DOI: 10.3745/KIPSTD.2009.16-D.2.169

주식 투자 추천 시스템을 위한 효율적인 저장 구조

하 유 민[†] · 김 상 욱[††] · 박 상 현[†††] · 임 승 환[††††]

요 약

규칙 탐사는 주어진 데이터베이스로부터 빈번하게 발생하는 패턴들을 발견하는 연산이다. 규칙 탐사 연산을 이용하여 주식 데이터베이스로부터 유용한 규칙들을 발견하고 이를 토대로 주식 투자자들에게 주식의 매매를 적절한 시점에 추천할 수 있다. 본 논문에서는 이러한 주식 투자 시스템에서 질의를 효율적으로 처리하기 위한 저장 구조에 관하여 논의한다. 먼저, 주식 투자 추천을 지원하기 위한 다섯 가지 저장 구조들을 제안하고, 각 구조들의 특징과 장단점을 비교한다. 또한, 실제 주가 데이터를 이용한 실험을 통하여 제안된 저장 구조들의 성능을 검증한다. 실험 결과에 의하면, 히스토그램을 이용한 저장 구조의 경우, 기존의 기법에 비하여 질의 처리 성능이 약 170배 개선되는 것으로 나타났다.

키워드 : 시계열 데이터, 규칙 탐사, 주식 투자 추천

Efficient Storage Structures for a Stock Investment Recommendation System

You-Min Ha[†] · Sang-Wook Kim[††] · Sanghyun Park[†††] · Seung-Hwan Lim[††††]

ABSTRACT

Rule discovery is an operation that discovers patterns frequently occurring in a given database. Rule discovery makes it possible to find useful rules from a stock database, thereby recommending buying or selling times to stock investors. In this paper, we discuss storage structures for efficient processing of queries in a system that recommends stock investments. First, we propose five storage structures for efficient recommending of stock investments. Next, we discuss their characteristics, advantages, and disadvantages. Then, we verify their performances by extensive experiments with real-life stock data. The results show that the histogram-based structure improves the query performance of the previous one up to about 170 times.

Keywords : Time-Series Data, Rule Discovery, Recommending Stock Investments

1. 서 론

시계열 데이터(time-series data)는 시간의 흐름에 따라 객체의 변화를 관측하여 얻어진 값들의 리스트이다[1, 2, 3, 4, 9]. 시계열 데이터의 대표적인 예로서 주가의 변화를 기록한 주가 데이터(stock data)를 들 수 있다. 이러한 시계열 데이터의 임의의 시점의 값은 이전까지의 값들이 보인 변화의 경향에 의해 영향을 받는다[5, 10]. 따라서 시계열 데이터로부터 규칙을 발견하고, 이를 이용하여 미래에 출현할 값을 예측할 수 있다. 이를 주식 투자에 적용하여 주가 데이터의 분석을 통해서 지수의 흐름, 주가의 변화 시점, 거래 시세 등을 예측하여 주식의 매매를 적절한 시점에 추천한다면, 주식 투자자들의 성공적인 주식 투자를 기대할 수 있을 것이다.

주식 투자자들에게 자동적으로 주식 투자를 추천하기 위해서는 주식 투자자들이 원하는 다양한 투자 조건들을 고려해야 한다. 주식 투자자들은 손실 위험이 크더라도 많은 수익을 얻을 수 있는 공격적인 투자를 원하거나, 적은 수익을 얻더라도 손실을 최소화하는 안정적인 투자를 원할 수 있다. 따라서 주식 투자 추천 시스템에서는 각 투자자들이 설정한 투자 조건을 만족하는 경우에 해당 종목을 투자자에게 자동적으로 추천해 줄 수 있어야 한다.

기존의 DBMS들을 이용하여 주식 투자 추천 시스템을 구성하는 것도 가능하지만, 이러한 방식은 질의에 대한 빠른 응답을 보장할 수 없다는 단점을 갖고 있다. 따라서 참고 문헌[6]에서는 규칙 탐사를 기반으로 하여 전술한 요건들을 만족하는 주식 투자에 특화된 시스템을 제안하였다. 이 시스템은 주가 데이터에서 빈번하게 발생하는 패턴들을 발견하고, 각 패턴을 지지하는 과거의 주가 데이터를 참조하

† 준 회 원 : 연세대학교 컴퓨터과학전공 석사
†† 종신회원 : 한양대학교 정보통신학부 교수
††† 종신회원 : 연세대학교 컴퓨터과학과 부교수(교신저자)
†††† 준 회 원 : 한양대학교 전자통신컴퓨터공학과 박사과정
논문접수 : 2008년 9월 9일
수 정 일 : 1차 2008년 12월 23일
심사완료 : 2009년 1월 11일

여 해당 빈번 패턴 발생 이후의 변화 경향을 예측한다. 각 투자자는 자신이 원하는 투자 조건을 질의의 형태로 입력할 수 있으며, 예측 결과가 이러한 조건을 만족하면 해당 투자자에게 매수/매도를 추천한다.

이 시스템에서는 하나의 질의가 실행될 때마다 디스크에 저장된 대량의 주가 데이터 내에서 질의 처리에 필요한 부분을 읽고, 이를 분석하여 추천 값을 결정한다. 따라서 이러한 처리 방식은 랜덤 디스크 액세스(random disk access)를 많이 유발 하는데, 이는 전체 시스템의 처리 성능을 떨어뜨리는 가장 큰 요인으로 작용한다. 특히, 시스템을 이용하는 많은 투자자들이 다수의 관심 종목에 대하여 질의 처리를 요청할 수 있으므로, 이를 보다 효과적으로 처리하기 위한 방안이 필요하다. 따라서 본 논문에서는 이러한 질의를 처리할 때 발생하는 디스크 액세스 수 및 CPU 계산을 줄일 수 있는 다양한 저장 구조를 제안하고, 이들의 성능을 평가한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 이전 연구에서 제안한 규칙 모델에 관하여 간략히 설명한다. 제 3장에서는 본 연구에서 제안하는 시스템이 실행되는 환경의 여러 가지 제약 조건으로 인해 발생하는 문제들을 정의한다. 제 4장에서는 이러한 문제들을 해결하는 방법들을 제안한다. 제 5장에서는 실험을 통하여 제안한 방법들의 성능을 검증한다. 끝으로, 제 6장에서 본 논문을 요약하고 결론을 내린다.

2. 주가 데이터 예측 모델

이 장에서는 참고 문헌[6]에서 제안된 규칙 모델과 질의 모델에 대하여 간략하게 설명한다.

2.1 규칙 모델

규칙 모델에서 규칙은 규칙 헤드(rule head)와 규칙 바디(rule body)로 구성된다[11,12]. 규칙 헤드는 시간에 따라 변하는 주가 데이터에서 빈번하게 발견된 패턴이며, 규칙 바디는 일정 시간 간격이 지난 후 주가 변화 양상을 가리킨다. 이를 좀 더 명확히 정의하면 다음과 같다.

$$H \xrightarrow{t} B(s,c)$$

여기서, H는 규칙 헤드이며, B는 규칙 바디이다. 이 규칙은 H에 해당되는 사건이 발생한 후, t 시간이 흐른 후에는 B에 해당되는 사건이 발생하였음을 의미한다.

주가 변화의 패턴이 규칙으로서 가치를 가지기 위해서는 과거에 발생하였던 많은 패턴들이 규칙과 부합하여야 한다. s는 아래와 같이 정의되는 지지도(support)로서 H에 해당되는 패턴 P가 과거에 발생하였던 상대 빈도를 표현한다. 즉, 규칙 헤드 H와 매치하는 실제 주가 변화 패턴이 얼마나 많이 발생하였는가를 나타내는 척도이다.

$$s(H) = \frac{H\text{와 매치되는 패턴들의 발생 수}}{H\text{와 매치되는 패턴과 길이가 동일한 모든 패턴의 발생 수}} \times 100$$

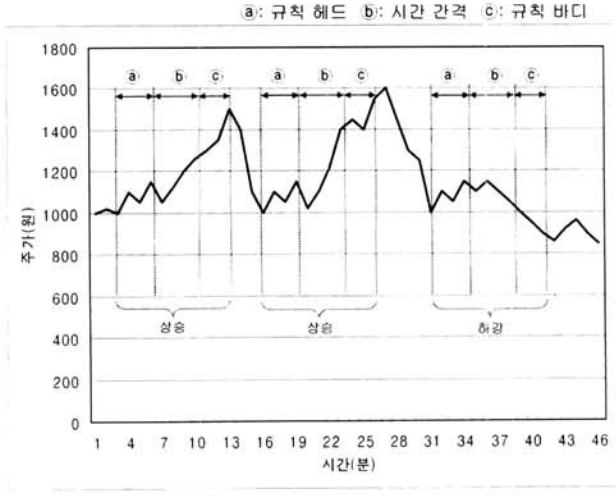
또한, 규칙으로서 가치를 가지기 위해서는 H와 매치하는 과거 패턴들이 B 구간 내에서 일정한 경향을 보여야 한다. c는 아래와 같이 정의되는 신뢰도(confidence)로서 H와 매치하는 과거 패턴들 중 얼마나 많은 수가 규칙 바디 B를 위한 조건을 만족시키는가를 표현한다.

$$c(H,B) = \frac{H\text{와 매치되며, } B\text{의 조건을 만족시키는 패턴의 발생 수}}{H\text{와 매치되는 패턴의 발생 수}} \times 100$$

본 논문에서는 과거의 주가 데이터베이스를 분석함으로써 지지도와 신뢰도가 사전에 지정된 값 이상인 규칙들을 탐사하고, 투자자의 관심 종목의 최근 주가 변화 패턴이 탐사된 어떤 규칙의 헤드 H와 매치됨이 발견되면, 해당 규칙의 바디 B를 참조하여 해당 종목에 대한 투자 유형을 투자자에게 추천하는 기법을 제안한다. 투자 유형은 '매수', '매도', '보유', '무추천' 등이 있을 수 있다. 투자 유형은 규칙 바디에 의하여 결정되며, 규칙 바디에 대한 조건은 투자자의 투자 성향에 따라 달라질 수 있다.

예를 들어, (그림 1)은 어떤 종목의 주가 변화를 나타낸 것이다. 이 종목의 주가 데이터에서 ⓐ 형태의 패턴이 3회 발생하였고, 이 패턴이 발생한 이후 일정한 시간 ⓑ 만큼 지난 후의 규칙 바디 ⓒ에서 주가가 오른 횟수가 2회, 내린 횟수가 1회였다고 하자. 이로 미루어, 다음에 이와 같은 패턴이 다시 발생한다면 주가가 오를 확률이 높으므로, 예측값 'BUY'를 추천하게 된다. 위의 그림에서는 3회 발생한 패턴을 예로 들어 설명하였으나, 실제로는 현재까지의 주가 변화 데이터 중 최소 지지도 이상의 발생 빈도를 갖는 빈번 패턴(frequent pattern)[7,8] 들을 규칙 헤드로 사용한다.

본 응용에서 H는 (그림 1)에 나타난 예제의 ⓐ 구간에서와 같은 특정 주가 변화 패턴 P의 발생과 대응되는 사건이다. 또한, B는 ⓒ 구간 내에서 발생하는 주가의 특성을 요약하는 사건이다. 예를 들어, 위의 (그림 1)의 예제에서 B는 "상승"으로 표현될 수 있다. 투자자는 자신이 추천 받기를 원하는 투자 유형과 관련하여 이 ⓒ 구간 내 주가 특성에



(그림 1) 규칙 모델 예제

관한 구체적인 조건을 명시할 수 있다. 이를 규칙 바디의 조건이라 명명한다. 이 조건은 구간 내 주가 특성이 어떠한 경향을 보일 때 이를 상승으로 간주할 것인가 하는 조건을 나타낸다. 위의 예제에서 투자자는 ⓐ 구간의 마지막 주가 대비 ⓒ 구간에서의 평균 주가 상승률이 10% 이상 되는 것을 규칙 바디의 조건으로 설정할 수 있으며, 이 경우 그림 1의 주가 변화 형태는 의미 있는 규칙으로 생성될 수 있다. 이와 같이, 이러한 규칙 바디의 조건은 투자자의 성향에 따라 달라질 수 있다.

주가 데이터는 실수값을 가지므로 빈번한 패턴이 발생할 가능성은 매우 낮다. 따라서 주가 변화율의 도메인을 다수의 구간들로 나누어, 실수값인 각 주가 변화율을 구간과 대응되는 문자로 변환한 후, 이로부터 빈번 패턴을 탐색하는 방법을 사용한다. 탐색된 빈번 패턴들은 매번 주가가 갱신될 때마다 다수의 질의들에 대하여 빠르게 검색되어야 하므로, 이들에 대한 인덱스를 구성하여 저장한다.

각 투자자는 자신의 투자 성향에 따라 주식 종목, 그 종목을 매도할 시점과 매수할 시점을 결정하는 주가 변화율의 최소/최대값, 예측할 구간의 길이 등을 정하여 질의를 작성한다. 이 값들은 해당 질의에 대한 규칙 바디의 특성을 결정한다.

2.2 질의 모델

[정의 1] 질의 Q

투자 추천을 요구하기 위하여 투자자가 정의하는 질의 Q의 형태는 다음과 같다.

$$Q = (I, T, BL, [\alpha, \beta], mC)$$

각각의 변수는 다음과 같은 의미를 가진다.

- I : 예측하려는 종목.
- T : 규칙 헤드와 규칙 바디 사이의 시간 간격.
- BL : 규칙 바디의 길이.
- [α, β] : 보유 변동률. (α와 β의 의미는 정의 2에서 설명)
- mC : 최소 신뢰도. (mC > 0.5 이며, 그 의미는 정의 2에서 설명) □

종목 I의 주가가 변화할 때마다 Q를 수행하며, 현재까지의 주가 변화가 빈번 패턴과 매치되었을 때 해당 종목에 대한 추천값을 반환한다.

질의 Q의 실행 결과 F(Q)는 다음과 같은 값을 가진다.

[정의 2] 질의 Q의 실행 결과 F(Q)

빈번 패턴이 발생한 각 사례(case)에 대하여, 규칙 헤드의 마지막 주가에 대한 규칙 바디의 주가 평균값의 증가 비율을 r이라 하면, 질의 Q=(I, T, BL, [α, β], mC) 의 실행 결과 F(Q)는 다음과 같이 정의된다.

F(Q) = X, X ∈ {SELL, HOLD, BUY, NONE} 이며, α는 'HOLD' 선택 하한선, β는 'HOLD' 선택 상한선이다. 이때, X의 결과로 올 수 있는 4개의 값들을 추천값이라 하며, 각각 다음과 같은 경우에 질의 Q의 추천값으로 결정된다.

- SELL: 종목 I의 모든 빈번 패턴에 대하여, $r \le \alpha$ 인 경우의 비율이 mC 이상일 때.
- HOLD: 종목 I의 모든 빈번 패턴에 대하여, $\alpha < r < \beta$ 인 경우의 비율이 mC 이상일 때.
- BUY: 종목 I의 모든 빈번 패턴에 대하여, $r \ge \beta$ 인 경우의 비율이 mC 이상일 때.
- NONE: 종목 I의 모든 빈번 패턴에 대하여, SELL, HOLD, BUY 어느 결과의 비율도 mC를 넘지 못했을 때.

이때, SELL, HOLD, BUY 3가지의 추천값 중 2개 이상의 추천값이 동시에 결정되는 경우를 방지하기 위하여, 정의 1에서와 같이 최소 신뢰도 mC는 0.5보다 큰 값을 입력하도록 제한을 둔다. 따라서 한 시점에서 F(Q)는 유일한 값을 추천한다. F(Q)의 값이 X인 경우, F(Q)=X 로 표시하고, '질의 Q를 실행한 결과, 추천값은 X이다' 라고 읽는다. □

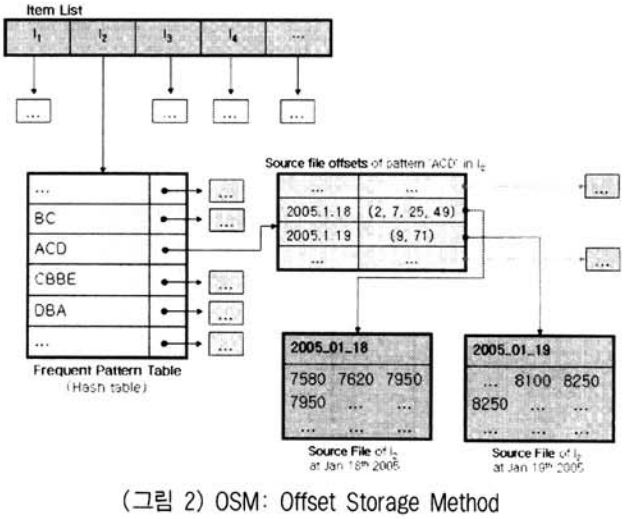
질의 Q를 결정하는 모든 변수값들은 투자자가 원하는 대로 등록할 수 있으므로, 이 질의 모델은 투자자들의 다양한 성향을 유연하게 수용할 수 있다는 장점을 가진다. 실험 결과, 이 주식 투자 추천 시스템은 70% 이상의 예측 정확도를 가지는 것으로 나타났다.

3. 주가 저장 구조

본 장에서는 제 2장에서 설명한 규칙 모델과 질의 모델을 실제로 구현한 시스템에서 이러한 모델을 대상으로 효율적으로 질의 처리를 하기 위한 저장 구조들을 제안하고, 장단점을 논의한다.

3.1 OSM: Offset Storage Method

현재 변화되는 주가가 빈번 패턴과 매치되면, 과거 주가 데이터에서 현재 발생한 빈번 패턴이 발견된 이후의 주가 변화를 분석하여 추천값을 결정하게 된다. 이러한 데이터 검색



(그림 2) OSM: Offset Storage Method

과정을 그림으로 나타내면 (그림 2)와 같다.

각 주식 종목에 대하여 과거에 각 빈번 패턴이 나타난 모든 위치들을 저장한 인덱스 파일과 수집된 원본 주가 데이터가 모두 디스크에 저장되어 있다. 만약, 빈번 패턴 'ACD'가 종목 I2 에서 발생하였다면, 종목 I2의 인덱스 파일을 읽어 빈번 패턴 'ACD'가 발생한 위치들을 모두 찾는다. 그 다음, 원본 주가 데이터에서 패턴 'ACD'가 발생한 모든 파일을 각각 읽어 각 발생에 대한 추천값을 계산하고, 이를 기반으로 최종 추천값을 결정한다. 이러한 저장 구조의 이름을 OSM(offset storage method)이라 명명한다.

OSM는 참고문헌[6]에서 기본적으로 사용하고 있는 구조로서, 간단하고 원본 주가 데이터에 대한 인덱스 파일만 작성하여 저장하면 된다. 따라서 원본 주가 데이터 이외에 별도로 저장되는 데이터의 크기가 작다는 장점이 있다. 그러나 주가 예측을 위하여 특정 패턴이 발생한 모든 주가 데이터를 읽어야 한다. 이때, 빈번 패턴은 전체 데이터베이스 내에서 분산되어 있으므로 질의 처리 과정에서 랜덤 디스크 액세스가 대단히 많이 발생하는 단점이 있다.

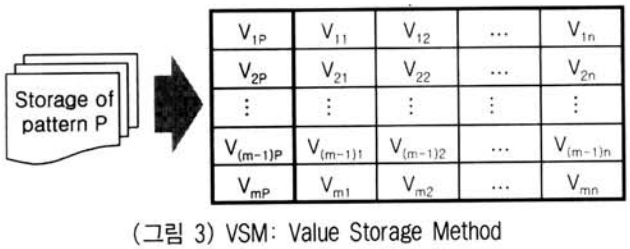
3.2 VSM: Value Storage Method

제 3.1절의 끝에서 언급한 단점을 해결하기 위한 방법은 함께 액세스 될 확률이 높은 데이터를 디스크 내에 밀집하여 저장하는 것이다. 이를 위하여 모든 빈번 패턴들에 대하여 각 패턴이 발생한 이후에 나타나는 일정 길이만큼의 주가들을 추출하여 함께 저장해 둔다. 이 결과, 질의 처리 시 이렇게 함께 저장된 데이터를 순차적으로 디스크로부터 액세스하게 되므로 랜덤 액세스를 하는 OSM과 비교하여 성능 향상을 기대할 수 있다. (그림 3)은 이 방법을 나타낸 것이다.

패턴 P가 발생한 이후의 주가들이 (그림 3)과 같이 저장되어 있다고 하자. m은 패턴 P가 발생한 회수, n은 T와 BL 값의 합의 최대값이다. 이때, V_{xy} 는 패턴 P가 x번째 발생한 위치에서 y번째 떨어진 주가를 의미한다. V_{mP}는 패턴 P의 마지막 주가를 가리킨다. 이와 같은 저장 구조를 사용하면, 패턴 P의 x번째 발생에 대한 주가 상승률 r_x를 다음과 같이 계산할 수 있다.

$$r_x = \frac{\dfrac{\sum_{i=t+1}^{t+bl+1} V_{xi}}{bl} - V_{xP}}{V_{xP}}$$

이러한 저장 구조를 VSM(value storage method)이라 명명한다. 어떤 빈번 패턴 P의 발생 회수가 m번이었다고 가정하면, (그림 3)과 같은 저장 구조를 사용하여 질의를 처리할 수 있다. 이 구조는 질의를 처리할 때 랜덤 디스크 액세스가 없다는 장점을 가진다.



(그림 3) VSM: Value Storage Method

3.3 ADSM: Accumulated Difference Storage Method

제 2장에서 설명한 질의 처리 과정을 살펴보면, 질의 처리에 사용되는 빈번 패턴이 과거의 주가 데이터에서 발생했던 사례마다 규칙 바디의 평균 주가를 하나하나 계산하고 있다. 주가 데이터를 저장할 때에 주가를 직접 저장하지 않고 미리 변화량의 누적값을 계산하여 저장해 두면, 질의 처리 시 규칙 바디의 평균 주가 계산 과정을 줄일 수 있다. (그림 4)에서 이와 같은 방법을 나타낸 것이다.

패턴 P가 발생한 이후의 주가 부분합이 위 그림과 같이 저장되어 있다고 하자. m은 패턴 P가 발생한 회수, n은 T와 BL 값의 합의 최대값이다. 이때, A_{xy}는 V_{x1}에서 V_{xy}까지의 변화량의 누적 값이며, 다음과 같이 계산된다.

$$A_{xy} = \sum_{i=1}^{y} V_{xi}$$

이때, 패턴 P의 x번째 발생에 대한 주가 상승률 r_x는 다음과 같이 계산할 수 있다.

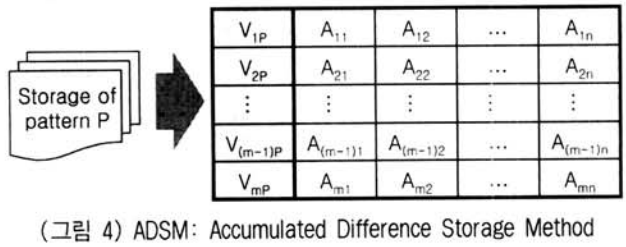
$$r_x = \frac{\left(A_{x(t+bl+1)} - A_{x(t+1)}\right)/bl - V_{xP}}{V_{xP}}$$

이러한 저장 구조를 VSM(value storage method)이라 명명한다. VSM을 사용할 경우, 패턴 P가 발생한 뒤로 t+1번째 주가로부터 t+bl+1 번째까지의 주가의 합을 빠르게 계산할 수 있다. 따라서 전체 주가 상승률의 계산량이 O(m×bl)에서 O(x) 로 감소하며, 저장 구조에서 세로로 3열의 값만 사용하므로 디스크 액세스 수도 감소하는 장점이 있다.

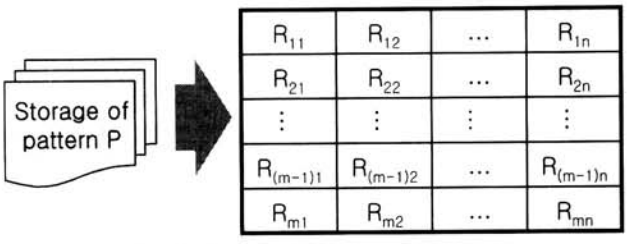
3.4 RSM: Ratio Storage Method

만약 질의의 t값이 고정되어 있다면, 미리 변화율을 계산하여 직접 저장해 둘 수 있다. 이 경우, 질의 처리에 소요되는 계산량을 대폭 감소시킬 수 있다. (그림 5)는 이러한 방법을 나타낸 것이다.

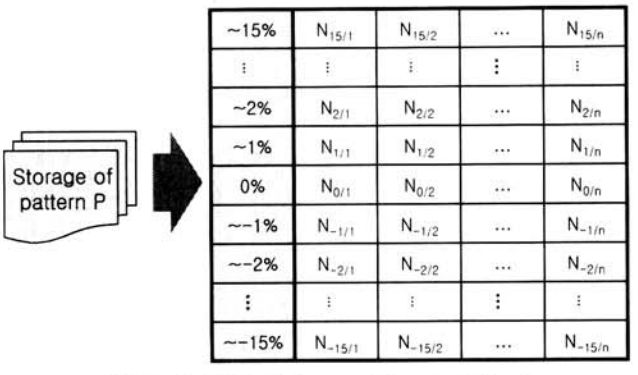
패턴 P가 발생한 이후의 주가 변화율이 위 그림과 같이 저장되어 있다고 하자. m은 패턴 P가 발생한 회수, n은 T



(그림 4) ADSM: Accumulated Difference Storage Method



(그림 5) RSM: Ratio Storage Method



(그림 6) HSM: Histogram Storage Method

와 BL 값의 합의 최대값이다. V_{mP}는 패턴 P의 마지막 주가를 가리킨다. 이때, R_{xy}는 다음과 같이 계산된다.

$$R_{xy} = \frac{\sum_{i=1}^{bl} V_{xi} \Big/ bl - V_{xP}}{V_{xP}}$$

t=0일 때, y=bl 인 모든 R_{xy}를 찾아 각각의 주가 상승률을 얻을 수 있다. 이러한 저장 구조를 RSM(ratio storage method)이라 명명한다. 미리 계산된 주가 변화율이 저장되어 있으므로 CPU 계산량이 대폭 감소하며, 저장 구조에서 세로로 1열의 값만 필요하기 때문에 디스크 액세스 수를 더욱 줄일 수 있다. 그러나 고정된 t값에 대해서만 RSM을 작성할 수 있기 때문에, 다양한 t값을 가지는 질의들을 모두 처리하려면 자료 구조의 크기가 커져야 한다는 것이 단점이다.

3.5 HSM: Histogram Storage Method

제 3.4절에서 언급한 바와 같이, 시간 간격 t와 규칙 바디의 길이 bl 의 값이 결정된다면, 각 빈번 패턴이 발생했을 때의 주가 변화율을 직접 계산하는 것이 가능하다. 제 2장의 추천 과정에서는 이렇게 구한 변화율 R_x 값을 사용자 질의에 포함된 인수 중 하나인 [α, β] 값과 비교하여 추천값을 결정한다. 따라서 α와 β 값을 경계로 하여 주가 변화율이 α보다 작은 경우, β보다 큰 경우, α와 β 사이에 있는 경우의 비율을 각각 얻을 수 있다면, 추천값을 결정하는데 충분한 정보가 된다. 따라서 제 3.4절의 RSM과 같이 변화율을 직접 계산한 다음, 이 값들을 사용하여 히스토그램을 구성하면, 규칙의 처리가 가능하다. 이러한 방법을 나타내면 (그림 6)과 같다.

(a) 0.001일 때

0.001	0.002
20	10

(b) 0.002일 때
Scaling을 하면 15,15로 나뉨

0.001	0.002
30	
15	15

(그림 7) 히스토그램의 범위에 따른 오차

패턴 P가 발생한 이후의 주가 변화율의 발생 회수가 위 그림과 같은 히스토그램으로 저장되어 있다고 하자. 히스토그램의 구간은 주가 변화율을 나눈 것이며, 이 예에서는 -15%에서 15%까지의 구간을 1% 단위로 나누었다. 이때, 히스토그램을 나누는 구간은 미리 결정된 값을 사용한다. n은 T와 BL 값의 합의 최대값이다. 이때, $N_{r/i}$는 패턴 P가 발생한 이후의 주가 변화율이 r-1% 초과 r% 이하인 경우의 회수를 가리킨다.

히스토그램을 사용한 위와 같은 저장 구조를 HSM(histogram storage method)이라 명명한다. HSM을 사용할 경우, 원본 데이터가 아무리 증가하더라도 히스토그램의 크기는 증가하지 않는다. 따라서 계속 누적되는 주가 데이터의 증가에 따른 공간 제약이 해결되는 장점이 있다. 또한, 전체 히스토그램의 크기가 고정되므로, 계산량은 O(n)에서 O(1)로 감소한다. 그러나 HSM은 RSM과 마찬가지로 고정된 t 값에 대해서만 데이터 구조를 작성할 수 있다는 단점이 있다. 하지만 다른 데이터 구조에 비해 데이터의 크기가 훨씬 작으므로, 질의가 예상되는 다양한 t값에 대하여 각각 히스토그램을 작성하여 처리함으로써 이러한 문제를 해결할 수 있다. 그러나 데이터의 가용 범위를 구간들로 나누고, 구간별 발생 회수만 저장하므로 최소 분할 구간 크기 이하의 질의가 입력되면 오차가 발생한다. (그림 7)은 그 예를 나타낸 것이다.

히스토그램의 분할 구간 크기를 0.001(0.1%) 단위로 잡았을 때, 어떤 패턴의 주가 변화율 발생 횟수가 그림 (a)와 같았다고 하자. 분할 구간 크기를 0.002(0.2%) 단위로 바꾸어 잡을 경우, 같은 데이터가 [0.002 → 30] 으로 표현된다. 질의의 보유 변동률에 0.001(0.1%) 값이 포함되어 있다면, 이 히스토그램으로는 질의를 정확히 처리할 수 없다.

HSM은 히스토그램을 이용하여 데이터를 손실 압축하므로 이와 같이 정확성의 문제가 발생한다. 이에 대한 해결책은 1) 사용자가 질의를 입력할 때 최소 분할 구간 크기 단위로만 보유 변동률을 입력할 수 있도록 하거나, 2) 비례식을 사용하여 주가 변화율이 한 구간 안에 균일하게 분포되어 있다고 가정하고 계산하는 방법이 있다.

(그림 7)(b)는 비례식을 사용하는 방법의 예를 나타낸 것이다. 원본 데이터에서 주가 변화율이 0.001인 경우가 20회, 0.002인 경우가 10회 발생하였을 때, 히스토그램의 분할 구간 크기가 0.002였다면 이 데이터는 [0.002 → 30] 으로 표현된다. 질의들 중 보유 변동률이 0.001인 것이 있을 경우, 주가 변화율이 균일하게 분포되어 있다고 가정하면 (그림 7)(b)와 같이 [0.001→15], [0.002→15] 로 계산할 수 있다. 원본 데이터와 비교하면 오차가 발생하나, 분할 구간의 크기가 질의의 보유 변동률 단위보다 클 경우에도 질의 처리가

가능하다는 장점을 가진다. 본 연구에서는 비례식을 사용하여 주가 변화율을 계산하고, 이를 이용하여 오차를 줄이는 방법을 채택한다.

4. 성능 평가

본 장에서는 제 3장에서 제안한 저장 구조들을 대상으로 질의 처리 성능을 평가한다. 제 4.1절에서는 성능 평가를 위한 실험 환경을 설명하고, 제 4.2절에서는 실험 결과를 분석한다.

4.1 실험 환경

본 연구에서는 성능 분석을 위하여 3개월 분량의 실제 한국의 주가 데이터베이스 KOSPI[13]를 사용하였다. 빈번 패턴이 발생한 이후의 최대 주가 길이는 20으로 하였으며, 다음과 같은 조건으로 905개 종목에 대하여 종목당 각각 108개의 질의들을 생성하였다.

- α: -0.003, -0.002, -0.001 중 한 값
- β: 0.001, 0.002, 0.003 중 한 값
- T: 0 으로 고정
- bodyLen: 1, 3, 5 중 한 값
- 최소신뢰도: 50%, 60%, 70%, 80% 중 한 값

실험을 위한 환경으로는 2.4GHz Pentium IV 2.4GHz 프로세서에 1GB 메모리를 장착한 PC와 MS 윈도우 2003 서버 운영체제를 사용하였다.

본 실험에서 성능 평가의 대상으로 선정한 저장 구조들은 원본 데이터를 가공하지 않고 저장하는 OSM, 함께 요청될 데이터들을 밀집하여 저장하는 VSM, 주가 변화량의 누적값을 저장하는 ADSM, 예측 시점을 고정하여 주가의 변화량을 저장하는 RSM, 히스토그램을 사용하는 HSM의 다섯 가지이다.

4.2 실험 결과

본 논문에서는 세 가지 종류의 실험을 수행하였다. 실험 1에서는 각 저장 구조들이 필요로 하는 디스크 공간의 크기를 비교하였다. 실험 2에서는 각 저장 구조들을 이용하여 질의처리를 수행하였을 때에 걸린 시간을 비교하였다. 실험 3에서는 원본 데이터의 크기를 변경하며 수행 시간을 비교함으로써 각 저장구조들의 확장성을 검증하였다.

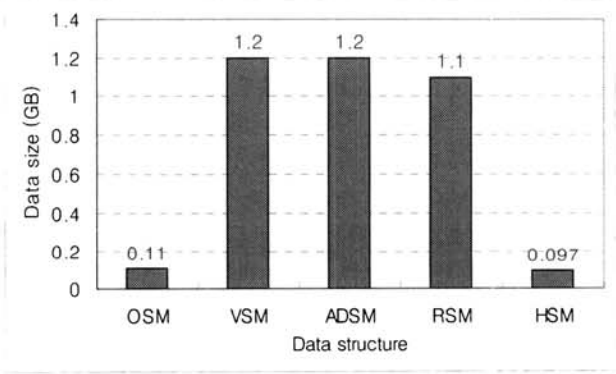
(그림 8)은 실험 1의 결과를 보인 것이다. 가로축은 각 저장 구조들을 나타내고, 세로축은 각 저장 구조들이 필요로 하는 디스크 공간의 크기를 기가 바이트 단위로 나타낸 것이다.

실험 결과, 각 저장 구조들이 사용한 디스크 공간은 OSM은 0.11, VSM은 1.2, ADSM은 1.2, RSM은 1.1기가 바이트로 나타났다. VSM, ADSM, RSM은 원본 데이터를 가공하지 않고 저장한 OSM에 비해 약 12배 많은 디스크 공간을 사용하였다. 반면에 HSM은 OSM 보다도 오히려 적은 디스크 공간을 사용하였다. 따라서 다양한 예측시점에 대하여 각각 HSM 저장 구조를 생성함으로써, HSM의 단점인 특정 예측 시점만을 지원한다는 문제점을 극복할 수 있다.

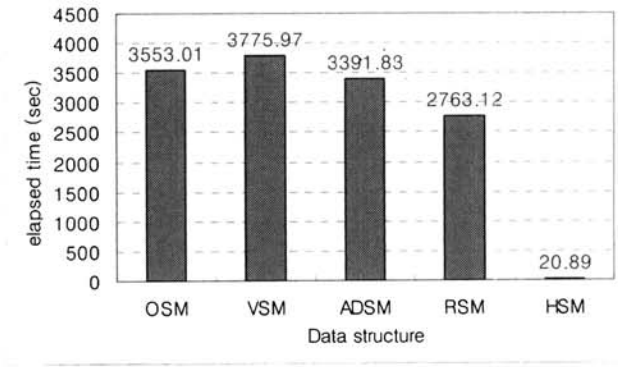
(그림 9)는 실험 2의 결과를 보인 것이다. 가로축은 각 저장 구조들을 나타내고, 세로축은 각 저장 구조들의 질의 처리에 소요된 시간을 초 단위로 나타낸 것이다.

실험 결과, 질의 처리에 걸린 시간은 OSM은 3553.01, VSM은 3775.97, ADSM은 3391.83, RSM은 2763.12, HSM은 20.89초로 나타났다. VSM의 경우, 원본 데이터를 가공하지 않고 저장하는 OSM에 비하여 오히려 수행 시간이 약 1.6배 증가하였다. 이는 주가 데이터를 순차적으로 저장한 경우, 서로 다른 빈번 패턴들에 대하여 다른 파일을 읽어야 하므로 디스크 캐시가 적중할 확률이 그만큼 낮아지기 때문이다. ADSM의 경우, OSM에 비하여 약 1.05배, VSM에 비하여 약 1.11배의 수행시간 감소를 보였다. RSM의 경우, OSM에 비하여 약 1.29배, VSM에 비하여 약 1.37배, ADSM에 비하여 약 1.23배의 수행시간 감소를 보였다. HSM의 경우, OSM에 비하여 약 170.08배, VSM에 비하여 약 180.75배, ADSM에 비하여 약 162.37배, RSM에 비하여 약 132.27배의 수행시간 감소를 보였다.

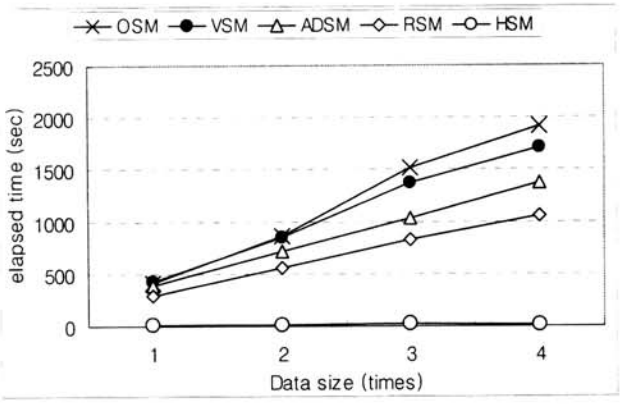
실험 3에서는 원본 데이터의 크기 증가에 따른 각 저장 구조들의 수행 시간을 비교하기 위하여 원본 데이터를 2배, 3배, 4배로 복사함으로써 크기를 증가시킨 데이터를 사용하였다. (그림 10)은 실험 3의 결과를 보인 것이다. 가로축은 사용한 데이터의 크기를 원본 데이터의 크기에 대비하여 배



(그림 8) 저장 구조에 따른 저장 공간의 크기



(그림 9) 저장 구조에 따른 전체 질의 처리 시간



(그림 10) 데이터 크기 증가에 따른 처리 시간의 변화

율로서 나타낸 것이다. 세로축은 각 데이터 크기에 대하여 각 저장 구조들이 질의 처리에 소요된 시간을 초 단위로 나타낸 것이다.

실험 결과, 원본 데이터의 크기가 증가함에 따라 HSM을 제외한 나머지 네 개의 저장 구조를 이용한 질의 처리 시간은 선형적으로 증가하였다. 이것은 원본 데이터의 크기가 증가할수록 저장 구조를 위한 저장 공간의 크기가 함께 증가하게 되어 질의 처리 시간의 증가를 초래하기 때문이다. 반면, 히스토그램을 저장하는 HSM은 원본 데이터 크기에 큰 영향을 받지 않고 일정한 질의 처리 시간을 보였다. 이것은 원본 데이터의 크기가 증가하더라도 히스토그램의 크기는 증가하지 않기 때문이다.

실험 결과를 종합하면, 히스토그램을 사용하는 HSM이 디스크 공간, 질의 처리 시간, 확장성 측면에서 가장 우수한 성능을 보이는 것으로 나타났다.

5. 결 론

본 연구에서는 주가를 예측하는 규칙 모델을 사용하는 실시간 규칙 추천 시스템에서 각 질의를 빠르게 처리하기 위하여, 다섯 가지 저장 구조들을 제안하였다. 제안된 데이터 저장 구조들은 선행 연구의 질의 처리 방법과 동일한 처리 결과를 얻으면서도 디스크 액세스 수를 감소시켜 질의 처리 시간을 줄이는 것을 목표로 고안되었으며, 정확도, 저장 공간, 처리 성능 측면에서 각각 다른 특징을 가진다. 실험을 통한 성능 평가를 수행함으로써 제안한 저장 구조들을 서로 비교 분석하였다. 실험 결과에 의하면, 히스토그램을 사용하는 HSM이 가장 좋은 질의 처리 성능을 보였으며, 원본 데이터의 크기가 증가하더라도 질의 처리 성능이 크게 변화하지 않는 것으로 나타났다.

감사의 글

본 연구는 지식경제부 및 정보통신연구진흥원의 대학IT연구센터지원사업(IITA-2008-C1090-0801-0040)과 한국과학재단의 2008 특정기초연구사업의 지원(No. R01-2008-000-20872-0)을 받았습니다.

참 고 문 헌

[1] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient Similarity Search in Sequence Databases," In Proc. Int'l. Conf. on Foundations of Data Organization and Algorithms, FODO, pp. 69-84, Oct., 1993.

[2] S. W. Kim, S. H. Park, and W. W. Chu, "An Index-Based Approach for Similarity Search Supporting Time Warping in Large Sequence Databases," In Proc. Int'l. Conf. on Data Engineering, IEEE, pp.607-614, 2001.

[3] W. K. Loh, S. W. Kim, and K. Y. Whang, "A Subsequence Matching Algorithm that Supports Normalization Transform in Time-Series Databases," Data Mining and Knowledge Discovery Journal, Vol.9, No.1, pp.5-28, July, 2004.

[4] S. H. Park et al., "Efficient Searches for Similar Subsequences of Difference Lengths in Sequence Databases," In Proc. Int'l. Conf. on Data Engineering, IEEE ICDE, pp.23-32, 2000.

[5] P. Bloomfield, Fourier Analysis of Time Series, Wiley, 2000.

[6] You-min Ha, Sanhyun Park, Sang-Wook Kim, Jung-Im Won, and Jee-Hee Yoon, "Rule Discovery and Matching in Stock Databases," 32nd Annual IEEE International Computer Software and Applications Conference(COMPSAC 2008), pp.192-198, 2008.

[7] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," In Proc. Int'l. Conf. on Very Large Data Bases, VLDB, pp.487-499, 1994.

[8] R. Agrawal and R. Srikant, "Mining Sequential Patterns," In Proc. Int'l. Conf. on Data Engineering, IEEE ICDE, pp.3-14, 1995.

[9] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast Subsequence Matching in Time-series Databases," In Proc. Int'l. Conf. on Management of Data, ACM SIGMOD, pp.419-429, May, 1994.

[10] T. Anderson, "The Statistical Analysis of Time Series," Wiley, 1971.

[11] G. Das, K.-I. Lin, H. Mannila, Gopal Renganathan, and Padhraic Smyth, "Rule Discovery from Time Series," In Proc. Int'l. Conf. on Knowledge Discovery and Datamining, pp.16-22, 1998.

[12] S. Park and W. W. Chu, "Discovering and Matching Elastic Rules From Sequence Databases," in Fundamenta Informaticae, Vol.47, No.1-2, pp.75-90, Aug-Sept, 2001.

[13] Koscom Data Mall, http://datamall.koscom.co.kr, 2005.

하 유 민

e-mail : ymha@cs.yonsei.ac.kr
2004년 연세대학교 컴퓨터과학과(학사)
2007년 연세대학교 컴퓨터과학전공(석사)
관심분야 : 데이터마이닝, 데이터베이스, 임베디드 시스템

김 상 욱

e-mail : wook@hanyang.ac.kr
1989년 2월 서울대학교 컴퓨터공학과(학사)
1991년 2월 한국과학기술원 전산학과(석사)
1994년 2월 한국과학기술원 전산학사(박사)
1991년 7월~1991년 8월 미국 Stanford University, Computer Science Department, 방문연구원
1994년 3월~1995년 2월 KAIST 정보전자연구소 전문연구원
1999년 8월~2000년 8월 미국 IBM T.J. Watson Research Center, Post-Doc.
1995년 3월~2003년 2월 강원대학교 정보통신공학과 부교수
2003년 3월~현 재 한양대학교 정보통신학부 교수
2009년 1월~현 재 미국 Carnegie Mellon University, Visiting Scholar
관심분야 : 데이터베이스 시스템, 저장 시스템, 트랜잭션 관리, 데이터 마이닝, 멀티미디어 정보 검색, 공간 데이터베이스/GIS, 주기억장치 데이터베이스, 이동 객체 데이터베이스/텔레매틱스, 사회 연결망 분석, 웹 데이터 분석

박 상 현

e-mail : sanghyun@cs.yonsei.ac.kr
1989년 서울대학교 컴퓨터공학과(학사)
1991년 서울대학교 컴퓨터공학과(공학석사)
2001년 University of California, Los Angeles (UCLA) 대학원 컴퓨터과학과(공학박사)
2002년~2003년 포항공과대학교 컴퓨터공학과 조교수
2003년~2006년 연세대학교 컴퓨터과학과 조교수
2006년~현 재 연세대학교 컴퓨터과학과 부교수
관심분야 : 데이터베이스, 데이터 마이닝, 바이오인포매틱스, 임베디드 시스템

임 승 환

e-mail : shlim@agape.hanyang.ac.kr
2003년 한양대학교 전자컴퓨터공학부(학사)
2005년 한양대학교 정보통신대학원(공학석사)
2005년~현 재 한양대학교 전자통신컴퓨터공학과 박사과정
관심분야 : 데이터베이스, 데이터 마이닝, 사회 연결망 분석