Spark GraphX를 활용한 개인 추천 시스템 개발

김성숙*, 박기진**, Sun Lu***
*한국종합공해시험연구소
**아주대학교 공과대학 융합시스템공학과
***아주대학교 일반대학원 산업공학과
e-mail: sungsook87@gmail.com, {kiejin, sunlu1124}@ajou.ac.kr

A Development of Personalized Recommendation System using Spark GraphX

Sungsook Kim*, Kiejin Park**, Sun Lu***
*Korea Air Pollution Synthesis Research Center
**Dept. of Integrative Systems Engineering, College of Engineering, Ajou University
***Dept. of Industrial Engineering, Graduate School, Ajou University

요 약

소셜 데이터는 인터넷 상의 수 많은 개인과 개인의 상호 작용에 의하여 연결되어 있으며, 이러한 데이터를 분석하여, 분석 대상에 내재하고 있는 구조와 특성을 파악하는 일은 중요하다. 특히, 개인 추천을 위해서는 개별 데이터들의 관계 그래프를 활용하여 빠르고 정확하게 추천 값을 도출하는 것이 효율적이다. 하지만, 기존 추천 기법으로는 신규 사용자와 아이템이 끊임없이 등장하는 상황을 즉각적으로 반영하기가 어렵고, 또한 많은 결측값을 포함하는 sparse 한 데이터일 경우에는 추천 시스템의 연산 공간과 시간에 많은 제약이 있다. 이에 본 논문에서는 Spark GraphX 를 활용한 개인 추천 시스템을 설계 및 개발하였으며, 이를 통하여 사용자와 아이템간에 내재하는 복합 요인이 반영된 그래프 기반 추천을 실행하여, 개인 추천 결과의 우수성을 확인하였다.

1. 서론

오늘날 SNS(Social Network Service)는 수 많은 개인과 개인의 상호 작용에 의하여 연결되어 있으며, 여기서 발생하는 소셜 데이터는 표면화(Explicit) 되지는 않지만 분석 대상에 내재(Implicit)하는 구조와 특성을 파악하는데 중요한 단서가 된다. 특히, 개인 추천을 위해서는 개별 데이터들의 관계 그래프를 활용하여 빠르고 정확하게 추천 값을 도출하는 것이 효율적이다[1].

하지만, 기존 추천 기법으로는 신규 사용자와 아이템이 끊임없이 등장하는 상황을 즉각적으로 반영하기가 어렵고, 또한 많은 결측값을 포함하는 sparse한 데이터일 경우에는 추천 시스템의 연산 공간과 시간에 많은 제약이 있다[2]. 이에 본 연구에서는 클러스터 컴퓨팅[3] 환경에 적합한 대용량 데이터 분산 저장 및 그래프 병렬 처리에 효율적인 Spark GraphX[4]를 활용하여, 개인 추천 시스템을 설계 및 개발하였다.

본 논문은 총 5개의 장으로 구성되어 있으며, 제2장에서는 개인 추천을 위한 분산 메모리 그래프 처리 기법을 기술했으며, 제3장에서는 Spark GraphX 기반 대용량 분산 그래프 처리 시스템 구조를 설명하였으며, 제4장에서는 개발된 시스템에 대한 성능 평가를 수행하였다. 마지막으로, 제5장에서는 결론과 후속 연구에 대해 언급하였다.

2. 분산 메모리 그래프 처리: GraphX

본 분산 메모리 그래프 처리에서는 기존 추천 시스템 모델에서 통용되는 '사용자' 는 Source Vertex로, '아이템' 은 Destination Vertex로, 사용자가 부여한 점수(Score)는 Edge 값으로

모델링하였다. 두 Vertex 간의 내재하는 구조를 파악하기 위해 그래프 분산 병렬 처리 알고리즘인 SVD++ 기법[5]을 적용하였으며, 추천값 예측 시 특정 사용자와 연관된 아이템에 내재하는 피드백[6]을 추가함으로써 정확도를 높이고자 하였다. 여기서 피드백이란 아이템 서로 간에 주고 받는 영향력을 의미한다. 또한, 그래프 표현 기법을 적용함으로써 대용량 데이터가 Sparse할 경우 효율적인 연산이 가능하도록 하였다.

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathrm{q_i^T}\left(\mathrm{p_u} + |\mathrm{N(u)}|^{-\frac{1}{2}} \sum_{\mathrm{j}\in\mathrm{N(u)}} \mathrm{y_j}\right) \qquad (1)$$

<식 1>의 사용자 u의 아이템 i에 대한 예측값(평점, $\hat{\mathrm{r}}_{\mathrm{ui}}$) 계산시에, 평점 평균값 μ , 사용자 편향치 $\mathrm{b_u}$,아이템 편향치 $\mathrm{b_i}$ 의 합인 $\mu + b_i + b_u$ 에 아이템의 잠재 특성(Latent Feature) $\mathrm{q_i}$ 와 사용자 잠재 특성 $\mathrm{p_u}$ 가 적용되며, 추가로 SVD++에서는 특정 사용자가 선택한 아이템 집합(N(u))에 속한 아이템들의 잠재 특성($\mathrm{y_j}$)이 별도로 반영된다.

3. Spark GraphX 기반 개인 추천 시스템

개인 추천을 위해서 분산 메모리 그래프 프로세싱이 가능하도록 Spark RDD[7] 타입 데이터를 Vertex와 Edge로 구성된 그래프 구조로 변환하여 처리하였다. <그림 1>은 입력 Vertex가 분할(Partition)된 후, 관련된 Vertex 간의 메시지 전달(Join)을 통하여 새로운 출력 Vertex로 변환되는 과정이다. 이와 같이, 모델링된 그래프의 변환을 통해 새로운 그래프를 생성할 경우, 특정 사용자의 예측값을 계산할 때, 출력 Vertex에 이미 필요한 계산이 포함되어 있으므로 효율적이다. 특히, GraphX에서는 반복적인 데이터 처리가 주를 이룰 경우에는 중간(Intermediate) 데이터를 메모리에 저장한 후 계속적으로 사용하기 때문에 더욱 효과적이다.

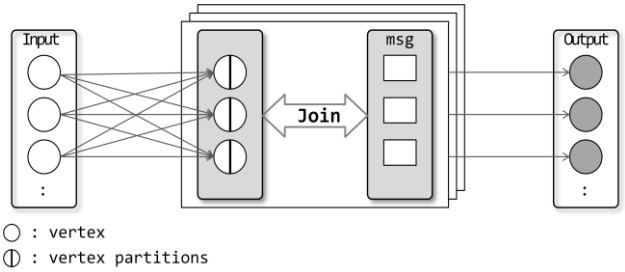

그림 1. Vertex 변환 과정

본 논문의 개인 추천 시스템 구현을 위해 1 대의 Master와 총 8 대의 Worker 노드로 구성된 Hadoop-Spark 기반 클러스터(그림 2 참조)에 Hadoop 3.0 과 Spark 2.3 분산 메모리 그래프 처리 플랫폼을 구축하였다. Master 노드의 Driver 메모리는 12 GB, 각 Worker 노드에서 동작하는 Executor 메모리는 64GB, Executor 당 코어는 4개로 GraphX 런타임 파라미터를 설정하였다. 한편 Spark를 활용한 그래프 질의 처리는 분산 메모리 상에서 작동하기 때문에 디스크에서 읽어오는 기존 대용량 데이터 처리 시스템들 보다 훨씬 빠르게 수행된다.

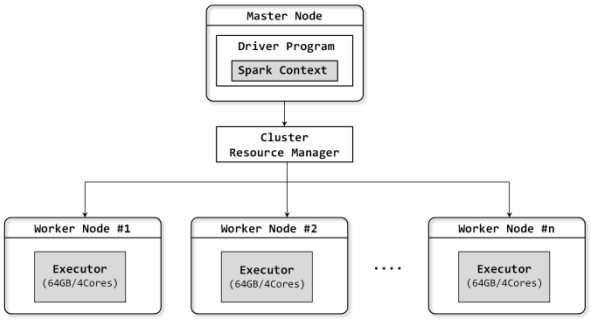

그림 2. 개인 추천 시스템 구조

4. 성능 평가

4.1 입력 데이터 셋

본 성능 평가 실험에서 사용한 SNS 데이터는 약 1 GB 크기인 뉴스 웹사이트 데이터이며, 사용자가 뉴스를 등록하면 다른 사용자들이 해당 글에 부여한 점수에 따라 뉴스의 순위가 변동하게 된다. 입력 데이터는 Key와 Value로 구성된 json 타입이며, 총 22개의 속성을 가지며, 이들 중 'author', 'parent_id', 'score' 항목 만을 뽑아 실험에 사용하였다. 'author' 항목은 Source Vertex, 'parent_id' 항목은 Destination Vertex, 그래프 Edge 값으로는 'score' 항목을 사용하였다. 실험 입력 데이터에서 Training을 위해 80%, Testing을 위해 20%의 데이터 셋을 할당했다.

4.2 결과 분석

성능 평가 척도로서 <식 2>의 평균 제곱근 오차(Root Mean Square Error)를 채택하였으며, RMSE는 추천 모델의 정확도를 표현하는데 적합하고, 수치가 작을수록 추천 성능이 좋게 된다.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{r}_i - r_i)^2} \qquad (2)$$

학습 모델에 필요한 매개변수들의 최적값을 찾기 위하여 반복적인 실험을 수행하였다. 예를 들면 학습 반복 회수 매개변수의 경우, 내재 요인 개수를 2로 고정하였을 때, <그림 3> 과 같이 RMSE 지표 향상이 미미해지는 지점(10회 부근)을 찾을 수 있다.

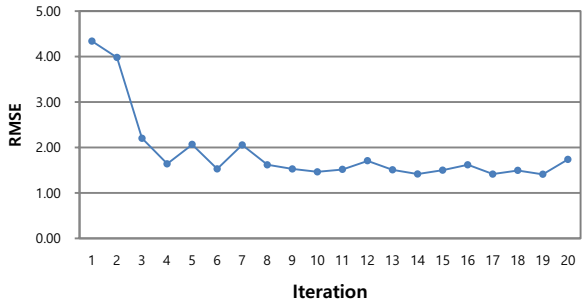

그림 3. 학습 반복 회수에 따른 RMSE 변화

잠재 특성(y_j) 차원 수(Rank)가 너무 많으면 과적합 문제가 발생하는 현상이 발생한다. 따라서, 해당 모델의 적절한 Rank 값을 찾는 것도 학습의 중요한 부분이라고 할 수 있다.

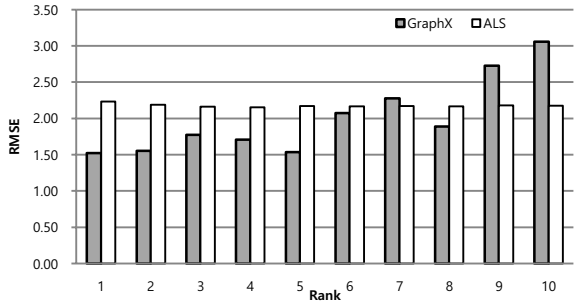

그림 4. 잠재 특성 차원(Rank)에 따른 RMSE 비교

<그림4>에서는 Rank가 5일 때 RMSE가 가장 좋았으며, 이를 통해 기존 ALS(Alternating Least Squares) 기법과 비교시 적절한 Rank 값을 사용할 경우 예측값에 대한 RMSE가 개선됨을 알 수 있다.

5. 결론 및 후속 연구

본 논문에서는 Spark GraphX를 활용한 개인 추천 시스템의 성능을 분석하였다. 결과로써 기존 추천 시스템에 사용되는 행렬기반 모델(Matrix Factorization) 보다 추천 성능이 더 우수했으며, 특히 대용량의 SNS 데이터에서 개인화 추천은 Sparse한 데이터가 많이 존재하기 때문에 해당 Vertex와 이와 관계된 데이터들의 처리만으로도 추천이 가능하게 되었다.

후속 연구로는 대량의 SNS 데이터 내에 존재하는 함축적인 상호 관계의 특성들을 정량화할 수 있는 방안들을 모색하고, 이를 추천 시스템에 반영할 수 있도록 할 예정이다.

참고 문헌

[1] K. Park, C. Baek and et. al., “A Development of Streaming Big Data Analysis System Using In-memory Cluster Computing Framework: Spark,” *LNEE*, Vol. 393, pp. 157-163, 2016.

[2] S. Kang and K. Park, “Improving Top-K Contents Recommendation Performance by Considering Bandwagon Effect: Using Hadoop-Spark Framework,” *LNEE*, Vol. 474, pp. 137-142, 2018.

[3] V. Vavilapalli, A. Murthy, and et. al., “Apache Hadoop YARN: Yet Another Resource Negotiator,” *Proc. of the 4th annual Symposium on Cloud Computing ACM*, pp. 5:1-5:16, 2013.

[4] J. Gonzalez, R. Xin, and et. al., “GraphX: Graph Processing in a Distributed Dataflow Framework,” *Proc. of the 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’14)*, pp. 599-613, 2014.

[5] Y. Koren, “Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model,” *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’08)*, pp. 426-434, 2008.

[6] Y. Hu, Y. Koren, and C. Volinsky, “Collaborative Filtering for Implicit Feedback Datasets,” *Proc. of the 8th IEEE International Conference on Data Mining*, pp. 263-272, 2008.

[7] M. Zaharia, M. Chowdhury, and et. al., “Resilient Distributed Datasets: A fault-tolerant abstraction for in-memory cluster computing,” *Proc. of the 9th USENIX conference on Networked Systems Design and Implementation*, 2012.