Journal of the Korea Academia-Industrial cooperation Society
Vol. 12, No. 5 pp. 2375-2382, 2011

DOI: 10.5762/KAIS.2011.12.5.2375

웹 개인화를 위한 웹사용자 클러스터링 알고리즘에 관한 연구

이해각[1*]
[1]순천향대학교 컴퓨터공학과

A Study on Web-User Clustering Algorithm for Web Personalization

Hae-Kag Lee[1*]
[1]**Department of Computer Engineering, Soonchunhyang University**

요 약 웹사이트 운영이 비즈니스 모델로서의 성공을 거두기 위한 가장 중요한 요소 중 하나는 웹사용자의 성향을 분석하여 이를 효율적으로 이용하는 것이다. 사용자 분석을 통하여 사용자들에게 웹사이트의 가치를 효율적으로 전달하고 이를 통하여 운영자는 충분한 수익을 거둘 수 있다. 이러한 점에서 웹 사이트를 이용하는 사용자들의 취향과 행동방식을 얻어내려는 웹 방문 패턴 발견으로써의 사용자 클러스터링은 매우 중요하다. 또한 얻어진 사용자의 클러스터링 정보는 웹 개인화나 웹 사이트를 재구성하는데 필수적이다.
본 논문에서는 사용자 웹 방문 데이터를 정제하고 분류하여 그 특성에 따라 사용자들을 몇 개의 그룹으로 클러스터링 하기 위한 알고리즘이 제안된다. 알고리즘은 2단계로 구성되는데 첫 번째 단계는 초기해를 구하는 단계로서, 패스의 사이각을 이용하여 유사도를 측정하고 이 유사도에 따라 K개의 사용자 그룹으로 분류하여 초기해를 구한다. 두 번째 단계는 첫 번째 단계에서 구한 초기해를 개선하여 최적해를 찾는 과정으로서 하이퍼플레인을 이용하여 클러스터링하는 개량된 K-평균알고리즘을 제안한다. 또한 실험을 통하여 기존의 방법과 비교하여 제안된 알고리즘의 효율성과 패스 특성이 보다 정확하게 계산된 클러스터링이 구현됨을 확인할 수 있다.

Abstract The user clustering for web navigation pattern discovery is very useful to get preference and behavior pattern of users for web pages. In addition, the information by the user clustering is very essential for web personalization or customer grouping.
In this paper, an algorithm for clustering the web navigation path of users is proposed and then some special navigation patterns can be recognized by the algorithm. The proposed algorithm has two clustering phases. In the first phase, all paths are classified into k-groups on the bases of the their similarities. The initial solution obtained in the first phase is not global optimum but it gives a good and feasible initial solution for the second phase. In the second phase, the first phase solution is improved by revising the k-means algorithm. In the revised K-means algorithm, grouping the paths is performed by the hyperplane instead of the distance between a path and a group center. Experimental results show that the proposed method is more efficient.

Key Words : Data Mining, Clustering, Hyperplane, K-Means Algorithm

1. 서론

최근 인터넷에는 컨텐츠 산업의 붐을 타고 많은 기업들이 자사의 사이트를 개설하여 소비자에게 보다 직접적이고 적극적으로 마케팅을 하고 있다. 이러한 추세는 최근 들어 더욱 활성화 되고 있는데 기업은 웹사이트 구축을 통하여 각자의 수익모델을 만들고 유료 컨텐츠를 개설하고 차별화된 서비스를 통하여 수익을 극대화하려 노력하고 있다. 특히 수익 구조를 인터넷 매매에 갖고 있는 기업 즉, 전자상거래 기반의 기업의 경우 온라인 상의 소

*교신저자 : 이해각(lhk7083@sch.ac.kr)
접수일 11년 02월 18일 수정일 (1차 11년 04월 17일, 2차 11년 04월 25일) 게재확정일 11년 05월 12일

비자들에 대해서 실구매까지 이어지도록 많은 투자와 전략을 기획하게 된다. 기업이 인터넷 상에서 사업을 확장하고 많은 투자를 함에 있어서 실질적으로 드러나지 않는 인터넷 사용자들을 어떻게 파악하고 그들의 반응과 관심분야에 대한 정보를 도출해 내는가는 매우 어려운 문제이다. 또한 최근 들어 온라인 소비자들의 성향은 더욱 까다로워지고 그들의 관심사는 매우 빠르게 변화하고 있다.

이처럼 웹사이트 상에서 이루어지는 소비자들의 행동양식을 분석하고 그 특성에 따라 그룹화하고, 각각의 취향에 맞는 서비스를 제공하는 것은 웹사이트의 수익 창출을 위하여 매우 중요하다. 경우에 따라, 사용자 행동양식에 대한 정보 획득을 위해 웹방문 패턴인식이 필요한데, 이 때 웹 로그데이터 분석법을 활용한다. 최근 들어 웹사이트 개인화(Web Personalization)에 관한 연구가 활발히 진행되고 있다[1-4]. 웹 개인화는 클러스터링과 같은 데이터마이닝 기법을 이용하여 개개의 사용자에게 흥미를 가질만한 URL들의 집합을 예측하는 것이라 할 수 있다.

웹 로그는 서버에 대한 접근에 관련된 데이터와 참고자료에 대한 데이터 등이 있는데, 데이터양이 접속횟수에 비례하여 계속적으로 증가하므로 분석에 있어서 대용량의 처리가 필요하고, 처리에 대한 효율성을 제고할 필요가 있다.

이러한 대용량의 데이터를 일반질의(Query), OLAP(On-Line Analysis Processing), 다차원 분석(Multi-Dimensional Analysis) 등과 같은 기존의 조회 도구들은 구조적인 데이터베이스 조회도구로서 데이터에 숨겨진 패턴, 관계, 경향 등의 의사결정에 필요한 지식정보에는 한계를 가지고 있다. 반면, 데이터 마이닝(Data Mining)은 통계 및 수학적 기법뿐만 아니라 신경망 등을 비롯한 여러 가지 패턴인식기법 등을 사용하여 데이터 속에 내재된 정보를 찾아 의사결정이나 정보시스템에 활용할 수 있도록 지원해 준다. 또한 얻어낼 수 있는 정보의 형태도 다양하다. 그러므로 기존 조회도구와 더불어 데이터 마이닝을 이용함으로써 보다 심도 있는 정보를 획득할 수 있다[5-7].

본 논문에서는 웹 로그 데이터의 분석을 위하여 데이터 마이닝 기법 중에 클러스터링 기법을 적용하여 웹 사이트 방문패턴에 따른 사용자 클러스터링 알고리즘을 제안한다. 데이터마이닝 기법 중 클러스터링에 대한 대표적인 알고리즘은 K-평균알고리즘이다. 그러나 K-평균알고리즘은 초기해(Initial Solution)에 의존한 해가 구해지는데 초기해에 따라 최적해가 아닌 비최적해에 수렴할 우려가 있다(2.3절 그림 4 참조). 또한, K-평균알고리즘을 웹방문 패턴에 그대로 적용하면, 방문데이터에 대한 특징벡터(Feature Vector)의 차수(Degree)가 증가할 경우 매우 많은 계산량을 가지게 되어 웹방문 패턴 발견기법으로 활용하기가 어렵다[8]. 이것은 K-평균 알고리즘이 유사도 측정도구로서 벡터 간의 거리계산에 의존하기 때문이다. 본 논문은 다차원 공간벡터 상에서의 하이퍼플레인(Hyperplane)을 이용하여 계산량을 대폭 줄일 수 있는 개량된 K-평균 알고리즘을 제안한다.

2. 기존의 방법에 대한 고찰

2.1 전처리 과정

웹방문에 대한 각 패스의 패턴을 알아내기 위해서 우선 로그 파일의 데이터로부터 원하는 정보만을 취하여 입력데이터로 사용한다.

```
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2001-10-16 08:59:57
#Fields: date time c-ip cs-username s-sitename s-computername s-ip s-port
cs-method cs-uri-stem cs-uri-query sc-status sc-win32-status sc-bytes
cs-bytes time-taken cs-version cs-host cs(User-Agent) cs(Cookie)
cs(Referer)
2001-10-16 08:59:57 192.168.50.151 - W3SVC1 ISA 211.112.217.47 80
GET /Default.htm - 304 0 196 280 70 HTTP/1.1 isa-cse.sch.ac.kr
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+98) - -
2001-10-16 08:59:57 192.168.50.151 - W3SVC1 ISA 211.112.217.47 80
GET /pz_chromeless_2.1.js - 304 0 140 336 30 HTTP/1.1 isa-cse.sch.ac.kr
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+98) -
http://isa-cse.sch.ac.kr/
2001-10-16 08:59:57 192.168.50.151 - W3SVC1 ISA 211.112.217.47 80
GET /image/enter.gif - 304 0 140 331 20 HTTP/1.1 isa-cse.sch.ac.kr
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+98) -
http://isa-cse.sch.ac.kr/
2001-10-16 09:00:04 192.168.50.151 - W3SVC1 ISA 211.112.217.47 80
GET /main.htm - 200 0 1428 367 100 HTTP/1.1 isa-cse.sch.ac.kr
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+98) -
```

[그림 1] 전처리되지 않은 IIS 웹 로그 의 예

그림 1은 웹로그의 예이며 로그 파일은 하나의 파일에 기간별(일, 주, 월)로 분리하여 저장할 수도 있다.

웹 서버의 로그로부터 얻어지는 초기 데이터는 전처리되어 데이터마이닝에 용이하도록 변환되어 재구성된다. 전처리 과정은 데이터 정제, 사용자 구분, 세션 구분, 세션 보정, 패스보정, 형식화 단계로 나눌 수 있다.

	방문 페이지 및 머문 시간
path1:	a(20) → d(7) → e(6)
path2:	b(15) → d(5) → c(5) → f(8)

[그림 2] 전처리 된 웹방문 패스의 예

2.2 유사도 측정에 의한 패스 클러스터링

두 개의 웹방문 접근경로를 각각 path1과 path2라고 했을 때, 두 개의 패스에 대한 유사도 측정하기 위해 우선 특징벡터(Feature Vector)를 정의한다. 특징벡터는 방문한 페이지 목록, 방문 순서, 방문 후 머문 시간의 요소로 정의된다[1]. 그림 3은 특징벡터의 생성 예제이다.

벡터 요소	a	b	c	d	e	f	a d	b d	c f	d c	d e	a d e	b d c	d c f
path1	20	0	0	7	6	0	27	0	0	0	13	33	0	0
path2	0	15	5	5	0	8	0	20	13	10	0	0	25	18

[그림 3] 그림 2에 대한 특징벡터의 예

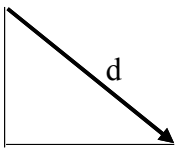
각각의 패스에 대해 발생하는 서브시퀀스에 대해 특징벡터를 생성하고 이에 대한 여부를 이용하여 패스를 분류하는 것은 시간적으로나 공간적인 문제에 있어서 매우 어려운 문제이다.

참고문헌[9]에서는 각 페이지에 대한 접근 유무에 따라서 벡터를 정의하고 유클리드에 의한 클러스터링을 시도하였다. 이 방법은 많은 페이지를 가지는 웹 사이트에 대해서는 효율적이지 못하다. 참고문헌[10]에서는 Kohenen 알고리즘을 이용하여 클러스터링을 하였는데 이 방법은 시간 흐름적인 표현을 하기에 부족함이 있다[11]. 참고문헌[1]에서는 AprioriAll 알고리즘을 이용하고 내용 페이지를 구분하여 클러스터링을 하여 빠른 탐색시간을 보장받을 수 있으나 전체적인 패스의 정보를 표현하기가 어렵다.

참고문헌[4]에서 제안한 패스간의 유사도를 측정하는 방법은 두 특징벡터의 사이각을 계산하고 이를 두 특징벡터의 유사도로 활용함으로써, 위에서 언급한 시간 흐름의 표현에 대한 문제점과 패스정보의 전체를 반영하지 못하는 문제점에 대한 해결책을 제시하였다.

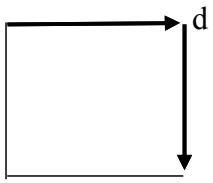
2.3. K-평균 알고리즘

클러스터링은 데이터를 유사한 특징을 가진 몇 개의 집단으로 그룹화하여 분할하는 것을 말한다. K-평균 알고리즘은 데이터 마이닝의 클러스터링 작업의 대표적인 기법이다[5,6,8,12]. 이 방법은 특징벡터 사이의 거리를 이용해 주어진 기준을 최적화하도록 구현된다. 유사한 특성을 갖는 레코드들은 서로 근접하여 위치한다는 가정에 근거하여 거리에 의한 클러스터링을 하는데, 거리에 대한 정의로 사용되는 대표적인 것인 유클리드 거리와 맨하탄 거리(일명 City-Block 거리라고도 함)에 대한 정의는 다음 그림 4 ~ 그림 5와 같다[5].



$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{N} (x_{ik} - x_{jk})^2}$$

[그림 4] 유클리드 거리에 대한 정의



$$d(x_i, x_j) = \sum_{k=1}^{N} |x_{ik} - x_{jk}|$$

[그림 5] 맨하탄 거리에 대한 정의

K-평균 알고리즘은 주어진 데이터들을 유사도에 따라 K개의 그룹으로 나누는 알고리즘을 말하며 구체적인 과정은 다음과 같다.

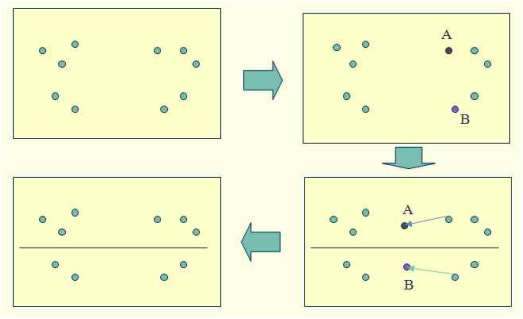
① 몇 개의 그룹을 생성할 것인지 K를 결정
② 각 그룹의 초기 중심값을 결정
③ 모든 데이터를 각 그룹의 중심 값에 대하여 가장 짧은 거리를 갖는 그룹에 할당
④ 각 그룹에 속하는 모든 데이터에 대한 평균값을 구해 새로운 중심값으로 결정하고 기존의 중심값과 새로운 중심값의 차이 계산하여 그 차이가 0에 근접할 때까지 ③부터 다음 단계 계산을 반복함

K-평균 알고리즘은 여러 클러스터링 방법 중에서 대용량 데이터를 빠르게 처리할 수 있으며, 그 알고리즘도 비교적 간단하다. 그러나 단점은 초기해에 의존한 해가 구해져 최적해가 아닌 비최적해에 수렴하는 경우가 수 있다.

그림 6은 초기해의 잘못 설정으로 알고리즘이 비최적해에 수렴되는 경우를 보여주고 있다. 그림 6에서 A, B는 알고리즘 중간단계에서의 클러스터 중심 값을 의미한다. 이와 같이 잘못 구해진 비최적해는 지역 최적해(Local Optimal Solution)의 일종이며 이러한 오류를 방지하기 위해서는 초기해의 설정이 중요함을 알 수 있다.

K-평균 알고리즘의 또 다른 문제점은 각 단계별 거리에 대한 계산량이 매우 많다는 것이다. 각 데이터는 각

클러스터의 중심으로부터 떨어진 계산을 통하여 가장 작은 거리를 가지는 클러스터에 속하게 하는 방식으로 해를 개선해나가는 방식인데 이 과정에서 많은 계산을 요구하게 된다. 이러한 단점을 극복하기 위하여 본 논문에서는 거리 계산 없이 클러스터링 하는 하이퍼플레인(Hyperplane)을 이용한 클러스터링 기법을 제안한다.



[그림 6] 비최적해에 수렴하는 K-평균 클러스터링 알고리즘의 예

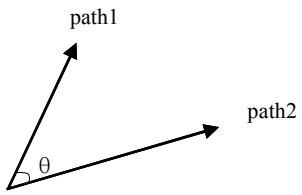
3. 제안 알고리즘

본 논문에서 제안하는 알고리즘은 특징벡터들의 사이각의 크기를 유사도로 사용한 클러스터링을 하여 초기해를 구하는 1단계 과정과 이 초기해를 개선하여 최적해를 찾는 2단계 과정으로 구성된다.

3.1 유사도에 따른 초기해 설정

2.3절에서 언급한 바와 같이 K-평균 알고리즘은 지역최적해가 구해지는 단점이 있다. 이는 알고리즘이 잘못된 초기해를 가지고 시작하는 경우에 발생한다. 본 논문은 이러한 문제점을 해결하기 위하여 우수한 초기해를 구하고, 이를 개선하여 최적해(Global Optimal Solution)를 방법을 제안한다.

1단계 클러스터링으로써 2개의 패스에 대한 유사성 비교는 패스사이의 각을 이용하며 이것은 그림 7과 같이 벡터의 내적(Inner Product) 계산을 통하여 측정할 수 있다.



[그림 7] 2개의 패스간의 사이각 θ

$$\cos(\theta_{path1,path2}) = \frac{path1 \cdot path2}{\| path1 \| \ \| path2 \|}$$

위에서 $\| path1 \|$ 과 $\| path2 \|$ 는 각각 path1과 path2의 크기(Norm)을 의미한다. 설명을 위하여 5개의 패스간의 각에 대한 예를 보자.

$$\cos(\theta pass_1, pass_2) = \begin{bmatrix} 1 & 0.087 & 0.350 & 0.212 & 0.407 \\ 0.087 & 1 & 0.019 & 0.100 & 0.284 \\ 0.350 & 0.019 & 1 & 0.381 & 0.145 \\ 0.212 & 0.100 & 0.381 & 1 & 0.305 \\ 0.407 & 0.284 & 0.145 & 0.305 & 1 \end{bmatrix}$$

내적 계산을 통하여 패스 행렬 결과를 얻으면 cos(Θ path1, path2)를 이루는 모든 요소는 0~1사이의 값을 갖게 되며 이 값이 클수록 유사성이 크다고 할 수 있다. 만약 이 값이 0인 경우 두 패스가 유사성이 없는 관계를 나타내고, 1인 경우 두 패스가 동일함을 나타낸다. 이 중에서 cos(Θ path1, path2)에 범위를 지정하여 유사성이 높은 것들끼리 클러스터링 한다. 참고문헌[4]에서는 지정된 범위에 해당하는 요소에 대하여 1로 표시하고 나머지를 0으로 표시한 후 행렬을 변형시켜 그룹화하도록 제안하였다. 그림 8은 $0.36<\cos(\Theta)\leq 1.0$ 의 범위에 해당하는 요소에 대하여 1로 정의하고 나머지는 0으로 정의하였을 경우, 각 패스 쌍에 대한 유사도를 나타내는 멤버쉽 행렬과 변환된 행렬의 예이다.

path	1	2	3	4	5
1	1	0	0	0	1
2	0	1	0	0	0
3	0	0	1	1	0
4	0	0	1	1	0
5	1	0	0	0	1

⇒

path	1	5	3	4	2
1	1	1	0	0	0
5	1	1	0	0	0
3	0	0	1	1	0
4	0	0	1	1	0
2	0	0	0	0	1

[그림 8] 멤버쉽 행렬과 변환된 행렬의 예

그림 8에서 각 패스의 유사도에 따라 1단계 클러스터링 한 결과는 다음과 같다.

Cluster A	path1, path5
Cluster B	path3, path4
Cluster C	path2

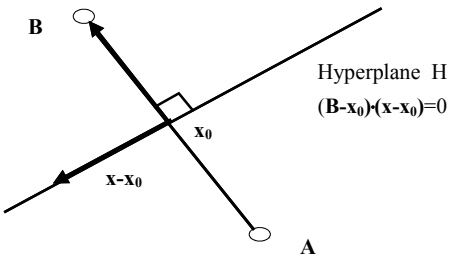
그러나 이 결과는 최적해가 아니며 여러 클러스터에 중복되는 패스가 발생되고 그룹이 모호한 경우가 대부분이다. 이 결과를 초기해로 하는 두 번째 단계의 개선과정

을 통하여 최적해가 구해지며 3.2절에서 자세히 논의한다. 해를 구하는 과정을 이와 같이 2단계로 구성하는 이유는 기존의 K-평균 알고리즘의 해가 임의의 초기해에 의존하여 구해지는 과정에서 비최적해에 수렴하는 경우가 있는데, 본 논문의 방법은 매우 우수한 초기해를 사용함으로써 그 단점을 보완할 수 있다. 또한 초기해를 구하는 과정이 매우 빠른 시간 내에 구해질 수가 있다.

그룹이 형성되면 각 특징벡터들은 그룹별로 해시 체이닝 기법(Hash Chaining Technique)을 사용하여 저장한다. 이것은 다음 단계 클러스터링을 위한 저장을 위해서이며 그룹별로 분할 저장함으로써 저장 공간의 효율을 높일 수 있고 해시 체이닝 기법을 사용하기 때문에 선형 탐색에 비하여 매우 빠른 탐색시간을 확보할 수 있는 장점이 있다.

3.2 하이퍼플레인을 이용한 클러스터링

하이퍼플레인은 2차원 평면 상에서의 직선, 3차원 공간상에서의 평면처럼 n차원의 벡터 공간을 두 개로 분할하는 기하학적인 의미를 가지고 있는 초평면이다. 이러한 하이퍼플레인은 방정식 $\sum_{j=1}^{n} p_j x_j = k$를 만족하는 점들의 집합 벡터 $\mathbf{x} = (x_1, x_2, x_3, \ldots x_n)$로 구성되어 있다. 여기서 $\mathbf{p}$는 n차원 공간 E^n 안의 0이 아닌 벡터이고 k는 스칼라이다. 또한 p는 하이퍼플레인과 수직을 이룬다. 만약 $\mathbf{x_0} \in H$이면 $\mathbf{p} \cdot \mathbf{x_0} = k$이고 $\mathbf{x} \in H$인 어떤 x에 대해서 $\mathbf{p} \cdot \mathbf{x} = k$를 취한다(· 는 벡터의 내적연산을 의미한다. 즉, 하이퍼플레인은 $\mathbf{p} \cdot (\mathbf{x} - \mathbf{x_0}) = 0$을 만족시키는 점들의 집합으로 표현할 수 있으며 여기서 $\mathbf{x_0}$는 하이퍼플레인 상의 고정된 점이다[13,14].

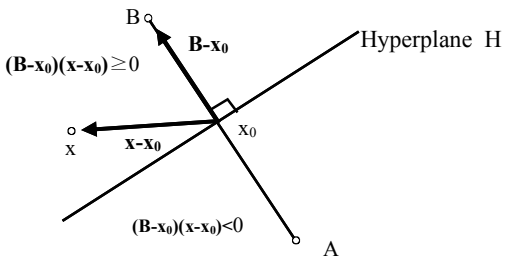


[그림 9] 하이퍼플레인의 정의

하이퍼플레인은 E^n을 2개의 영역으로 나누며 이것을 반공간(halfspaces)이라고 부른다. 따라서 $\mathbf{x}$가 어느 영역에 속하는지는 다음과 같이 결정한다. $\{\mathbf{x} : \mathbf{p} \cdot (\mathbf{x}-\mathbf{x_0}) \geq 0\}$이면 하이퍼플레인을 중심으로 p방향 쪽의 영역에 속하게 되고, $\{\mathbf{x} : \mathbf{p} \cdot (\mathbf{x}-\mathbf{x_0}) \leq 0\}$이면 p와 반대 방향 쪽의 영역에 속하게 된다. 이 같은 사실은 여러 개의 벡터를 그 특징에 따라 두 개의 그룹으로 분할할 수 있는 근거가 된다.

하이퍼플레인은 그림 10과 같이 하나의 그룹의 중심점 A와 또 다른 그룹의 중심점 B의 중간지점인 $\mathbf{x_0}$를 수직으로 지나며 $(\mathbf{B}-\mathbf{x_0}) \cdot (\mathbf{x}-\mathbf{x_0}) = 0$을 만족한다. 따라서 하이퍼플레인은 중심점 A와 중심점 B를 중심으로 공간을 이등분하여 두 개의 반공간을 구성하게 된다.

그림 10에서 보는 바와 같이 하이퍼플레인에 의한 클러스터링은 한 개체가 두 개의 중심값에 대하여 $(\mathbf{B}-\mathbf{x_0}) \cdot (\mathbf{x}-\mathbf{x_0}) \geq 0$를 만족하면 B그룹에, $(\mathbf{B}-\mathbf{x_0}) \cdot (\mathbf{x}-\mathbf{x_0}) < 0$를 만족하면 A그룹에 속하게 되는 방법이다.



[그림 10] 두 지점 A와 B사이의 하이퍼플레인에 의한 임의의 값 x의 그룹 결정

유사도 측정에 의한 패스의 1단계 클러스터링 후 K-평균 알고리즘을 통한 2단계 클러스터링을 한다. 그러나 K-평균 알고리즘은 많은 횟수의 계산과 비교를 하게 된다. 이러한 단점을 줄이는 방법으로 하이퍼플레인을 이용한 방법을 사용하게 되었다. 하이퍼플레인을 이용하면 2개의 그룹 중심값을 기준으로 $(\mathbf{B}-\mathbf{x_0}) \cdot (\mathbf{x}-\mathbf{x_0})$ 값을 계산하여 해당 개체의 2개의 그룹 중심값과의 근접도를 결정할 수가 있다. 이때 모든 가능한 그룹쌍의 조합에 대한 하이퍼플레인에 의한 그룹 결정이 이루어지는 것이 아니라, 비교하여 우위인 것과 비교하지 않은 나머지 그룹의 중심값을 분할하는 하이퍼플레인에 의한 그룹 결정을 함으로써 비교횟수를 현격하게 줄일 수 있다.

하이퍼플레인을 이용한 클러스터링 알고리즘은 다음과 같다.

① 몇 개의 그룹을 생성할 것인지 k를 결정.
② 각 그룹의 중심 값으로 임의의 값을 할당.
③ 2개로 이루어진 그룹 쌍(예를 들어, A, B)의 하이퍼플레인과 수직으로 교차하는 x_0를 구한다.
④ 각각의 데이터 x에 대해 $B-x_0$ 벡터와 $x-x_0$벡터의 내

적을 구하여 각 그룹 쌍에서 소속 그룹을 결정, 전체 그룹 쌍에 대하여 반복

$(B-x_0)\cdot(x-x_0) > 0 \rightarrow B$

$(B-x_0)\cdot(x-x_0) \leq 0 \rightarrow A$

⑤ 가장 많이 포함되는 그룹을 소속그룹 결정

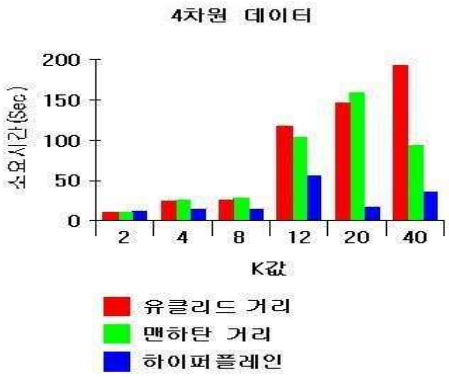
⑥ 각 그룹에 속하는 모든 데이터에 대한 평균값을 구해 새로운 중심 값으로 결정

⑦ 기존의 중심 값과의 차이를 계산

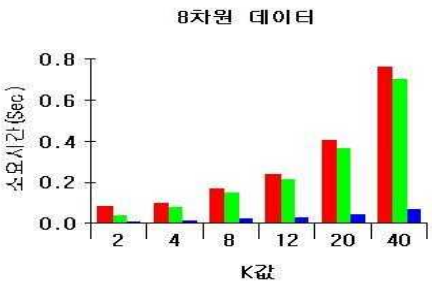
⑧ 중심 값의 차이가 0에 근접할 때까지 ③부터 반복

4. 실험 및 알고리즘의 성능 분석

K-평균 알고리즘의 거리에 대한 대표적인 정의로서 유클리드 거리와 맨하탄 거리를 이용한 기존의 클러스터링과 하이퍼플레인을 이용한 클러스터링을 비교 실험하였다. 4000개의 데이터를 가지고 실험하였으며, 4차원과 8차원 데이터에 대하여 K값을 2, 4, 8, 12, 20, 40으로 변화시키면서 소속그룹 결정에 대한 Loop 당 소요시간, Loop수, 그리고 전체 소요시간을 측정하였다. 그 결과를 그림 11 ~ 그림 16에서 그래프로 제시하였다.

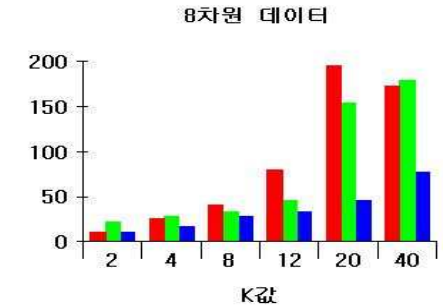


[그림 11] 4차원 데이터 Loop 당 소요시간



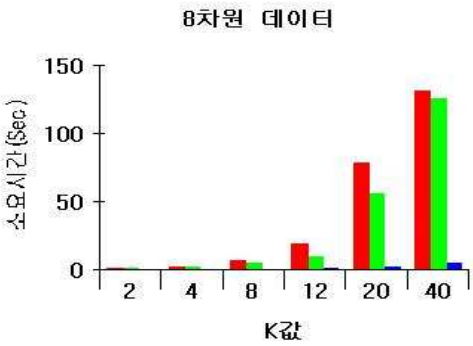
[그림 12] 8차원 데이터 Loop 당 소요시간

[그림 13] 4차원 데이터 Loop 수



[그림 14] 8차원 데이터 Loop 수

[그림 15] 4차원 데이터 전체 소요시간



[그림 16] 8차원 데이터 전체 소요시간

위 결과를 볼 때, 하이퍼플레인을 이용한 클러스터링은 기존의 유클리드 거리나 맨하탄 거리에 의한 클러스터링과 비교했을 때 Loop 당 소요시간이나 전체 소요시간에 대하여 큰 효율성을 보이는 것을 알 수 있다. 이러한 결과는 하이퍼플레인을 이용한 클러스터링이 기존의 유클리드 거리 혹은 맨하탄 거리 계산방법에 비하여 계산이 적기 때문에 나타나는 결과라고 할 수 있다. 다시 말해, 두 그룹 중심점을 양분하는 하이퍼플레인을 정의하고 각 데이터가 양분된 부분공간 중 어디에 속하는지만 판단하면 되기 때문에(그림 10 참조) 각 벡터의 거리를 모두 계산해야하는 기존의 방법보다 적은 계산 양과 계산 시간을 필요로 한다. 또한, 두 점 사이의 거리를 할 때 $(\mathbf{B}-\mathbf{x}_0)$를 모든 레코드에 대하여 반복적으로 사용하게 되는데, 이러한 반복되는 부분을 저장하여 이용하기 때문에 산술 계산의 양을 줄일 수 있다.

또한 실험을 통해서 발견한 사항은 유클리드 거리나 맨하탄 거리와 비교했을 때 하이퍼플레인을 이용한 클러스터링에서 Loop 횟수가 적은 것을 알 수 있다. 이것은 1단계의 우수한 초기해를 가지고 알고리즘을 수행한 결과이다.

5. 결론

웹사용자 클러스터링에 대하여 본 논문에서는 사용자 웹 방문 패턴을 인식하고 이를 클러스터링하기 위하여 사용자의 세션별로 방문 패스에 대한 서브시퀀스를 형성하여 이를 특징벡터로 사용하였다. 따라서 방문 패스에 내재되어 있는 고유의 순차적인 흐름과 부분적인 특성을 파악할 수 있었다. 비최적해에 수렴하는 것을 방지하기 위하여 첫 번째 단계 우수한 초기해를 구하고 이를 2단계에서 개선하는 방식의 2단계 알고리즘을 제안하였다. 2단계에서는 기존의 K-평균 알고리즘의 많은 계산량을 줄이기 위하여 하이퍼플레인을 이용한 클러스터링 기법을 도입하여 알고리즘의 효율성 향상을 높였다.

또한 1단계 클러스터링으로 그룹화 된 결과를 그룹별로 해시 체이닝을 이용하여 저장하여 탐색과 계산의 효율을 높이는 결과를 얻을 수 있었다.

참고문헌

[1] 김종달, “웹 로그에서 웹 방문 패턴을 이용한 사용자 웹 방문 패스 클러스터링”, 포항공과대학 석사학위논문, 2002.

[2] Baraglia, R. Silvestri, F. "Dynamic personalization of web sites without user intervention", In Communication of the ACM 50(2): 63-67, 2007.

[3] Cooley, R. Mobasher, B. and Srivastave, J. “Web Mining: Information and Pattern Discovery on the World Wide Web” In Proceedings of the 9th IEEE International Conference on Tool with Artificial Intelligence, 1997.

[4] Cyrus Shahabi, Amir M. Zarkesh, Jafar Adibi, and Vishal Shah "Knowledge Discovery from Users Web-Page Navigation", RIDE, 1997.

[5] 강현철, 한상태, 최종후, 김은석, 김미경, “데이터마이닝(방법론 및 활용)”, 자유아카데미, 2001. 2

[6] 장남식, 홍성완, 장재호. “(성공적인 지식경영을 위한 핵심정보기술) 데이터 마이닝”, 대청, 2000.

[7] 조재희, 박성진, “데이터 웨어하우징과 OLAP”, 대청, 2000.

[8] Vassilvitskii, S. "How Slow is the k-means Method?". Proceedings of the 22nd Symposium on Computational Geometry (SoCG),. 2006.

[9] Olfa Nasraoui, Hichem Frigui, Raghu Krishnapuram, Anupam Joshi, "EXTRACTING WEB USER PROFILES USING RELATIONAL COMPETITIVE FUZZY CLUSTERING", Intl. J. Artificial Intelligence Tools, 2000.

[10] Jan Kerkhofs, Prof. Dr. Koen Vanhoof, Danny Pannemans, "Web Usage Mining on Proxy Servers: A Case Study", http://www. docstoc.com/docs/28616441/Web-Usage-Mining-on-Proxy-Servers-A-Case-Study

[11] Cottrell M., Fort J.C., Pagès G., Two or three things that we know about the Kohonen algorithm, in Proc of ESANN, M. Verleysen ED., D Facto, Bruxelles, 1994.

[12] Chris Ding and Xiaofeng He. "K-means Clustering via Principal Component Analysis". Proc. of Int'l Conf. Machine Learning (ICML 2004), pp 225-232, 2004.

[13] Charles W. Curtis Linear Algebra, page 62, Allyn & Bacon, Boston, 1968.

[14] Mokhtar S. Bazaraa, John J. Jarvis, "LINEAR PROGRAMMING AND NETWORK FLOWS", JOHN WILEY & SONS.

이 해 각(Hae-Kag Lee) [정회원]

- 1987년 2월 : 한국과학기술원 산업공학과 (산업공학 석사)
- 1992년 8월 : 한국과학기술원 산업공학과 (산업공학 박사)
- 1992년 9월 ~ 현재 : 순천향대학교 컴퓨터공학과 교수

<관심분야>
데이터베이스, 데이터 마이닝, 공장자동화