대용량 영화 데이터를 활용한 벡터 유사도 기반 영화 추천 시스템

신동걸○, 이승형, 조진성
경희대학교 컴퓨터공학과
sindong008@naver.com, shlee7@khu.ac.kr, chojs@khu.ac.kr

Vector Similarity based Movie Recommend System using a Large-scale Movie Data

DongGeol Shin○, SeungHyung Lee, JinSung Cho
Department of Computer Engineering, Kyung Hee University

요 약

본 논문에서는 대용량 영화 데이터를 활용한 벡터 유사도 기반 영화 추천 시스템을 제안한다. 대용량 영화 데이터는 하둡을 통해 데이터 분석에 사용될 데이터를 선별한다. 선별된 데이터는 R프로그래밍 언어를 통해 비교 사용자들과의 유사도를 구하고 영화 예측 평점을 계산한다. 이 유사도와 예측 평점을 바탕으로 추천 영화를 제공한다. 제안하는 시스템을 통하여 사용자는 감상하지 않은 영화들의 예측 평점과 추천 영화를 제공받아 영화 선택을 보다 편리하게 할 수 있다.

1. 서 론

최근 IT기술의 발전으로 소셜 네트워크 등의 대용량 데이터가 축적되고 있으며, 이와 같은 데이터들로부터 가치 있는 정보를 빠르게 찾아내는 빅데이터 기술이 개발되고 있다. 빅데이터 기술은 현재 상용화 단계에 이르러 기존 IT분야뿐만 아니라 금융, 의료, 마케팅 분야 등 거의 모든 산업분야에서 다양하게 활용되고 있다.

다양한 빅데이터 활용 분야 중, 마케팅 분야에서는 사용자의 상품 구매 이력, 상품 선호도 정보 등의 정보를 분석하여 사용자가 원하는 상품을 정확하게 예측하고 추천해줌으로써 더 쉽게 원하는 아이템을 찾을 수 있도록 한다. 특히, 최근 영화와 IPTV시장의 성장으로 이와 같은 빅데이터를 이용한 추천 기술에 더욱 많은 관심이 집중되고 있다.

본 논문에서는 대용량의 영화 평점 데이터를 분석하여 사용자가 선호할 영화를 추천하는 시스템을 제안한다. 제안하는 시스템은 사용자의 영화별 별점을 부여하는 패턴과 유사하게 평점을 부여하는 유사 사용자들을 찾아, 사용자가 아직 감상하지 않은 영화들 가운데 유사 사용자들이 높게 평점을 부여한 영화를 추천한다. 본 논문에서는 또한 빅데이터의 병렬 분석을 통한 고속 추천을 위하여 대표적인 빅데이터 프레임워크이 하둡(Hadoop)과 R 환경에서의 설계 방안을 함께 제시한다.

2. 관련 연구

2.1 아파치 하둡

아파치 하둡 (Apache Hadoop)[1][2]은 노드 클러스터에서 대량의 데이터를 빠르게 처리할 수 있는 오픈소스 분산 프레임워크이다. 이 프레임워크는 분산처리 시스템인 구글 파일 시스템을 대체할 수 있는 하둡 분산 파일 시스템(HDFS: Hadoop Distributed File System)과 맵리듀스 패러다임의 구현을 제공한다.

하둡은 하둡 공통 패키지로 구성되어 있다. 이 패키지에는 하둡 파일 시스템(HDFS), OS 수준 앱스트랙션(OS level abstractions) 그리고 맵리듀스(MapReduce) 엔진이 포함되어 있다.

작은 하둡 클러스터에는 하나의 마스터와 여러 워커 노드들로 구성 되어 있다. 마스터 노드들은 잡트렉커(JobTracker),테스크트렉커(TaskTracker),네임노드(NameNode), 데이터노드(DataNode)로 구성 된다. 슬레이브 또는 워커 노드(Worker Node)는 데이터노드와 테스크트렉커로서 동작을 한다.

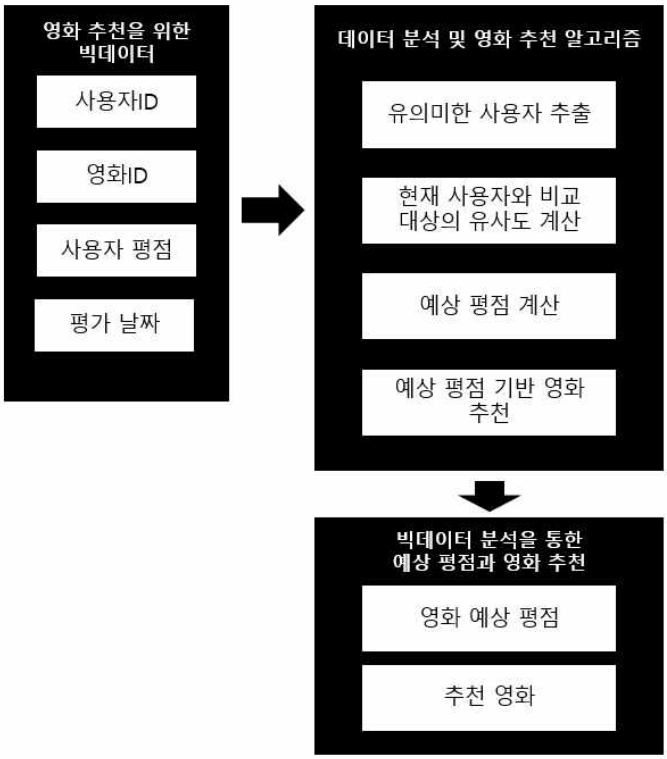
대용량 하둡 클러스터에는 HDFS가 파일 시스템 인덱스를 관장하기 위한 네임노드 전담 서버를 통해 관리된다. 그리고 2차 네임노드는 네임노드의 메모리 구조 스냅샷을 만들어서 파일시스템에 장애나 데이터의 손실을 줄여준다. 단독 잡트렉커 서버는 작업 스케줄링을 관리한다.

2.2 R 프로그래밍 언어

R 프로그래밍 언어[3][4]는 통계 계산과 그래픽을 위한 프로그래밍 언어이자 소프트웨어 환경이다. R은 통계 소프트웨어 개발과 자료 분석에 널리 사용되고 있으며, 패키지 개발이 용이하여 통계학자들 사이에서 통계 소프트웨어 개발에 많이 쓰이고 있다.

3. 빅데이터 기반 영화추천

[그림 1]은 사용자가 영화 평점을 매긴 빅데이터 정보를 통한 영화 추천 알고리즘의 전체 흐름도이다. 준비된 빅데이터 정보를 입력 받으면 하둡을 통하여 실제 분석에 사용될 데이터와 사용되지 않을 데이터를 구별한다. 이 구별이 끝난 데이터는 통계 프로그래밍 언어 R을 통해서 원하는 영화의 예상 평점을 계산한다. 이렇게 예상된 평점을 사용자에게 출력한다.



[그림 1] 제안하는 추천 알고리즘 흐름도

3.1 영화 추천을 위한 빅데이터

본 논문에서 제안하는 빅데이터 기반 영화추천 알고리즘에서는 Netflix prize[5]에서 사용된 데이터를 사용한다. 데이터는 총 1.95GB의 string data로 100,480,507개의 평가로 480,189명의 사용자가 17770가지 영화에 매긴 평점이다. 데이터 파일은 각각의 영화를 평가한 사용자들의 사용자ID, 사용자의 평점, 평가 날짜가 적힌 17770개의 파일이 존재한다.

3.2 데이터 분석 및 영화 추천 알고리즘

3.2.1 유의미한 사용자 추출

Netflix prize 데이터는 17770개의 각 영화별로 사용자가 매긴 평점이 저장되어 있지만 극소수의 영화에만 평점을 부여한 사용자들은 실제 추천의 정확성에 도움이 되지 않을 뿐만 아니라, 수행 시간의 증가를 초래한다. 따라서 Netflix prize 데이터를 사용자 기준으로 정렬한 후 데이터의 신뢰성을 높이기 위하여 k개 이상의 영화에 평점을 준 사용자만을 추천 알고리즘에 사용할 데이터로 다시 선정한다.

이 과정을 효율적으로 처리하기 위하여 데이터를 하둡을 통해 분산 처리한다. 17770개의 영화 파일을 n개의 노드에 분산 저장한 후, 각 노드의 맵퍼에서 사용자별 영화 평점으로 정렬을 수행한다. 그리고 리듀서에서 각 맴퍼의 부분적인 사용자 영화 평점 데이터를 모아 완성된 형태의 사용자별 영화 평점 데이터를 생성한다.

3.2.2 현재 사용자와 비교 사용자의 유사도 계산

사용자와 유사한 평점 부여 패턴을 갖는 유사 사용자를 찾기 위하여 아래와 같은 평점 유사도 수식을 사용한다. 이 수식은 코사인 유사도 공식으로 내적 공간의 두 벡터 간 각도의 코사인 값을 이용하여 측정된 벡터간의 유사한 정도를 의미한다. 각도가 0°일 때의 코사인 값은 1이며, 180°로 완전히 반대 반향인 경우 -1의 값을 갖는다. 따라서 계산된 유사도는 -1에서 1까지의 값을 가지며, -1은 서로 완전히 반대되는 경우, 0은 서로 독립적인 경우, 1은 서로 완전히 같은 경우를 의미한다.

$$\text{similarity} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

A_i는 현재 사용자의 영화 평점, B_i는 비교 대상 사용자의 영화 평점이다. 즉, B_i는 모든 비교 대상자이므로 이 과정을 비교 대상 사용자의 수만큼 반복하며 유사도를 계산한다. 유사도가 1에 가까운 값이 나올 경우 현재 사용자와 비슷한 취향을 갖은 비교 대상이며 -1에 가까운 값이 나올 경우 현재 사용자와 반대 취향을 갖고 있음을 의미한다. 0에 가까운 값이 나온 비교 대상자는 현재 사용자와 독립적인 관계임으로 이후 예상 평점을 계산할 때 의미가 없는 사용자이다.

이 평점 유사도 수식은 R프로그래밍 언어를 통해 구현한다. 또한 3.2.1 절의 유의미한 사용자 추출과 유사하게 n개의 노드에서 맴리듀스 패러다임으로 분산처리 한다.

3.2.3 예상 평점 계산

비교 대상들의 평점과 3.2.2의 과정을 통해 구한 유사도를 가중치로 하여 현재 사용자 영화별 예상 평점을 계산한다.

$$\text{estimate rating} = av + \frac{(x_1 - av_1) \times s_1 + \cdots + (x_n - av_n) \times s_n}{s_1 + s_2 + \cdots + s_n}$$

$x_1, x_2, \cdots, x_n$은 비교 대상들의 현재 영화에 대한 평점을 의미, av는 현재 사용자의 영화에 대한 평균 평점을 의미, $av_1, av_2, \cdots, av_n$은 비교 대상들의 영화에 대한 평균 평점을 의미, $s_1, s_2, \cdots, s_n$은 비교 대상들과 현재 사용자의 평점 유사도를 의미한다.

[표 1]은 현재 사용자의 특정 영화에 대한 예상 평점 계산 예시이다. 현재 사용자의 평균 평점은 3.5라고 가정

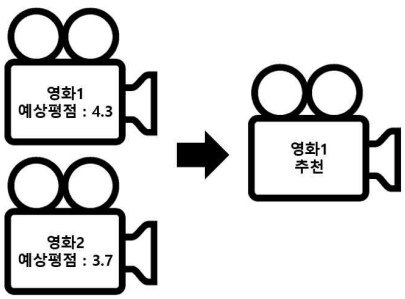
하면, 3.5 + {(5 - 2) * 0.7 + (3 - 3.5) * 0.5 + (5 - 3.7) * 0.5 + (2-2.5) * (-0.1) + (4 - 3.5) * 0.3} / (0.7 + 0.5 + 0.5 - 0.1 + 0.3) = 4.43이므로 4.4라는 예상 평점을 구할 수 있다.

[표 1] 예상 평점 계산 예시

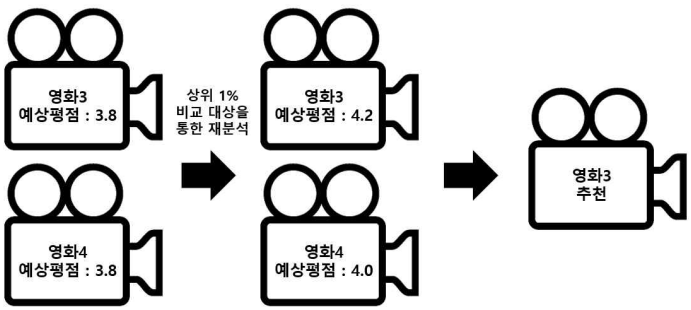
	대상1	대상2	대상3	대상4	대상5
평점	5	3	5	2	4
평균평점	2	3.5	3.7	2.5	3.5
유사도	0.7	0.5	0.5	-0.1	0.3
결과 값	2.1	-0.25	0.65	0.05	0.15

3.2.4 예상 평점 기반 영화 추천

3.2.3절의 수식들을 활용하여 사용자의 영화별 예상 평점을 구할 수 있다. [그림 2]와 같은 방식으로 예상 평점을 통해 데이터에 있는 17770개의 영화중에서 높은 평점이 나온 영화부터 추천한다. 평점이 동일한 영화의 경유 [그림 3]과 같은 방식으로 현재 사용자와 유사도가 높은 상위 1%(4800명) 비교 대상자의 영화 평점을 참고하여 우선순위를 정한다. 상위 1% 비교 대상자만을 통해 3.2.3와 같은 방식으로 예상 평점 계산한 후 더 높은 평점이 나온 영화를 먼저 추천한다.



[그림 2] 예상 평점에 따른 영화 추천



[그림 3] 예상 평점이 동일할 경우 영화 추천

3.3 빅데이터 분석을 통한 예상 평점과 영화 추천

사용자의 요청이 있는 경우 3.2의 과정을 통해 얻은 결과를 사용자에게 제공한다. 사용자에게 출력되는 정보는 다음과 같다.

사용자가 영화를 선택할 경우 3.2.2와 3.2.3의 과정을 통해 예상 평점을 계산한 후 사용자에게 제공한다.

사용자가 영화 추천을 원할 경우 3.2.2와 3.2.3의 과정을 통해 모든 영화에 대한 예측 평점을 계산한다. 이 결과 값을 바탕으로 3.2.4의 과정을 통해 추천 영화 순위를 결정하고 해당 영화들 중 상위에 랭크되는 영화들을 평점 순위대로 사용자에게 제공한다.

4. 결론

본 연구는 빅데이터를 기반으로 해서 사용자에게 영화를 추천해주고 영화의 예상평점을 제공하는 알고리즘을 설계하는데 목적이 있다. 제공된 빅데이터는 하둡을 통해 데이터 분석에 사용될 데이터를 선별한 후 프로그래밍 언어 R을 통해 선별된 데이터를 계산하고 예상 평점을 예측하여 사용자에게 제공되도록 한다.

본 영화 추천 알고리즘에서 구현을 통해 중점을 두고자 하는 부분은 사용자의 취향에 정확한 영화를 추천해주고 정확한 예상 평점을 제공하는데 있다.

* “본 연구는 미래창조과학부 및 정보통신기술진흥센터(IITP)에서 지원하는 서울어코드활성화지원사업의 연구결과로 수행되었음” (R0613-16-1203)

참 고 문 헌

[1] White Tom, Hadoop The Definitive Guide, Yahoo Press, 2012

[2] Garry Turkington, Hadoop: Data Processing and Modelling, Packt Publishing, 2016

[3] O’Reilly Media, Hands-On Programming with R: Write Your Own Functions and Simulations, O’Reilly Media, 2014

[4] Michael J. Crawley, Statistics: An Introduction Using R, John Wiley & Sons Inc, 2014

[5] http://www.netflixprize.com/rules.html