하둡에서 개인 성향을 이용한 영화 추천시스템

김선호, 김세준, 모하영, 김채린, 박규태, 박두순
순천향대학교 컴퓨터소프트웨어공학과
e-mail : sunho529@gmail.com

A Movie Recommender Systems using Personal Disposition in Hadoop

Sun-Ho Kim, Se-Jun Kim, Ha-Young Mo, Chae-Reen Kim, Gyu-Tae Park, Doo-Soon Park
Dept. of Computer Software Engineering, Soonchunhyang University

요 약

정보의 폭발적인 증가로 인해 사용자들은 오히려 원하는 정보를 빠른 시간에 얻는 것이 힘들어졌다. 따라서 이 문제를 해결하기 위한 다양한 방식의 새로운 서비스들이 제공되고 있다. 추천 시스템 중에서 영화를 추천해주는 방법에는 사용되는 알고리즘에는 협업필터링 방법이 가장 성공한 알고리즘으로 사용되고 있다. 협업 필터링 방법은 사용자가 자발적으로 입력한 선호도 평가치를 바탕으로 추천 하고자 하는 사용자와 취향이 비슷하다고 판단되는 사람들 즉, 최근접 이웃을 구하고 최근접 이웃의 선호도 평가치를 바탕으로 사용자에게 영화를 추천을 해주는 기법이다. 그러나 협업 필터링에는 몇 가지 대표적인 문제점이 있으며 희박성 및 확장성, 투명성이 있다.

본 논문에서는 영화 추천 시스템에서의 협업필터링의 희박성 문제를 보완하고자 개개인의 성향을 반영하여 효율이 좋은 추천 방법을 제안하고 하둡에서 성능평가를 하였다.

1. 서론

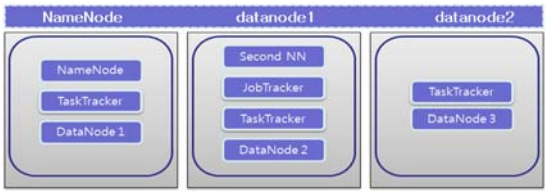
정보의 홍수 속에서 사용자들은 정보가 너무 많기에 자신이 원하는 정보를 찾거나 상품을 선택하여 구매하고자 하는 데에 어려움을 겪고 있다. 서비스 제공자는 이러한 문제점을 해결하고자 노력해 왔다. 수많은 콘텐츠와 서비스가 범람하는 현대에 개인화 서비스 중에서 사용자의 원하는 것을 제공 받는 것이 점차 중요하게 부각됨에 따라 사용자가 선호할 것이라고 판단되는 상품이나 서비스를 미리 판단하여 사용자에게 적절하게 제공해 주는 추천 시스템의 중요성 또한 크게 부각되고 있다. 추천 시스템의 활용사례로서는 Amazon에서 시작하여, 이후 현재까지 수많은 업체들이 개인화 서비스를 제공하고 있으며, 최근까지 다양한 방법의 시스템이 개발되었다.

이들 중에서 협업 필터링(Collaborative Filtering)은 여러 방법 중에서 가장 성공적인 방법으로 알려져 있으며, 이를 활용한 웹페이지, 영화, 논문, 신문기사 추천 등의 다양한 적용사례를 가지고 있다[1]. 협업 필터링을 이용한 추천 시스템에서 사용자들은 이용항목들에 대한 선호도 평가 점수를 부여한다. 그러면 시스템은 사용자들의 평가 점수를 이용해서 선호도에 따른 사용자들 간의 유사도를 구하고 특정 사용자와 유사한 개인성향을 가진 다른 사용자들의 평가 정보를 바탕으로 아직 평가하지 않은 항목들에 대한 평가 점수를 예측한다[2].

본 논문에서는 영화 추천 시스템에 주로 사용되는 협업 필터링의 문제점 중에서 희박성 문제를 보완 및 해결하기 위한 방법으로 개인 성향을 이용하는 방법 중 하나인 방법인 인구통계학적 데이터에서 개인 성향을 파악하고 이를 협업필터링의 최근접 이웃을 구성하는 입력 데이터로 사용한다. 일반적으로 개인 성향들은 요소가 많으면 많을수록 개개인의 특징을 더 잘 알 수 있기에 알 수 있는 모든 개인 성향 데이터를 이용하는 것이 최적의 성능을 나타낸다고 할 수 있으나 이를 최적화 하여 최소한의 개인 성향을 이용하여 최대의 효과를 나타내는 방법을 제안하고 제안된 방법을 구현한다. 현재 빅데이터 처리에서 가장 화두로 떠오르고 있는 병렬처리 시스템인 하둡 시스템을 구축하고 적용하여 성능평가를 실시한다.

2. 하둡에서의 개인화 영화 추천 시스템

[그림 1]은 하둡 시스템의 구성을 나타낸다.



[그림 1] 하둡 구성 환경

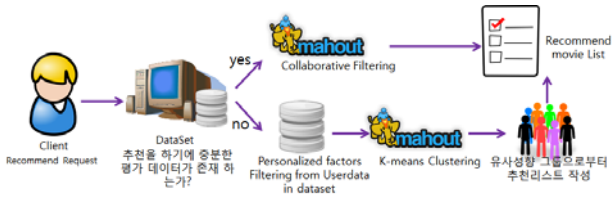
하둡 시스템은 일반적으로 홀 수로 구성하게 되며 가장 기본적인 구성인 3개의 구성이라 할 수 있다. 네임노드는 네임노드 역할과 더불어 데이터 노드의 역할을 담당하며 데이터노드1에서 세컨드 네임노드와 잡 트래커를 맡게 된다. 네임노드와 잡 트래케너는 서로 다른 곳에서 운영하는 것이 일반적인 운영 방법이다. 이는 두가지 주요한 데몬을 하나의 서버에 존재하게 하여 가중되는 부담을 분산하고자 하는 목적을 가진다. 마지막으로 데이터 노드에는 현재 별도의 구성없이 순수한 데이터 노드로서의 역할을 담당하고 있다. 하둡 시스템에서 데이터는 일반적으로 3copy 본을 가지게 됨으로 네임노드 또한 데이터 노드로서의 역할을 같이 하게 된다. [표1]은 하둡 시스템의 구성환경을 보여준다.

[표 1] 하둡 시스템 구성 환경

시스템 구성요소	시스템 세부 내용
운영체제	Cent OS
Java	1.7.0_04
Hadoop	1.0.3
Maven	3.0.5
Mahout	4.0.0

하둡상에서 데이터마이닝 알고리즘을 맵 리듀스로 작동하도록 제공하는 오픈소스인 머하웃을 작동시키기 위해서는 자바 가상머신 이외에 별도의 가상 머신인 메이븐을 필요로 한다.

하둡에서 개인화 영화 추천 시스템의 구성도는 [그림 2]와 같으며 데이터 마이닝 연산은 하둡에서 병렬로 처리되는 데이터마이닝 오픈소스인 머하웃을 이용하여 연산을 처리한다.



[그림 2] 하둡에서 개인화 영화 추천 시스템 구성도

추천을 위한 충분한 평가치가 존재한다면 머하웃에서 제공하는 협업필터링 방식을 이용하여 추천 리스트를 구성하고 그렇지 못한 경우 개인화 요인을 이용 제안한 최적의 개인화 요인들을 데이터로 하여 k평균 군집화를 사용하여 최근접 이웃을 구성하고 추천 영화 리스트를 작성하여 사용자에게 제공하게 된다. 개인화 요인을 바탕으로 한 추천 리스트는 [그림 3]과 같다.


영화 추천 결과

"Saint of Fort Washington, The (1993)"
Someone Else's America (1995)
"Wrong Trousers, The (1993)"
"Close Shave, A (1995)"
Casablanca (1942)
Maya Lin: A Strong Clear Vision (1994)
Wallace &Gromit: The Best of Aardman Animation (1996)
Rear Window (1954)
12 Angry Men (1957)
One Flew Over the Cuckoo's Nest (1975)

[그림 3] 하둡에서 구현한 추천 리스트

[그림 3]을 가지고 기존의 추천[3] 결과와 비교하면 동일하게 추천된 영화가 5개로 50%의 적중률 밖에 되지 않는 것을 볼 수 있다. 어떠한 차이점이 이러한 저조한 적중률을 가지는 두 결과를 가지고 왔는지 알아보기 위하여 k-평균 군집분석에서 입력으로 주어지는 k값을 변화시켜 보았으며 또한 임계치를 수정하여 반복적인 실험을 하였다. 실험 중에서 적중률이 비교적 높은 4가지에 방법에 대한 결과는 [표 2]와 같다.

[표 2] 적중률이 비교적 높은 4가지 결과

임계치 0.001 군집군 개수 k=7

Prefontaine (1997)
"Saint of Fort Washington, The (1993)"
Everest (1998)
Casablanca (1942)
"Wrong Trousers, The (1993)"
"Close Shave, A (1995)"
Maya Lin: A Strong Clear Vision (1994)
Wallace & Gromit: The Best of Aardman Animation (1 996)
Rear Window (1954)
Citizen Kane (1941)

임계치 0.001 군집군 개수 k=6

"Saint of Fort Washington, The (1993)"
Maya Lin: A Strong Clear Vision (1994)
"Close Shave, A (1995)"
"Wrong Trousers, The (1993)"
Casablanca (1942)
Rear Window (1954)
Wallace & Gromit: The Best of Aardman Animation (1996)
Dr. Strangelove or: How I Learned to Stop Worrying and Lo ve the Bomb (1963)
"Manchurian Candidate, The (1962)"
Citizen Kane (1941)

임계치 0.001 군집군 개수 k=8

"Saint of Fort Washington, The (1993)"
Maya Lin: A Strong Clear Vision (1994)
"Wrong Trousers, The (1993)"
"Close Shave, A (1995)"
Casablanca (1942)
Rear Window (1954)
Wallace & Gromit: The Best of Aardman Animation (1996)
Dr. Strangelove or: How I Learned to Stop Worrying and Lo ve the Bomb (1963)
"Manchurian Candidate, The (1962)"
Citizen Kane (1941)

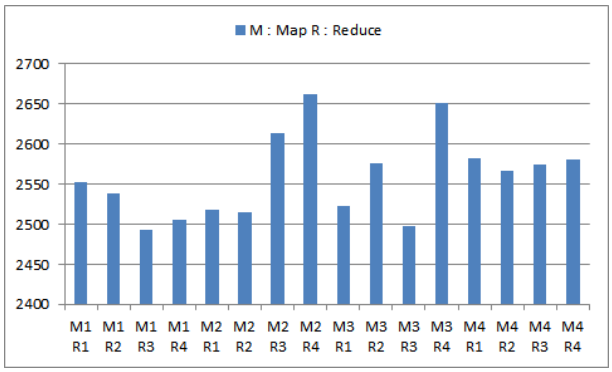
임계치 0.01 군집군 개수 k=6

"Saint of Fort Washington, The (1993)"
Maya Lin: A Strong Clear Vision (1994)
"Close Shave, A (1995)"
"Wrong Trousers, The (1993)"
Casablanca (1942)
Rear Window (1954)
Wallace & Gromit: The Best of Aardman Animation (1996)
Dr. Strangelove or: How I Learned to Stop Worrying and Lo ve the Bomb (1963)
"Manchurian Candidate, The (1962)"
Citizen Kane (1941)

[표 2]의 시계방향으로 결과를 살펴보면 첫 번째 결과는 10개중 6개가 적중한 60% 적중률을 보여 주었고 두 번째 결과를 보면 10개중 8개가 적중하여 80%의 적중률을 세 번째와 네 번째도 마찬가지의 8개의 적중을 보여 주었는데 결과를 보면 비교적 적중률이 높은 3가지는 임계치의 변화보다는 군집군의 개수에 따라 영향을 많이 받는 것을 알 수 있다. 따라서 하둡에서 구현한 것과 기존의 구현 결

과의 차이는 임계치의 차이라고 하기 보다는 군집군의 개수 의한 차이임을 확인 할 수 있다.

성능평가로는 하둡의 환경 설정에서 맵과 리듀스에 사용되는 cpu의 각각의 코어수를 변경 하는 것으로서 본 시스템의 최대 코어 개수인 4개를 기준으로 평가하였다. 일반적으로 맵과 리듀스에서 사용되는 코어의 개수가 많으면 많을수록 성능이 더 좋아진다고 할 수 있다. [그림 4]는 맵과 리듀스에서 사용되는 코어 수 변경에 따른 시간을 10회 측정하여 그 평균치들을 그래프로 나타낸 그림이다.



[그림 4] 하둡에서 맵 리듀스 코어 수 변경에 따른 시간 측정

[그림 4]를 보면 맵과 리듀스의 코어 수에 따라 시간이 다르게 측정되는 것을 볼 수 있는데 기존의 시스템은 맵과 리듀스에서 코어 1개만을 사용하는 첫 번째 그래프이고 이후 변경을 통해 측정한 결과이다. 맵과 리듀스를 4개까지 늘리면서 16개의 경우에 대해서 측정하였지만 일반적인 식을 이끌어 내지는 못하였다. 그것은 테스트한 데이터가 하둡에 잘맞지 않는 경우이거나 하둡에서 맵과 리듀스의 코어수가 훨씬 큰 경우에 결과가 잘 나타날 수 있다.

3. 결론

협업 필터링은 충분한 평가치가 존재하지 않을 때 추천의 만족도가 떨어지는 희박성 문제, 사용자가 늘어나면 늘어날수록 데이터의 연산이 기하급수적으로 늘어나는 확장성 문제 등을 가지고 있다. 희박성 문제를 보완하고자 본 논문에서는 개인의 성향을 반영하여 MovieLens 데이터를 이용, 최적의 개인화 요인을 찾고 이를 바탕으로 평가치 데이터가 충분하지 못할 경우 개인화 요인의 데이터를 이용, k-평균 군집화 방법을 이용하여 구하고 이를 바탕으로 영화를 추천하는 방식을 제안한다.

하둡에서는 알고리즘이 하둡에서 처리되는 과정인 맵/리듀스 과정에 맞게 변형 되어야 하는데 데이터 마이닝에서 주로 사용하는 알고리즘 들을 맵/리듀스로 변경하여 제공하는 오픈소스인 머하웃을 이용하였다. 하둡상에서 추천된 결과와 기존 시스템에서 추천된 결과에서 초기에는 적중률이 높지 못한 문제점이 있었으나 k값을 변경하는 여러 실험을 통해 적중률을 80%까지 높일 수 있었다. 이 결과를 바탕으로 영화 추천을 통한 교류 활성화에 기여할 수 있을 것이다. 또한 성능평가로서 코어수의 변화를 통하여 자료의 특성과 양에 맞추어 코어수를 조절하는 것이 성능에 영향을 미치는 것을 확인하였다. 성능평가를 통해 일반적인 식을 구하지는 못하였는데 이는 그것은 테스트한 데이터가 하둡에 잘 맞지 않는 경우이거나 하둡에서 맵과 리듀스의 코어수가 훨씬 큰 경우에 결과가 잘 나타날 수 있다. 향후에는 이러한 면을 고려해서 연구가 진행되어야 할 것이다.

참고문헌

[1] P. Resnick, N. lacovou, M. Sushak, P. Bergstrom, and J. Riedl, “GroupLens: an open architecture for collaborative filtering of Netnews.” In Proceedings of Computer Supported Cooperative Work Conference, 2001, pp.175-186

[2] Woon-hae Jeong, Se-jun Kim, Doo-soon Park and Jin Kwak, “Performance Improvement of a Movie Recommendation System based on Personal Propensity and Secure Collaborative Filtering”, Journal of Information Processing Systems Volume 9, Number 1, 2013

[3] 김세준, 박두순, 홍민, “하둡을 이용한 개인화 영화 추천 시스템”, 제40회 한국정보처리학회 추계학술발표대회 논문집 제20권 제2호, 2013