[1)]하둡 기반 빈발 시퀀스 추출기 개발

박준하*, 이병희*, 박상재*, 이정준*
*한국산업기술대학교 컴퓨터공학과
e-mail: junha2001@nate.com, qudgml89@kpu.ac.kr,
psjcompany@nate.com, jjlee@kpu.ac.kr

Development of Frequent Sequence Extractor Based on Hadoop

Joon-Ha Park*, Byung-Hee Lee*, Sang-Jae Park*, Jeong-Joon Lee*
*Dept. of Computer Engineering, Korea Polytechnic University

요 약

최근 증권, 센서, 기후, 의료 분야 등에서 수많은 시계열 데이터들이 쏟아져 나오고 있고, 이러한 시계열 빅 데이터를 통해 의미를 찾아내고자 하는 시계열 해석 및 분석, 예측 작업의 수요가 증가하고 있다. 시계열 해석 및 분석, 예측 작업을 하기 위해서 사용 될 수 있는 기초 작업은 유사한 시계열 시퀀스를 찾아내는 유사 시퀀스 매칭과 이러한 매칭을 통해 특정 시계열 데이터의 하나의 특징이 되는 빈발 시퀀스 추출 기술이 필요하다. 본 논문에서는 이러한 시계열 빅 데이터에서 유사 시퀀스 매칭을 이용한 빈발 시퀀스 추출 문제를 효율적으로 해결하는 빈발 시퀀스 추출기(Frequent Sequence Extractor)를 개발 및 구현하였다. 또한 분산처리 플랫폼인 하둡을 이용한 데이터 파싱을 사용하여, 각 분야별 시계열 데이터를 분석하는 전문가에게 효율적인 분산처리 효과를 제공한다.

1. 서론

시계열 해석(time series analysis)은 일정 시간 간격으로 배치된 데이터들의 수열을 일컫는 시계열(time series)을 해석하고 이해하는데 쓰이는 여러 가지 방법을 연구하는 분야이다[1]. 최근 증권, 센서, 기후, 의료 분야 등에서 수많은 시계열 데이터들이 쏟아져 나오고 있고, 이러한 시계열 빅 데이터를 통해 의미를 찾아내고자 하는 시계열 해석 및 분석, 예측 작업의 수요가 증가하고 있다.

시계열 분석에 사용되는 여러 기술 중 빈발 시퀀스 추출(Frequent Sequence Extraction)[2, 3]은 유사한 시계열 서브 시퀀스를 찾아내는 유사 시퀀스 매칭[4] 기술과 유사 시퀀스 매칭 결과를 유사도로 표현하는 작업을 통하여 해당 시퀀스에서 가장 자주 발생하는 빈발 패턴을 찾아내고 저장한다. 이를 통해 특정 시계열 데이터에 경향 또는 추이를 파악할 수 있으며 이후 예측에 활용할 시 중요한 역할을 할 수 있는 지표가 된다.

기존에 사용되는 시계열 분석 방법은 시계열 데이터로부터 분석하고자 하는 특정 값들을 추출한 다음 시계열 분석을 사용한다[1, 2, 3, 5]. 시계열 데이터로부터 특정 값들을 추출하는 작업은 시계열 분석에 있어서 꼭 필요한 필수조건이다. 하지만 이러한 작업은 시계열 데이터가 대용량의 빅 데이터로 점차 커지고 있는 상황에서 많은 비용과 시간을 소모하게 된다.

1) 본 연구는 지식경제부의 지원을 받는 정보통신표준화 및 인증 지원사업의 연구결과로 수행되었음

본 논문에서 개발 및 구현한 시스템은 대용량의 시계열 데이터에서 하둡(Hadoop)을 이용한 효율적인 분산 처리 기술을 접목하여 대용량에서 빈발 시퀀스를 추출할 값들의 추출을 효율적으로 개선하고, 시계열 분석을 하고자하는 시계열 분석가들에게 효율적인 시계열 분석 알고리즘을 사용할 수 있도록 도와주는 FSE(Frequent Sequence Extractor)를 구현하였다.

본 논문에서는 관련 연구에서 시계열 분석과 관련된 기존 연구들과 효율적인 분산처리 기술인 하둡의 맵리듀스(MapReduce)에 대해서 알아본다. 이후 세부 설계 및 구현에서는 본 논문에서 제안한 시스템에 세부적인 설계와 구현된 엔진을 살펴본다.

2. 관련 연구

2.1.시계열(Time Series)과 서브 시퀀스(Subsequence)

시계열 데이터는 각 시간별로 측정한 실수 값들의 시퀀스로, 그 예로는 주식 데이터, 환율 데이터, 날씨 변동 데이터 등이 있다. 시계열은 최근 빅 데이터가 화두가 되고 있는 현 시점에서 수십억 개 이상의 시퀀스로 이루어질 정도로 그 데이터가 커지고 있다[6].

서브시퀀스란 시계열 데이터들의 집합을 하나의 시퀀스라고 칭하고, 특정 길이로 이루어지도록 자른 형태를 말한다. 예를 들어 시작 위치가 a이고, 길이 x로 이루어진 서브 시퀀스 S는 $S[a, a+x-1]$의 구조를 갖게 된다[7].

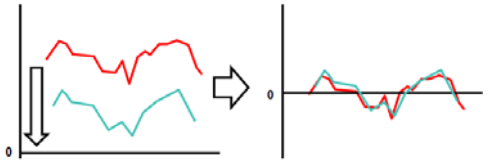
2.2. 유사 시퀀스 매칭(Similar Sequence Matching)

유사 시퀀스 매칭이란 길이가 같은 두 개의 서브시퀀스가 주어졌을 때, 두 서브시퀀스 간의 유클라디안 거리를 이용하여 비교하는 방법이다.

정의. ***Match*** 양의 실수 R(거리라고 부름) 시계열 T의 길이가 같은 서브시퀀스 C, M 이 주어지고, 그 두 서브시퀀스들 간의 유클라디안 거리를 $D(C,M)$ 이라고 할 때, $D(C,M) < R$ 이면, 서브시퀀스 C와 서브시퀀스 M은 매칭 되었다고 한다[8, 11].

2.2.1 S.E. 매칭(Shift Elimination Matching)

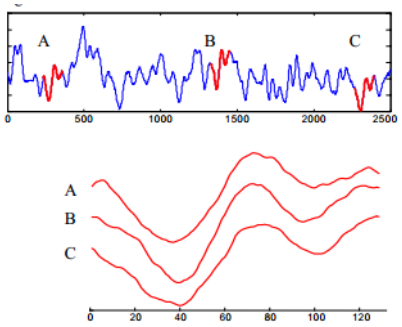
S.E. 매칭이란 두 개의 길이가 같은 서브시퀀스에서 유클라디안 거리를 비교할 때, 서브시퀀스의 각 벡터 값들의 값의 높이를 줄이는 작업이다. 이는 서브시퀀스의각 벡터 값들을 평균으로 빼는 과정을 통해 이루어진다. <그림 1>은 두 개의 길이가 같은 서브시퀀스의 S.E. 변환 과정을 보여주고 있다.



<그림 1> S.E 변환을 통한 매칭 기술

2.2.2 빈발 시퀀스 추출(Frequent Sequence Extraction)

시계열 데이터의 빈발 시퀀스는 해당 시계열 시퀀스에 가장 많이 나타나는 대표적인 서브시퀀스이다[11, 12, 13]. <그림 2 >에서는 시계열 시퀀스에서 가장 많이 발생한 3개의 시퀀스를 찾아낸 것을 나타내고 있다.



<그림 2> (위) 시계열 시퀀스에서 찾아낸 3개의 서브시퀀스[11]. (아래) 3개의 서브시퀀스를 확대한 모습[11].

2.3 하둡의 맵리듀스(MapReduce on Hadoop)

맵리듀스는 대용량 데이터를 병렬로 처리하기 위한 소프트웨어 프레임 워크이다. 신뢰할 수 없는 많은 저가의 장비로 구성된 클러스터 환경에서 페타 바이트 이상의 대용량 데이터를 병렬로 처리하기 위한 함수형 프로그래밍 기법으로 맵(Map)과 리듀스(Reduce)라는 함수를 기반으로 구성된다[9, 10]. 본 논문에서 구현한 FSE는 대용량의 시계열 데이터를 파싱하기 위해서 하둡의 맵리듀스를 사용한다.

2.3. 개발환경

제 2 절 에서는 본 논문에서 제안하는 시스템인 FSE를 개발하기 위하여 사용된 개발 환경과 플랫폼 그리고 외부 라이브러리 대해 설명한다. 개발 또는 실험을 통하여 <표 1> 와 같은 환경을 구축하고 사용하였다.

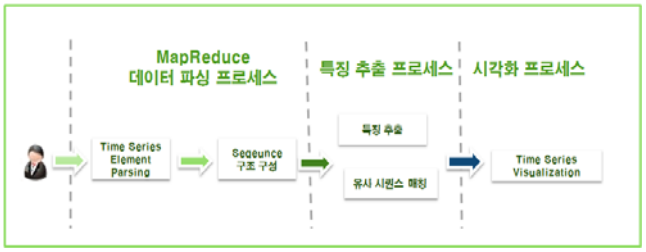
운영체제	Microsoft Windows 7 64 bit
언어	Java SE 1.7.0_25
플랫폼	Apache Hadoop 1.0.4 Java Virtual Machine
툴	Eclipse, Maven Plugin
외부 라이브러리	JLogger, Hadoop, Java Swing, JUnit

<표 1> 개발 환경 정리

3. 세부 설계 및 구현

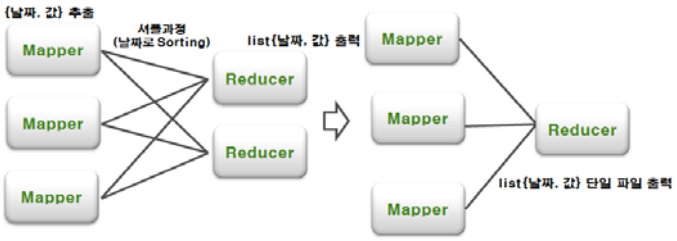
3.1 세부 설계

시퀀스에서 특징을 추출하기 위해서는 크게 3가지 프로세스를 수행하여야만 한다. 첫 번째로 특정 시계열 Raw 파일에서부터 특징을 분석할 값들을 추출하는 데이터 파싱 부분, 두 번째로는 첫 번째에서 파싱된 데이터로부터 그 속성을 파악하고 Sequence 자료구조 인스턴스에 원하는 값을 인-메모리로 저장한 뒤, 유사 시퀀스 매칭, 유사도 변환, 특징 추출 작업을 수행한다. 세 번째로는 추출된 특징을 시각화하는 프로세스를 거쳐 사용자들에게 추출된 특징을 차트로 보여준다. <그림 3> 은 위에서 설명한 세 가지 프로세스가 어떠한 방식으로 진행되는지 보여주고 있다.



<그림 3> FSE 전체 프로세스

첫 번째 데이터 파싱 프로세스에서는 시계열 파일로부터 원하는 {날짜, 값}의 데이터를 추출한다. 이 데이터를 추출하는 과정은 시계열 파일의 크기에 비례하며, 이 파일의 크기에 영향을 주는 것은 각 Row의 애트리뷰트의 개수이다. 빅 시계열 데이터 파싱 시, 노드를 더 많이 구성할 수록 데이터 추출 속도가 향상 된다. FSE의 데이터 파싱 프로세스에서 하둡의 맵리듀스는 <그림 4> 과 같이 사용된다.



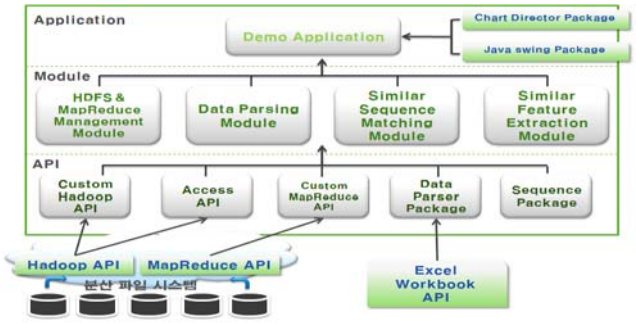
<그림 4> SFEE의 데이터 파싱에 관한 맵리듀스 구성

두 번째 과정에서는 첫 번째 과정에서 추출된 {날짜,값} 결과를 이용하여 유사 시퀀스 매칭, 유사도 변환, 특징 추출과정을 수행한다. 유사 시퀀스 매칭을 하기 위해서 사용자는 슬라이딩 윈도우 크기와 원하는 분석의 범위를 지정해주어야만 한다. 본 논문에서 구현한 엔진은 유사 시퀀스 매칭을 위하여 S.E.매칭 기법을 이용한다. S.E.매칭 과정을 통하여 얻어진 유클라디안 거리를 유사도로 변환하는데, 이때 유사도로 변환하기 위하여 지수유사도를 사용한다. 지수유사도는 유클라디안 거리와 같이 그 거리 값이 무한대로 표현되는 값을 백분율로 표현하여 가독성 있는 유사성을 나타내기 위해 고안되었다. 지수 유사도 공식은 다음과 같다.

고안된 지수 유사도 변환 공식 두 개의 길이가 w로 같은 서브시퀀스 C, M 의 유클라디안 거리가 $D(C,M)$일 때, 지수 유사도 S는 다음과 같다.

$$S(C,M) = e^{-\frac{D(C,M)}{w}}$$

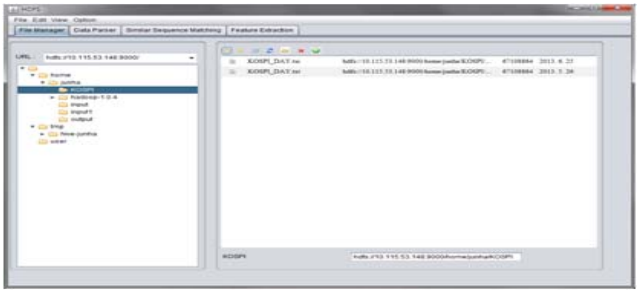
지수유사도 변환을 거친 값들은 거리행렬로 만들어진다. 이 거리행렬을 통하여 빈발 시퀀스를 추출한다. 거리행렬이 주어져 있을 때, 특징 추출 알고리즘을 거쳐 특징으로 추출된 서브시퀀스의 리스트를 리턴 한다. 세 번째 과정인 시각화 프로세스는 JAVA의 JFreeChart를 특징을 시각화하는 도구로 사용하여 진행된다. FSE에서는 시계열 분석가 또는 개발자가 FSE의 클라우드 컴퓨팅 파워를 사용하는 각종 빈발 시퀀스 추출관련 작업을 이용하기 용이하도록 하부 클래스들을 추상화하여 API 형식으로 제공한다 (<그림 5>).



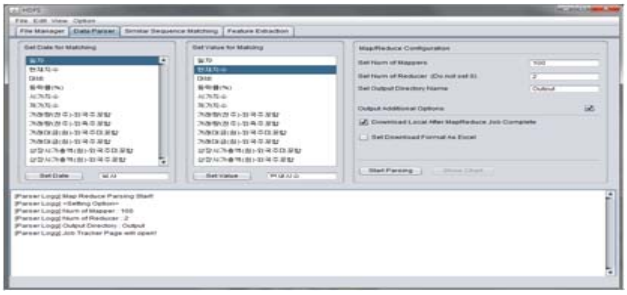
<그림 5> FSE 모듈 구성도

3.2 구현

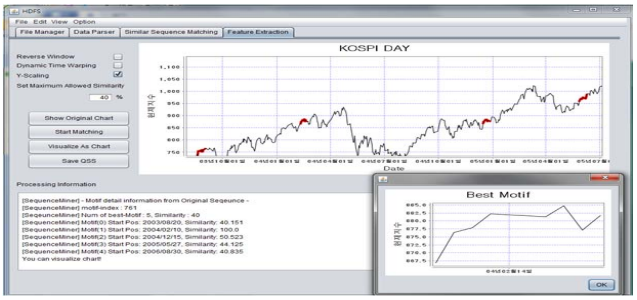
본 절은 FSE의 구현된 모습을 나타낸다. <그림 6>, <그림 7>, <그림 8>은 구현된 FSE의 API를 이용하여 만들어진 데모 어플리케이션이다. 데모를 만들기 위하여 JAVA의 Swing UI 패키지를 외부라이브러리로 사용하였다.



<그림 6> FSE의 하둡 파일시스템 브라우저 UI 구현



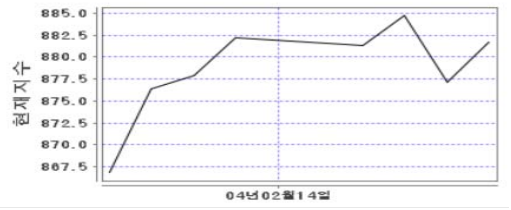
<그림 7> FSE의 데이터 파싱



<그림 8> FSE 특징 추출 화면

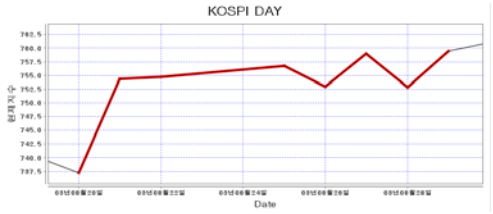
<그림 6>는 하둡을 이용한 파일시스템 접근을 UI로 구현한 화면이다. 이렇게 하둡 파일 시스템(HDFS)에 접근해야하는 이유는 시계열 빅 데이터를 분산으로 처리하기 위해서 시계열 빅 데이터를 각 노드별로 분산 저장해야 하기 때문이다. <그림 7>는 데이터 파싱이 어떤 방식으로 이루어지는 지를 보여주는 화면의 모습이다. <그림 8>는 추출된 데이터로부터 유사 시퀀스 매칭 알고리즘을 사용하여 빈발 시퀀스를 추출한 뒤 추출된 서브시퀀스들을 시각화한 화면이다.

<그림 9>, <그림 10>, <그림 11> 은 금융데이터를 테스트 데이터로 사용하였을 때의 결과 화면이다. 길이가 8인 빈발 시퀀스의 결과이다. <그림 9>의 질의로 허용유사도 40%이상인 서브시퀀스를 검색하였을 때 다음과 같이 5개의 매칭 결과가 나타났다. <그림11>은 5개의 매칭 결과 중 유사도가 가장 낮은 40.151%인 서브시퀀스의 결과이다.



<그림 9> FSE으로 금융데이터에서 추출한 8길이의 특징

<그림 10> 금융데이터에서의 8길이와 매칭 된 결과 화면



<그림 11> <그림 7>의 특징과 매칭 되는 결과 (허용 유사도 40.141)

4. 결론 및 향후 연구

본 논문에서는 이러한 시계열 빅 데이터에서 유사 시퀀스 매칭을 이용한 시퀀스 빈발 시퀀스 추출 문제를 효율적으로 해결하고자, 각 분야별 시계열 데이터를 분석하는 전문가들이 이러한 문제 하에서 효율적인 분산처리 효과를 기대할 수 있는 빈발 시퀀스 추출기(Frequent Sequence Extractor)을 개발 및 구현하였다. 또한 유클라디안 거리를 직접적으로 사용하는 기존의 방법 대신에 지수 유사도를 통한 거리 변환을 이용하여 그 값을 백분율로 표현하여 그 유사성의 척도를 직관적으로 판단 할 수 있도록 하였다. 대용량의 데이터를 처리하기위한 하둡 시스템과 시계열 분석을 위한 알고리즘 및 데이터 처리 과정의 결합을 통해 대용량의 시계열 분석이 가능하도록 하는 시스템을 설계 및 구현했다는 점에 의의를 둔다.

향후 과제로 데이터 파싱에 의한 시계열 데이터에 대한 유사 시퀀스 매칭 알고리즘 수행 시간 또한 시계열 데이터의 크기가 커짐에 따라 비례하게 되는데 이를 위해서 효율적으로 분산으로 처리하여 알고리즘 수행 시간 속도를 개선하는 연구가 이루어져야한다. 또한 유사 시퀀스의 허용유사도를 지정하는 명확한 수학적 계산을 통해 특정 시퀀스 또는 길이의 맞는 허용 유사도를 찾기 위한 방법에 관한 연구가 실험적으로 이루어져야한다.

참고문헌

[1] JD Hamilton "Time Series Analysis," Cambirdge Univ Press, 1994.

[2] Agrawal, R., Faloutos, C., and Swami, A., "Efficient similarity search in sequence databases,". the 4th Conference on Foundations of Data Organization and Algorithms. Chicago, IL. pp.69-84, 1993.

[3] Agrawal, R., Psaila, G., Wimmers, E. L., and Zait, M., "Querying shapes of histories,". the 21th Conference on Very Large Databases. Zurich, Switzerland, pp.502-514, 1995.

[4] HS Lim, KY Whang and YS Moon, "Similar Sequence Matching Supporting Variable-length and variable-tolerance Continuous Queries on Time-Series Data Stream Information Sciences,", the Conference on Information Sciences, pp.1461-1478, 2008.

[5] G.Peter Zhang., "Neural network forecasting for seasonal and trend time series,", European Journal of Operational Research, Vol.160, pp.501-514, 2005.

[6] Hegland, M., Clarke, W., and Kahn, M., "Mining the MACHO dataset,", Computer Physics Communication, Vol.142, pp.22-28, 2002.

[7] C Faloutsos, M Ranganathan and Y manolopoulos., "Fast Subsequence Matching in time-series databases," 1994.

[8] Chan, Kin-Pong, and Ada Wai-Chee Fu., "Efficient time series matching by wavelets.," Data Engineering, 1999. Proceedings., 15th International Conference on. IEEE, 1999.

[9] 김형준, 조준호, 안성화, 김병준 지음, 클라우드 컴퓨팅 구현기술, 에이콘, 2010.

[10] 톰 화이트 저(김우현, 심탁길 역), Hadoop 완벽가이드, 한빛미디어, 2011.

[11] Lonardi, J. L .E. K and Patel, P., "Finding motifs in time series". the 2nd Workshop on Temporal Data Mining. pp.53-68, 2002.

[12] Das, G., Lin, K., Mannila, H., Renganathan, G., and Smyth, P., "Rule discovery from time series". the 4th int'l Conference on Knowledge Discovery and Data Mining. New Yor, NY, Aug 27-31. pp.16-22, 1998.

[13] Hopper, F., "Discovery of temporal patterns - learning rules about the qualitative behavior of time series". the 5th European Conference on Principles and Practive of Knowledge Discovery in Databases. Freiburg, Germany, pp.192-203, 2001.