시계열 데이터로부터 경향성을 이용한 순차패턴의 탐색

오용생*, 남도원**, 장지숙**, 이동하***, 이전영**

* 포항공과대학교 정보통신학과

** 포항공과대학교 전자컴퓨터공학부

*** 포항공과대학교 정보통신연구소

(albatros, irene, jihan,dongha, jeon)@white.postech.ac.kr

Rule discovery for sequential patterns of trend from Time-Series

Young-Saeng Oh*, Do-Won Nam**, Ji-suk Jang **, Dong-Ha Lee***, Jeon-Young Lee**

* Dept. of Computer and Communications Engineering, POSTECH

** Division of Electrical and Computer Engineering, POSTECH

*** POSTECH Information Research Laboratories, POSTECH

요 약(Abstract)

데이터마이닝 분야에서 시계열 데이터(time-series data)내에서 숨어 있는 순차패턴의 발견은 상품(Items)이나 어떤 사건(Event)과 같이 데이터의 특징이 명확한 대상에 대한 연구는 많이 되어왔으나 수치 값을 가지는 시계열 데이터에서 이들 내부에 숨어있는 패턴을 발견하는 것은 최근에 관심을 가지게 되었다. 우리는 시계열 데이터를 시간적 변화에 따라 값의 변화 경향(Trend)이 같은 데이터 그룹을 패턴 요소인 벡터(Vector)로 표현하여 이들을 이용해서 흥미로운 패턴들을 발견한다. 이와 같은 벡터적인 표현으로 우리는 벡터들 간의 포함관계를 적용해 모든 가능한 형태의 패턴 발견을 목적으로 한다. 또한 경향성을 가진 패턴 요소를 사건(Event)과 같이 취급함으로써 다양한 종류의 시계열 데이터가 동시에 발생될 때 이들 상호간에 연관된 시간적 패턴을 찾을 수 있다. 따라서 이 연구에서 제안하는 경향성을 기초로 한 순차패턴의 탐색은 기업내부의 판매실적의 변화 패턴이나, 고객의 구매 행동분석에 적용이 가능하리라 여겨진다.

Key word : 시계열 데이터(Time Series), 순차패턴(Sequential Pattern), 경향성(Trend),CRM

1. 서론

데이터마이닝 분야에서 시계열 데이터(time-series data)내의 데이터들이 가지는 순차패턴을 발견하는 연구는 상품판매점에서 고객의 구매 패턴을 발견하는 것부터 증권에서 유사한 종목의 주식가격의 시간적 패턴 분석, 엔지니어링 분야에서 설비의 고장분석에 이르기까지 다양한 적용범위를 가지고 있다. 예를 들어 서점에서 일정한 기간동안 판매된 정보를 조사해 보니 "삼국지를 구입한 사람은 몇주일 안에 수호지를 구입한다"와 같은 순차패턴 정보를 발견하는 것으로 시간적 순서를 가진 사건이나 트랜젝션(transaction)의 발생을 분석하여 이들이 자주 발생되는 사건이나 트랜젝션이 어떤 순서로 발생되는 것인지를 찾는 것이다. 그러나 상품(Items)이나 어떤 사건(Event)과 같이 데이터의 특징이 명확한 대상들 간에 발생되는 순차패턴의 발견은 많이 연구되어왔으나 엔지니어링 데이터와 같이 패턴을 구성하는 데이터가 다양한 수치 값을 가지는 분야에서의 순차패턴 발견은 최근[4]에서야 관심대상이 되어왔다. 수치 값과 같은 데이터로 이루어진 데이터 집합에서 어떤 패턴을 발견하기 위해서는 먼저 데이터의 특

징을 발견하여야 한다. 즉 기존에 상품이나 사건의 경우는 타임시리즈에 존재하는 데이터 집합이 $I=\{i_1,i_2,...,i_m\}$과 같이 m개의 문자로 표현될 수 있는 데이터 집합이나, 수치값을 가진 데이터의 경우 $I=\{-\infty,0,+\infty\}$의 값의 범위를 가지므로 단순히 이들 값의 순서에 따른 패턴을 발견하는 것은 불가능하고 무의미 한 것이다. 이러한 분야에서의 순차패턴의 발견은 먼저 데이터 값 자체보다는 데이터가 가지는 시간적인 변화 특성을 기준으로 패턴을 발견하여야 한다. 예를 들어 어떤 상품의 할부금액의 변화 패턴을 분석한 결과 "2개월간 이정한 수준 이상의 할부금액 감소가 발생되면 향후 3개월 내에 할부금액의 급격한 증가가 나타난다"와 같은 패턴을 발견하기 위해서는 각 데이터가 발생되는 시점의 값이 중요하기 보다는 이들 값이 시간적으로 연속해서 만들어내는 특징에 의해 발생되는 패턴을 찾는 것이 보다 의미가 있을 것이다. 우리의 연구는 위와 같이 시계열 데이터 내의 데이터가 값의 범위가 정해지지 않은 데이터 집합일 경우 이들 값이 만들어 내는 특징을 발견하고 또한 이들 내부에 숨어있는 모든 순차패턴을 발견하는 것을 목적으로 한다.

본 논문의 구성은 제 2장에서 기존에 수행된 시계열 데이터 내에서 순차패턴 발견에 관한 연구를 알아보고 제 3장에서 시계열 데이터를 경향성 형태로 변환하여 패턴 발견을 위한 데이터 정의 방법과 제 4장에서 정의된 데이터에서 패턴을 발견하는 방법을 살펴본 후 제 5장에서 결론 및 향후 연구방향을 제시한다.

2. 관련된 연구

수치값을 가진 데이터와 관련된 시계열 데이터에서 순차패턴에 관한 연구는 최근에 발표[4] 되었는데 이 방법은 기존의 연구 중 에피소드 방법[3]을 일반적인 수치 값을 가지는 타임시리즈에 적용하는 시도를 하였다. 이들은 먼저 시계열 데이터를 사용자가 지정한 윈도우 w 크기를 가지는 데이터 집합으로 구분하여 시계열 데이터베이스 내의 데이터 특징을 구분하였다. 즉 시계열 데이터 집합을 s 라 할 경우 $s=\{x_1,x_2,...,x_n\}$ 이고 이들을 각각 크기 w인 부분적 시계열로 나누므로 부분적 시계열 데이터를 s_i라 할 경우 $s_i=\{x_i,...,x_{i+w+1}\}$가 된다. 따라서 최초의 시계열 데이터 집합은 윈도우 w로 구성된 $s=\{s_1,s_2,...,s_{n-w+1}\}$으로 변환된다. 이들 각각의 부분적 시계열 데이터는 패턴탐색 시 보다 높은 지지도(Support)를 얻기 위해 먼저 클러스터링을 한다. 예를 들어 $k-means$ 방법을 사용한다면 위의 데이터 집합을 k개의 클러스터로 구분하고 이들 각각의 클러스터에 알파벳 이름을 붙인다. 따라서 클러스터의 개수 k 만큼의 알파벳들로 구성된 시계열 데이터가 생성되고 이들을 이용하여 기존의 에피소드 방법을 이용하여 패턴을 발견한다.

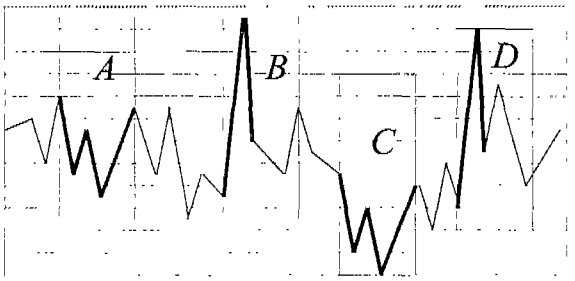
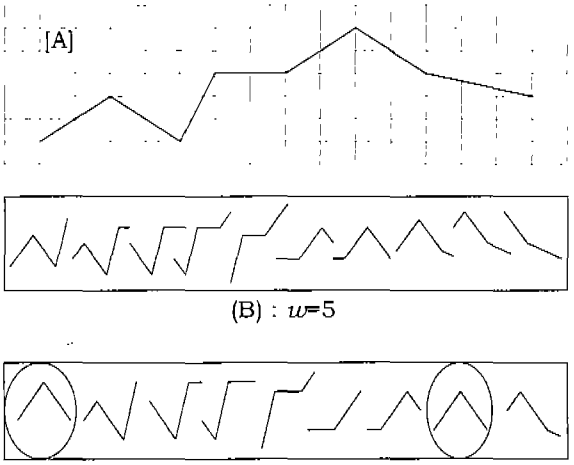


그림 [1]

위의 그림[1]은 시계열데이타에서 윈도우(window)를 이동하면서 패턴을 찾는 것을 보여주는 것으로 전체 시계열 데이터를 윈도우를 이동하면서 윈도우 크기의 이미지(image) 패턴으로 보고 같은 이미지가 있는 지를 계속 찾는 방식을 주로 사용한다. 위의 그림에서는 패턴 A와 C가 같은 형태로 자주 발견된는 것을 알 수 있다. 그러나 위와 같이 윈도우를 이동하면서 패턴을 찾는 방법에는 몇가지 문제점이 있다.



(B) : w=5

(C) : w=4

그림[2]

가장 자주 발생되는 문제점으로는 사용자가 패턴 발견

을 위해 정하는 윈도우의 크기 w에 따라 발견되지 않는 패턴이 존재한다는 것이다.

보다 간단한 예를 들기 위해 시계열 데이터가 그림[2]-A 와 같이 있을 경우 윈도우의 크기를 4 와 5 로 할 경우 이때 나타나는 패턴의 종류는 그림[2]-B 와 그림[2]-C 와 같다. 그림에서 보아서 알 수 있듯이 윈도우크기를 5 로 할 경우는 반복되는 패턴을 발견할 수 없다. 그러나 윈도우 크기를 4 로 할 경우는 상승 후 하락하는 패턴이 발견된다. 이와 같이 윈도우크기에 많은 영향을 받으므로 사용자가 다양한 값을 적용하면서 계속 반복적으로 찾아야한다.

두번째 문제점으로는 데이터가 가지는 경향성이나 시간적 지속성을 고려하지 않고 단지 정해진 윈도우를 이미지(image)형태로 취급하고 클러스터링(clustering)방법에 따라 유사한 그룹으로 분류한다는점이다. 그러나 일반적으로 어떤 시계열 데이터를 그래프로 표현하고 패턴을 발견할 경우에 생각할 수 있는 것은 주변의 다른 그래프의 모양보다 뚜렷이 구분이 되고 자주 발생되는 패턴을 먼저 고려한 다는 것이다. 즉 패턴의 특징이 될 수 있는 것은 값의 변화가 다른 것보다 크고 어느 정도의 지속성을 가진 형태의 패턴이 될 수 있다. 따라서 단순히 이미지 형태로 패턴을 비교하는 것은 의미없는 패턴을 발견하는 결과를 만든다.

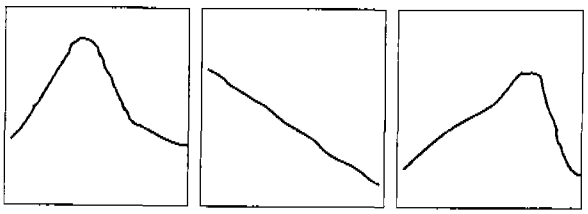
따라서 우리는 기존 연구의 문제점을 극복하고 수치값을 가진 데이터 집합의 시계열 데이터 에서 보다 일반적이고 가능한 모든 종류의 순차 패턴을 발견하기 위해 다음과 같은 특징을 반영한 알고리즘을 제시한다. 먼저 기존의 윈도우를 적용한 패턴 요소를 발견하는 방법이 아닌 데이터 내에 있는 시간적으로 이웃하는 데이터 값의 변화 경향성을 기준으로 패턴요소를 발견한다. 이와 같은 방법은 기존 연구[10]에서 시계열 데이터에서 발생하는 트렌드 순서가 같은 모양의 데이터 탐색과 유사할 수 있으나 , 기존 방법에서는 사전에 데이터의 트렌드 성격을 미리 정하여 놓고 이들의 존재여부를 파악하나 , 우리는 이웃하는 데이터의 연관성을 계산하여 이들을 패턴의 요소로 처리한다. 따라서 윈도우를 사용하지 않기 때문에 기존 연구의 문제점이 해결된다. 두 번째로는 같은 트렌드(예를 들어 상승 후 하강)에 대해서도 이들이 만들어지는 값에 따라 새로운 패턴을 발견한다. 예를 들어 A 사의 지난 10 년간의 장난감의 월별 판매 패턴을 보니까 11 월 대비 12 월이 보다 높은 패턴을 보이고 이러한 현상이 매년 일정한 증가율로 계속적으로 차이가 커질 경우 기존 방법에서는 발견하지 못하였다. 그러나 우리는 패턴 요소를 트렌드 성격을 나타내는 벡터로 표현하여 벡터가 가지는 포함관계를 고려한 순차 패턴의 검색으로 보다 많은 패턴 정보를 밝힐 수 있다.

3. 문제의 정의

3.1 순차패턴 요소의 발견

순차 데이터를 가진 시계열 데이터베이스 $\boldsymbol{D}$가 주어졌을 때 이들 내부 데이터들이 가지는 값의 범위가 $I=\{-\infty,0,+\infty\}$일 경우 각각의 값을 기초로 순차패턴을 발견하는 것은 충분한 지지도(support)를 얻기 힘들고 앞에서 설명한 것과 같이 의미없는 패턴을 발견하므로 이들을 적당한 방법으로 데이터 그룹을 만들어 순차패턴의 패턴 요소를 만들어야 한다. 이와 같은 방법을 적용하기 위한 방법으로는 윈도우 크기만큼 일정한 간격으로 서로 겹치면서 데이터를 구분하는 방법과 최근에 연구된 방법[6]과 같이 타임시리즈의 데이터를 $\boldsymbol{k}$개의 샘플링을 통하여 점을 구하고 이들을 직선으로 연결하는 조각으로 만들어 유사한 모양의 트렌드를 찾는 방법이 등이 있다

그러나 이와 같은 방법은 유사성을 측정하는 방법에서 시간축의 작은 변화에도 전혀 다른 데이터 그룹으로 묶이거나 또는 하나의 패턴 요소를 반복적으로 표현하여 중복 현상의 발생 등 몇 가지 문제점있다.



w1 w2 w3

그림 [3]

예를 들어 위의 그림[3]의 3 가지 형태의 윈도우 데이터가 있을 경우 이들 각각의 최저 값과 최고 값이 110,212 일 때 이들이 상호간의 유사성을 측정하기 위해 가장 일반적으로 사용하는 측정방법(Euclidean Distance)를 사용하여 계산하면 W1 과 W2 가 보다 적

은 값을 가지는 결과를 만든다. 그 것은 값을 비교시 같은 시점에 발생된 데이터간에 만 비교하므로 시간적 외곡이 약간만 있어도 전혀 다른 결과를 만들내기 때문이다. 이러한 것을 해결하기 위해 많은 방법(Time Warp method)이 연구되었으나 시간적 비용(time cost)이 과다하게 소요된다. 따라서 우리는 이러한 시간적 외곡을 쉽게 처리 할 수 있고 시간적으로 이웃하는 데이터의 변화 경향성을 패턴 요소로 만드는 방법을 선택한다. 기존의 연구와 가장 큰 차이점으로는 윈도우를 사용하지 않고 데이터가 가지는 패턴 요소를 만든다는 것이다.

즉 타임시리즈 데이터 $D=\{x_1,x_2,...,x_n\}$ 에서 $\{x_i,x_{i+1},...,x_{i+m}\}$ 의 데이터가 시간에 따른 값의 변화가 상관관계가 있다면 같은 상관계수 값(correlation coefficient) 을 가지는 데이터를 하나의 패턴으로 하는 것이다. 일반적으로 일련의 데이터에서 상관계수를 구

$$r=\frac{m(\sum_{i=1}^{m}t_ix_i)-(\sum_{i=1}^{m}t_i)(\sum_{i=1}^{m}x_i)}{\sqrt{\left[m(\sum_{i=1}^{m}t_i^2)-(\sum_{i=1}^{m}t_i)^2\right]*\left[m(\sum_{i=1}^{m}x_i^2)-(\sum_{i=1}^{m}x_i)^2\right]}}$$

하는 식은 Pearson 의 상관계수 식[1]을 사용한다.

식[1] r : Pearson's 의 상관계수

위의 식에서 $1\le t_i\le m$ 인 범위를 가지며 x_i 는 타임시리즈 내의 값을 나타낸다.

따라서 사용자가 정한 상관계수 값을 λ 라 할 경우 값 λ 이상을 가지는 데이터의 그룹을 하나의 패턴요소로 보고 이들 다양한 패턴요소가 시간적 순서를 가지고 만들어내는 순차패턴을 찾는다. 상관계수 값을 만족하는 t 번째 패턴요소를 s_i 라 할 때 일반적인 정의 식으로 로 표현하면 다음과 같다.

정의 1:

$$s_i=\{x_i,x_{i+1},...,x_{i+m}\},\{s_i\parallel r_i\mid\ge\lambda\}_i$$

예를 들어 $\boldsymbol{D}$ = { 157 , 169 , 179 , 188, 168 , 154 , 152 , 162 , 172 , 182 , 192 , 200 , 210 , 220 , 240 } 이고 λ =0.9 일 경우 이들의 패턴요소는 다음과 같이 3 개로 표현된다. s_1 ={157,169,179,188}, s_2 ={188,168,154}, s_3 ={152, 162, 172, 182, 192, 200, 210, 220, 240} 이들은 모든 상관계수 값이 모두 0.9 이상을 가지므로 조건을 만족하는 패턴요소가 된다.

패턴요소를 이웃하는 값의 변화를 나타내는 벡터로 표현할 때 값이 형성되는 위치를 고려해야 한다. 예를 들어 미국의 다우존스 주가는 1940 년에 약 150 포인트이었으나 지금 2000 년에는 약 1 만포인트를 웃돌고 있다. 그러나 과거와 현재의 주요한 주가 패턴은 유사한 경향성을 가지고 있다. 따라서 150 포인트에서 15 포인트 주가 상승은 1 만 포인트에서는 15 포인트가 아니 1000 포인트의 효과와 같다. 그러나 기존의 방법에서는 이런 차를 구분하는 방법을 적용하지 않고 단지 정규화(normalizing)만을 사용하여 1 만포인트에서 같은 값인 15 포인트 값을 형성하는 것을 같은 패턴으로 처리하였다. 우리는 이와 같은 문제점을 해결하기 위해 이웃하는 데이터간의 값의 편차가 누적되어 만들어내는 변화가 위에서 정의한 상관계수 값을 만족하는 것을 하나의 패턴으로 하였다.

즉 $d_i=\{x_i,x_{i+1},...,x_{i+m}\}$ 의 순차 데이터에서 변화율 함수 $\phi_\nabla(d_i)=\{\nabla_1,\nabla_2,...,\nabla_{m-1}\}$ 로 데이터를 변환하는데 이때 $\nabla_i=(x_{i+1}-x_i)/x_i$ 를 나타낸다. 따라서 이들 변화값 편차들이 나타내는 누적된 데이터는 다음과 같다. 그러므로 이들이 하나의 패턴 요소가 되는 트렌드를 만들기 위해서는

$$\phi_g(\phi_\nabla)=\{\sum_{i=1}^{1}\nabla_i,\sum_{i=1}^{2}\nabla_i,...,\sum_{i=1}^{m-1}\nabla_i\}=\{\phi_1,\phi_2,...,\phi_{m-1}\}$$

$$f_v(\phi_g)=\{\phi_1,\phi_2,...,\phi_{m-1}\}=v_i\ \{v_i\parallel r_i\mid>\mid\lambda\mid\}$$

를 성립하여야 한다.

이렇게 만들어진 데이터 집합을 보다 다루기 편리하게 하기 위해 벡터의 형태로 표현한다. 즉 기울기(α) 와 시작위치 값(St) 및 기간(d) 값을 가지는 벡터적인 표현으로 표기한다. 따라서 하나의 벡터는 다음과 같다.

$$v_i(s_i)=(\alpha_i,St_i,d_i)$$

여기서 기울기 α_i 는 다음의 식을 이용하여 구하며 기간 $d_i=x_{i+m}.t-x_i.t$, $St_i=x_i.t$ 이다

$$\alpha=\frac{(m\sum_{i=1}^{m}t_ix_i)-(\sum_{i=1}^{m}t_i)(\sum_{i=1}^{m}x_i)}{(m\sum_{i=1}^{m}t_i^2)-(\sum_{i=1}^{m}x_i)^2}$$

식 2 : 기울기 값

이와 같이 표현하므로 인해 기존의 윈도우 방식보다는 데이터의 특징을 보다 잘 반영할 수 있고 전체 데이

터를 다루는 것보다는 보다 압축된 데이터를 다루게 된다.

위와 같이 벡터의 표현으로 만들어진 순차패턴의 패턴요소는 아주 다양한 값의 범위를 가진다. 즉 기울기(α) 값이 가지는 다양성과, 기간(d)도 같은 기울기 값에서도 아주 많은 종류가 발견될 것이다. 따라서 이들 값 자체만으로는 자주 발생되는 공통적인 패턴요소를 발견하기 힘들므로 충분한 지지도(support)를 얻는 패턴요소를 발견하기 위해 위에서 만들어진 벡터리스트 전체에 대해 그룹화 작업을 해야 한다. 이것은 일반적으로 생각할 수 있는 간단한 방법으로 처리가 가능하다. 즉 기울기를 기준으로 0 부터 +90 까지 및 -90 까지의 각도에 대하여 사용자가 일정한 간격으로 나누고 이 범위에 포함되는 모든 기울기는 같은 값을 가지도록 변환하는 것이다. 우리는 여기서 기간을 고려하지 않았는데 그것은 다음에 설명되는 패턴요소간의 연관성 정의에 따라 고려된다. 따라서 모든 벡터들은 소속되는 그룹의 기울기(α)값을 대표값으로 하는 벡터로 이루어진 집합이 된다.

3.2 패턴 요소의 연관성 정의

우리는 타임시리즈 각 데이터 값을 일련의 연속되는 값들이 사용자가 정의한 상관계수 값 이상을 가지는 벡터로 표현하였고 또한 이렇게 표현된 벡터들은 패턴발견을 위해 충분한 지지도(support)를 얻기 위해 그룹화을 통해 패턴이 되는 기준벡터를 발견하였다. 우리는 이들 통해 순차 패턴을 발견한다. 그런데 이렇게 만들어진 m 개의 패턴요소 벡터의 집합 $I_V = \{v_1, v_2, ..., v_m\}$ 에 있는 기준 벡터는 $v_i = (\alpha_i, *, d_i)$ 로 시작점을 가지지 않는 벡터들 이다. 그리고 이들 표현된 기준 벡터들은 기울기가 같은 두개의 벡터 v_i, v_j 간에 다음과 같은 정의[1]가 성립한다.

정의 1:

$$\exists v_i, v_j,\ \{v_i.\alpha = v_j.\alpha \wedge d_i \le d_j \mid v_i \in v_j\}$$

즉 $v_i = (\alpha_i, *, d_i), v_j = (\alpha_j, *, d_j)$ 두 벡터가 기울기(α)가 같고 $d_i \le d_j$ 일 경우 벡터의 합에 따라 기간(d_j)이 큰 벡터는 $v_j = (\alpha_i, *, d_i) + (\alpha_i, *, d_j - d_i)$ 와 같다. 따라서 서로 다른 패턴요소 일지라도 기울기가 같으면 그 내부에는 기간의 크기가 작은 벡터가 포함된 것으로 볼 수 있다.

즉 $v_j = v_i.d_i + v_\nabla.d_\nabla$ 가 되며 이것은 벡터 v_j 가 발생된 기간에 발생된 것이므로 $v_j = \{(v_i v_\nabla), (v_\nabla v_i)\}$ 의 집합으로 볼 수 있다. v_∇ 벡터가 먼저 발생되고 다음에 v_i 가 이후에 발생된 $v_j = \{(v_i v_\nabla)\}$ 또는 반대의 경우 $v_j = \{(v_\nabla v_i)\}$ 가 동시에 존재한다. 또한 위의 정의 1 은 다음과 같은 정의 2 로 확장이 가능하다.

정의 2: $\exists v_i, v_j,\ n > 0$ 일경우

$$\{v_i.\alpha = v_j.\alpha \wedge d_j = n \cdot d_i \mid v_j = \bigcup_n v_i\}$$

위의 정의는 벡터의 곱의 기본적인 성질에 따라 만족하는 것으로 $v_j = v_{i(1)}.d_i + v_{i(2)}.d_i + ... + v_{i(n)}.d_i$ 와 같다. 따라서 우리는 $v_j = \{(v_{i(1)} v_{i(2)} ... v_{i(n)})\}$ 으로 벡터 v_j 를 표현한다.

예를 들어 $s_i = \{v_1 v_2 v_3\}, s_j = \{v_4 v_5 v_6\}$ 두개의 순차패턴에서 v_1, v_4 가 같은 기울기를 가지고 기간 $d_4 = d_1 + d_\nabla$ 이며 $v_6.\alpha = v_3.\alpha$, $d_6 = d_3 + d_\nabla$ 인 관계를 가지는 경우 위의 정의에 따라 이들의 벡터들의 의미적인 포함관계를 이용하여 집합으로 표시하면 다음과 같다.

$$s_j = \{((v_1 v_\nabla), (v_\nabla v_1))(v_2)((v_3 v_\nabla), (v_\nabla v_3))\}$$

따라서 $s_j = \{v_\nabla v_1 v_2 v_3 v_\nabla\}$ 의 순서를 가지는 벡터리스트를 가지므로 $s_i \in s_j$ 의 관계가 성립하게 된다.

4. 순차패턴 발견

순차패턴을 발견하는 것은 기존의 여러 다양한 알고리즘을 대부분 적용할 수 있으나 위에서 정의된 벡터들이 가지는 의미적인 포함관계를 추가한 탐색알고리즘이 필요하다. 쭉 기존연구[2]에서 GSP 알고리즘을 적용하지만 별도의 계층구조를 위한 데이터베이스가 필요하지 않고 또한 벡터들이 형성하는 값에 의한 일련의 벡터들이 가지는 값의 변화을을 같이 탐색하는 보다 일반적인 알고리즘을 제시한다.

4.1 패턴 후보의 생성

우리는 기존[1]에 AprioriAll 알고리즘과 유사한 방식으로 전체 시계열 데이터의 순차패턴 중 빈번히 발생되는 패턴요소를 이용해서 순차패턴이 되는 후보

(candidate)를 만들어 가능한 모든 순차패턴을 발견한다. 그러나 이때 기존 방법과 다른점은 벡터들이 가지는 포함관계를 함께 고려하면서 순차패턴을 찾는 것이다 .

패턴후보를 만드는 과정은 크게 두 가지로 정리될 수 있다. **k**개의 벡터를 가지고 만들어지는 순차패턴을 L_k 라 할 때 L_k 를 찾기 위해 L_{k-1} 인 순차패턴을 이용하여 C_k 인 순차패턴 후보(candidate) 를 만드는 과정과 만들어진 C_k 에 대하여 사용자가 지정한 최소 지지도(minimum support) 값을 만족하지 않는 순차패턴후보를 제거(prune) 하는 과정이다. 여기서 우리는 최소 지지도(minimum support)의 정의를 벡터화된 타임시리즈 내에 있는 벡터 중에서 부분적 순차패턴의 비율로 정의 한다. 일반적으로 순차패턴 후보를 만드는 방법은 알고리즘 [1]과 같다. 즉 바로 이전에 만들어진 L_{k-1} 를 s_1 이라 하고 같은 L_{k-1} 를 s_2 라 할 경우 s_1 와 s_2 를 상호 결합(Join)하는데 기존 방식과 다른점은 s_1 의 2 번째부터 와 s_2 의 첫번째부터 순차적으로 발생되는 벡터가 s_2 의 **k**-2 번째 까지 같은 튜플에 대하여 s_1 의 모든 벡터에 추가로 s_2 의 마지막 벡터를 첨가한 결과를 패턴후보로 한다.

INSERT INTO C_k;
SELECT $s_1.v_1,...,s_1.v_{k-1},s_2.v_{k-1}$;
FROM $L_{k-1}\ s_1, L_{k-1}\ s_2$;
WHERE
$s_1.v_2 = s_2.v_1\ and\ ...\ and\ s_1.v_{k-1} = s_2.v_{k-2}$;

알고리즘 [1]

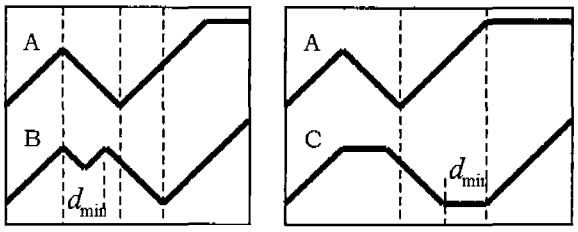
위의 알고리즘은 보다 의미 있는 패턴 검색을 위해 순차패턴이 만들어내는 최대기간 d_{max} 을 주어 각 벡터들로 구성되는 패턴이 가지는 전체 시간의 합이 d_{max} 를 넘지 않을 때까지 순차패턴을 찾아나간다.

4.2 순차패턴 발견

위의 알고리즘[1]에서 만들어진 패턴후보 C_k 는 이전에 발견된 자주 발견되는 부분적 순차패턴을 이용하여 가능한 모든 순차 패턴을 만들었다. 이와 같이 만들어진 패턴 후보는 다시 데이터베이스를 읽어 각 패턴후보에 대한 발견된 횟수(count)를 증가하면서 사용자가 준 최소 지지도 이상의 값을 가지는 패턴만을 남기고 지지도 값을 넘지 않는 패턴후보는 제거(prune)하는 과정을 거치게 된다. 이와 같은 패턴후보 제거과정에서 C_k 에 있는 패턴후보가 순차패턴 데이터에 존재하는지를 파악하는 기준은 앞의 정의[1],[2]를 이용한다. 우리는 앞에서 시계열 데이터를 벡터로 표현할 때 사용자가 지정한 기울기 단위로 값이 같은 그룹으로 묶어 표현하였다. 그러나 같은 기울기값을 가지는 벡터는 또한 기간(d)값이 다름에 따라 다양한 종류의 벡터가 만들어진다. 따라서 앞의 정의[1][2]를 사용함에 있어서 연속된 벡터가 만들어내는 순차패턴을 찾을 때 기간의 편차를 나타내는 v_∇ 에 대하여 고려를 해야하는 문제가 남아있다. 위의 3.2 절의 마지막에 예를 보인 패턴의 비교는 3 개의 벡터로 구성된 패턴중 v_2, v_5 는 같은 길이와 기울기를 가지는 벡터이었으나 만일 v_5 의 기간이 v_2 보다 d_∇ 만큼 클경우는 v_∇ 를 고려하지 않을 경우 전혀 다른 패턴이 같은 패턴으로 간주될 수 있다. 따라서 d_∇ 값의 허용범위는 사용자가 지정하는 값 d_{min} 범위 이내의 값에 따라 고려한다. 즉 두개의 순차패턴이 같다는 것은 이들 순차패턴을 만드는 패턴요소인 각각의 벡터의 기울기가 같고 바로 이웃하는 벡터와의 시간적 거리가 $d_\nabla <= d_{min}$ 일 경우만 성립하는 것이다. 예를 들어 앞의 3.2 절의 마지막 예제의 경우 v_5 의 기간이 v_2 보다 d_∇ 만큼 클경우 이들이 만들어내는 순차패턴의 집합은 다음과 같다.

$s_j = \{((v_1 v_\nabla),(v_\nabla v_1))((v_\nabla v_2),(v_2 v_\nabla))((v_3 v_\nabla),(v_\nabla v_3))\}$
$s_j = \{(v_\nabla v_1 v_\nabla v_2 v_3 v_\nabla)\}$ 또는 $s_j = \{(v_\nabla v_1 v_2 v_\nabla v_3 v_\nabla)\}$

즉 벡터 v_5 가 v_4 와 v_6 사이에 있으면서 d_∇ 만큼의 기간차이로 인해 시간적 편차를 나타내는 벡터 v_∇ 가 패턴요소로 존재한다. 만일 $d_\nabla <= d_{min}$ 를 만족한다면 두개의 순차패턴은 같은 순차턴을 만족하는 것이나 $d_\nabla > d_{min}$ 경우는 v_∇ 는 별개의 패턴요소이므로 다른 패턴으로 본다. 우리는 이러한 이웃하는 패턴요소 v_∇ 를 다른 기울기를 가지고 단지 $d <= d_{min}$ 인 패턴요소 벡터에도 적용하므로써 패턴요소들간에 나타나는 비정상적(outlier) 패턴요소의 발생에도 확장하여 적용할 수 있다. 다음 그림[4]는 기울기 값이 다른 패턴이 순차패턴 내부에 존재하는 경우를 나타낸다.



그림[4]

기존의 연구에서는 패턴 A,B,C는 서로 다른 별개의 패턴으로 처리되었다. 그러나 d_{min}을 고려할 경우는 패턴 A,B,C 는 모두 같은 순차패턴으로 발견된다. 즉 패턴을 구성하는 벡터가 d_{min}이내에 발생된 다면 중간에 다른 패턴요소가 있다 하더라도 같은 순차패턴의 형태로 취급한다.

L_1 ={large 1-vector};
for (k=2; $L_{k-1} \neq 0$ or Len(L_{k-1}) $< d_{max}$;k++) do
begin
C_k = New candidates generated form L_{k-1} ;
// Call Algorithm [1]
foreach subsequence s in the vectored-data sequence do
Increment the count of all candidates in C_k that are contained in s
// Consider included pattern element vector
L_1 = Candidates in C_k with minimum support
end

알고리즘[2]

우리는 지금까지 순차패턴요소가 이웃하는 패턴요소간의 포함관계를 고려하여 같은 순차패턴으로 지속되는지를 판단하는 기준을 제시하였다. 따라서 패턴을 찾는 방법은 크게 2가지로 구분될 수 있다. 먼저 각 패턴요소인 벡터 내부에 발생될 수 있는 모든 벡터의 형태를 가정하는 과정과 지지도 값을 만족하는 패턴후보에 대하여 같은 패턴의 형태가 데이터베이스 내부에 존재하는지 찾는 과정으로 정리된다. 따라서 위와 같은 포함관계를 고려한 패턴탐색은 기존의 AprioriAll 알고리즘[2]와 같으며 단지 패턴후보 C_k에 대한 벡터로 변환된 시계열 데이터에서의 탐색 시 위의 포함관계만을 적용한다.

5. 결론 및 향후 연구방향

본연구는 데이터베이스에서 수치 값의 형태로 구성된 순차패턴을 발견하는 방법을 제시하였다. 기존의 윈도우 방법보다 이해되기 쉽고 빠르게 순차패턴을 발견하며 또한 적용하고자 하는 응용분야 전문가가 직접 패턴을 발견할 경우 발견하고자 하는 패턴의 패턴요소인 벡터에 대하여 기울기나 벡터의 길이(시간)를 제한 조건으로 적용하므로서 불필요한 패턴의 발견을 제거할 수 있고 보다 의미있는 패턴만을 발견할 수 있다. 이와 같은 순차패턴의 발견에 관한 연구는 최근에 기업내부에서 발생되는 다양한 형태의 시계열 데이터를 기업외부의 데이터와 상호 연관성을 고려하여 변화 패턴을 발견하는 곳에 적용이 가능할 것으로 본다. 특히 패턴요소를 벡터로 표현하므로서 일상적으로 발생되는 것과는 다른 패턴의 발견이 용이하고 적용분야에 따라 이러한 발견은 보다 의미있는 정보가 될 것이다.

참고문헌

[1] R. Agrawal, R.Srikant. *Mining Sequential Patterns. In 11TH International Conference on Data Engineering* (ICDE'95)

[2] R. Srikant, R. Agrawal. Mining Sequential Patterns: Generalizations and Performance Improvements . *In 5th International Conference Extending Database Technology, Mar 1996*

[3] H. Mannila, H. Tovonen, A. Verkamo, Discovering frequent episodes in sequence. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (KDD'95)

[4] G. Das, K.-I, Lin, H. Mannila . Rule Discovery from Time Series . In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining* (KDD'98)

[5] H. Mannila, H. Toivonen. Discovering generalized episodes using minimal

occurrences. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (KDD'96)

[6] R. Agrawal, K.-P. Lin, H.S. Sawhney K. Sim. Fast Similarity Search in the Presence of Noise, Scaling , and Translation in Time-Series Databases. *In proceedings of the 21ST International Conference on Very Large Data Bases* (VLDB'95)

[7] E. Keogh, M.Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Proceedings of the 4rd International Conference on Knowledge Discovery and Data Mining* (KDD'98)

[8] E. Keogh, M. Pazzani. An Indexing Scheme for Fast Similarity Search in Large Time Series Database. *In Proceeding of the 11th International Conference on Scientific and Statistical Database Management* 1999

[9] C. Faloutsos, M. Ranganathan, Y. Manolopoulos. Fast Subsequence Matching in Time-Series Databases. *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data.*

[10] R. Agrawal, G. Psaila, E.L. Wimmers. *Querying Shapes of Histories. In proceedings of the 21st International Conference on Very Large Data Bases* (VLDB'95)

[11] R. Agrawal, C. Faloutsos, A. Swami. Efficient Similarity Search In Sequence *Databases . Foundations of Data Organization and Algorithms, 4th International Conference,* (FODO'93)

[12] J. Eamonn , E. Keogh, M. Pazzani. Scaling up Dynamic Time Warping to Massive Datasets. *Principles and Practice of Knowledge Discovery in Databases* (PKDD99)