한국지능시스템학회 논문지 2010, Vol. 20, No. 1, pp. 94-100

패턴의 변화를 가지는 연속성 데이터를 위한 스트리밍 의사결정나무

Streaming Decision Tree for Continuity Data with Changed Pattern

윤태복* · 심학준* · 이지형* · 최영미**

Taebok Yoon*, HakJoon Sim*, Jee-Hyong Lee* and Young-Mee Choi**

*성균관대학교 컴퓨터공학과

**성결대학교 멀티미디어공학부

요 약

데이터 마이닝(Data Mining)은 환경으로부터 수집된 데이터에서 패턴을 추출하고 의미 있는 정보를 발견하기 위하여 주로 사용된다. 하지만, 기존의 방법은 데이터의 수집이 완료된 상태에서 분석하는 것을 기반으로 하고 있으며, 시간의 흐름에 따른 패턴의 변화를 반영하기 어렵다. 본 논문은 연속성(Continuity data), 대량성(Large scale) 그리고 패턴의 가변성(Changed pattern)과 같은 특성을 가지는 스트림 데이터(Stream Data)의 분석을 위한 스트리밍 의사결정 나무(Streaming Decision Tree : SDT) 방법을 소개한다. SDT는 연속적으로 발생하는 데이터를 블록으로 정의하고, 각 블록은 의사결정나무 학습 방법을 이용하여 규칙을 추출한다. 추출된 규칙은 발생 시간, 빈도 그리고 모순 등을 고려하여 결합하였다. 실험에서는 시계열 데이터를 이용하여 분석하였고, 적절한 결과를 확인하였다.

키워드 : 스트리밍 데이터 마이닝, 연속성 데이터 분석, 스트리밍 의사결정 나무

Abstract

Data Mining is mainly used for pattern extracting and information discovery from collected data. However previous methods is difficult to reflect changing patterns with time. In this paper, we introduce Streaming Decision Tree(SDT) analyzing data with continuity, large scale, and changed patterns. SDT defines continuity data as blocks and extracts rules using a Decision Tree's learning method. The extracted rules are combined considering time of occurrence, frequency, and contradiction. In experiment, we applied time series data and confirmed resonable result.

Key Words : Streaming Data Mining, Continuity Data Analysis, Streaming Decision Tree

1. 서 론

유비쿼터스 컴퓨팅이나 지능형 교육 시스템 등과 같은 지능적인 서비스를 필요로 하는 환경에서 데이터 마이닝은 유용하게 사용되고 있다. 데이터 마이닝은 수집된 데이터로부터 의미 있는 정보를 추출하기 위해 사용되는데, 데이터를 수집하고 분석하여 패턴을 추출하여, 그 정보를 기반으로 적절한 서비스를 제공하는 것인 일반적인 패러다임이라 하겠다[1]. 하지만, 기존의 데이터 마이닝 방법은 수집이 완료된 상태에서 데이터를 분석하는 것을 기반으로 하고 있다. 더욱이 수집된 데이터 안에서 시간의 흐름에 따라 변하는 패턴이 존재하는 경우, 이를 반영한 분석 방법은 찾아보기 쉽지 않다. 예를 들어 휴대전화 통신사에서 사용자의 송수신 정보를 이용하여 통화 패턴을 분석한다고 할 때, 단위 시간당 발생하는 통화정보의 양은 매우 방대할 것이다. 또는, CCTV와 같은 영상 감시 장치로부터 실시간으로 이상 징후를 판단한다고 할 때, 짧은 시간에 축적되는 방대한 양의 데이터에서 의미 있는 결과를 얻는 것은 어려운 일이다. 이뿐만 아니라 주식 시장의 주가 변동률, 원자력 발전소의 원자로 온도 모니터링과 같은 센서 네트워크 환경 그리고 네트워크 환경에서 패킷 분석을 통한 서비스 분야에 이르기까지 매우 다양하다[4].

또한 이런 연속성을 가지는 데이터의 특징 중에는 시간의 흐름에 따라 패턴의 변화가 발생 할 수 있는데, 이와 같은 연속성, 대량성 그리고 패턴의 변화 등의 특성을 가진 데이터를 스트림 데이터(Stream Data)라고 하며, 그 특징은 다음과 같다.

- 대량성(Large Scale) : 데이터의 크기가 너무 방대하여

접수일자 : 2009년 11월 30일
완료일자 : 2010년 2월 3일
감사의 글 : 본 논문은 2009년도 문화체육관광부 재원으로 한국콘텐츠진흥원의 지원을 받아 수행된 차세대 게임 전문 교육기관 지원 사업의 연구 결과입니다. 연구비 지원에 감사드립니다.
"본 논문은 본 학회 2009년도 추계학술대회에서 선정된 우수논문입니다."

분석이 쉽지 않고, 분석한다고 해도 해석이 어렵다.

- 연속성(Continuity) : 데이터의 수집이 완료된 상태가 아니라 지속적으로 데이터가 축적되고 있다.
- 가변성(Changed Pattern) : 데이터는 시간의 흐름에 따라 패턴이 변하는 모습을 보인다.

앞서 설명한바와 같이 시간의 흐름에 따라 지속적으로 데이터가 축적되는 것은 분석에 많은 어려움 가져온다. 또한 실시간 가공의 어려움을 해결하기 위하여 단순하게 저장 공간에 누적하는 것은 분석 할 수 없을 정도의 방대한 양의 데이터로 만들게 된다. 또한 시간의 흐름에 따라 수집된 데이터는 패턴의 변화를 가질 수 있으며, 패턴의 변화에 따른 적절한 서비스가 요구된다. 예를 들어 지능형 이러닝 환경에서 학습자의 학습 수준, 성향에 따라 맞춤형 학습 컨텐츠를 제공해 준다고 가정하자. 이때 먼저 선행되어야 하는 과정은 학습자의 상태를 파악하는 과정이 요구 된다. 이를 우리는 학습자 인지과정이라고 말하며, 인지과정의 결과물을 학습자 모델이라고 한다. 하지만, 이 학습자 모델은 시간이 지남에 따라 변하게 된다. 학습 과정의 흐름에 따라 지적 수준이 향상되고 이에 따른 학습 성향도 변할 수 있기 때문이다. 본 논문은 대량성, 연속성 및 가변성의 성질을 가지는 스트림 데이터의 분석을 위한 스트리밍 의사결정 나무(Streaming Decision Tree : SDT)를 제안한다. SDT는 연속적으로 발생하는 데이터를 분석하기 위해 데이터를 임계 구역(Threshold block : TB)으로 구분하고, 의사결정나무 방법을 이용하여 TB를 분석하여 패턴(규칙)을 추출한다. 다수 구간의 TB에서 분석된 결과는 발생 시간(Time), 빈도(Frequence) 그리고 모순(Conflict)을 고려하여 결합하였다. 실험에서는 UCI Machine Learning Repository[6]의 Waveform데이터를 이용하였고, 적절한 결과를 확인하였다.

2. 관련 연구

스트리밍 데이터 분석을 위한 스트리밍 데이터 마이닝(Streaming Data Mining) 기술은 다양하게 연구되고 있다. Aggarwal[10]는 스트림 데이터를 분석하기 위한 다양한 방법을 소개하고 있다. 그는 스트림 데이터를 분석하기 위하여 군집(Clustering), 분류(Classification), 빈도 패턴 추출, 패턴 변화, 인덱싱 기법, 슬라이딩 기법 등 여러 가지 유용한 방법을 조사하였다. Babcock[2] 등은 스트림 데이터 환경에서 모델 생성과 이슈에 대하여 소개하였고, Jain[4]은 다양한 환경에서의 스트림 데이터 발생을 소개하고, 통계적 방법을 이용한 스트림 데이터 분석을 제안하였다. 김진화와 민진영[3]은 연속 발생 데이터의 실시간 데이터 분석을 위해 슬라이딩 윈도우(Sliding Window)를 이용하여 분할하고 의사결정나무 방법을 이용하여 분석하였다. 하지만, 각각의 슬라이딩 윈도우에서 분석을 통하여 만들어진 규칙을 통합하는 과정이 명확하지 않고 단순하게 처리하여 효율성이 낮다. Golab[5] 등은 스트림 데이터의 저장/관리를 위한 다양한 방법을 제시하였다. Domingos[7]은 연속적인 데이터 발생 환경에서 효과적인 분석을 위한 VFDT(Very Fast Decision Tree)을 제안하였다. VFDT는 호페딩 트리 방법에 기반을 두고 있으며, 매우 빠른 데이터의 발생에 효과적으로 처리 할 수 있는 방법을 제안한다 하지만, 연속성 데이터의 분석을 고려할 뿐, 시간에 따른 데이터 패턴의 변화는 고려하기 어렵다는 문제를 가지고 있다. Hashemi와 Yang[9]은 패턴이 변하고, 노이즈(Noise) 데이터를 포함하고 있는 스트림 데이터를 분석 할 수 있는 FlexDT(Flexible decision tree) 방법을 제안하였다. 이 방법은 VFDT를 이용하였고, 퍼지 이론을 이용하여 노이즈 데이터를 처리하는 방법을 소개하였다.

기존의 방법들은 스트림 데이터의 연속성에 초점을 맞춰 문제를 해결하였다. 스트림 데이터의 연속성, 대량성 그리고 가변성의 특징을 모두 고려한 분석 방법은 찾아보기 힘들다.

3. 데이터의 패턴 변화

본 논문은 연속성, 대량성 및 패턴의 변화를 가지는 데이터를 스트림 데이터라고 정의하였다. 여기서 패턴의 변화요소는 연속성이나 대량성 요소와 다른 성격을 가진다. 연속성과 대량성이 데이터가 가지는 외적 성질이라고 한다면, 패턴의 변화는 내적인 성질을 의미한다. 데이터 패턴의 변화는 연속성 또는 시계열 정보를 가지는 데이터에서 순서나 시간의 흐름에 따라 패턴의 변화를 가지는 데이터라고 정의할 수 있다. 이런 데이터는 도메인에 따라 다양하게 확인할 수 있는데, 예를 들어 지능형 이러닝 환경에서는 학습자의 지식수준을 들 수 있고, 게임 환경에서는 게임 플레이어의 게임 운영 수준, 인터페이스 친숙도에 따른 조작 능력 등을 말한다. 실생활에서는 사용자 관심의 변화나 다수 사용자의 성향 변화 등을 들 수 있다. 이와 같은 패턴은 그 변하는 형태에 따라 일반형, 선형 변환형, 구간 반복형, 비 예측형으로 구분하여 나눌 있다(표 1).

표 1. 패턴이 변화는 데이터의 분류
Table 1. Type of Changing Pattern data

구 분	설 명
일반형	패턴의 변화가 없는 일관된 형태
선형 변화형	패턴이 일관되게 변화하는 형태
구간 반복형	과거 패턴이 반복적으로 나타나는 형태
비 예측형	패턴이 없이 변화하는 형태

일반적으로 데이터 분석에 가장 많이 사용되는 형태가 일반형이다. 패턴이 일관되게 포함되어 있어서 임의의 구간 데이터를 이용하여 분석해도 결과가 동일하다. 선형 변화형은 분석 방법과 목적이 기존 일반형과 차이를 가진다. 과거의 정보를 기반으로 미래를 추이/추정하는 것이다. 예를 들어 일반형은 “과거에 1, 2, 3을 선택했다는 패턴을 찾았다”면, “미래에도 1, 2, 3 선택할 것이다”라고 대답하지만, 선형 변화형은 “과거에 1, 2, 3을 선택했다”면, “미래에는 4, 5를 선택할 것이다”라고 대답하는 형태를 말한다. 구간 반복형은 일반형의 반복적 형태라고 할 수 있고, 비 예측형은 넓은 의미에서는 일반형과 유사하지만, 데이터가 일관되지 못하여 패턴을 찾기 힘든 형태라 할 수 있다. 본 논문에서는 구간 반복형 형태의 데이터 변화에 대하여 구체적으로 논의하겠다. 그림 1은 패턴 변화의 형태를 그림으로 표현하였다.

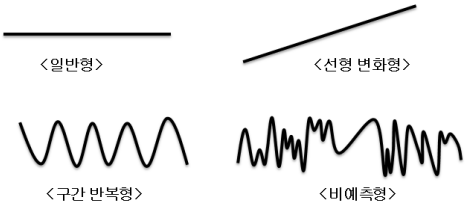


그림 1. 다양한 패턴 변화 데이터
Fig. 1. Variety of Changing Pattern data

3. 스트리밍 의사결정나무 (Streaming Decision Tree : SDT)

스트리밍 의사결정나무(Streaming Decision Tree : SDT)는 연속성과 대량성의 특징을 가지는 데이터에서 패턴의 변화를 고려한 분석방법이다. SDT를 이용하기 위해서는 먼저 적용하려는 도메인(Ubiquitous Computing, Game, e-Learning, etc.)을 이해하고 해결하려는 문제에 따라 속성을 정의한다. 정의된 속성에 따라 분석에 사용될 데이터를 수집한다. SDT를 위한 분석에는 그림 2와 같이 4가지 단계를 거치게 된다.

첫째, 데이터 수집과 임계 구역 설정은 연속적으로 발생하는 데이터에서 임계 구역(Threshold block:TB) 단위로 저장/관리하는 것을 의미한다. 임계 구역의 크기는 시간이나 데이터의 크기로 지정할 수 있으며, 적용하려는 분야의 특성에 따라 사용자가 선택하여 결정한다. 예를 들어 어떤 환경에서는 1초에 10MByte가 수집되고, 또 다른 환경에서는 1초에 10KByte가 수집된다고 할 때, 동일한 임계 구역을 지정하는 것은 비효율적일 것이다.

둘째, 의사결정나무 분석은 임계 구역으로 모아진 데이터를 이용하여 패턴을 추출하는 과정이다. SDT에서는 C4.5[8] 방법을 이용하였으며, 분석 결과는 If-then 형식의 규칙으로 변환하여 저장한다.

셋째, 규칙의 결합(Confluence of Rules)은 각 구역 단위 분석을 통하여 얻어진 규칙을 통합하는 과정이다. 규칙의 결합 과정에서는 발생 빈도수(Appearance Frequency), 최근 지속 시간(Duration Time) 그리고 오류 빈도수(Conflict Frequency)를 이용하여 규칙의 유효성을 수치화 한다.

넷째, 모델 생성은 누적하여 발생한 규칙 집합에 대하여 유효성 수치를 기반으로 신뢰성을 판단하고, 규칙의 형태(일치형, 포함형, 모순형)를 고려하여 규칙을 결합한 최종 결과물이다.

앞서 설명한 네 가지 단계 중에서 임계 구역에 따른 데이터 수집과 분석, 그리고 규칙의 결합에 대하여 보다 구체적으로 설명하겠다.

3.1 임계 구역과 작업 흐름

SDT를 위해서는 먼저 데이터의 양과 발생 간격 및 시간 등을 고려하여 임계 구역(Threshold block : TB)의 크기를 결정해야 한다. 수집되는 데이터가 TB의 크기를 만족할 때, 의사결정나무를 이용하여 규칙을 추출한다. 그림 3은 과거에서 현재까지 시간의 흐름에 따라 발생하는 데이터와 그 데이터를 임계 구역으로 묶어서 규칙을 생성하고, 생성된 규칙을 결합하는 과정을 보여주고 있다.

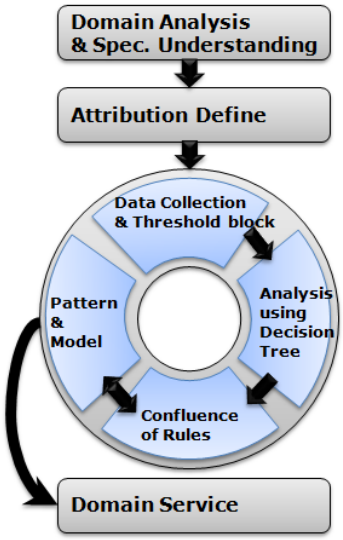


그림 2. SDT를 위한 작업 흐름도
Fig. 2. Workflow for SDT

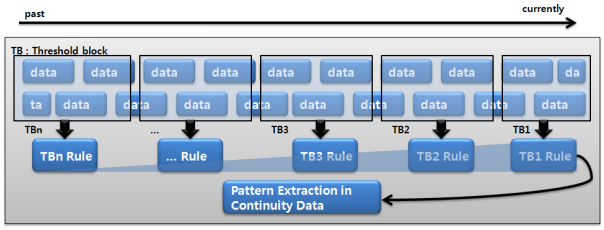


그림 3. 임계 구역과 규칙의 결합
Fig. 3. Threshold block(TB) and Rule Confluence

3.2 규칙의 결합

규칙의 결합(Rule Confluence)은 임계 구역 별로 발생한 규칙들을 결합하는 과정을 의미한다. 규칙을 결합할 때는 규칙간의 모순이나 포함 관계를 고려할 수 있어야 하고, 각각의 규칙에 대하여 신뢰 정도를 측정할 수 있어야 한다. 또한, 패턴의 변화를 반영하기 위해 규칙의 발생 시기(Aging)를 반영할 수 있어야 한다. 규칙과 규칙간의 유사형태에 따라 일치형, 포함형 그리고 모순형으로 분류하였다. 일치형의 경우는 비교 하고자 하는 두 규칙이 일치하거나 유사성이 높아, 둘 중에 하나를 선택하여 사용하는 경우를 이야기 한다(그림 4).

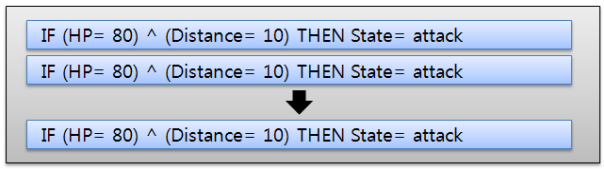


그림 4. 두 규칙의 일치형 사례
Fig. 4. Example of Concurrence type

포함형은 하나의 규칙이 다른 규칙에 일부 또는 전체가 포함되는 경우를 의미한다. 하나의 규칙에는 여러 개의 조건을 포함하게 되는데, 예를 들어 "IF (HP=<50)^(Distance=<40)^(Item=<2) THEN State=chase" 규칙에서

는 “HP=<50”, “Distance=<40” 그리고 “Item=<2”와 같이 세 개의 조건을 가지고, “State=chase“ 결과를 가진다. 규칙 사이에 포함되어 있는 조건들이 일방적으로 한쪽에 포함될 수 도 있고, 부분적으로 포함관계를 가질 수 도 있다. 아래 예제는 부분적인 포함관계를 보이는 규칙의 예를 보여주고 있다. 첫 번째 규칙의 첫 번째 조건인 ”HP=<50“와 두 번째 규칙의 첫 번째 조건인 ”HP=<60“에서 공통영역으로 계산하여 ”HP=<50“을 이용하고, 첫 번째 규칙의 두 번째 조건과 두 번째 규칙의 두 번째 조건인 ”Distance=<40“와 ”Distance=<30“에서도 마찬가지로 공통 영역인 ”Distance=<30“이 선택된다. 첫 번째 규칙의 세 번째 조건인 ”tem=<2”의 경우 두 번째 규칙에 의해 없어도 만족하기 때문에 제거된다(그림 5).

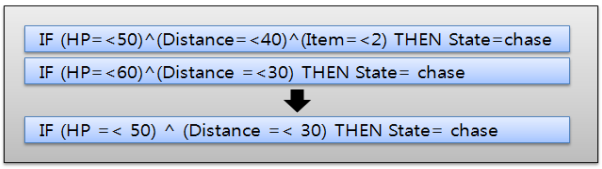


그림 5. 두 규칙의 포함형 사례
Fig. 5. Example of Inclusion type

모순형은 두 개의 규칙에 포함되어 있는 조건들은 같으나 결과가 상이하여 모순(Conflict)가 발생한 경우이다(그림 6). 이럴 때는 규칙이 가지고 있는 신뢰도를 이용하여 높은 값을 가지는 규칙을 남겨둔다. 규칙의 신뢰도는 발생 빈도수, 최근 지속 시간, 오류 빈도수를 이용하여 계산하며, 다음 장에서 구체적으로 소개하겠다.

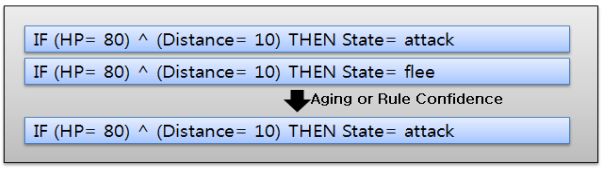


그림 6. 두 규칙의 모순형 사례
Fig. 6. Example of Conflict type

3.3 규칙의 신뢰도(Rule Confidence)

규칙 신뢰도는 모든 규칙에 대하여 빈도수, 최근 지속 시간 그리고 모순을 반영하여 규칙에 가중치를 부여한다. 발생 빈도수는 각각의 임계 구역별 분석을 통해 생성된 규칙들에서 얼마나 자주 동일한 규칙이 발생했는지 여부이다. 최근 지속시간은 과거에 발생했던 규칙 보다는 현재에 가까운 시간에 발생한 규칙을 보다 의미 있게 사용하기 위한 것으로 규칙에 나이(Aging)라고 생각할 수 있다. 그리고 오류 빈도수는 과거에서 현재까지 발생했던 규칙 중에서 모순이 발생했던 경우에 대하여 벌점(Penalty)을 주기 위한 것이다. 규칙의 가중치 부여를 위한 수식 **RuleConf**는 다음과 같다.

$$RuleConf = \frac{AF}{AF_{\max}} + \frac{DT}{DT_{\max}} - CF$$

AF : 발생 빈도수(Appearance Frequency)
DT : 최근 지속 시간(Duration Time)
CF : 오류 빈도수(Conflict Frequency)

여기서 AF는 발생빈도수를 의미하며 임계 구역 TB #n에서 얼마나 자주 발생 했는지를 파악하기 위한 것으로 수치가 높을 수 록 신뢰도가 높다고 할 수 있다. DT는 최근 지속 시간으로 최근 발생 빈도가 높을수록 높은 가중치를 부여하기 위함이다.

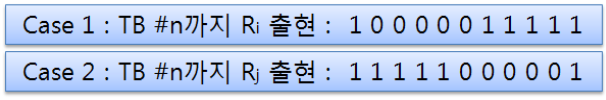


그림 7. 규칙 발생 빈도에 따른 비교
Fig. 7. Comparison of frequency

그림 7은 규칙의 발생 빈도수는 동일하나 발생된 시간적 차이가 다른 경우를 설명하기 위한 예이다. 예를 들어 Case 1과 Case 2 두 가지 경우가 존재한다. Case 1에서는 R_i라는 규칙이 처음 한번 발생하고 5번 동안 발생하지 않다가, 최근 5회 발생하였다. Case 2의 경우 R_j라는 규칙이 과거 5회 발생하였고, 최근에는 1회 발생 하였다. 이때 발생 빈도수만 놓고 비교 할 때는 동일한 결과를 가져온다. 하지만, 최근에 발생한 규칙에 높은 가중치를 부여한다면 당연히 Case 1이 의미 있는 규칙일 것이다. 그래서 어떤 규칙이 TB #n과 TB #n-1 에서 존재하면, DT는 1증가 하고 존재하지 않으면, DT는 2배 감소한다. 마지막으로 CF는 오류 빈도수를 의미하며 얼마나 자주 TB #n에서 오류가 발생했는지를 측정하여 가중치 반영하며 수치가 높을수록 신뢰도가 낮게 된다.

그림 8은 앞서 설명한 규칙 결합 과정을 예를 들어 설명한다. $TB_{\#1}$과 $TB_{\#2}$는 각각 3개의 규칙을 가지고 있다. 모든 규칙의 AF 값은 1로 초기화 되며, $TB_{\#12}$에서 일치형일 경우 AF 값은 2로 바뀌고, 최근에 발생하였으므로 DT의 값도 1씩 증가한다. $TB_{\#12}$와 $TB_{\#3}$에서는 두 번째 규칙이 모순이 발생하여 CF값이 1증가 한 것을 확인 할 수 있다. 또한 최근에 발생하지 않았기 때문에 DT의 값이 반으로 줄었다.

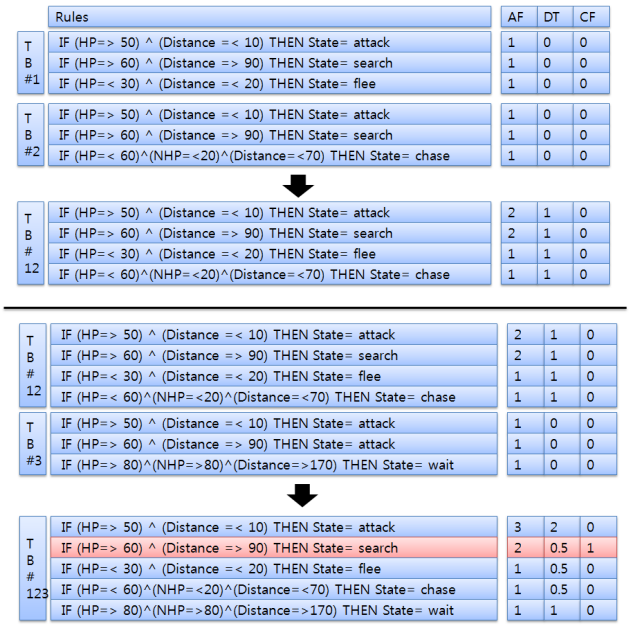


그림 8. 규칙 결합 예제
Fig. 8. Example of Rule Confluence

$TB_{\#123}$에서 두 번째 규칙의 신뢰도를 계산하면 아래와 같이 -0.09 값을 얻게 된다. 수집데이터 특성이나 도메인 환경에 따라 신뢰도 수치의 임계값을 정하여 규칙의 사용 여부를 결정한다.

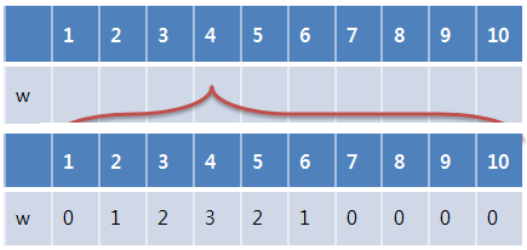
$$RuleConf = \frac{AF}{AF_{\max}} + \frac{DT}{DT_{\max}} - CF$$

$$= \frac{2}{3} + \frac{0.5}{2} - 1 = 0.66 + 0.25 - 1 = -0.09$$

4. 실 험

4.1 실험 환경 소개 및 데이터 생성

제안하는 방법의 실험을 위해 UCI Repository[6]의 Waveform 데이터와 같은 생성 방법을 이용하였다. 10개의 속성과 참과 거짓의 값을 가지는 클래스(Class)에 대하여 100개의 사례(Instance)를 20회 생성하였다. 1회 200개의 생성 데이터는 앞서 설명했던 임계 구역(Threshold block)과 같은 개념이다. 각 임계구역은 속성의 중요도를 부여하여 매번 패턴이 변화하는 효과가 있도록 하였다. 그림 9는 10개의 속성 중에서 4번째 속성의 의미를 높게 부여하는 모습을 보이고 있다. 속성 2는 1, 속성 3은 2, 속성 4는 3의 가중치(w)를 부여하여 순차적으로 의미가 있도록 하였다. 이와 같은 방법으로 모든 속성에 대하여 순회하면서 패턴의 변화를 가지는 데이터 집단을 임의 생성하였다.



	1	2	3	4	5	6	7	8	9	10
w	0	1	2	3	2	1	0	0	0	0

그림 9. Wave data 생성 예
Fig. 9. Example of Wave data

전체 4000개의 데이터를 생성했으며, 각각 200개 단위로 중요하게 여기는 속성이 다르게 구성하였다. 그림 10은 각 집단에서 가장 중요한 속성을 나타내고 있다. 1~200에서는 속성 7, 201~400은 속성 6 ... 3801~4000은 속성 8이 중요한 의미를 가진다.

Instance	200	400	600	800	1000	1200	1400	1600	1800	2000
Main Attrib.	A7	A6	A5	A4	A3	A2	A1	A10	A9	A8
Instance	2200	2400	2600	2800	3000	3200	3400	3600	3800	4000
Main Attrib.	A7	A6	A5	A4	A3	A2	A1	A10	A9	A8

그림 10. 패턴 변화를 가지는 Wave data
Fig. 10. Changing Patterns in Wave data

4.2 인공 데이터를 이용한 비교 실험

제안하는 방법의 유효성을 확인하기 위하여 패턴의 변화를 가지는 인공 데이터를 이용하여 세 가지 실험을 하였다. 시계열 순서를 가지는 4000개의 데이터에서 시간 t_n까지 학습에 사용하고, t_n이후 m 만큼을 검증에 사용하였다. 여기서 n의 값은 100, 500, 1000 ... 3500, 3800으로 그림 11과 같이 지정하였고, m의 값은 150으로 하였다. 첫 번째 실험(Normal DT_1)은 0~t_n 까지 학습하고, t_n 이후 150개를 테스트로 이용하였다. 이 방법은 일반적인 데이터 마이닝 방법으로 과거 정보를 기반으로 얼마나 미래 상황에 잘 대응하는지를 파악하기 위함이다. 본 실험에서는 과거의 데이터가 일관되지 않고 시간의 흐름에 따라 변하는 모습을 보이므로, 전통적인 분석 방법으로 얼마나 잘 대응하는지를 비교하였다.

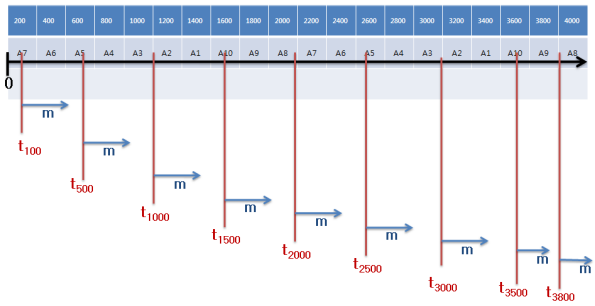


그림 11. 실험 1을 위한 데이터 설계
Fig. 11. Design of data for Experiment #1

두 번째 실험은 첫 번째 실험과 유사하나 학습 데이터의 범위를 축소시켰다. 즉, 수집된 데이터에서 학습 데이터 선별시 0~t_n 까지 선별하는 것이 아니라 p값만큼만 사용하였다. 과거 데이터를 모두 이용하여 미래를 예측하는 것보다는 패턴의 변화를 가지는 데이터에서는 가까운 과거에 대해서만 학습하고 미래를 예측하는 것인 더 좋은 결과를 가질 수 있기 때문이다. 그림 12는 p값을 반영한 실험 방법(Normal DT_2)에 대하여 보여주고 있다.

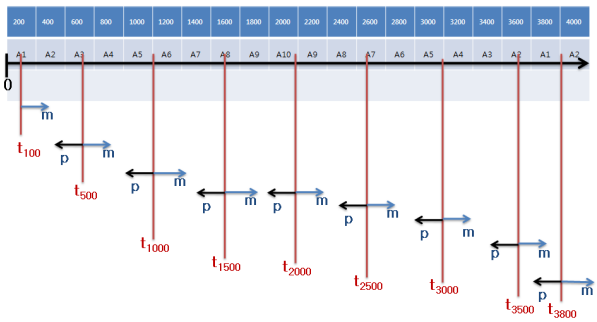


그림 12. 실험 2를 위한 데이터 설계
Fig. 12. Design of data for Experiment #2

세 번째 실험은 제안하는 방법을 이용하여 t_n까지의 데이터를 각각 분할하여 임계구역으로 정의하고, 분석에 사용하였다. 임계 구역 크기는 100으로 하였다. 각 블록의 규칙들은 신뢰도 값(**RuleConf**)을 기반으로 규칙들을 결합하여 최종 모델로 활용하였다. 세가지 실험 내용을 정리하면 표 2와 같다.

표 2. 실험 설계
Table 2. Design of Experiment

	실험명	Training set	Test set
1	Normal DT_1	0~t_n	t_n~m, m=150
2	Normal DT_2	p~t_n ,p=150	t_n~m, m=150
3	Streaming DT	0~t_n, SDT	t_n~m, m=150

세 가지 실험에서 규칙의 개수와 에러율을 비교하였다. 규칙의 개수에서, Normal DT_2은 매번 150개의 데이터만 학습에 반영하기 때문에 규칙의 개수가 30개를 넘지 않았다. Normal DT_1와 Streaming DT는 학습 데이터가 증가할수록 선형적으로 증가하는 모습을 보이고 있으나, DT_1보다 제안하는 방법인 Streaming DT가 규칙의 개수가 더 작게 나타는 것을 확인 할 수 있다(그림 13).

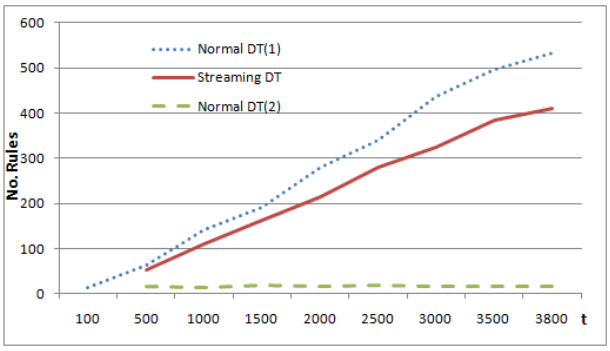


그림 13. 규칙의 크기 비교
Fig. 13. Comparison of Rule size

평균 에러율은 Normal DT_1은 29.1%, Normal DT_2 는 28.8% 그리고 Streaming DT는 26.2% 으로 나타났다. Normal DT_1은 처음보다 패턴이 변화되는 시점에서 에러율이 높아지는 모습을 보여고. 시간이 지나면서 조금씩 다시 낮아졌다. 하지만, 규칙의 개수가 지속적으로 증가하는 모습을 보여, 결과 해석의 어려움을 가져왔다. Normal DT_2는 직전 150를 이용하기 때문에 에러율의 편차가 크게 발생하는 모습을 보이고 있다. 제안하는 방법인 Streaming DT는 초기에는 비슷한 결과를 보이지만 시간이 지남에 따라, 기존의 방법에 비하여 좋은 결과를 보였다(그림 14).

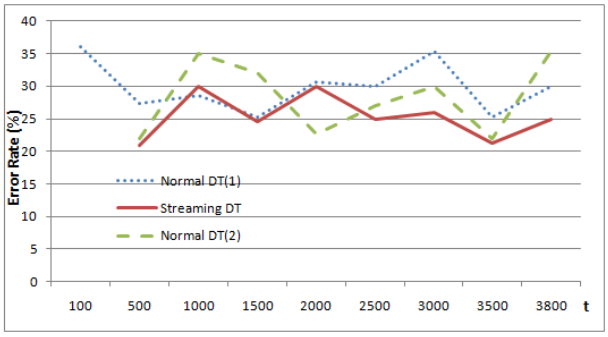


그림 14. 에러율 비교
Fig. 14. Comparison of Error rate

5. 결론 및 향후 연구

정보통신 기술의 발달과 함께 여러 환경에서 수집되는 데이터의 크기나 형태 및 속성이 다양한 모습을 보이고 있다. 본 논문은 패턴의 변화를 가지는 연속성 데이터에서, 효과적으로 분석할 수 있는 방법을 소개하였다. 또한, 임계 구역에 따라 개별적으로 발생한 패턴을 위해 규칙의 결합(Rule Confluence) 방법을 제안하였다.

실험에서는 시간의 흐름에 따라 변화를 가지는 데이터를 비교 분석 하였으며, 기존 의사결정나무 방법보다 규칙의 개수가 감소하면서 평균 에러율이 낮아진 모습을 확인 할 수 있었다.

향후 연구로는 임계영역에서 추출된 규칙의 결합을 위한 보다 효과적인 방법과, 제안하는 방법의 실제 도메인에 적용한 실용적 실험이 필요하겠다.

참 고 문 헌

[1] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, "Knowledge Discorvery and Data Mining : Towards a Unifying Framework", *KDD-96*, 1996.

[2] B. Babcock, S. Babu, M. Datar, R. Motwani, J. Widom, "Models and issues in data stream systems", *ACM SIGMOD-SIGACT-SIGART Symposium on principles of database systems*, 2002.

[3] 김진화, 민진영, "연속발생 데이터를 위한 실시간 데이터 마이닝 기법", *한국경영과학회지*, 2004.

[4] A. Jain, "Statistical Mining in Data Streams", *Ph.D. Dissertation, University of California, Santa Barbara*, 2006.

[5] L. Golab, M. Tamer Ozsu, "Issues in Data Stream Management", *SIGMOD Record*, Vol. 32, No. 2, 2003.

[6] UCI Machine Learning Repository Web site : http://archive.ics.uci.edu/ml/

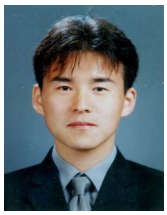
[7] P. Domingos and G. Hulten. "Mining high-speed data streams", *In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.

[8] J. Ross Quinlan, "*C4.5: Programs for Machine Learning*", Morgan Kaufmann, 1992.

[9] S. Hashemi and Y. Yang, "Flexible decision tree for data stream classification in the presence of concept change, noise and missing values", *Data Mining and Knowledge Discovery*, Vol. 19, No. 1, 2009.

[10] C. C. Aggarwal, "*Data Streams Models and Algorithms*, Chapter 1 : AN INTRODUCTION TO DATA STREAMS", Springer US, 2007.

저 자 소 개



윤태복(Taebok Yoon)
2001년 : 공주대학교 전자계산학과(학사)
2005년 : 성균관대학교 컴퓨터공학(석사)
2007년 : 성균관대학교 컴퓨터공학 (박사수료)

관심분야 : 사용자 모델링, 게임인공지능
Phone : 031-290-7987
Fax : 031-299-4637
E-mail : tbyoon@skku.edu

심학준(HakJoon Sim)
2009년 : 동서대학교 네트워크공학과(학사)
2009년~현재 : 성균관대학교 컴퓨터공학 (석사과정)

관심분야 : 기계학습, Web Intelligence
E-mail : trainto@skku.edu

이지형(Jee-Hyong Lee)
1993년 : 한국과학기술원 전산학과(학사)
1995년 : 한국과학기술원 전산학과(석사)
1999년 : 한국과학기술원 전산학과(박사)
2002년~현재 : 성균관대학교 정보통신 공학부 부교수

관심분야 : 지능시스템, 기계학습, 온톨로지
E-mail : jhlee@ece.skku.ac.kr

최영미(Young-Mee Choi)
1979년 : 이화여자대학교 수학과(학사)
1981년 : 이화여자대학교 전산학(석사)
1993년 : 아주대학교 컴퓨터공학과(박사)
1994년~현재 : 성결대학교 멀티미디어공학부 교수

관심분야 : 게임인공지능, 교육용멀티미디어콘텐츠
Phone : 031-467-8164
E-mail : choiym@sungkyul.ac.kr