

모바일 환경에서 데이터 마이닝을 적용한 하이브리드 데이터 브로드캐스트 기법

김한수⁰ 황인준
아주대학교 정보통신전문대학원 정보통신공학과
{kfreeman, ehwang}@ajou.ac.kr

Hybrid broadcast system using data mining in mobile computing environment

Hansu Kim⁰ Eenjun Hwang
Graduate School of Information and Communication, Ajou University

요약

모바일 컴퓨팅 환경에서 서버와 클라이언트 간에 데이터 전송은 대역폭의 차이로 인해 비대칭적 통신 구조를 이룬다. 이와 같은 제한된 환경에서 정보 전달은 한번의 전송으로 다수의 클라이언트에게 서비스를 제공하는 브로드캐스트 기법이 효과적이다. 하지만 브로드캐스트 방식은 전송되는 데이터의 양이 많을 경우 클라이언트 측에서는 필요한 서비스를 제공받기 위해 높은 접근대기 시간을 갖는 문제점을 가지고 있다.

본 논문에서는 클라이언트 요청 정보를 프로파일 형태로 서버측에 저장하고 이를 데이터 마이닝 기법을 사용해서 자주 요청되면서 서로 연관성이 높은 데이터 패턴을 추출하여 전송함으로써 접근대기 시간을 줄일 수 있는 기법을 제안한다. 또한 추출된 패턴도 정기적으로 브로드캐스트 함으로써 클라이언트 측에서 예견적인 프리패치와 캐쉬 재배치를 통해 배터리 소비량을 줄이는 방법을 제안한다.

1. 서론

최근 무선 통신과 모바일 단말기 기술이 급속히 발달함에 따라서 모바일 컴퓨팅 기술이 점차 전세계적으로 보편화되어 가고 있다. 하지만 아직까지 모바일 컴퓨팅 환경은 기존 유선 컴퓨팅에 비해 잦은 접속 단절, 낮은 대역폭, 적은 배터리 용량 등 많은 제약사항들을 갖고 있다. 이와 같은 제한된 환경에서 브로드캐스트를 통한 데이터 전송 방식은 클라이언트측의 배터리 용량과 주파수 대역협소의 문제를 극복하는데 있어 효과적이다[1].

그러나 서버에서 클라이언트 측으로 단방향 데이터 전송으로 채널을 통해 공유된 클라이언트의 수가 많을 경우 각 클라이언트는 필요한 데이터들을 서비스 받기 위해 높은 대기 시간을 소비하게 된다.

따라서 본 논문에서는 브로드캐스트 스케줄을 구성하기 위해 클라이언트의 이전 요청 프로파일을 사용자에 따라 독립된 세션단위로 분할하여 저장한다. 이 파일을 데이터 마이닝 기법을 적용하여 자주 발생되면서 연관성이 높은 데이터 집합을 추출하여 브로드캐스트 데이터를 구성한다. 또한 추출된 패턴 규칙을 정기적으로 전송하여 클라이언트 측에서 지능적으로 프리패치 및 캐쉬 재배치하는 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 브로드캐스트 방식 및 적용된 마이닝 기법에 대해 알아보고 3장에서 서버에 기록된 프로파일에서 순차규칙 추출과정을 다룬다. 4장에서는 추출된 순차규칙을 이용한 클라이언트 프리패치와 캐쉬 재배치 전략을 제안하고, 5장에서는 제안하는 시스템 구성도와 브로드캐스트 시나리오를 설명한다. 마지막으로 6장에서 결론 및 향후과제를 제시한다.

2. 관련 연구

모바일 환경에서 브로드캐스트를 통한 데이터 전송방식은 다음과 같이 세 방식으로 나뉜다. i) 푸시(Push) 기반 접근 방식은 클라이언트의 직접적인 요청에 관계없이 서버에서 미리 정의된 사용자 프로파일에 따라 전송하는 방식이다[2,3]. 그러나 현재 클라이언트의 요구에 관계없는 데이터를 브로드캐스트 할 경우 불필요한 전송으로 인해 접근대기 시간이 커질 수 있는 단점을 갖게 된다. 이와 같은 문제점을 해결하기 위해 ii) 풀(Pull) 기반 접근 방식은 일반적인 서버-클라이언트 구조를 이루며 명시적인 요청에 따라 데이터를 전송함으로써 대역폭을 줄일 수 있다[4,5]. 그러나 이 방식은 서버가 항상 각 클라이언트들의 요청을 모니터링 해야 하기 때문에 클라이언트들의 요청 데이터가 많을 경우 서버의 로드가 증가하는 단점이 있다. 따라서 최근에는 위 두 방식의 단점을 보완하고 장점을 결합한 iii) 하이브리드(Hybrid) 방식이 많이 연구되고 있다[6,7,8].

서버측에 저장된 클라이언트들의 이전 요청 프로파일을 분석하여 데이터 패턴 규칙을 추출하기 위해 적용된 각 마이닝 기법을 요약하면 다음과 같다. 첫째 연관규칙은 다량의 데이터 중에서 자주 발생되면서 연관성이 높은 데이터 집합을 발견하는 것이다[9]. 둘째 순차패턴은 연관성을 지니는 데이터 집합중에서 이들간에 순차성을 찾는 것이다 [10].

3. 순차규칙 기반 브로드캐스트 스케줄

프로파일은 접근 IP주소, 정보 액세스 시간, 접근 방법 등과 같은 다양한 정보들을 포함하고 있다. 이 파일을 전처리 과정을 거쳐 각 유저 ID에 따라 요청한 시간순으로 분할하여 저장하게 된다. 여기에서 명시된 시간범위내 서비스 요청 정보를 분할하기 위해 세션개념을 사용하였다. 전체 데이터베이스는 세션들의 집합으로 구성된다.

이로부터 순차규칙을 추출하기 위한 정의는 다음과 같다. 데이터 집합 $I = \{i_1, i_2, \dots, i_m\}$ 와 세션집합 $S = \{s_1, s_2, \dots, s_n\}$ 로 구성된 데이터베이스 D는, $\forall S \in D, S \subseteq I$ 인 관계에 있다. 세션집합에서 데이터들의 발생순서가 크기 k개의 p_1, p_2, \dots, p_k 의 순서로 발생 할 확률(p)을 순차패턴이라 정의한다.

순차패턴 $p = \langle p_1, p_2, p_3, \dots, p_k \rangle$ 의 가능한 순차규칙은
 $\langle p_1 \rangle \Rightarrow \langle p_2, p_3, \dots, p_k \rangle$,
 $\langle p_1, p_2 \rangle \Rightarrow \langle p_3, p_4, \dots, p_k \rangle$,
 ...

$\langle p_1, p_2, \dots, p_{k-1} \rangle \Rightarrow \langle p_k \rangle$ 이 될 수 있다. 하나의 순차패턴 $P_n = \langle p_1, p_2, \dots, p_n \rangle \Rightarrow \langle p_{n+1}, p_{n+2}, \dots, p_k \rangle$ 이 $0 < n < k$ 범위내에 있을 때 지지도 $\langle p_1, p_2, \dots, p_n \rangle$ 를 가지면서 또한 $\langle p_1, p_2, \dots, p_k \rangle$ 의 지지도를 가질 때 신뢰도값은 다음과 같다.
 신뢰도(P_n) = 지지도($\langle p_1, p_2, \dots, p_k \rangle$) / 지지도($\langle p_1, p_2, \dots, p_n \rangle$) * 100로 구해진다.

표1. 추출된 순차규칙의 예

Sequence	Rules	Support(%)	Confidence(%)	Weight
<d, a, c>	<d>=><a,c>	2.0	60	140
	<d,a>=><c>	1.5	80	
<e, f, b>	<e>=><b,f>	1.5	70	150
	<e,b>=><f>	1.0	80	
<a, d, c>	<a>=><d,c>	1.5	60	130
	<a,d>=><c>	1.0	70	
<e, b, f>	<e>=><f,b>	2.0	80	170
	<e,f>=>	1.5	90	

이와 같은 정의를 따르는 브로드캐스트 스케줄을 간단한 예를 통해 설명하면 다음과 같다.

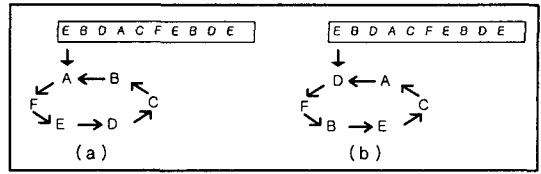


그림1. 브로드캐스트 스케줄 구성 예

브로드캐스트 스케줄을 위한 데이터 집합이 {A,B,C,D,E,F}이라 하자. 표 1은 추출된 순차규칙의 예이다. 이 중에서 {d, a, c}, {a, d, c}와 {e, f, b}, {e, b, f}간에는 서로 유사성이 있다. 이 경우 브로드캐스트 되기 위해 유사한 집합들 간에 선택이 필요 하게된다. 이를 위해 각 집합 간 신뢰도값의 합을 가중치(Weight)값으로 둔다. 따라서 유사한 그룹중에서 가중치값이 높은 <d, a, c> 와 <e, b, f>가 선택되어진다. 그림 1은 이와 같은 방법으로 브로드캐스트 스케줄을 구성한 것이다. 1-(a)는 순차규칙을 사용하지 않은 예이고 1-(b)는 순차규칙을 사용한 예이다. 브로드캐스트 디스크(broadcast disk)가 시계 반대방향으로 회전하고 각 데이터가 서비스되는 시간간격이 1초라고 하면 1-(a), 1-(b)의 접근대기는 다음과 같다.

1-(a): 4 + 3 + 2 + 3 + 2 + 3 + 5 + 3 + 2 + 3 = 30 초
 1-(b): 3 + 1 + 2 + 1 + 1 + 3 + 4 + 1 + 2 + 1 = 19 초
 로 1-(b)의 접근대기 시간이 적은 것을 알수있다.

4. 순차규칙 기반 프리페치와 캐쉬 재배치 전략

클라이언트 프리페치와 캐쉬 재배치 전략은 모바일 환경에서 제한된 대역폭과 캐쉬 용량을 고려할 때 중요한 사항이다. 본 장에서는 순차규칙을 이용한 프리페치와 캐쉬 재배치 전략을 제안한다. 브로드캐스트 서버는 데이터 전송과 함께 순차규칙에 의해 추출된 패턴도 정기적으로 전송한다. 그러므로 클라이언트가 브로드캐스트 되는 데이터의 순서를 알게된다. 제안하는 프리페치 슈도코드는 그림 2와 같다. 이 슈도코드는 현재까지 브로드캐스트 서버측에 저장된 클라이언트 요청 프로파일을 분석해서 앞으로 사용될 확률이 높은 패턴을 추론한다. 생성된 규칙의 수가 많을 경우 신뢰도 값에 따라 내림차순으로 정렬함으로써 생성패턴의 수를 제한하게 된다.

```

prediction_items=0;
foreach sequential rule a=>b do
    if there is a match for a in request_items then
        prediction_items = prediction_items U a2
    endif
endifor
sort prefetch_items in decreasing order of the confidence
if broadcast_item is not in cache then
    if broadcast_item is in prediction_item but not cache then
        pvalue=rule_conf(broadcast_item)
        if minimum pvalue of cached_items < pvalue then
            prefetching
        endif
    endif
endif
    
```

그림2. 프리페치 슈도코드

클라이언트들은 캐쉬 재배치를 위해 순차규칙에 의해 추론된 패턴과 현재 캐쉬에 저장된 데이터들을 비교해 추론된 패턴보다 적은 신뢰도값을 저장한 데이터들을 재배치 한다.

캐쉬 재배치 슈도코드는 그림 3 과 같다.

```

if cached_items == prediction_items then
  for each data item in
    prediction_items ∩ cached_item do
    replace the data item supported by a rule with the least confidence
else
  replace a data item in cached_items-inferred_items using their
  pvalues
endif
    
```

그림3. 캐쉬 재배치 슈도코드

이와 같은 슈도코드들을 간단한 예로 설명하면 그림 4와 같다. 클라이언트가 서비스 받기를 원하는 처음 세개의 데이터 E, B, D가 스냅샷으로 선택되었다. 캐쉬에는 C, F가 저장되어 있고 브로드캐스트 스트림은 A, B, C, D, E, F를 반복적으로 브로드캐스트 한다. 클라이언트는 현 시점에서 E를 서비스 받기 위해 브로드캐스트 스트림을 모니터링 하다가 E가 전송되기 전에 B를 지능적으로 클라이언트 캐쉬에 저장한다.

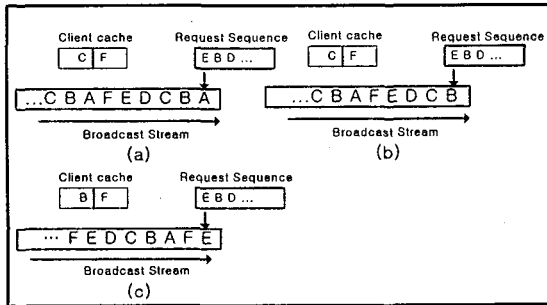


그림4. 순차패턴 기반 프리패치와 캐쉬 재배치 예

5. 브로드캐스트 시스템 시나리오

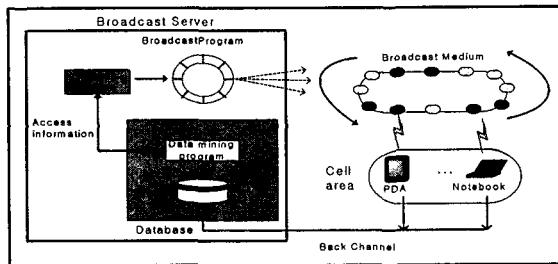


그림5. 전체적인 시스템 구성도

먼저 모바일 사용자들은 자신의 모바일 단말기를 통해 서비스를 이용하는 동안 필요한 정보가 단말기의 로컬 캐쉬에 있는지를 검색하고 한다. 필요한 데이터가 없을 경우 브로드캐스트를 일정시간 모니터링 한 후, 백채널을 통해 브로드캐스트 서버에게 서비스 받기 원하는 요청 정보를 보낸다. 브로드캐스트 서버는 요청 대기큐를 통해 들어온 요청 들을 전처리 과정을 거친후 각 사용자별 독립된 세션 단위 프로파일로 저장하게 된다. 이 파일을 데이터 마이닝

프로그램을 이용해 자주 요청되면서 서로 연관성이 높은 데이터들을 추출하여 브로드캐스트 디스크를 구성하여 전송한다. 이때, 마이닝 프로그램으로 추출된 패턴 또한 정기적으로 브로드캐스트 디스크에 추가하여 전송함으로써 클라이언트가 앞으로 전송 되어질 데이터들의 순서를 인식하게한다. 이를 이용해 클라이언트측 에서는 자신이 필요로 하지 않는 데이터가 전송되는 시간동안은 에너지 소모량이 적은 휴지 상태(doze mode)를 유지하고 필요한 데이터가 전송시점에 예견적으로 데이터를 프리패치 받아 로컬 캐쉬에 적재하면서 서비스를 이용해 나가게 된다.

6. 결론 및 향후 계획

모바일 컴퓨팅 환경의 제약조건 속에서 브로드캐스팅 기법은 효과적인 방법으로 이와 관련된 연구들이 활발하게 진행되고 있다. 본 논문에서는 브로드캐스트되는 데이터들을 구성하기 위해 데이터 마이닝 기술을 적용하고, 추출된 패턴 정보를 스케줄에 포함시켜 전송함으로써 클라이언트 측에서 지능적으로 필요한 데이터들을 프리패치 받고 캐쉬 재배치를 사용할 수 있는 방법을 제안하였다. 향후연구 과제로는 제안한 기법과 기존 하이브리드 스케줄링 방식 및 클라이언트 프리패치 캐쉬 재배치 알고리즘과 성능을 비교 평가할 계획이다.

참고 문헌

- [1] Jin Jing, Abdelsalam Helal, Ahmed Elmagarmid, "Client-Server Computing in Mobile Environments," ACM Computing Surveys, Vol.31, No.2, pp. 117-157,1999.
- [2] Acharya, S., " Broadcast Disks: Data Management for Asymmetric Communications Environments," SIGMOD Conference,199-210,1995.
- [3] C.J.Su and L. Tassiulas. "Designing Broadcast Schedules for Information Dissemination through Broadcasting". Technical report, Institute For Systems Research, University of Maryland, College Park, December 1996.
- [4] Aksoy, D. Franklin M. "Scheduling for Large-Scale On-Demand Data Broadcasting," INFOCOM '98 vol.2 651-659 1998.
- [5] Xuan, P. "Broadcast on demand: efficient and timely dissemination of data in mobile environments," Real-Time Technology and Applications Symposium,38-48,1997.
- [6] Hu, J.H., "A Novel Push-and-Pull Hybrid Data-Broadcast Scheme for Wireless Information Networks," ICC 200, vol.3, 1778-1782,2000.
- [7] K. Stathatos, N. Roussopoulos, and J.S. Baras, "Adaptive Data Broadcast in Hybrid Networks," Proc.23rd VLDB Conf.,1997.
- [8] K. Stathatos, N. Roussopoulos, and J.S. Baras. "Adaptive data broadcast in hybrid networks," In Proceedings of the 23 rd VLDB Conference, pages 326-335, Athens, Greece, August 1997.
- [9] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proceedings. 20th Int'l Conf. VLDB, Sept. 1994.
- [10] R.Agrawal and R. Srikant, " Mining Sequential Patterns," Proceeding. Int'l Conf. Data Eng.(ICDE), Mar.1995.