The Journal of IIBC. Vol. 14, No. 1, pp.101-107, Feb. 28, 2014. pISSN 2288-0879, eISSN 2288-2618

http://dx.doi.org/10.7236/JIIBC.2014.14.1.101
JIIBC 2014-1-13

클러스터링 기반 협업 필터링 알고리즘을 사용한 분산 추천 시스템

Distributed Recommendation System Using Clustering-based Collaborative Filtering Algorithm

조현제*, 이필규**

Hyun-Je Jo*, Phill-Kyu Rhee**

요 약 본 논문에서는 협업 필터링 알고리즘을 클러스터링 기반으로 분산 환경에서 구현하여, 추천을 위한 수행 시간을 최적화 하는 방법에 대한 제안을 한다. 하둡 기반으로 시스템을 구성하였고, 분산 Min-hash 클러스터링 기반의 협업 필터링 방법을 제안하고, 이를 기반으로 분산 추천 시스템을 구성하였다. 분산 사용자 기반 협업 필터링 기법을 사용하여 무비렌즈 (Movie Lens)의 영화 평점 데이터를 기반으로 각각의 사용자에게 알맞은 영화를 추천해주는 분산 추천 시스템을 구현하고 실험을 통하여 성능의 우수성을 검증하였다.

Abstract This paper presents an efficient distributed recommendation system using clustering collaborative filtering algorithm in distributed computing environments. The system was built based on Hadoop distributed computing platform, where distributed Min-hash clustering algorithm is combined with user based collaborative filtering algorithm to optimize recommendation performance. Experiments using Movie Lens benchmark data show that the proposed system can reduce the execution time for recommendation compare to sequential system.

Key Words : Recommendation System, Collaborative Filtering, Min-hash Clustering, Hadoop

Ⅰ. 서론

요즘은 데이터의 시대라고 할 수 있다. 수많은 양의 텍스트, 사진, 동영상 등 정말 많은 데이터가 웹상에 존재하고 쉽게 접근 가능하다. 뿐만 아니라 그러한 데이터를 이용하거나, 클릭하여 감상한 경우에도 그 사실 자체가 또 다른 데이터가 되어 존재하게 된다. 때문에 인터넷을 이용하는 사람들은 인터넷에 존재하는 모든 데이터를 일일이 찾아서 검토할 수는 없기 때문에 각각의 사용자에 맞는, 개인화된 데이터를 제공받아야 할 필요성이 있다.

협업 필터링 (collaborative filtering) 은 데이터를 사용자의 속성에 맞게 추천 해 주는 알고리즘 중 가장 널리 쓰이는 알고리즘 중 하나이다[3]. 협업 필터링은, 비슷한 행동을 한 둘 이상의 사용자들은 수행 할 다음 행동 역시 비슷할 것이라는 전제에서 시작한다. 이 전제를 바탕으로 기존 사용자들의 행동을 분석하여 새로운 사용자에게 아이템을 추천해 주거나, 새로운 물품을 구매하고자 하는 사용자에게 그 사용자가 원할만한 아이템을 추천해 준다.

본 논문에서는 협업 필터링 알고리즘을 클러스터링

*준회원, 인하대학교 컴퓨터정보공학과
**정회원, 인하대학교 컴퓨터정보공학과
접수일자 2014년 1월 17일, 수정완료 2014년 2월 3일
게재확정일자 2014년 2월 7일

Received: 17 January, 2014 / Revised: 3 February, 2014
Accepted: 7 February, 2014
**Corresponding Author: pkrhee@inha.ac.kr
Dept. of computer engineering, Inha University, Korea

기반으로 분산 환경에서 구현하여서, 추천의 효과를 최적화 하는 방법에 대한 제안을 한다. 하둡(Hadoop)[6] 기반으로 시스템을 구성하였고, 분산 환경에서 추천 성능의 최적화를 위하여 분산병렬 Min-hash 클러스터링 기법을 사용하였다[5]. 그리고 여러 종류의 협업 필터링 기법 중에서 분산 환경에 가장 적합한 사용자 기반 협업 필터링 (user-based collaborative filtering) 기법을 사용하여 무비렌즈 (Movie Lens) 의 영화 평점 데이터[7]를 기반으로 각각의 사용자에게 알맞은 영화를 추천해주는 알고리즘을 구현하고 실험을 통하여 성능의 우수성을 검증하였다.

II. 관련 연구

1. 추천 시스템

Paul Renick은 추천 시스템을 "데이터를 시스템이 통합하여 적절하게 사용자에게 보여 주는 것" 이라 정의하였다[1]. 다시 말하면, 추천 시스템은 시스템이 가지고 있는 데이터를 사용자의 특성에 맞게, 그 사용자가 좋아할 만한 내용을 각각의 사용자에게 맞게 개인화하여 제공하는 것을 말한다. 이러한 추천 시스템은 Netflix의 영화 추천 [12], Google의 뉴스 추천[13], YouTube의 비디오 추천 [14] 등 다양한 분야에서 사용자에게 좀 더 양질의 컨텐츠를 제공하기 위해 사용된다.

이러한 추천 시스템이 갖추어져 있다면, 컨텐츠를 제공하는 공급자와, 컨텐츠를 소비하는 사용자 모두 좋은 결과를 얻게 된다. 컨텐츠를 소비하는 사용자 입장에서는 관심이 있을 가능성이 높은 컨텐츠를 추천받기 때문에 수많은 정보를 모두 검토할 필요가 없고, 컨텐츠를 제공하는 공급자 입장에서는 사용자 개개인에 대한 맞춤 컨텐츠를 제공하게 됨으로써 더욱 높은 수익을 창출할 수 있다.

가장 먼저 시작된 추천 알고리즘은 단순히 가장 인기가 많은 컨텐츠를 사용자에게 추천해주는 것이었다. 하지만 이러한 방식은 사용자의 정보를 전혀 고려하지 않기 때문에 개인화된 추천이라고 할 수 없었다. 이후, 추천의 성능을 개선하고자 많은 연구가 진행되었는데 최근 가장 많이 사용되는 기술은 협업 필터링이다.

2. 협업 필터링

협업 필터링에는 몇 가지 접근 방식이 있다. 첫 번째는 기존에 사람들이 아이템에 대한 평가를 기준으로 추천을 해 주는 Memory-based 방식, 두 번째는 데이터를 특정 확률적 모델에 적용하여 아이템을 추천해 주는 Model-based 방식, 마지막 세 번째는 사용자가 아이템에 대해 평가한 데이터뿐만이 아니라 다른 정보도 같이 활용하는 Hybrid 방식이 있다.

Memory-based 협업 필터링에는 비슷한 행동을 한 사용자를 클러스터링하여 목표 사용자가 속하는 군집에서 다른 사람들이 높은 점수의 평가를 부여한 아이템을 추천 해 주는 사용자 기반의 협업 필터링이 있으며, 사용자 기반이 아니라 아이템 사이의 연관성을 파악하여 비슷한 아이템의 군집을 생성하고, 그 군집 안에서 목표 사용자의 행동 기록을 기반으로 다른 아이템을 추천해 주는 아이템 기반 협업 필터링 기법이 있다.[3]

Model-based 협업 필터링은 데이터를 확률적 모델 등에 적용하여 추천을 수행하는 방식인데, 주로 활용되는 기법으로는 Bayesian-belief Network 기반의 협업 필터링, 클러스터링 기반의 협업 필터링, Markov Decision Process 기반의 협업 필터링 등이 존재한다.

Hybrid 방식은 앞서 설명한 Memory-based 방식과, Model-based 방식을 혼합하여 구성하거나, 혹은[2]와 같이 다른 알고리즘을 결합한 경우를 일컫는다.[3] 표 1 은 협업 필터링의 종류를 요약하였다.

표 1. 협업 필터링 알고리즘의 종류[3]
Table 1. Types of collaborative filtering algorithms

종류	구현 기법
Memory-based CF	Neighbor-based Item/User-based
Model-based CF	Bayesian belief nets CF Clustering CF Regression-based CF MDP-Based CF Latent CF Dimensionality reduction
Hybrid Recommender	Content-based CF Combining algorithms

3. 맵리듀스

맵리듀스 (MapReduce)는 아파치 하둡 (Apache Hadoop) 기반에서 동작하는 프로그램들의 알고리즘 수

행 로직 스타일을 일컫는다[4]. 맵리듀스는 Map과 Reduce라는 두 개의 과정으로 이루어져 있다. Map 과정에서는 데이터를 병렬로 읽고, Reduce에서는 그 결과를 하나로 합치는 과정이다. Map과 Reduce 과정은 각각 Mapper와 Reducer 객체가 수행을 하고, Mapper와 Reducer의 입력과 출력은 <Key, Value> 형태로 이루어져 있다.

가장 먼저 Mapper에서는 파일을 읽을 때, <읽은 줄 수, 텍스트> 형태로 읽는다. 그리고 읽은 텍스트 값을 이용하여 사용자가 정의한 Map 로직을 수행하고, 그 결과를 <Key, Value> 형태로 변환하여 출력한다. Mapper에서 출력된 <Key, Value> 쌍은 Reducer로 전달되기 전에 Shuffle 이라는 과정을 통과한다. Shuffle 과정에서는 Mapper에서 출력된 모든 <Key, Value> 쌍들이 같은 Key끼리 정렬되고, Key별로 섞인다. Mapper에서 한번 Map함수를 수행하는 단위가 입력 텍스트의 한 줄 단위였다면, Reducer에서는 한 개의 정렬된 Key에 대한 Value의 리스트 형태이다. 따라서 Reducer의 입력 형태는 <Key, List<Value>> 형태가 된다. 그리고 Reducer에서는 Mapper에서와 마찬가지로 사용자가 정의한 Reduce 함수를 수행하고, 그 결과를 출력함으로써 전체적인 맵리듀스 과정이 종료된다.

하지만, 모든 과정이 Mapper 한 번과 Reducer 한 번으로 끝나는 것이 아니다. 알고리즘의 형태, 구현 방식에 따라 다양한 형태의 맵리듀스 로직이 구성될 수 있다.

4. Min-hash 클러스터링

Min-hash 클러스터링은 간단한 Hash 함수를 이용한 확률 기반의 클러스터링 기법이다.[5]

$$h(x) = ax + b \pmod{k} \qquad (1)$$

와 같이 간단한 형태의 Hash 함수를 사용하기 때문에 속도가 빠르며, 여러 반복 횟수를 필요로 하는 K-means 클러스터링이나, K-NN 클러스터링 기법보다 수행 시간이 매우 짧다는 장점이 있다. 사용자가 어떤 상품을 구매한 기록을 이용해 Min-hash 클러스터링을 수행한다고 가정하면 다음과 같은 과정으로 진행된다.

Min-hash 클러스터링은 User가 구매한 Item들의 id를 각각의 hash 함수에 대입하고, 그 결과 중 가장 작은 값을 선택하면서 수행된다. 그리고 그 가장 작은 값을 signature value 라 하며, 여러 개의 signature value가 그 사용자가 속한 클러스터의 ID가 된다. Min-hash 클러스터링에는 몇 개의 파라미터가 존재한다. 위 식 1에 등장하는 변수 a, b, k와, Hash 함수 그룹의 개수를 나타내는 q, 한 개의 Hash 함수 그룹에 존재하는 Hash 함수의 개수인 p이다.

한 User가 아이템을 구매 한 로그에 대해서, 각각의 Hash 함수 그룹별로 Hashing을 수행한다. 예를 들어 한 Hash 함수 그룹에 두 개의 Hash 함수가 존재하고, 각 Hash 함수가

$$h_1(x) = 4x + 5 \pmod{17}$$
$$h_2(x) = 3x + 7 \pmod{11}$$

일 때, 사용자가 구매한 아이템의 목록

User: 6, 7, 13, 15, 10, 3

에 대해서 Hash를 수행하게 된다.

함수 h_1의 결과로는 <12, 16, 6, 14, 11, 0>이 되고, 함수 h_2의 결과로는 <3, 6, 2, 8, 4, 5> 이 된다. 이 두 개의 결과 리스트 중에서 각각의 최솟값은 0과 2며, 그 두 최솟값을 고정길이 레코드로 변환(예: 00000002)하면 그 사

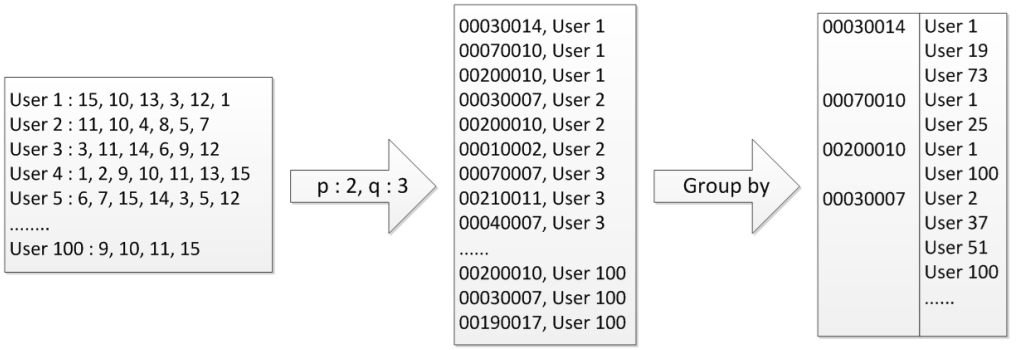


그림 1. Min-hash Clustering의 과정
Fig. 1. Process of Min-hash clustering

용자가 속하게 되는 클러스터가 00000002가 된다.

위 예제를 $p=2, q=3$인 경우로 확장하여 클러스터링을 수행해 보면 위 그림 1과 같다.

Ⅲ. 제안하는 방법

이 논문에서 제시하는 분산 추천 시스템의 수행 순서는 다음과 같다. 가장 먼저 하둡 기반의 데이터를 알맞은 형태로 정제하여 데이터를 불러오고, 그 데이터를 앞서 설명한 Min-hash 클러스터링 기법을 이용하여 사용자별로 분류한다. 그리고 각각의 분류된 클러스터에 존재하는 사용자들끼리의 유사도를 계산한다. 계산된 유사도를 바탕으로 Weighed-sum 알고리즘을 이용해 사용자의 선호도가 기록되지 않은 아이템에 대해 사용자의 선호도를 예측한다. 마지막으로, 그 결과를 종합함으로써 최종적으로 사용자에게 추천 될 아이템을 결정한다.

1. 병렬 Min-hash 클러스터링

앞서 설명한 Min-hash 클러스터링은 하둡 기반의 분산 환경에서 효율적으로 수행될 수 있다. 따라서 이 문서에서는 Min-hash 클러스터링 알고리즘을 수행할 때, 기존의 일반적인 로직 형태가 아니라 맵리듀스 기반의 로직 형태로 변경하여 하둡 기반에서 수행하였다. Min-hash 클러스터링은 하나의 Map 과정과 하나의 Reduce 과정으로 구성된다.

Map 과정에서는 입력 데이터에 대해 Hash를 수행한다. 그림 2는 Min-hash 클러스터링의 Map 과정에 대한 예시이다. 위 그림에서 Mapper의 입력 텍스트의 형태는 <1::1193 5,661 3,914 3>과 같은 형태이다. 이 입력 형태는 <사용자-아이디::아이템-아이디 선호도, 아이템-아이디 선호도> 형태이다. 만약 Hash 함수의 그룹의 개수가 5개이며, 각각의 그룹에 있는 Hash 함수의 개수가 3개라면, 그룹의 ID는 한 개의 데이터마다 5개가 생성 될 것이며, 각각 그룹의 ID는 3개의 각각의 Hash 함수로 해싱을 수행한 결과 중 가장 낮은 값들로 이루어진 조합일 것이다. 첫 번째 Hash 함수 그룹에 속해있는 Hash 함수가 각각

$$h_1 = 7x + 9 \pmod{39}$$
$$h_2 = 5x + 17 \pmod{53}$$
$$h_3 = 13x + 3 \pmod{67}$$

라면, 첫 번째 Hash 함수 그룹에 의해 생성된 그룹 ID는 첫 데이터 <1::1193 5,661 3,914 3>에 대해서는 <112917>이라는 그룹 ID가 생성된다. 이와 같은 작업을 모든 Hash 함수 그룹과 모든 데이터에 대해 수행한다. 각각의 데이터에 대해서 Min-hash Clustering을 수행한 결과를 Reducer로 보낸다.

Reduce 과정에서는 그림 3에서와 같이 앞서 Mapper에서 진행된 Min-hash 결과를 클러스터링 하는 역할을 수행한다. Reduce 과정 이전에 앞서 언급하였던 것과 같이 Shuffle 과정에서 모든 Min-hash 결과가 Key(그룹 ID)별로 정렬된다. Reducer에서는 각각의 그룹 ID에 속한 데이터들의 대푯값을 정하여, 그 대푯값을 해당 클러스터의 대푯값으로 설정한다.

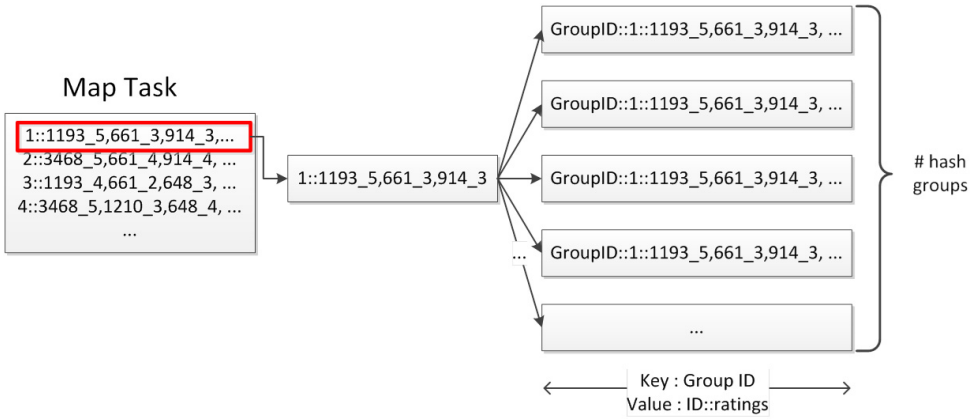


그림 2. Min-hash 클러스터링 Map 과정
Fig. 2. Map process of Min-hash clustering

2. 유사도 계산 알고리즘

사용자와 사용자 사이의 유사도를 계산할 때는 Pearson-Correlation을 사용하며, 유사도를 판단하고자 하는 두 대상을 그래프로 표현하였을 때, 얼마나 직선 형태로 표현 할 수 있는가에 대한 정도를 나타낸다. 사용자 u와 사용자 v 사이의 유사도 $W^c_{u,v}$는 다음 수식을 이용하여 계산한다.

$$W^c_{u,v} = \frac{\sum_{i \in I}(r^c_{u,i} - \overline{r^c_u})(r^c_{v,i} - \overline{r^c_v})}{\sqrt{\sum_{i \in I}(r^c_{u,i} - \overline{r^c_u})^2}\sqrt{\sum_{i \in I}(r^c_{v,i} - \overline{r^c_v})^2}} \quad (2)$$

여기서 c는 병렬 Min-hash를 통해 생성된 클러스터를 나타낸다.

I는 아이템, $r^c_{u,i}$는 클러스터 c에 속한 사용자 u의 아이템 i에 대한 평점, $\overline{r^c_u}$는 클러스터 c에 속한 사용자 u의 모든 아이템에 대한 평점의 평균이다. Pearson-Correlation 유사도 대신에, Vector-cosine 유사도, Jaccard-Coefficient 유사도 등을 사용 할 수 있다.

3. 사용자 선호도 예측 알고리즘

사용자와 사용자 사이의 유사도를 이용하여 사용자가 선호도를 매기지 않은 아이템에 대해 사용자가 얼마만큼의 선호도를 가질지에 대해 예측한다. 이 과정에서 Weighted Sum기법을 활용한다.

클러스터 c에 속하는 사용자 a의 아이템 i에 대한 선호도 예측 값 $P^c_{a,i}$는, 클러스터 c에 속하는 사용자 u가 아이템 i에 평가한 선호도 $r^c_{u,i}$를 이용하여 다음과 같이 정의된다.

$$P^c_{a,i} = \overline{r^c_a} + \frac{\sum_{u \in U}(r^c_{u,i} - \overline{r^c_u}) \cdot W^c_{a,u}}{\sum_{u \in U}|W^c_{a,u}|} \quad (3)$$

Ⅳ. 실험 결과 몇 결론

본 논문에서 작성된 맵리듀스 알고리즘은 아파치 하둡 1.2.1 기반으로 작성되었으며, CPU 2vCore, RAM 4GB 성능의 KT UCloud의 가상 인스턴스 3개를 클러스터로 형성하여 실험하였다. 본 논문에서는 사용자의 수를 증가시키면서 제안한 알고리즘의 수행시간 변화를 측정하였다.

본 실험에서 제안한 분산 추천 알고리즘에 의하여, 사용자의 수가 증가하더라도 사용자 한 명당 추천을 제공하는데 소요된 시간이 선형적으로 증가하지 않음을 확인할 수 있어서 제안하는 방법의 효용성을 검증하였다.

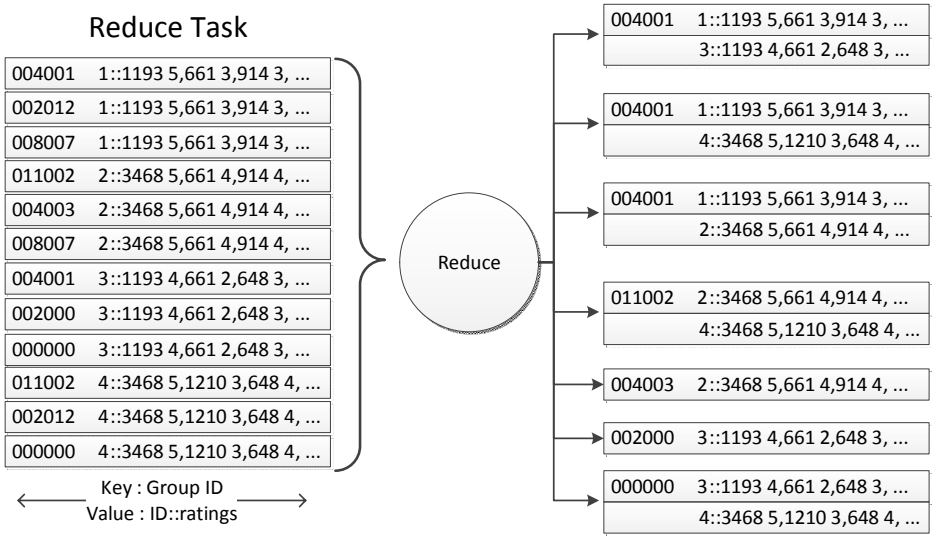


그림 3. Min-hash 클러스터링의 Reduce 과정
Fig. 3. Reduce process of Min-hash clustering

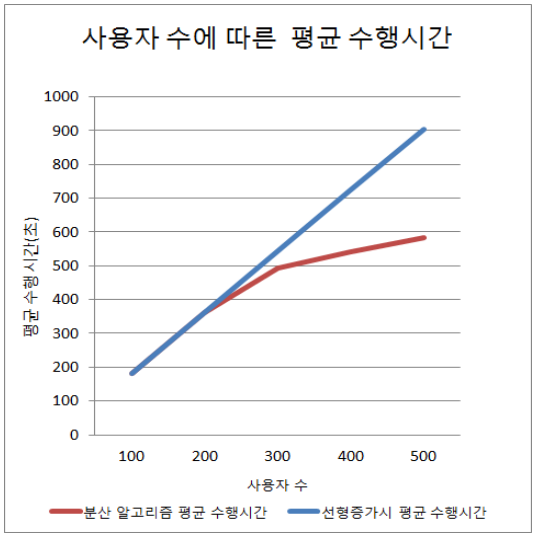


그림 4. 제안된 분산 협업적 필터링 기반의 추천 시스템 수행 시간에 대한 성능 분석

Fig. 4. The performance analysis in terms of execution time for the proposed distributed recommendation system based on distributed collaborative filtering algorithm

References

[1] Paul Renick and Hal R. Varian, "Recommender System," Communications of the ACM" Vol 40, No.3, March. 1997

[2] Yan Shen, Hak-Chul Shin, "Reinforcement Learning Algorithm Based Hybrid Filtering Image Recommender System",The Journal of The Institute of Internet, Broadcasting and Communication, VOl. 12 No. 3, June 2012

[3] Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. Advances in Artificial Intelligence, 2009(12)

[4] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1)

[5] Das, A. S., Datar, M., Garg, A., & Rajaram, S. (2007). Google news personalization: scalable online collaborative filtering, 271-280.

[6] http://hadoop.apache.org/

[7] http://grouplens.org/datasets/movielens/

[8] Gong, S. (2010). A collaborative filtering recommendation algorithm based on user clustering and item clustering. Journal of Software, 5(7), 745?752. doi:10.4304/jsw.5.7.745-752

[9] G Smith Linden B.; York, J. (n.d.). Amazon.com Recommendations: Item-to-item Collaborative Filtering.

[10] Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., et al. (2010). The YouTube video recommendation system, 293-296. doi:10.1145/1787275.1787324

[11] Karypis, G. "Evaluation of item-based top-n recommendation algorithms", in Proceedings of the International Conferrence on Information and Knowledge Management (CIKM'01), pp.247-254, Atlanta, Ga, USA, November 2001.

[12] http://www.netflix.com/

[13] https://news.google.com/

[14] http://www.youtube.com/

[15] H. Lee, J. Kwon, "A New Distributed Graph Data Storage System for Large-Scale Recommender Engines", Journal of Korean Institute of Information Technology, Vol. 11, No. 7, pp. 139-149, July 31, 2013.

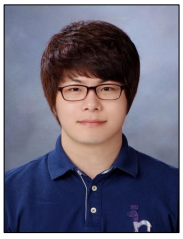
[16] Seok-Jong Yu, "Comprehensive Temporal Filter for Expanded Collaborative Filtering Algorithm", Journal of Korean Institute of Information Technology, Vol. 11, No. 11, pp. 173-179, Nov. 30, 2013.

[17] Kitae Hwang, "Genre-based Collaborative Filtering Movie Recommendation",The Journal of The Institute of Internet, Broadcasting and Communication, VOl. 10, No. 3, June 2010

『이 논문은 인하대학교의 지원에 의하여 연구되었음.』
『This work was supported by INHA UNIVERSITY Research Grant.』

저자 소개

조 현 제(준회원)



- 2013년 2월 : 인하대학교 컴퓨터정보공학과 졸업(학사)
- 2013년 3월 ~ 현재 : 인하대학교 컴퓨터정보공학과 (석사)

<관심분야: Big Data 컴퓨팅, 추천 시스템, Machine Learning>

이 필 규(정회원)

- 1982년 2월 : 서울대학교(학사)/전기공학
- 1986년 8월 : East Texas State Univ.(석사)/전산학
- 1990년 12월 : Univ. of Louisiana (박사)/전산학
- 현재 : 인하대학교 컴퓨터정보공학과 정교수

<관심분야: 컴퓨터 비전, 패턴 인식, Big Data 컴퓨팅, Machine Learning>