태그가 있는 소셜 미디어 데이터를 위한 맵리듀스 기반의 군집화 알고리즘

김세영○, 김도현, 이재환, 이세희, 정성원
서강대학교 컴퓨터공학과
xptr33@gmail.com, rla3864@naver.com, entaroid@naver.com, jaydenlee@sogang.ac.kr,
jungsung@sogang.ac.kr

A MapReduce-based Clustering Algorithm for Social Media Data with Tags

Saeyoung Kim○, Dohyun Kim, Jaehwan Lee, Sehee Lee, Sungwon Jung
Department of Computer Science and Engineering, Sogang University

요 약

페이스북, 인스타그램과 같은 소셜미디어 매체들의 사용량이 폭발적으로 증가함에 따라 텍스트, 이미지, 영상과 같은 복잡한 구조를 가지진 방대한 양의 데이터들이 생성되었으며 이를 분석하기 위한 기법에 많은 관심이 집중되고 있다. 하지만 기존 기법들은 소셜미디어 매체 데이터의 특성에 적합한 군집화 방법이 아니기 때문에 군집 결과를 도출하는 데에 있어 낮은 정확도와 느린 속도가 문제로 나타나고있다. 본 논문에서는 이를 해결하기 위해 벡터 태그 정보를 기반으로 하는 데이터들 간의 새로운 유사도 측정기법을 제안하여 정확도를 향상시키며 맵리듀스 환경을 통해 군집과정을 분산병렬처리 함으로써 속도를 향상한다. 실험 결과는 인공데이터와 실제데이터를 사용하여 k-mode와 비교를 통해 알고리즘의 높은 성능을 나타냄을 보인다.

1. 서 론

페이스북, 인스타그램과 같은 소셜미디어 매체들의 사용량이 폭발적으로 증가함에 따라 소셜미디어상에서 생성된 데이터의 분석 기법에 많은 관심이 집중되고 있다. 그런데 최근 소셜 미디어 데이터가 텍스트 기반의 데이터에서 이미지, 영상 등의 멀티미디어 데이터를 포함한 복잡한 구조를 가지게 되면서 기존의 군집화 알고리즘을 사용한 소셜 미디어 데이터 분석은 다음과 같은 문제들에 직면하게 되었다. 첫째, 오디오나 이미지 등의 멀티미디어 데이터가 포함된 소셜미디어 포스트 데이터간의 유사도 계산 시간이 매우 크다. 둘째, 포스트들이 가지는 의미나 목적에 의한 분류가 어렵다. 셋째, 대용량 소셜미디어 데이터가 갖는 특성에 적합한 군집화 기법이 제안되지 않았다.

이와 같은 문제를 해결하기 위하여, 본 논문에서는 소셜 미디어 포스트에 포함된 태그(tag) 벡터의 분석을 통한 유사도 측정기법을 새롭게 제안하고, 그 결과를 바탕으로 대용량 소셜미디어 데이터를 분석하기 위한 맵리듀스 기반의 새로운 군집화 알고리즘을 제안한다.

논문의 구성은 다음과 같다. 2장에서는 태그 기반의 소셜미디어 데이터 유사도 측정 기법을 제안하고 3장에서는 대용량 태그 벡터 데이터를 위한 맵리듀스 기반의 군집화 알고리즘을 제안한다. 그리고 4장에서는 실험을 통해 제안된 기법에 대한 성능 분석을 진행하고 마지막 5장에서 결론을 맺도록 한다.

2. 태그가 있는 소셜미디어 데이터간의 유사도 측정법

소셜 미디어 데이터의 태그 벡터는 사용자가 자신의 포스트를 위해 정의한 키워드의 집합이다. 예를 들어 어떤 포스트가 "#여행객 #도시 #배낭여행" 이라는 태그를 가지고 있다고 하자. 이때 "#여행객"과 같은 키워드를 태그이고 세 개의 키워드로 이루어진 "#여행객, #도시, #배낭여행"이 태그 벡터이다. 태그 벡터는 해당 포스트의 의미에 대한 메타 데이터로 활용될 수 있다. 이와 같은 태그 벡터에 대한 유사도 측정은 포스트에 포함된 다양한 멀티미디어 컨텐츠에 대한 복잡한 유사도 계산을 대신하여 빠른 시간 내에 군집화를 수행 할 수 있도록 할 뿐 아니라, 사용자의 의도에 따른 의미상의 데이터 분류를 가능하게 한다.

그런데 기존에 정의된 유사도 측정 기법들은 태그 벡터간의 유사도를 측정하기에 부적합한 부분이 많다. 예를 들어 범주 속성 데이터의 유사도 측정에 가장 많이 사용되는 해밍 거리(Hamming distance)의 경우, 두 벡터에서 동일한 순서에 위치한 태그끼리 비교하여 단어가 정확히 일치하는 경우에만 두 벡터를 유사하다고 판단한다. 그러나 사용자가 입력하는 태그는 특정한 순서가 없으며, 비슷한 의미를 가지는 다양한 단어가 사용될 수 있기 때문에 이와 같은 단순 비교로는 정확한 유사도를 판단하기 어렵다. 따라서 효율적으로 소셜미디어 데이터를 군집화하기 위해서는 새로운 유사도 측정법이 필요하다.

2.1 부분 태그 간의 유사도 가중치

소셜미디어 데이터가 가지고 있는 태그들은 띄어쓰기 없이 하나 이상의 단어가 연속적으로 이어져있다. 예를 들면 "#배낭여행" 태그는 "배낭"이라는 단어와 "여행"이라는 단어가 이어져서 하나의 태그를 구성한다. 그런데 #배낭여행 과 #여행 은 의미상으로 깊은 관계가 있지만, 기존의 유사도 측정 기법은 이와 같은 의미 관계를 고려하지 않는다. 태그의 의미상의 관계를 반영하기 위하여, 우리는 한 태그가 다른 태그에 완전히 포함되는 경우 이 두 태그를 "부분 태그"라고 하고, 부분 태그 간의 유사도에는 가중치를 부여한다. 두 부분 태그간의 유사도 가중치는 다음과 같이 정의한다.

정의 1. 부분 태그 간의 유사도 가중치

두 태그 w_k와 w_l에 대하여, w_k와 w_l이 부분 태그 관계에 있을 때, 부분 태그 가중치 $sw(k,l)$은 다음과 같이 정의된다.

$$sw(k,l) = 1 - \frac{|len(w_l) - len(w_k)|}{\max(len(w_l), len(w_k))},$$

이때 $len(w_k)$는 태그 w_k의 길이이다.

예를 들어 "#배낭여행" 과 "#여행"의 경우, 전체 길이는 4, 겹치는 부분의 길이는 2이고, 태그간 유사도 가중치는 0.5이다.

2.2 태그 순서 가중치

소셜미디어 포스트를 분석해보면 일반적으로 해당 포스트의 태그들 중 일반적이고 중요한 태그들이 앞쪽에 위치한다. 이러한 사실을 유사도 분석에 반영하기 위하여 태그의 위치에 따른 순서 가중치를 다음과 같이 정의한다.

정의 2. 태그 순서 가중치

$$pw(k,l) = 1 - \frac{l}{len(k)}$$

정의 2에서 len(I)는 벡터 I의 크기이고 l은 태그의 위치이다.

예를들어 “#배낭여행”이라는 태그가 열 개의 태그중 첫 번째에 나타나는 태그 벡터가 있고 열 개의 태그중 일곱 번째에 나타나는 태그 벡터가 있다고 하자. 이때 각각의 가중치는 0.9와 0.3으로 첫 번째 포스트가 두 번째 포스트 보다 배낭여행과 더 관련이 있는 포스트이다.

2.3 부분 태그와 태그 순서 가중치를 반영한 유사도 측정 기법

2.1에서 정의된 부분 태그간의 유사도 가중치와 2.2에서 정의한 태그 순서 가중치를 적용한 두 태그 백터의 유사도는 다음과 같이 정의한다.

정의 3. 부분 태그와 태그 순서 가중치를 반영한 유사도

$$S(i,j)=\sum_{k=0}^{cnt(i)}\sum_{l=0}^{cnt(j)} sw(k,l)pw(k,l)$$

I_i의 태그들의 개수를 cnt(i)라고 한다.

3. 맵리듀스 기반의 태그 데이터 군집화 알고리즘

기존의 범주속성 데이터를 군집화하는 대표적인 알고리즘에는 k-mode[1]가 있다. k-mode는 군집 안에서 각 차원에서 가장 빈번하게 등장하는 값들의 벡터를 대푯값으로한다. 하지만 소셜미디어 데이터의 태그 벡터는 길이가 고정되어있지 않고 속성값이 차원에 상관없이 등장할 수 있다. 그렇기 때문에 속성별로 대푯값을 바탕으로 군집의 중심을 설정하는 방법은 태그 벡터의 군집화에는 적합하지 않다.

3.1 태그 벡터 데이터의 군집 중심값

소셜미디어 데이터의 태그 벡터는 길이가 고정되어있지 않고 속성값이 차원에 상관없이 등장할 수 있다. 따라서 군집에 속한 데이터의 태그들이 전체 데이터 벡터 중에서 등장한 횟수를 측정한다. 그리고 임계값 이상의 등장 횟수를 가지는 태그들로 이루어진 벡터를 군집의 중심을 삼는다. 이때 많이 등장한 태그일수록 그 군집을 가장 잘 표현하는 태그라고 할 수 있으며, 정의 2와 같이 많이 등장한 횟수대로 벡터를 정렬한다.

3.2 군집화 후 센트로이드 업데이트

k-means[4] 및 k-mode의 군집화 기법은 초기 군집 중심값의 선정 방법이 전체 성능에 큰 영향을 미친다. 우리는 먼저 전체 데이터에 속한 태그 벡터를 탐색하면서 사용된 태그의 빈도수를 측정한다. 그리고 가장 많이 사용된 태그를 바탕으로 k개의 초기 군집 중심값을 구한다. 초기 중심값 설정 알고리즘은 다음과 같다.

Algorithm: 초기 센트로이드 설정 알고리즘

Input: I := 군집화의 대상이 되는 이미지 리스트. 리스트 길이는 N. 각 항목($I[i]$)은 여러 개의 태그들을 포함하며, 다음과 같은 속성들이 있다.

- $taglist$: 이미지 $I[i]$의 태그 리스트
- $freqsum$: 이미지 $I[i]$의 태그 빈도수 총합
- rfs : 생성된 센트로이드들 중 이미 등장한 태그의 점수들을 $freqsum$에서 모두 제거한 실제 $freqsum$ ($I[i].rfs \le I[i].freqsum$)

H := I 가 갖고 있는 모든 태그들에 대한 빈도수(등장 횟수)를 분석한 테이블.(key, $value$) = (tag, $frequency$) 형식으로, $H[tag]$로 tag에 대한 빈도수를 참조 가능.

Output: C := 군집화 기준이 되는 센트로이드 리스트. $C \subset I$ 이며, 초기 리스트 길이는 0.

```
for i=1 to N:
    for tag in I[i].taglist:
        I[i].freqsum += H[tag]   ; 각 이미지의 빈도수 총합 계산
I.sort(key=freqsum, reverse=True)   ; 빈도수 총합을 기준으로 내림차순 정렬
C[1]=I[1]   ; 첫 번째 센트로이드 설정
while i=2 to N and len(C) != M:   ; 모든 이미지들을 방문하여 적합한 센트로이드 후보 생성
    max_freq=I[i].rfs
    CC=i   ; 다음 유력 센트로이드 후보
    while j=i+1 to N and freqtotal < I[j].freqsum:   ;
        if max_freq < I[j].rfs:
            max_freq = I[j].rfs
            CC=j
    C[i]=I[CC]
return C
```

3.3 TBKmedoid using MapReduce

빅데이터는 그 용량이 매우 크기 때문에 단일 머신에서 메모리 기반의 알고리즘을 이용하여 군집화를 하는 데에는 많은 어려움이 있다[2][3]. 그래서 본 논문에서는 빅데이터를 군집화하기 위하여 분산병렬처리 환경인 맵리듀스를 이용한 알고리즘을 개발하였다. 이때 맵퍼는 각각의 데이터를 가장 가까운 센트로이드의 군집에 할당시켜주는 작업을 하고, 리듀서는 새로운 센트로이드를 갱신 시켜주는 작업을 한다.

먼저 데이터들은 n개로 나뉘어져서 n개의 맵퍼에 전역적으로 브로드캐스팅된다. 그후 각각의 맵퍼는 k개의 센트로이드의 정보를 받아서, 센트로이드와 데이터간의 거리계산을 병렬적으로 수행한다. 그리고 각 데이터는 가장 가까운 센트로이드가 있는 군집에 할당된다. 그 다음, 같은 군집에 속한 데이터는 같은 리듀서로 모인다. 그 후, 각 리듀서들은 각 군집의 새로운 센트로이드를 계산한다. 이 과정은 그림 1에 묘사되어 있다. 그림 1과 같은 과정은 더 이상 센트로이드의 변화가 없을 때 까지 반복되거나 사용자가 설정한 반복 횟수에 도달하면 멈추게 된다.

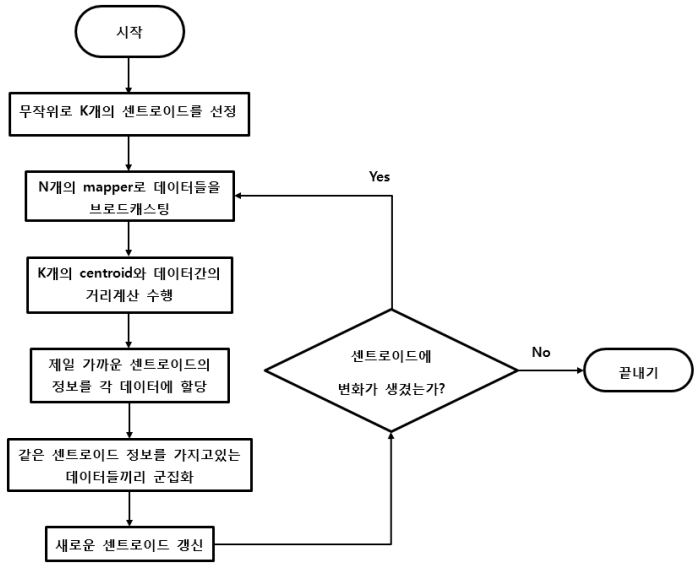

그림 1 TBKmedoid 알고리즘 흐름도

4. 성능평가

4.1. Synthetic Dataset

TBKmedoid의 성능을 확인하기 위하여 인공적으로 10차원의 데이터를 생성하였다. 태그로 이루어진 범주속성 데이터를 생성하기 위하여 먼저 각 차원이 하나의 단어로 이루어진 범주속성 데이터를 생성한다. 그리고 각 벡터가 가진 단어들에 같은 레벨에 속한 다른 벡터가 가진 단어들중 하나 이상을 이어서 태

그 벡터 데이터를 생성하였다. 그리고 각 데이터의 벡터의 개수와 군집의 개수정보는 표1에 설명되어있다.

	데이터 개수	군집의 개수
DS1	10000	100
DS2	30000	300
DS3	50000	500

표 1 Synthetic Dataset 정보

각 군집의 대푯값을 업데이트 하면서 군집화하는 알고리즘은 생성한 데이터를 이용하여 초기 군집 중심값 설정에 의하여 반복 횟수가 감소하고 그에따른 시간과 정확도를 K-mode와 TBKmedoid의 시간과 정확도를 측정하였다.

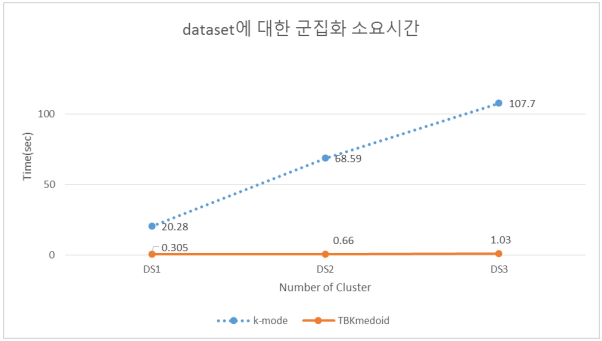

그림 2 dataset을 군집화 하는데 걸리는 시간

그림 2를 보면 k-mode는 DS1를 군집화하는데 걸린 시간이 20.28초다. 그리고 데이터의 개수가 20000개 증가한 DS2에 대한 시간은 68.59초, DS2에서 20000개가 증가한 DS3에서는 107.7초가 걸렸다. 데이터의 개수가 증가하고 그에따라 cluster의 개수가 증가하자 소요시간이 기하급수적으로 증가하였다. 특히 DS3를 k-mode로 군집화 할 땐 반복이 끝나지 않아 100번째에서 강제로 정지시켰다. 반면 TBKmedoid는 DS1, DS2, DS3 각각 0.305초, 0.66초, 1.03초가 걸렸다.

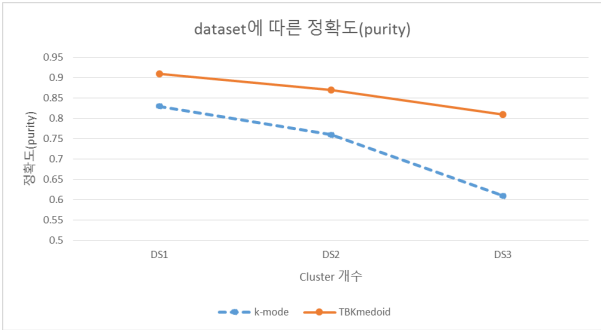

그림 3 Synthetic dataset 실험 결과의 정확도

이번에는 정확도를 측정해 봤다. 군집화 결과에 대한 정확한 성능 측정을 위하여 puriy측정하였다. 그 결과 그림 3과 같은 그래프를 얻을 수 있었다. 그림 3을 보면 두 알고리즘 모두 데이터가 증가하자 정확도가 떨어졌다. 그러나 k-mode의 경우 정확도가 더 낮고 더 가파르게 떨어진다는 사실을 알 수 있다. 본 실험을 통해서 TBKmedoid가 더 빠르고 정확하게 군집화가 가능하다는 사실을 알 수 있다.

4.2. Real Dataset

우리는 실제 데이터에서 우리가 제안한 군집화 알고리즘의 성능을 측정하기 위하여 인스타그램 데이터를 수집하여 실험하였다. 각 데이터셋의 개수와 군집의 개수는 표 2와 같다.

	데이터 개수	군집의 개수
RDS1	10000	100
RDS2	20000	200
RDS3	30000	300
RDS4	40000	400

표 2 Real Dataset 정보

실제 데이터셋은 초기에 군집화가 이루어져 있지 않기 때문에 순수도(purity)를 측정할 수 없다. 그래서 실제 데이터셋에 대한 군집화하는데 걸린 시간을 측정해본 결과 그림 4와같다. 모두 10초 이내로 군집화가 이루어졌다.

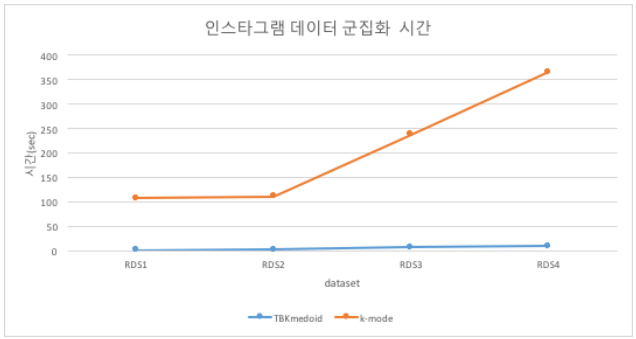

그림 4 인스타그램 데이터 군집화 시간

5. 결론

본 논문에서는 소셜미디어 데이터를 군집화 하기 위하여 태그를 활용한 유사도 측정법을 개발하였고, 이에 맞는 군집화 알고리즘을 제안했다. 실험 결과 인공데이터와 실제 데이터에서 시간이 더 빠르게 나타났다. 실제 데이터는 레벨링이 되어있지 않아 정확도를 측정하지 못했으나 인공 데이터실험을 바탕으로 실제 데이터에서도 정확한 군집화가 가능할 것 이라고 예상된다. 기존의 군집화 알고리즘 보다 정확한 군집화가 가능하지만, 군집의 개수가 증가하면 정확도가 떨어지는 문제가 발생한다. 군집의 개수가 많은 데이터도 좀 더 정확하게 군집화를 가능하게하는 후속 연구가 필요하다.

"본 연구는 미래창조과학부 및 정보통신기술진흥센터의 서울어코드활성화지원사업의 연구결과로 수행되었음" (IITP-2015-R0613-15-1174)

참고문헌

[1] Huang, Z; Ng, MKP, "A fuzzy k-modes algorithm for clustering categorical data", IEEE Transactions on Fuzzy Systems, 1999, v. 7 n. 4, p. 446-452

[2] Jefferey Dean, Sanjay Ghemawat, "MapReduce: simplified data processing on large clusters," In Proceedings of OSDI, 2004.

[3] Spyros Blanas, Jignesh M. Patel, Vuk Ercegovac, Jun Rao, Eugene J. Shekita, and Yuanyuan Tian, "A comparison of join algorithms for log processing in mapreduce." Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, 2010.

[4] P. Lingras, and C. West "Interval Set Clustering of Web Users with Rough K-means", Journal of Intelligent Information System, Vol. 23, No. 1, 2004.