이 통합되어서 하이브리드 방식으로 적용되기도 하였는데, 의료진단[Rissland *et al.*, 1993; Fathi-Torbaghan and Meyer, 1995; Reategui *et al.*, 1997], 건물 결함진단[Watson and Abdullah, 1994], 통신 네트워크 진단[Jiang *et al.*, 1995], 사무용 설비의 고장진단[이재식과 김영길, 1998] 등의 분야에 적용되었다.

2. 문제 영역

본 연구는 복사기를 중심으로 하여 각종 사무 기기를 제조·판매하는 A사의 설비 고장진단을 대상으로 한다. A사는 다양한 모델의 제품들을 판매하고, 판매된 제품들에서 발생하는 이상(EM : Equipment Malfunction)을 수리하며, 제품과 관련된 소모품을 공급하는 역할도 하고 있다. A사의 고객의 경우에는 제품의 기본적인 성능 뿐 아니라 사후 써어비스에 대한 점을 중요한 요소로 생각하기 때문에 고객을 위한 써어비스가 A사의 주요 관리 대상이 되고 있다.

만약 설비에 이상이 있다고 판단될 경우 대부분의 고객은 전문적 지식이나 경험이 부족하기 때문에 스스로 문제를 해결하기보다는 써어비스센터로 고장에 대한 신고(Call)와 써어비스 요청을 하고, 회사에서는 신고를 접수한 후, 즉 Call 발생 후 수리 기술자를 파견하여 소비자의 문제를 해결하여야 한다. 그러나 고객의 수가 점차 증가하고, 판매되어 현장에서 사용되고 있는 제품의 수가 증가함에 따라 고객의 수리에 대한 기술자의 파견이 제대로 이루어지지 못하고 있는 실정이다. 이러한 상황 때문에 많은 고객들이 불만을 가지게 되고 고객의 써어비스 만족도가 점차로 떨어지게 된다 회사는 이러한 문제점을 해결하기 위해 예방점검이라는 제도를 마련하여 고장이 일어나지 않은 기계에 대해서도 미리 방문하여 기계를 점검해 주고 손상된 부분이나 고장이 예상되는 부분에 대해 조치를 해 주는 써어비스를 현재 진행하고 있다. 이상의 과정을 그림으로 도식화하면 <그림 2.1>과 같다.

본 연구에서 다루는 데이터는 4종류의 기종, 4,044대의 기계에 대한 총 31,146건의 방문사례로 구성되어 있다. 이들 중 예방점검은 <표 2.1>에서 보는 바와 같이 927대의 기계, 2,613건에 대해서만 수행되었다. 이는 전체 데이터의 약 8.4%에 해당하는 수치로서, EM 방문 즉 고객의 Call을 받고 방문한 경우의 38.8%에 비하여 극히 미미한 예방점검만이 수행되고 있음을 알 수 있다.



<그림 2.1> 설비 이상 처리 및 데이터 축적 과정

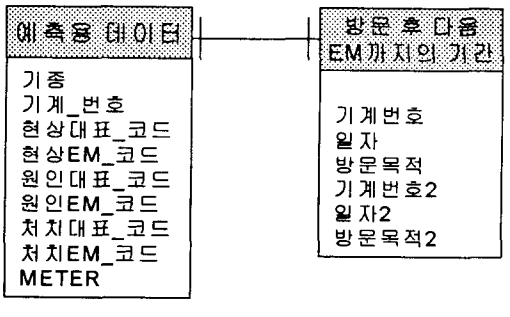
<표 2.1> 방문 목적별 발생수와 방문 목적내용

코드	내용	발생수	%
01	EM	12110	38.8
02	예방점검	2613	8.4
03	점검의뢰	541	1.7
04	재방문	3310	10.6
05	MOD	19	0.0
06	고객COVER	9748	31.3
07	영업활동	136	0.4
08	설치	2266	7.3
09	철수	115	0.4
10	이동설치	160	0.5
11	이동철수	12	0.0
12	CONVERSION	113	0.4
13	FIELD 정비	2	0.0
14	O/H 설치	0	0.0
15	O/H 철수	1	0.0
총 건수		31146	100.0

효과적인 예방점검 계획을 수립하기 위해서는 고장이 발생할 시기를 예측할 수 있어야 한다. 예방점검이 효과적으로 수행되어야만 고객의 Call이 감소할 것이며 고객의 만족도는 향상될 것이다.

3. 데이터 집합 및 사전 처리

본 연구에 사용된 데이터는 A사에서 이루어지는 고객 방문에 관한 정보를 바탕으로 하였다. 1996년 7월 1일부터 1998년 1월 6일까지의 기간 동안 고객의 요청이 발생한 후 수리 기술자가 파견되어 고장을 수리하기까지의 과정에서 수집되는 정보들을 데이터 베이스에 축적한 데이터로서 'TSCS_방문'(TSCS : Technical Service Control System)과 '방문'으로 구성되어 있다. 'TSCS_방문'은 설비의 고장에 대한 데이터로서 12개의 속성으로 구성되어 있고, '방문'은 방문과 직접적으로 관련된 정보를 나타내는 데이터로서 41개의 속성으로 구성되어 있다. 관련 개체 관계도(ERD : Entity Relationship Diagram)는 <그림 3.1>, <그림 3.2>와 같다. 이 데이터들은 총 31,146건의 양에 비해 많은 양의 데이터가 Missing Value를 가지고 있거나 잘못된 값을 가지고 있기 때문에 연구에 적합한 데이터베이스로 만들기 위해 전처리 과정을 거쳐 <그림 3.3>과 같은 예측용 데이터베이스를 추출하였다.



<그림 3.3> 예측을 위한 데이터베이스 구조

'TSCS_방문'의 속성 12개와 '방문'의 속성 41개를 하나의 데이터베이스로 통합한 후 중복되는 속성인 '순번'과 '방문목적'을 삭제하면 총 51개의 속성이 모이게 된다. 이 51개의 속성 중 고장시기의 예측을 위해서 어떠한 속성을 선택할 것이냐는 예측의 적중률을 높이는데 많은 영향을 미친다. 주요 속성의 선택은 설명력이 약한 쓸모 없는 속성들은 생략함으로써, 저장되는 속성의 수를 줄이고 사례베이스의 크기를 줄이는 효과가 있다[Lee, 1994]. 만일 사례를 구성하는 모든 속성을 사용한다면, 사례의 증가에 따라 점차로 사례베이스가 커져서 사례에 대한 처리에 컴퓨터 능력이 소요되는 정도가 커져 시스템의 속도가 매우 떨어지게 되기 때문이다. 주요 속성들을 골라내기 위한 표준적인 방법은 존재하지 않으므로 이 연구에서는 영역지식에 의해 속성을 선정하는 방법과 각 속성의 값에 따라 Call 발생 횟수가 현저히 달라지는 속성을 선정하는 방법에 의해 속성을 선택하였다. 예측용 데이터베이스의 하나의 레코드는 <표 3.1>과 같은 구조로 이루어져 있다.

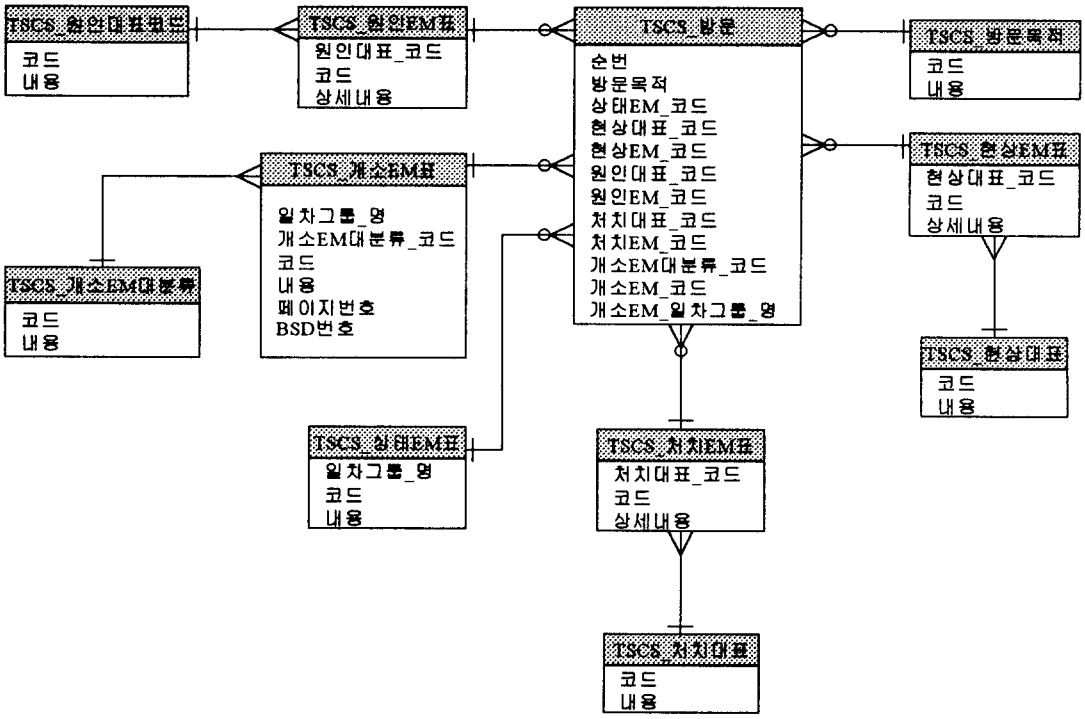
<표 3.1> 예측용 데이터의 레코드 구조

필드 명	데이터 형식
기종	문자열
기계_번호	문자열
현상대표_코드	문자열
현상EM_코드	문자열
원인대표_코드	문자열
원인EM_코드	문자열
처치대표_코드	문자열
처치EM_코드	문자열
METER	숫자
다음_고장시기	숫자

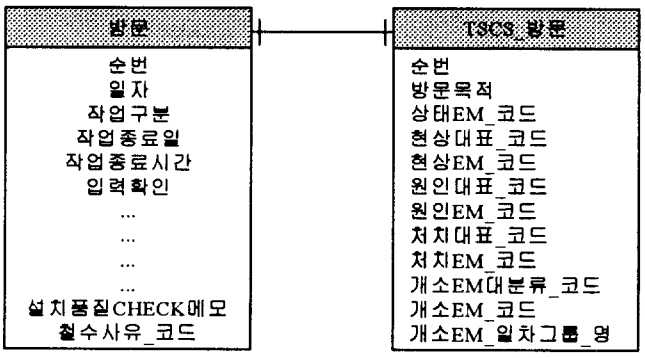
<표 3.1>에서 '대표_코드'라 함은 현상, 원인 또는 처치의 대분류 코드이다. 예를 들어 원인의 경우 수십 종류의 원인이 있을 수 있는데, 이러한 원인들을 설정치 불량, 기계적 불량 등 9종류로 대분류하여 코드를 부여하였다. '현상대표_코드' '원인대표_코드'와 '처치대표_코드'는 각각 그 세분류로 'EM_코드'를 가지고 있다. 'EM_코드'는 각 '대표_코드'에서 분류된 세부적인 내용들에 대한 코드이다. 'METER'라는 속성은 복사기의 사용량을 나타낸다. 기계가 사용될 때마다 그 양을 누적한 수치이다. 일반적으로 생각할 때 기계의 사용기간이 길어지거나, 사용회수가 증가하면 즉, 'METER' 수치가 증가하면 고장은 더욱 자주 발생할 것이다.

본 연구에서 예측하고자 하는 속성인 '다음_고장시기'란 현재 고장이 발생하여 점검을 받은 기계가 다음 언제에 고장이 일어날 것인가를 나타내는 속성을 뜻한다. '다음_고장시기'는 현재 가지고 있는 데이터에는 명시되어 있지 않으므로 '방문' 테이블로부터 각 기계의 Call이 발생한 일자와 바로 그 다음 Call이 발생한 일자와의 차이를 구함으로써 '다음_고장시기'를 산출하였다.

추출된 예측용 데이터베이스 안의 사례는 총 5,710건이었는데, 이 사례들 중에서 무작위로 추출된 100건의 사례를 Test Set으로, 또 다른 100건은 Training Set으로 사용하였으며 나머지 5,510건은 사례베이스로 사용하였다.



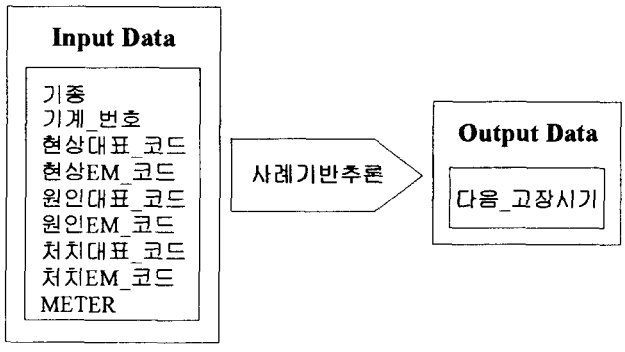
<그림 3.1> 'TSCS_방문' 및 관련 개체들간의 ERD



<그림 3.2> '방문'과 'TSCS_방문'의 ERD

4. 사례기반 추론

예측을 위하여 어떠한 기법을 적용할 것인가는 문제 영역의 특성이나 데이터의 특성에 따라 좌우될 수 있다. 본 연구의 대상이 되는 데이터는 문자열 형식의 속성과 숫자 형식의 속성을 모두 가지고 있으므로, 이 속성들에 적용 가능한 기법 중 하나인 사례기반 추론을 적용하여 <그림 4.1>과 같이 연구를 수행하였다.



<그림 4.1> 연구 모형

입력 속성으로는 <그림 4.1>에서 보는 바와 같이 9개의 속성을 사용하며, 출력 속성은 다음_고장시기이다. 입력 사례에 대하여, 사례베이스내의 사례 k의 총 유사도 점수(TSS : Total Similarity Score)는 다음과 같은 식으로 나타난다.

$$TSS_k = \sum_{i=1}^{9} SS_{ki} \qquad \text{for } k = 1, \ldots, K$$

SS_{ki}는 사례 k의 속성 i의 유사도 점수(Similarity Score)이며 10점을 만점으로 한다. 즉 모든 속성에 대해 완전히 일치할 경우 TSS는 90점이 되는 것이다.

각 입력 속성에 대한 세부적인 유사도 점수 부여방법을 살펴보면 다음과 같다. 기종·현상대표_코드·원인대표_코드·처치대표_코드 등은 일치할 때에는 10점, 일치하지 않을 때에는 0점을 부여하였다.

다음은 기계번호에 대한 유사도 점수이다. 본 연구에서 사용된 사례에는 4가지 기종에 대한 방문기록이 있는데, A, B, C 기종의 기계들은 10자리로 구성된 기계번호를 가지고 있으며, D 기종은 5자리나 6자리의 기계번호를 가지고 있다. 기계번호는 왼쪽에서 오른쪽으로 갈수록 대분류의 기계로부터 세분류의 기계로 나뉘어 감을 나타낸다. 기계번호에 대한 비교는 왼쪽에서부터 이루어진다. 왼쪽의 문자부터 하나씩 비교해 가는데, 만일 상위 왼쪽의 문자가 일치하지 않을 경우에는 하위 문자에 대해서는 비교하지 않는다. 기계번호의 유사도 점수를 M_{SS} 라고 한다면, M_{SS} 는 아래와 같이 계산된다.

$$M_{SS} = \frac{\text{일치한 문자열의 수}}{\text{문자열의 수}} \times 10$$

10을 곱한 이유는 각 입력 속성마다 할당된 10점 만점이라는 기준에 맞추기 위하여서이다. 이 때 만약 비교하는 두 문자열의 개수가 일치하지 않을 경우에는 두 문자열 중에서 문자열의 개수가 큰 것을 문자열의 수로 정한다. 기계번호 '000BS14370' 과 '000BU10470' 을 비교한다고 한다면 왼쪽으로부터 4번째까지는 일치하나 다섯번째 문자는 'U' 와 'S' 로 다르므로 그 이후에 일치하 문자열이 3개가 있지만 이것을 제외하고 가장 왼쪽부터 일치한 4개에 한해서 점수를 주게되는 것이다. 그 결과 이 기계번호에 대한 유사도 점수는 $\frac{4}{10} \times 10$ 으로 계산되어 4점을 받게 된다. 기계번호 '291839' 와 '29657' 이라는 기계를 비교한다고 한다면 일치 문자수는 2개이며, 문자열의 개수는 '291839' 의 6개와 '29657' 의 5개중 큰 수인 6개로 한다. 따라서 $\frac{4}{6} \times 10 = 6.7$ 점의 점수를 받는다.

METER 에 대한 유사도 점수는 전체 점수 10점에서 두 METER 간의 차이를 10점 만점으로 환산하여 아래와 같이 계산하려 하였으나,

$$10 - \frac{|\text{입력사례 } METER - \text{비교사례 } METER|}{METER\text{의 최대값인 } 58,443,469} \times 10$$

이러한 방식으로 유사도 점수를 부여 할 경우 나누어주는 METER의 최대값이 너무 크기 때문에 METER간의 차이를 충분히 반영하지 못하는 단점이 있다. 대안으로 선택된 방법은 METER의 차이를 구간으로 나누어 점수를 부여하는 방법이다. 아래의 <표 4.1>은 구간에 따른 유사도 점수를 보여주고 있는데, 여기서 MD란 입력사례 METER와 비교사례 METER의 차이이다.

<표 4.1> METER 차이의 유사도 점수

구간	유사도 점수
40,000,000< MD	0
35,000,000< MD <40,000,000	1
30,000,000< MD <35,000,000	2
25,000,000< MD <30,000,000	3
20,000,000< MD <25,000,000	4
15,000,000< MD <20,000,000	5
10,000,000< MD <15,000,000	6
5,000,000< MD <10,000,000	7
1,000,000< MD <5,000,000	8
500,000< MD <1,000,000	9
0< MD <500,000	10

각각의 EM_코드의 경우에는, 대표_코드가 일치하여야만 EM_코드의 비교가 가능하다. 따라서 대표_코드가 일치할 때에만 EM_코드에 대한 비교를 수행하여, 일치시에는 10점, 불일치 시에는 0점을 부여한다.

5. 다음 고장시기 예측 결과

다음_고장시기 예측을 위한 시스템(CBMTP : Case-based Malfunction Time Prediction)은 Visual Basic 6.0과 Access 97 데이터베이스 관리 시스템을 사용하여 Personal Computer에서 구현되었다.

연구의 첫 단계에서는 먼저 총유사도 점수가 가장 높은 사례를 찾은 다음, 그 사례가 가지고 있는 다음_고장시기를 예측값으로 추천하는 방식으로 수행하였다. 이 때 각 입력속성에 주어지는 가중치들을 변화시켜 가면서 가장 적중률이 좋을 때 어떤 가중치가 주어졌는지를 찾았다. 적중률이 가장 좋은 가중치를 찾은 다음에는 가중치는 변화시키지 않고 고정시킨 채, 추천하는 사례

를 찾는 방법에 변화를 주면서 여섯 가지의 실험을 수행하였다.

《실험 1》에서는 하나의 사례가 입력되었을 때 사례베이스 안에 있는 모든 사례들과 각각 비교하여 총유사도 점수를 구한다. 그 중 총유사도 점수가 가장 높은 사례가 가지고 있는 다음_고장시기의 값이 입력 사례의 다음_고장시기로 추천된다.

다음_고장시기가 가질 수 있는 값의 범위는 짧게는 1일에서부터 가장 긴 경우는 522일로 관찰되었다. 이는 본 연구에 쓰인 데이터가 1년 6개월간의 데이터를 수집한 것이기 때문이며, 사례의 수집 기간이 길어진다면 예측하고자 하는 값들의 범위는 더욱 넓어질 것이다. 따라서 예측된 다음_고장시기의 값이 정답과 정확히 일치하는 경우만을 맞는 것으로 보기에는 무리가 있으므로, 본 연구에서는 예측된 값과 정답이 일정한 오차 범위 안에 들어가면 맞은 것으로 간주하고 적중률을 구하였다. 오차의 범위를 정답의 앞·뒤 5일부터 시작하여 1일씩 증가시켜 가며 구한 적중률은 <표 5.1>과 같다.

<표 5.1> 《실험 1》의 적중률

오차범위	적중률	오차범위	적중률
±5	28	±14	47
±6	33	±15	48
±7	35	±16	48
±8	37	±17	50
±9	37	±18	50
±10	42	±19	51
±11	43	±20	52
±12	45	±25	55
±13	46	±30	58

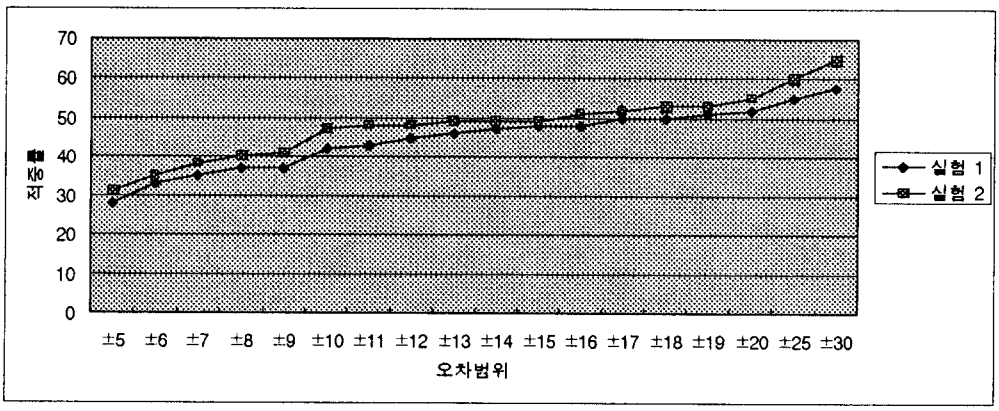
《실험 2》는 총유사도 점수가 높은 상위 3개의 사례를 찾아 그 중에서 가장 가까운 다음_고장시기를 추천하는 방식이다. 고장을 예측한다는 것은 고장이 일어나기 전에 미리 예방한다는 점이 중요하므로 정답보다 늦은 시기를 추천한다는 것은 무의미할 것이다. 그러므로 상위 3개 사례의 다음_고장시기의 값들 중 가장 가까운 값을 추천하는 것이 합리적이다. 이러한 방식으로 추천된 값에 대하여 《실험 1》과 같은 방법으로 구한 적중률은 <표 5.2>와 같다.

<표 5.2> 《실험 2》의 적중률

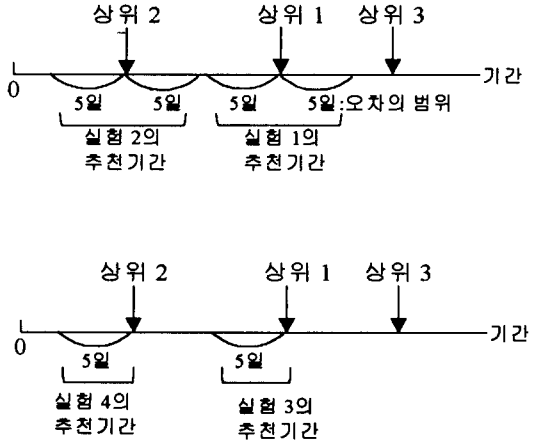
오차범위	적중률	오차범위	적중률
±5	31	±14	49
±6	35	±15	49
±7	38	±16	51
±8	40	±17	52
±9	41	±18	53
±10	47	±19	53
±11	48	±20	55
±12	48	±25	60
±13	49	±30	65

<표 5.1>과 <표 5.2>의 값들을 비교해 볼 때 상위 1개의 값을 추천하는 것보다는 상위 3개의 값 중 가장 가까운 것을 추천할 때 좀 더 나은 적중률을 보이는 것을 알 수 있다. 이 2개의 실험을 그래프로 나타내면 <그림 5.1>과 같다.

앞에서도 언급했듯이 다음_고장시기의 예측은 그 시기가 정답보다 이르다면 상관이 없으나 정답보다 늦은 값이 추천된다면 그 의미가 없다. 고장이 발생하기 전에 방문이 이루어져야 하며, 고장이 발생한 이후가 된다면 이미 고객의 고장 발생 Call이 일어났다고 볼 수 있기 때문이다. 위의 《실험 1》과 《실험 2》는 이러한 맥락에서 볼 때 적합하지 않다. 즉, 오차를 정답의 앞·뒤에서 모두 인정하지 말고, 정답의 앞쪽의 오차만을 인정하는 것이 더 합리적이다. 각 실험에 대한 이해를 돕고자 추천 방식과 오차 및 추천 기간을 그래프로 나타내면 <그림 5.2>와 같이 표현된다. 《실험 1》과 《실험 2》에서 추천 기간은 선택된 사례가 가지고 있는 값의 앞·뒤로 오차범위를 적용하여 정한 것인데 반해, 《실험 3》과 《실험 4》는 선택된 사례가 가지고 있는 값의 앞으로만 오차 범위를 적용하여 정한다. 또한 《실험 1》과 《실험 3》에서는 총유사도 점수가 가장 높은 상위 1의 사례만을 고려하여 다음_고장시기를 추천하는 것이고, 《실험 2》와 《실험 4》는 총유사도 점수 상위 1, 2, 3위의 다음_고장시기들 중 가장 작은 값을 추천하는 것이다. 예를 들어, <그림 5.2>의 《실험 2》와 《실험 4》에서는 상위 2위의 사례가 가장 작은 다음_고장시기 값을 가지고 있으므로, 이것이 추천되는 것이다.



<그림 5.1> 《실험 1》과 《실험 2》에 대한 그래프



<그림 5.2> 각 실험의 추천방법

<표 5.3>은 《실험 3》의 적중률이고, <표 5.4>는 《실험 4》의 적중률이다. 이 두 실험의 비교 그래프는 <그림 5.3>과 같다.

<표 5.3> 《실험 3》의 적중률

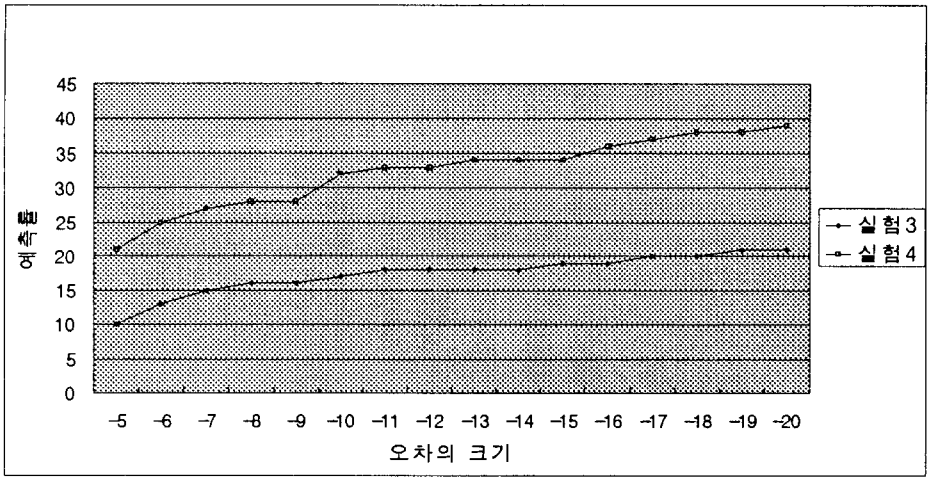
오차범위	적중률	오차범위	적중률
-5	10	-14	18
-6	13	-15	19
-7	15	-16	19
-8	16	-17	20
-9	16	-18	20
-10	17	-19	21
-11	18	-20	21
-12	18	-25	23
-13	18	-30	23

《실험 3》과 《실험 4》의 결과는 《실험 1》과 《실험 2》의 결과에 비해 낮은 적중률을 보이고 있다. 이는, 《실험 1》과 《실험 2》는 예측하고자 하는 시기의 앞이나 뒤쪽 어디의 오차범위에나 해당하면 적중한다고 본 것임에 반해, 《실험 3》과 《실험 4》는 예측하고자 하는 시기의 앞쪽 오차범위에 해당하여야만 적중한 것으로 간주하는 방식이기 때문에 낮은 적중률을 보일 수 밖에 없다고 생각된다. 《실험 3》과 《실험 4》의 관계는 《실험 1》과 《실험 2》의 관계와 마찬가지로 상위 1개에 의한 추천보다는 상위 3개를 이용한 추천이 훨씬 나은 적중률을 보임을 알 수 있다.

<표 5.4> 《실험 4》의 적중률

오차범위	적중률	오차범위	적중률
-5	21	-14	34
-6	25	-15	34
-7	27	-16	36
-8	28	-17	37
-9	28	-18	38
-10	32	-19	38
-11	33	-20	39
-12	33	-25	42
-13	34	-30	45

본 실험에서 이용한 사례 기반 시스템을 실무에 적용하기에는 단점이 있다. 기술자가 어떠한 한 사례가 들어올 때마다 매일 시스템을 이용하여 다음_고장시기를 예측하고, 이에 따라 일일 단위로 고장이 예상되는 바로 그 시점에 방문하는 것은 어려운 일이다. 따라서 실무진의 이용상의 편의나, 방문기간을 예측하고자 했던 목적에 보다 부합하는 시스템을 위하여서는 15일이나 월(月) 단위로 예측하는 것이 나을 것이라 생각된다. 《실험 5》와 《실험 6》에서는 이런 생각을 바탕으



<그림 5.3> 《실험 3》과 《실험 4》의 그래프

를 찾는 방법에 변화를 주면서 여섯 가지의 실험을 수행하였다.

《실험 1》에서는 하나의 사례가 입력되었을 때 사례베이스 안에 있는 모든 사례들과 각각 비교하여 총유사도 점수를 구한다. 그 중 총유사도 점수가 가장 높은 사례가 가지고 있는 다음_고장시기의 값이 입력 사례의 다음_고장시기로 추천된다.

다음_고장시기가 가질 수 있는 값의 범위는 짧게는 1일에서부터 가장 긴 경우는 522일로 관찰되었다. 이는 본 연구에 쓰인 데이터가 1년 6개월간의 데이터를 수집한 것이기 때문이며, 사례의 수집 기간이 길어진다면 예측하고자 하는 값들의 범위는 더욱 넓어질 것이다. 따라서 예측된 다음_고장시기의 값이 정답과 정확히 일치하는 경우만을 맞는 것으로 보기에는 무리가 있으므로, 본 연구에서는 예측된 값과 정답이 일정한 오차 범위 안에 들어가면 맞은 것으로 간주하고 적중률을 구하였다. 오차의 범위를 정답의 앞·뒤 5일부터 시작하여 1일씩 증가시켜 가며 구한 적중률은 <표 5.1>과 같다.

<표 5.1> 《실험 1》의 적중률

오차범위	적중률	오차범위	적중률
±5	28	±14	47
±6	33	±15	48
±7	35	±16	48
±8	37	±17	50
±9	37	±18	50
±10	42	±19	51
±11	43	±20	52
±12	45	±25	55
±13	46	±30	58

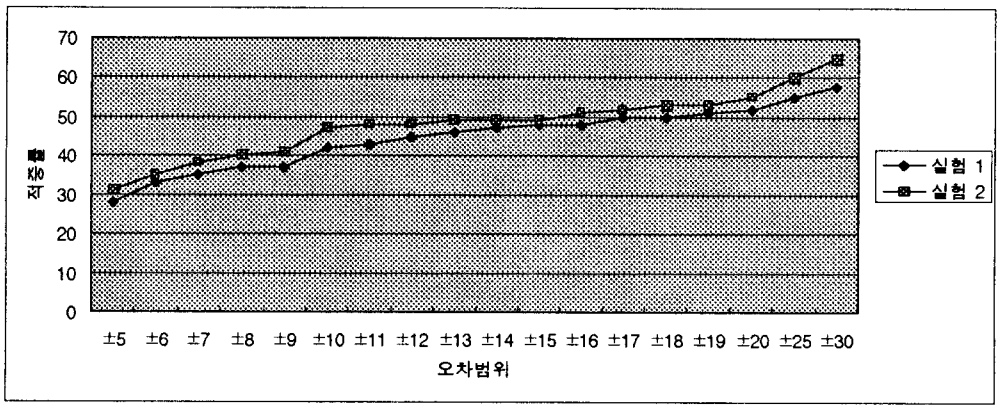
《실험 2》는 총유사도 점수가 높은 상위 3개의 사례를 찾아 그 중에서 가장 가까운 다음_고장시기를 추천하는 방식이다. 고장을 예측한다는 것은 고장이 일어나기 전에 미리 예방한다는 점이 중요하므로 정답보다 늦은 시기를 추천한다는 것은 무의미할 것이다. 그러므로 상위 3개 사례의 다음_고장시기의 값들 중 가장 가까운 값을 추천하는 것이 합리적이다. 이러한 방식으로 추천된 값에 대하여 《실험 1》과 같은 방법으로 구한 적중률은 <표 5.2>와 같다.

<표 5.2> 《실험 2》의 적중률

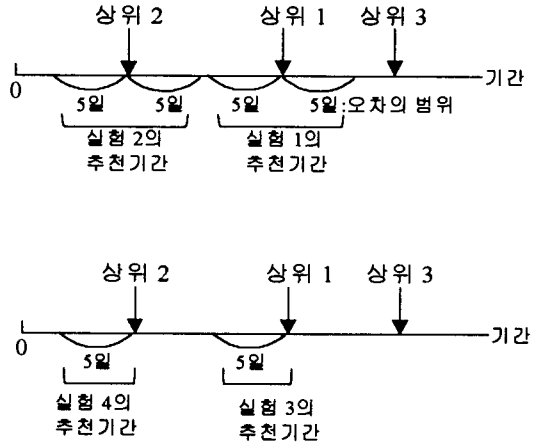
오차범위	적중률	오차범위	적중률
±5	31	±14	49
±6	35	±15	49
±7	38	±16	51
±8	40	±17	52
±9	41	±18	53
±10	47	±19	53
±11	48	±20	55
±12	48	±25	60
±13	49	±30	65

<표 5.1>과 <표 5.2>의 값들을 비교해 볼 때 상위 1개의 값을 추천하는 것보다는 상위 3개의 값 중 가장 가까운 것을 추천할 때 좀 더 나은 적중률을 보이는 것을 알 수 있다. 이 2개의 실험을 그래프로 나타내면 <그림 5.1>과 같다.

앞에서도 언급했듯이 다음_고장시기의 예측은 그 시기가 정답보다 이르다면 상관이 없으나 정답보다 늦은 값이 추천된다면 그 의미가 없다. 고장이 발생하기 전에 방문이 이루어져야 하며, 고장이 발생한 이후가 된다면 이미 고객의 고장 발생 Call이 일어났다고 볼 수 있기 때문이다. 위의 《실험 1》과 《실험 2》는 이러한 맥락에서 볼 때 적합하지 않다. 즉, 오차를 정답의 앞·뒤에서 모두 인정하지 말고, 정답의 앞쪽의 오차만을 인정하는 것이 더 합리적이다. 각 실험에 대한 이해를 돕고자 추천 방식과 오차 및 추천 기간을 그래프로 나타내면 <그림 5.2>와 같이 표현된다. 《실험 1》과 《실험 2》에서 추천 기간은 선택된 사례가 가지고 있는 값의 앞·뒤로 오차범위를 적용하여 정한 것인데 반해, 《실험 3》과 《실험 4》는 선택된 사례가 가지고 있는 값의 앞으로만 오차 범위를 적용하여 정한다. 또한 《실험 1》과 《실험 3》에서는 총유사도 점수가 가장 높은 상위 1의 사례만을 고려하여 다음_고장시기를 추천하는 것이고, 《실험 2》와 《실험 4》는 총유사도 점수 상위 1, 2, 3위의 다음_고장시기들 중 가장 작은 값을 추천하는 것이다. 예를 들어, <그림 5.2>의 《실험 2》와 《실험 4》에서는 상위 2위의 사례가 가장 작은 다음_고장시기 값을 가지고 있으므로, 이것이 추천되는 것이다.



<그림 5.1> 《실험 1》과 《실험 2》에 대한 그래프



<그림 5.2> 각 실험의 추천방법

<표 5.3>은 《실험 3》의 적중률이고, <표 5.4>는 《실험 4》의 적중률이다. 이 두 실험의 비교 그래프는 <그림 5.3>과 같다.

<표 5.3> 《실험 3》의 적중률

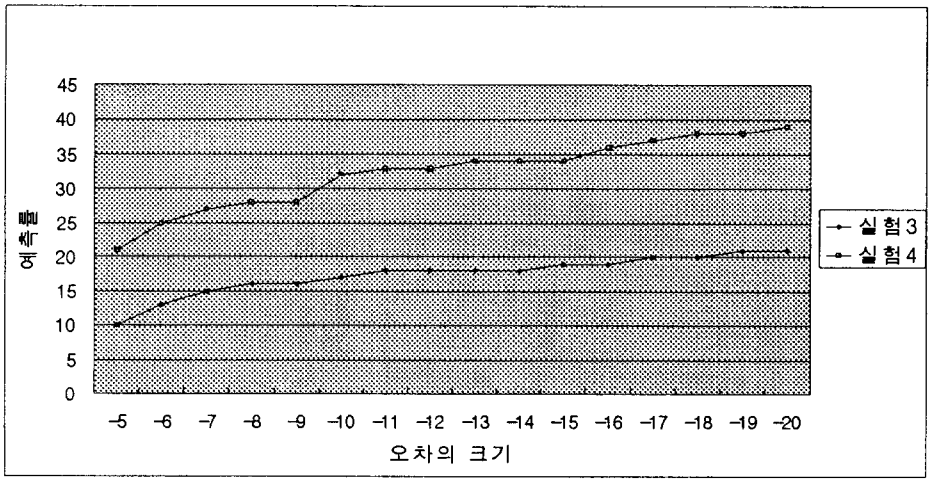
오차범위	적중률	오차범위	적중률
-5	10	-14	18
-6	13	-15	19
-7	15	-16	19
-8	16	-17	20
-9	16	-18	20
-10	17	-19	21
-11	18	-20	21
-12	18	-25	23
-13	18	-30	23

《실험 3》과 《실험 4》의 결과는 《실험 1》과 《실험 2》의 결과에 비해 낮은 적중률을 보이고 있다. 이는, 《실험 1》과 《실험 2》는 예측하고자 하는 시기의 앞이나 뒤쪽 어디의 오차범위에나 해당하면 적중한다고 본 것임에 반해, 《실험 3》과 《실험 4》는 예측하고자 하는 시기의 앞쪽 오차범위에 해당하여야만 적중한 것으로 간주하는 방식이기 때문에 낮은 적중률을 보일 수 밖에 없다고 생각된다. 《실험 3》과 《실험 4》의 관계는 《실험 1》과 《실험 2》의 관계와 마찬가지로 상위 1개에 의한 추천보다는 상위 3개를 이용한 추천이 훨씬 나은 적중률을 보임을 알 수 있다.

<표 5.4> 《실험 4》의 적중률

오차범위	적중률	오차범위	적중률
-5	21	-14	34
-6	25	-15	34
-7	27	-16	36
-8	28	-17	37
-9	28	-18	38
-10	32	-19	38
-11	33	-20	39
-12	33	-25	42
-13	34	-30	45

본 실험에서 이용한 사례 기반 시스템을 실무에 적용하기에는 단점이 있다. 기술자가 어떠한 한 사례가 들어올 때마다 매일 시스템을 이용하여 다음_고장시기를 예측하고, 이에 따라 일일 단위로 고장이 예상되는 바로 그 시점에 방문하는 것은 어려운 일이다. 따라서 실무진의 이용상의 편의나, 방문기간을 예측하고자 했던 목적에 보다 부합하는 시스템을 위하여서는 15일이나 월(月) 단위로 예측하는 것이 나을 것이라 생각된다. 《실험 5》와 《실험 6》에서는 이런 생각을 바탕으



<그림 5.3> 《실험 3》과 《실험 4》의 그래프

로 <표 3.2>에 있는 일일단위의 다음_고장시기 데이터를 15일 단위와 월 단위로 변환하여 실험을 하였다. 이 때 15일 단위로 예측한 《실험 5》의 결과는 43 %의 적중률을, 1개월 단위로 예측한 《실험 6》의 결과는 47%의 적중률을 보였다.

《실험 6》의 알고리즘을 구현한 모습이 <그림 5.4>의 CBMTP 이다. 이 시스템은 사용자가 고장의 시기를 알고 싶어하는 기계에 대한 정보를 예측하고자 하는 사례라고 되어있는 부문에 입력한 후 실행을 시키면 가장 유사한 사례를 찾아 그 사례의 내용들과 함께 고장이 일어날 시기를 월(月)단위로 추천해 주도록 되어 있다.

본 연구에서 수행한 6번의 실험 중에서 《실험 6》의 결과에 대하여 평가하여보면 다음과 같다. 《실험 6》의 적중률은 47%로서 이 결과를 수치만으로 평가한다면 그다지 높은 적중률을 보이지 않는 것처럼 보인다. 본 연구에서 예측하고자 하는 다음_고장시기는 실험에 사용된 사례베이스 내에서만 고려하여 보아도 1년 6개월 즉 18개월 중에서 하나를 맞추는 것이다. 이 중 고장이 일어난 월(月)의 이후에 대하여 예측을 하게 되므로, 총 18가지에서 일어난 월(n이라고 하자)을 맨 수가 경우의 수가 된다. 따라서 무작위로 예측하였을 때 예측한 결과가 맞을 확률은 불과

$$\sum_{i=1}^{100} \frac{1}{18-n_i} \div 100 \times 100(\%) = 18.9\%$$

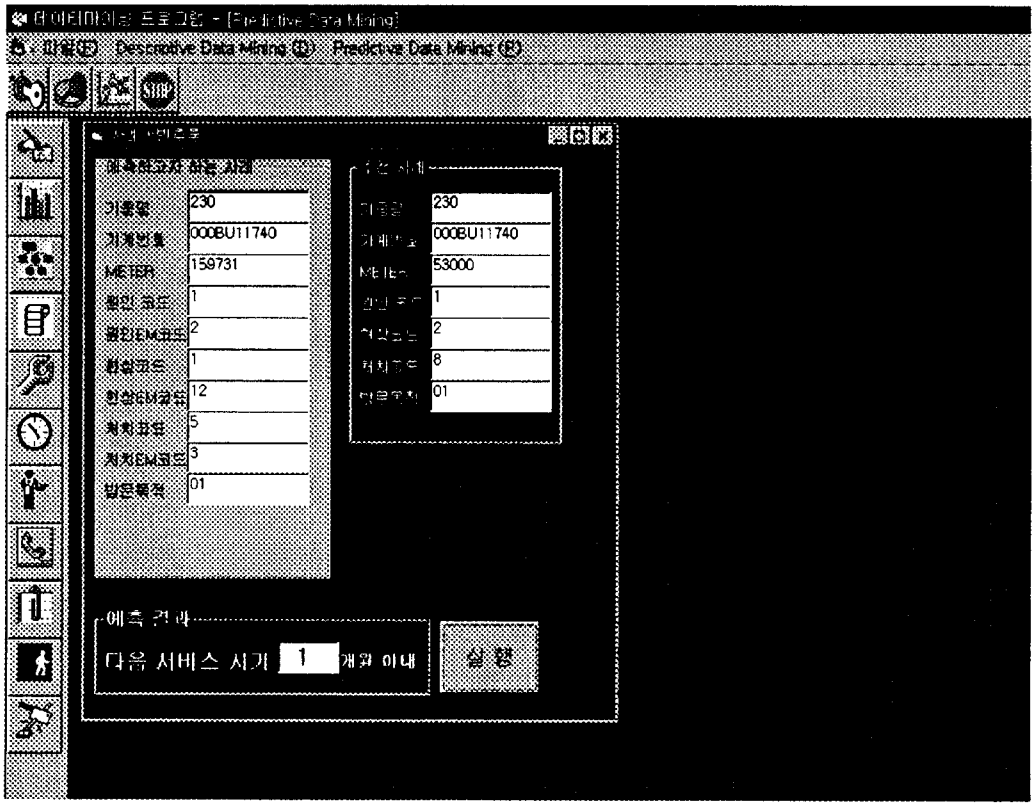
밖에 되지 않는다. 이 18.9%와 CBMTP의 47%를 비교해 보면 CBMTP의 적중률이 결코 낮은 수치라고 할 수 없다.

사람이 무작위로 다음_고장시기를 예측하고 그 정확성을 기대한다는 것은 어려운 일이다. 그러나 CBMTP를 이용한 예측은 이미 발생하였던 사례들을 수집하고 그것을 근거로 하여 다음_고장시기를 예측하기 때문에 일정 수준이상의 정확성을 주는 일관된 예측을 할 수 있다는 장점이 있다. 또한 예측에 소요되는 시간의 측면도 매우 중요한 평가 항목이 된다. CBMTP를 이용하여 100개의 사례를 예측하는 경우 평균 120분 정도의 시간이 소요되었다. 즉, 하나의 사례에 약 1분 정도의 시간이 소요되므로 신속성의 측면에서 CBMTP는 상당히 효과적임을 알 수 있다.

6. 결 론

본 연구에서는 설비에 이상이 발생하여 수리가 이루어진 후에, 그 설비의 다음 고장은 언제 발생할 것인가, 즉 다음_고장시기를 예측하는 문제를 다루었다. 다음_고장시기를 추천하는 방법에 따라 6가지의 실험을 실시하였는데, 다음_고장 시기를 예측하는 것은 일일 단위의 예측보다는 1개월 단위의 예측이 효과적이었으며, 적중률은 약 47%이었다.

본 예측시스템을 도입하였을 때의 효과를 살펴보면 다음과 같다. 첫째, 다음_고장시기 예측에 소요되는 시간을 줄이고 일관성을 유지한다. 둘째, 고장이 발생하기 전에 미리 예방점검을 함으로써 고객들이 중요하게 생각하는 사후 관리 측면에서 만족감을 주게 되므로 고객 써어비스 만족도가 향상되어 더 나은 이익을 창출할 수 있다. 셋째, 고장이 발생할 시기를 미리 알 수 있으므로, 수리를 위해 파견되는 기술자들의 배치 계획을 미



<그림 5.4> CBMTP 실행 화면

리 수립할 수 있어 효과적인 인적 자원 관리를 할 수 있으며, 인건비 측면에서도 이익이 기대된다.

본 연구의 한계 및 향후 연구 방향은 다음과 같다. 첫째, 사례 수집의 문제점을 들 수 있다. 여러 오류나 Missing Value를 가지고 있는 데이터가 전체 양에 비하여 상대적으로 많아 사용할 수 없는 사례가 많았으며, 예측하고자 하는 속성의 범위가 방대하고 예측에 사용된 속성들의 조합이 무수히 많다는 것에 비한다면 실험에 사용된 데이터는 극히 부족하다고 볼 수 있다. 둘째, 실험 결과 비교의 문제점을 들 수 있다. 본 연구의 결과를 다른 기법들 즉 통계적인 기법이나, 다른 인공지능 기법들을 이용하여 나온 결과와 함께 상대적인 성능 평가를 하였다면 보다 나은 연구가 되리라 생각된다. 셋째, 사례기반 추론 시스템의 입력 속성 선정에 과학적 접근을 시도하지 못했다. 본 연구에서는 도메인에 대한 지식으로 속성을 선정하였는데, 이는 Feature Clustering이나 요인분석, 판별분석, Genetic Algorithm, Neural Network 등을 이용한 속성 선정보다는 체계적인 접근방법이라 말할 수 없으므로 체계적이고 과학적인 접근을 시도해 볼 필요가 있다. 넷째, 다른 속성에 대한 예측 문제이다. 본 연구에서는 다음_고장시기라는 한가지 속성에 대해 예측을 하였으나, 다음 고장이 언제 어떤 원인에 의해서 일어날 것인지를 예측할 수 있다면 예방점검이 보다 효과적일 것이다.

참고문헌

[1] 이재식, 김영길, "규칙 및 사례기반의 하이브리드 고장진단 시스템," 한국전문가시스템학회지, 제 4권 1호 (1998), 115~131.

[2] 이재식, 전용준, "사례기반 추론에 근거한 설비이상 진단 시스템," 한국전문가시스템학회지, 제 1권, 2호 (1995), 85~102.

[3] Fathi-Torbaghan, M. and D. Meyer, "ICARUS : Integrating Rule-based and Case-based Reasoning on The Base of Unsharp Symptoms," *Proc. of the IEEE Int'l Conf. on Systems, Man and Cybernetics*, Vol. 3 (1995), 2424~2427.

[4] Jiang, S., D. Siboni, A. A. Rhissa, and G. Beuchot, "Intelligent and Integrated System of Network Fault Management: Artificial Intelligence Technologies and Hybrid Architectures," *Proc. of IEEE Singapore Int'l Conf. on Networks / Int'l Conf. on Information Engineering*, (1995), 265~268.

[5] Lee, H., "A Case-Based Forecasting System," 한국경영과학회지, 19권 2호 (1994), 134~152.

[6] Lee, J. S. and Y. X. Xon, "A Customer Service Process Innovation using the Integration of Data Base and Case Base," *Expert Systems with Applications*, Vol. 11, No. 4 (1996), 543~552.

[7] Magaldi, R. V., "CBR for Troubleshooting Aircraft on the Flight Line," *IEE Colloquium(Digest)*, March, I057 (1994), 6/1~6/9.

[8] Michel, M. and A. Eric, "Using Data Mining to Improve Feedback from Experience for Equipment in the Manufacturing and Transport Industries," *IEE Colloquium(Digest)*, I198 (1996), 1/1~1/9.

[9] Reategui, E. B., J. A. Campbell, and B. F. Leao, "Combining a Neural Network with Case-based Reasoning in a Diagnostic System," *Artificial Intelligence in Medicine*, January, Vol. 9, I1 (1997), 5~27.

[10] Rissland, E. L., J. J. Daniels, Z. B. Rubinstein, and D. B. Skalak, "Case-based Diagnostic Analysis in a Blackboard Architecture," *Proc. of the National Conf. on Artificial Intelligence*, (1993), 66~72.

[11] Rudiger, B., H. William, W. David, and W. John, "Case-based Reasoning System for Troubleshooting," *IEE Colloquium(Digest)*, March, I057 (1994), 5/1~5/9.

[12] Watson, I. and S. Abdullah, "Developing Case-based Reasoning Systems : A Case Study in Diagnosing Building Defects," *Proc. of the IEE Colloquium on Case-Based Reasoning : Prospects for Applications*, Digest No. 1994/057, London, UK, March 3 (1994), 1~3.