협력적 여과 시스템을 위한 효과적인 사용자 군집 알고리즘

고 수 정[†] · 임 기 욱[††] · 이 정 현[†††]

요 약

협력적 여과 시스템은 사용자가 검색하고 읽었던 웹문서를 기반으로 사용자 군집을 생성하여 웹문서의 정확한 추천을 가능하게 한다. 이러한 목적으로 설계된 다양한 알고리즘이 있으나 속도가 느리거나 정확도가 낮다는 등의 단점이 있다. 본 논문에서는 이러한 단점을 보완하기 위하여 협력적 여과 시스템을 위한 효과적인 사용자 군집 알고리즘인 CUG알고리즘을 제안한다. CUG 알고리즘은 사용자 군집을 생성하기 위해 Apriori 알고리즘, Naïve Bayes 알고리즘을 이용한다. Apriori 알고리즘은 연관 단어 지식 베이스를 구축하고, Naïve Bayes 알고리즘은 구축된 연관 단어 지식 베이스에 가중치를 추가하며, 사용자가 검색하여 읽은 웹문서를 클래스별로 분류한다. CUG알고리즘은 분류된 웹문서를 기반으로 하여 사용자 군집을 만든다. 이러한 방법으로 설계된 CUG 알고리즘은 사용자들이 사용할 문서를 미리 검색하여 저장함에 의해 정보검색의 효율성을 향상시키는데 사용될 수 있다. 본 논문에서 설계한 CUG 알고리즘의 성능을 평가하기 위하여 기존의 K-means 방법과 Gibbs샘플링 방법에 의한 군집과 비교한다.

Effective User Clustering Algorithm for Collaborative Filtering System

Su-Jeong Ko[†] · Kee-Wook Rim[††] · Jung-Hyun Lee[†††]

ABSTRACT

Grouping user into clusters based on the web documents they have retrieved allows accurate recommendations of new web documents. A variety of algorithms have previously been designed with shortcomings of slow speed or low accuracy. For the purpose of complementing these shortcomings in this paper, we propose CUG algorithm, which is effective user clustering algorithm for collaborative filtering system. CUG algorithm uses Apriori algorithm, Naïve Bayes algorithm in order to make clusters for users. Apriori algorithm constructs association word knowledge base. Naïve Bayes algorithm adds weight to the association word knowledge base and categorizes web documents retrieved by user into classes. CUG algorithm groups users into clusters based on these web documents. CUG algorithm designed through these methods can be used to improve the efficiency of information retrieval by prefetching documents for the users and storing them in a document database in the system. For the purpose of evaluating performance for CUG algorithm designed in this paper, we compare our algorithm with methods of K-means clustering and Gibbs sampling.

키워드 : 협력적필터링(Collaborative filtering system), 웹마이닝(Web mining), Apriori 알고리즘(Apriori algorithm), 지식 베이스(Knowledge base), CUG 알고리즘(CUG algorithm)

1. 서 론

대부분의 여과 시스템은 사용자 프로파일에 기반을 두고 정보를 여과한다. 이러한 시스템은 어떠한 정보를 여과할 것인가를 결정하기 위해 브라우저에서 사용자가 검색하는 행동을 관찰하여 사용자의 유형을 미리 추출한다. 그러나 이러한 시스템은 초기에는 설정된 정보가 없다는 문제점이 있다[10, 19]. 즉, 새로운 사용자는 프로파일에 아무런 정보가 없는 상태에서 검색을 시도해야 하므로 먼저 프로파일을 만들어야 한다는 번거로움을 겪어야 한다[7, 22]. 협력적 여과 시스템은 이러한 문제점을 해결하기 위해 여러 사용자들의 경험을 바탕으로 공통된 검색유형을 찾아서 초기의 프로파일로 활용한다[11, 15, 17]. 본 논문에서는 초기의 프로파일을 구성하기 위해서 연관 웹 문서 분류 정보와 사용자 순차 브라우징 패턴 정보를 이용하여 공통된 검색유형을 찾아 웹문서를 추천하는 동적 링킹 시스템[24]을 이용한다. 동적 링킹 시스템은 사용자가 하나의 문서를 검색했을 때 여러 관련 문서를 추천함에 따라 사용자에게 초기에 설정된 문서가 없을지라도 추천된 모든 문서를 초기의 프로파일에 활용함에 따라 프로파일을 만들기 위해 사용자의 정보를 설정해야 하는 번거로움을 겪지 않아도 된다.

본 논문에서는 협력적 여과 시스템[8, 12]을 위한 사용자 군집 방법에 초점을 둔다. 사용자를 군집하기 위한 기존의

† 정 회 원 : 인하대학교 대학원 전자계산학과
†† 종신회원 : 선문대학 산업공학과 교수
††† 종신회원 : 인하대학교 전자계산공학과 교수
논문접수 : 2000년 12월 27일, 심사완료 : 2001년 3월 19일

알고리즘은 각기 단점을 갖고 있다[20]. 예를 들어, EM[14]은 사용자를 군집하기 위해 편리한 방법이나 같은 군집에 속한 사용자들이 검색한 문서는 같은 클래스에 있어야 한다는 제한을 갖는다. 또한, K-means 군집 방법[4]은 빠르나 정확도가 낮으며 Gibbs 샘플링 방법[5]은 정확도는 높으나 복잡하기 때문에 속도가 느리다.

이러한 방법들의 단점을 보완하기 위하여 본 논문에서는 CUG 알고리즘을 제안한다. CUG 알고리즘은 사용자가 검색하여 읽은 웹문서를 기반으로 사용자들 간의 군집을 생성한다. CUG 알고리즘은 사용자의 군집을 만들기 위해서 먼저 연관 단어 지식 베이스를 구축한다[18]. 다음으로, 구축된 연관 단어 지식 베이스를 대상으로 Naïve Bayes 학습을 함으로써 가중치를 부여한다. Naïve Bayes 분류자[16, 23]는 가중치가 부여된 연관 단어 지식 베이스의 하나의 클래스로 사용자가 검색한 후 찾은 웹문서를 분류한다. CUG 알고리즘은 분류된 웹문서를 기반으로 하여 사용자 군집을 만든다.

제안된 CUG 알고리즘은 웹문서를 미리 찾아내고 저장함으로써 정보검색의 효율성을 향상시킬 수 있다. 본 논문에서 설계한 CUG 알고리즘의 성능을 평가하기 위하여 기존의 K-means 방법과 Gibbs샘플링 방법에 의한 군집과 비교한다.

2. 연관 단어 지식 베이스를 이용한 Naïve Bayes 알고리즘과 CUG 알고리즘

2.1 연관 단어 지식 베이스

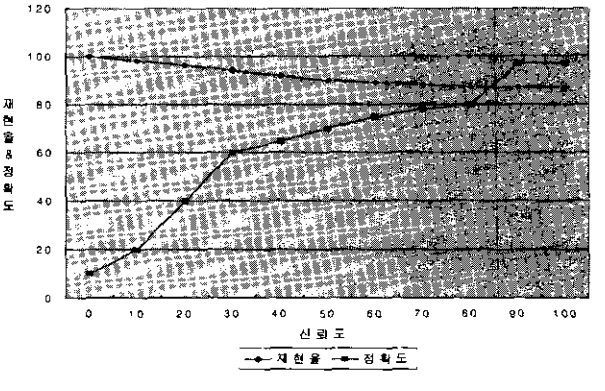
Apriori 알고리즘[1, 2]은 데이터 마이닝 기법을 이용하여 단어 간의 연관 규칙을 추출한다[21]. 연관 규칙은 "W1->W2"로 표현할 수 있는데, 여기서 W1과 W2는 문서로부터 추출된 단어로 항목들의 집합이다. 이와 같은 형태의 연관 규칙은 "W1을 포함하고 있는 데이터베이스 내의 트랜잭션은 W2도 함께 포함하고 있다"는 것을 의미한다. 이러한 경우에 사용되는 데이터베이스는 웹문서의 이름과 웹문서에서 추출된 명사들로 구성되는데, 알고리즘에 사용되는 빈발 단어 항목과 후보 단어 항목은 웹문서를 대상으로 형태소 분석을 통해 추출된 명사이다.

단어 간의 연관 규칙을 마이닝하는 단계는 두 단계로 구성된다. 첫 번째 단계에서는 최소의 지지도(min_support) 이상의 발생 지지도(transaction support)를 가지는 조합을 찾아 빈발 단어 항목을 구성한다. 두 번째 단계에서는 데이터베이스로부터 연관 규칙을 생성하기 위하여 빈발 단어 항목을 사용한다. 모든 빈발 단어 항목 (L)에 대해서 빈발 단어 항목의 모든 공집합이 아닌 부분집합들을 찾는다. 각각의 그러한 부분집합(A)에 대하여, 만약 support(A)에 대한 support(L)의 비율이 적어도 최소 신뢰도(min_confidence) 이상이면, A->(L-A)의 형태의 규칙을 출력한다. 이 규칙의 지지도는 support(L)이고, 신뢰도는 support(L)/support(A)이다.

연관 단어 쌍을 구성하기 위해서는 신뢰도(confidence)와 지지도(support)를 결정해야 한다. 신뢰도를 결정하기 위한 식 (1)은 다음과 같이 구해진다. 식 (1)은 단어W1과 W2의 모든 항목을 포함하고 있는 트랜잭션의 수를 단어 W1의 항목을 포함하고 있는 트랜잭션의 수로 나눈 결과 값을 나타낸다.

$$Confidence(W1->W2) = Pr(W2|W1) \quad (1)$$

(그림 1)은 100개의 웹문서를 대상으로 신뢰도를 다양하게 변화시켰을 때, 추출된 연관 단어에 대한 정확도와 재현율을 나타낸다. 100개의 웹문서는 본 논문이 실험을 위해 컴퓨터 분야의 웹문서를 8개의 클래스로 분류한 클래스 중에서 게임 클래스에 수집된 웹문서이다. 웹문서 수집은 웹문서 수집기를 이용한다. 마이닝된 결과에 대해 재현율과 정확도를 평가하는 기준은 영어 단어에 대한 시소러스인 WordNet[9]을 사용하여 평가하였다. 평가를 위해 WordNet에서 게임과 관련된 영어 단어의 동의어, 상의어, 하의어를 추출하였다. 추출된 단어를 한글로 번역하여 300개의 연관 단어를 구성하였다. 마이닝된 연관 단어가 이들 300개의 연관 단어에 포함되지 않을 경우 오류로 처리하였다. 정확도는 마이닝된 연관 단어 중에서 오류로 처리된 연관 단어의 비율을 나타낸다. 재현율은 마이닝된 연관 단어가 평가를 위해 구성된 연관 단어에 포함된 비율이다.



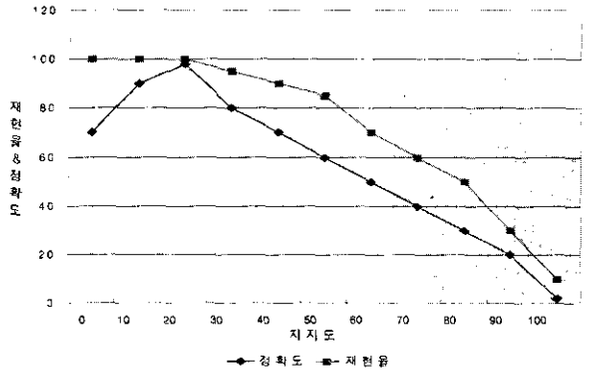
(그림 1) 신뢰도의 변화에 따른 재현율과 정확도

위의 그림은 신뢰도가 클수록 추출되는 연관 단어의 정확도는 높아지나 재현율은 낮아짐을 나타낸다. 그러나 85이상의 신뢰도에서는 재현율이 거의 일정하고 정확도는 높은 수치를 나타낸다. 따라서 가장 적합한 연관 단어를 추출하기 위해서는 신뢰도를 85이상으로 지정해야 한다.

지지도를 결정하기 위한 식 (2)는 전체 단어들의 쌍 중에 각 연관 단어의 출현 빈도를 나타낸다. 식 (2)는 단어 W1과 W2의 모든 항목을 포함하고 있는 트랜잭션의 수를 데이터베이스 내의 전체 트랜잭션의 수로 나눈 결과 값을 나타낸다.

$$Support(W1->W2) = Pr(W1 \cup W2) \quad (2)$$

지지도가 크다면 빈도수는 작으나 중요한 연관 단어가 생략될 수 있고, {기본&방식&이용&지정=>실행}과 같이 빈도수는 높지만 중요하지 않은 연관 단어가 추출된다. (그림 2)는 100개의 웹문서를 대상으로 지지도를 다양하게 변경시킴에 따른 정확도와 재현율의 변화를 나타낸다. 정확도와 재현율의 측정 기준은 신뢰도와 같다.



(그림 2) 지지도의 변화에 따른 재현율과 정확도

정확도와 재현율의 곡선이 일치하는 지점은 지지도가 22인 경우로, 이 지점에서 가장 적합한 연관 단어가 추출된다. 그러나, 지지도가 22이상인 경우에는 정확도와 재현율이 모두 낮아진다. 따라서 가장 신뢰할 만한 연관 단어를 추출하기 위해서는 22이하의 지지도를 지정해야 한다. 그러나 지지도를 0으로 한다면 클래스와 전혀 관계 없는 문서에서 연관 단어가 추출되므로 0보다 크도록 설정하여야 한다.

2.2 Naïve Bayes 알고리즘

Naïve Bayes 알고리즘은 학습 단계와 분류 단계를 통하여 웹문서를 분류할 수 있다. 이 알고리즘을 이용할 경우, 구축된 연관 단어 지식 베이스는 학습 단계와 분류 단계에서 모두 이용된다.

학습 단계에서는 지식 베이스의 연관 단어에 가중치를 부여한다. 가중치를 부여하기 위해서 우선 가중치를 부여하기 위한 훈련 집합을 구성한다. 구성된 훈련 집합으로부터 연관 단어를 마이닝한다. 마이닝된 연관 단어는 웹문서에 나타난 위치에 관계없이 독립적이라고 가정한다.

그러한 이유는 지식 베이스의 연관 단어에 가중치를 부여할 경우 마이닝된 연관 단어의 수만을 필요로 하기 때문이다. 이러한 가정하에 지식 베이스에 있는 연관 단어 aw_k의 가중치를 부여하기 위해서 식 (3)을 사용한다. 여기서, n은 훈련 집합으로부터 마이닝된 연관 단어의 전체 수이고, n_k는 전체 개수 n 중에서 지식 베이스에 있는 연관 단어 aw_k와 일치하는 단어의 수이다. 또한, classID는 지식 베이스에 있는 클래스의 번호이며, |AWKB|는 지식 베이스에 있는 전체 연관 단어의 수이다. N은 가중치를 더하기 위해 수집한 훈련 집합으로부터 마이닝된 연관 단어를 의미하므로 연관 단어 지식 베이스와는 관련이 없다. 따라서 분모에 연관 단어 지식 베이스의 크기를 추가함으로써 모든 훈련 문서로부터 마이닝된 연관 단어에 대한 정보를 표현하였다. 이에 따라 더욱 많은 정보를 사용하여 가중치를 부여함으로써 문서 분류의 정확도가 높아진다. 또한 분자에는 n_k에 1을 더하여 확률이 0이 되는 것을 예방하였다.

$$P(aw_k \mid classID) = \frac{n_k + 1}{n + |AWKB|} \tag{3}$$

연관 단어 지식 베이스는 식 (3)을 사용하여 가중치가 추가된다. 이러한 경우, 지식 베이스의 연관 단어가 훈련 집합에서 많이 출현할 경우 그 연관 단어에게는 높은 가중치가 부여되며, 드물게 출현할 경우 낮은 가중치가 부여된다.

분류 단계에서는 가중치가 부여된 연관 단어 지식 베이스를 사용하여 Naïve Bayes분류자에 의해 웹문서를 클래스별로 분류할 수 있다. 사용자가 검색하여 읽은 웹문서는 D={$daw_1, daw_2, \cdots, daw_n$}로 표현하여, {$daw_1, daw_2, \cdots, daw_n$}는 문서 D로부터 추출된 연관 단어이다. 또한 웹문서가 분류될 클래스는 classID로, 전체 클래스는 *classnum*로 표현한다. 또한 *P(class)*는 문서D의 전체 연관 단어 중 각 클래스에 포함된 연관 단어의 비율을 나타낸 것이다. 식 (4)는 이러한 표현을 사용하여 정의되며, 이 식에 의해 웹문서는 각 클래스로 레이블된다.

$$classID = \underset{class=1}{\overset{classnum}{MAX}} P(class) \sum_{i=1}^{n} P(daw_i \mid class) \tag{4}$$

분류된 문서를 사용하여 키워드 문서 데이터베이스를 구성할 수 있다. 키워드 문서 데이터베이스는 {userID, docID, classIDpos, classID}의 구조를 가지며, 여기서 classIDpos는 사용자가 검색하여 읽은 전체 문서 중에서 각 클래스로 분류된 문서의 비율을 의미하며, 식 (5)로 표현된다. 식 (5)에서 분모는 클래스{1, 2, ⋯, classID, ⋯, classnum}의 전체 클래스에 분류된 문서의 수를 의미한다. 분자는 클래스(classID)에 분류된 문서의 수를 나타낸다.

$$classIDpos = \frac{|doc|_{classID}}{\sum_{i=1}^{classnum} |doc|_i} \tag{5}$$

2.3 CUG 알고리즘

(그림 3)에서 제시된 CUG알고리즘은 분류된 문서를 기반으로 사용자의 군집을 생성한다.

(그림 3)의 CUG 알고리즘은 다음의 상황이 발생하는가를 항상 모니터링하고 있다가 두 가지 중 한 가지의 상황이 발생하면 사용자에 대해 새로운 그룹을 지정하고자 시도한다.

```
/* usergroupID : 사용자에게 배정된 groupID
 Similarity_Calculation( ) ; 그룹에 속한 사용자와 새로운 사
                             용자 간의 유사도 계산 함수
Group[group_num] : 전체 그룹을 순회하며 가장 큰 유사도를 갖는
                   그룹으로 사용자를 배정하기 위한 배열 */
Do while {(new user) .or. (existent user whose interest is
          changed)} ≠ ∅ begin
{
  /* 새로운 사용자-첫번째 그룹부터 시작*/
     j <- 1 ;
     If (new user)
        Then groupID <- 1 ;
  /* 기존에 있는 사용자- 현재의 그룹부터 시작 */
     If (existent user whose interest is changed)
        Then groupID <- current groupID ;
  /* 전체 그룹을 순회하며 적합한 그룹 탐색 */
  do while (group ≠ ∅) begin
  {
    P_sim = Similarity_Calculation( ) ;
    Case P_sim<similarity_threshold
       Go to next group ;
    /* 그룹을 변경하지 않을 경우 기존의 그룹이 배정됨*/
    Case (P_sim>=similarity_threshold .and. groupID = current
                 groupID)
     {user_groupID <- current groupID;
      Quit}
   /*유사도를 배열에 저장하고 다음 그룹으로 가서 유사도계산*/
    Otherwise
     {Group[groupID] <- P_sim;
     j++; Goto next group}
  }
enddo
   /* 사용자에게 그룹 배정 */
                                    group_num
 If j> 0 then user_groupID}<-  MAX Group[ i]
                                    i  1
                          //유사도가 가장 큰 그룹에 사용자배정
  else user_groupID <- ++group_num //적당한 그룹이 없으므로
                                   새로운 그룹을 생성
enddo
```

(그림 3) 사용자 군집을 위한 CUG알고리즘

첫 번째 상황은 새로운 사용자가 문서를 검색하여 읽은 경우이다. 이러한 경우, 동적 링킹 시스템에 의해 추천 받은 문서와 사용자가 검색한 문서로 임시의 프로파일을 작성하여, 첫 번째 그룹부터 시작하여 전체 그룹을 순회하며 그룹에 속해 있는 사용자들과 유사도를 계산한다. 사용자는 전체 그룹을 순회한 후에 평균 유사도가 임계값 이상인 그룹에 속하게 한다. 만일, 유사도가 임계값 이상인 그룹이 하나 이상일 경우, 사용자는 이러한 그룹 중에서 평균 유사도가 가장 큰 그룹으로 군집된다. 그룹에 속한 후에 그룹 사용자들의 공통된 경험을 바탕으로 추천된 문서를 기반으로 새로운 사용자의 초기 프로파일이 생성된다.

두 번째 상황은 사용자에 의해 검색된 문서가 추가되었거나 기간이 경과하여 삭제된 경우이다. 문서가 추가되거나 삭제되었다면 사용자가 검색하여 읽은 문서에 대한 정보를 저장하고 있는 키워드 문서 데이터베이스는 수정된다. 이때, CUG 알고리즘은 키워드 문서 데이터베이스의 내용이 수정된 사용자의 그룹을 변경할 필요성이 있는가를 탐색한다. 이 알고리즘은 전체 그룹을 탐색하기 전에 현재 사용자가 속한 그룹의 나머지 사용자들과 유사도를 계산한다. 계산 결과, 평균 유사도가 임계값 이상이면 다른 그룹을 탐색하지 않고 현재의 그룹을 사용자의 그룹으로 지정한다. 이에 따라 다른 그룹에 있는 사용자들과의 유사도를 계산하지 않음에 따라 사용자 군집을 위한 시간을 단축시킬 수 있다. 반면, 사용자와 나머지 사용자들과의 평균 유사도가 임계값 미만이면 전체 그룹을 순회하며 첫번째 경우와 같은 방법으로 새롭게 군집을 변경한다. 만약 그룹 에이전트가 전체 그룹을 순회하였음에도 불구하고, 사용자에게 배정할 그룹이 없다면 사용자를 배정할 새로운 그룹을 생성한다. (그림 4)는 (그림 3)에 속한 함수 *Similarity_Calculation( )*을 구체적으로 나타낸 그림이다.

Function ***Similarity_Calculation***( )

Step 1 : {Puser}를 행렬로 표현

$$PUSER = [\text{Pclass1pos}, \text{Pclass2pos}, \cdots, \text{Pclassnpos}]$$

$$GU = \begin{matrix} GU\ 1class\ 1\ pos & GU\ 2class\ 1\ pos & \cdots & GUusernumc\ lass\ 1\ pos \\ GU\ 1class\ 2\ pos & GU\ 2class\ 2\ pos & \cdots & GUusernumc\ lass\ 2\ pos \\ \vdots & \vdots & \vdots & \vdots \\ GU\ 1classnpos & GU\ 2classnpos & \cdots & GUusernumc\ lassnpos \end{matrix}$$

Step 2 : {*GU1, GU2, …, GUusernum*}를 행렬로 표현

Step 3 : {Puser}와 그룹 안에 있는 사용자와 유사도 계산

$$P\_sim = PUSER \otimes GU = \frac{1}{usernum} \sum_{j=1}^{usernum} V$$

$$V = \frac{\sum_{i=1}^{classnum} Pclassipos\ GUjclassipos}{\sum_{i=1}^{classnum} Pclassipos^2 + \sum_{i=1}^{classnum} GUjclassipos^2 + \sum_{i=1}^{classnum} Pclassipos\ GUjclassipos}$$

(그림 4) {Puser}와 그룹에 있는 사용자와의 유사도 계산 함수

(그림 4)에 나타난 함수 *Similarity_Calculation( )*은 사용자와 그룹에 있는 사용자와의 평균 유사도를 계산한다. 여기서, {Puser}은 새로운 사용자이거나 사용자 프로파일의 내용이 변경된 사용자를 의미한다. 이러한 사용자는 (그림 4)의 Step 1과 같이 표현한다. Step 1에서 *PUSER*는 {Puser}를 표현하는 *1 x classnum*의 행렬이다. 행렬 *PUSER*는 식 (5)에 의해 계산된 각 클래스에 분류된 문서의 비율값을 갖는다. {*GU1, GU2, …, GUusernum*}은 그룹 안에 속한 사용자들을 의미하며 (그림 4)의 Step 2와 같이 표현된다. *GU*는 {GU1, GU2, …, GU*usernum*}을 표현하는 *usernum x classnum*의 행렬이다. 여기서 *usernum*은 그룹 안에 속한 사용자의 수이며 *classnum*은 전체 클래스의 수이다.

(그림 4)의 Step 3에서는 {Puser}와 그룹 안에 있는 사용자와의 유사도를 구하기 위해 식 (6)의 자카드 계수를 사용한다[6]. 식 (6)은 user n과 user m이 각 클래스로 분류된 문서를 가지는 각각의 확률의 합에 대한 user n과 user m이

동일한 클래스에 공통으로 분류된 문서들이 나타날 확률의 비율을 나타낸다. 본 논문에서의 Ø연산자는 자카드 계수에 의한 식임을 나타낸다.

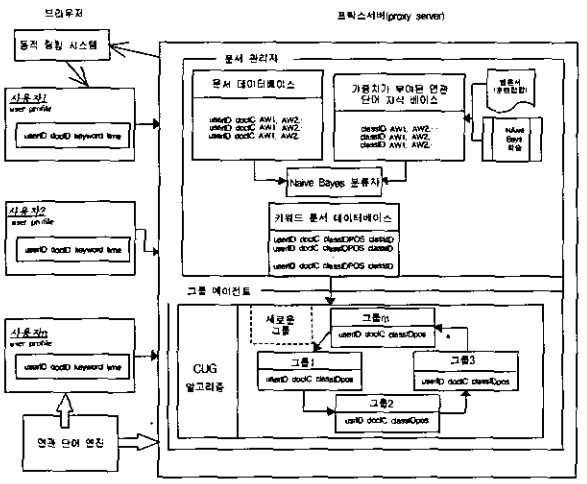
$$\{user\ n\} Ø \{user\ m\} = \#(user_n \cap user_m)/\#(user_n \cup user_m) \quad (6)$$

식 (6)을 이용하여 {Puser}와 그룹에 있는 사용자들{GU1, GU2, ···, GU*usernum*}의 평균 유사도는 식 (7)에 의해 표현한다.

$$P\_sim = PUSER Ø GU \quad (7)$$

3. 협력적 여과 시스템을 위한 설계 및 표현

(그림 5)는 CUG알고리즘을 이용한 협력적 여과 시스템의 구성도를 나타낸다.



(그림 5) 협력적 여과 시스템 구성도

3.1 연관 단어 엔진

연관 단어 엔진은 웹문서로부터 중요한 연관 단어를 추출하는 역할을 한다. 연관 단어는 2단계를 통하여 추출될 수 있다. 첫 번째 단계에서는 웹문서를 대상으로 형태소 분석을 한 결과로부터 명사만을 추출한다. 두 번째 단계에서는 Apriori 알고리즘을 사용하여 추출된 명사로부터 중요한 연관 단어를 마이닝한다. 이 단계에서 연관 단어를 마이닝하기 위해서는 어떠한 목적으로 연관 단어를 사용할 것인가에 따라 신뢰도와 지지도를 다르게 지정해야 한다. 신뢰도와 지지도를 다르게 지정함에 따라 마이닝되는 연관 단어는 다르기 때문이다.

3.2 동적 링킹 시스템

동적 링킹 시스템[24]은 사용자 브라우징 패턴 분석 시 데이터마이닝의 연관 규칙 결정 알고리즘이 아닌 순차 패턴 알고리즘인 Apriori_all[3]을 사용하여, 사용자들의 웹사이트 방문에 대한 순차적 특성을 고려하여, 사용자의 현재 세션과 가장 유사한 사용자 브라우징 패턴 정보를 생성한다. 또한 일정한 목적을 가지고 방문하는 사용자들이 찾고자 하는 문서와 가장 근접한 문서들을 추천하기 위해서 Bayesian분류자를 이용하여 연관 단어 지식 베이스의 클래스에 분류된 연관 웹 문서를 이용한다. 이러한 동적 링킹 시스템은 기존의 전체 사용자에 대한 정보를 이용하여 문서를 추천하는 첫 번째 경우와 그룹 내 사용자의 정보를 이용하여 문서를 추천하는 두 번째 경우로 구분된다.

첫 번째 경우는 최초로 사용자가 검색을 시작할 때 기존 모든 사용자의 브라우징 패턴 정보와 연관된 문서 정보를 이용하여 사용자에게 적절한 문서를 추천하여 임시의 사용자 프로파일을 구성하는 역할을 한다. 따라서 임시의 사용자 프로파일은 검색을 시작한 문서와 동적 링킹 시스템에 의해 추천된 모든 문서로 구성된다. CUG알고리즘은 임시의 사용자 프로파일을 이용하여 사용자를 그룹으로 군집한다.

두 번째 경우의 동적 링킹 시스템은 CUG 알고리즘에 의해 사용자가 한 그룹으로 군집된 후 사용자와 유사한 흥미를 갖는 그룹 내의 사용자들의 브라우징 패턴 정보와 분류된 연관 웹 문서를 이용하여 보다 구체적인 문서를 추천한다. 이를 기반으로 임시의 사용자 프로파일을 보다 구체적으로 추가하여 사용자의 초기 프로파일을 생성한다. 이에 따라 초기에 사용자의 정보가 있어야만 프로파일을 생성하는 기존의 여과 시스템의 단점을 해결한다.

3.3 가중치가 부여된 연관 단어 지식 베이스

연관 단어 지식 베이스를 만들기 위해서는 우선 훈련 집합을 만들어야 한다. 훈련 집합은 웹문서 수집기에 의해 수집된 웹문서로 구성한다. 연관 단어 엔진은 훈련 집합으로부터 연관 단어를 마이닝한다. Naïve Bayes 알고리즘[16]은 추출된 연관 단어를 대상으로 학습을 한다. 그 결과, 연관 단어 지식 베이스의 연관 단어에 가중치가 부여된다.

3.4 웹 브라우저와 프락시 서버

인터넷에 접속하고자 하는 일반 사용자는 넷스케이프나 익스플로러 등 상용의 웹 브라우저를 이용한다. 사용자가 (그림 5)의 협력적 여과 시스템을 이용하기 위해서는 단순히 브라우저의 환경설정을 인터넷에 직접 연결하는 대신 프락시 서버를 통하여 연결되도록 설정하면 가능하다. 이러한 경우 프락시 서버는 로그파일 안에 사용자에 대한 정보를 갖고 있다. 사용자가 HTTP서비스를 요구하면 인터넷에 바로 연결되는 것이 아니라 프락시 서버를 통하여 연결된다. 연결된 후, 사용자는 원하는 웹문서를 찾으며, 찾아진 문서는 연관 단어 엔진에 의해 키워드가 추출된다. 추출된 키워드는 사용자의 프로파일에 저장된다. (그림 5)의 문서 관리자는 수정된 사용자 프로파일의 내용을 문서 데이터베이스에 저장한다.

프락시 서버는 사용자의 이름과 요청된 URL, 문서를 읽은

시간을 로그파일에 기록하므로 사용자에 대한 정보를 로그파일에 저장할 수 있다. 이 논문에서는 URL을 docID라고 정의한다. 사용자 프로파일은 {userID, docID, 키워드, docID를 읽은 시간}의 형식으로 구성된다.

3.5 문서 관리자

문서 관리자는 사용자 프로파일과 문서 데이터베이스를 중개하며 문서 데이터베이스를 관리하는 역할을 한다. 문서 데이터베이스는 사용자가 검색한 문서에 대한 정보와 연관 단어 엔진에 의해 추출된 키워드를 모두 갖고 있다. 문서 데이터베이스는 {userID, docID, aw_1, $aw_2 \cdots aw_{awnum}$}의 구조를 갖는다. 이때 Naïve Bayes 분류자는 가중치가 부여된 연관 단어 지식 베이스를 사용하여 문서 데이터베이스에 있는 문서를 클래스별로 분류한다. 그 결과로, 키워드 문서 데이터베이스를 구성할 수 있다.

키워드 문서 데이터베이스는 {userID, docID, classIDpos, classID}의 구조를 가지며, 여기서 classIDpos는 식 (5)에 의해 계산된다.

3.6 그룹 에이전트

그룹 에이전트는 CUG 알고리즘에 의해 키워드 문서 데이터베이스의 정보를 이용하여 사용자 간의 군집을 만든다. 새로운 사용자는 검색을 시작한 문서와 동적 링킹 시스템을 이용하여 추천된 문서로 사용자의 임시 프로파일을 {userID, docID, 키워드, docID를 읽은 시간}의 형식으로 구성된다. 추천된 문서의 {docID를 읽은 시간}은 사용자가 검색을 시작한 문서와 동일하게 작성한다. 사용자의 임시 프로파일은 문서 데이터베이스와 키워드 문서 데이터베이스의 내용을 변동시키며, CUG 알고리즘은 변동된 내용을 기반으로 기존의 모든 그룹을 순회하며 어느 그룹이 사용자에게 적합한지를 조사한다. 사용자의 임시 프로파일을 사용한 결과로, 새로운 사용자는 그룹으로 분류된다. 그룹으로 군집된 사용자는 그룹 내의 사용자의 프로파일 정보를 기반으로 하는 동적 링킹 시스템에 의해 새로운 문서를 추천 받을 수 있으며 이에 따라 사용자의 임시 프로파일을 수정하여 초기 프로파일을 생성한다. 따라서 유사한 흥미를 갖는 여러 사용자들의 경험을 바탕으로 공통된 검색유형을 찾아서 사용자의 프로파일로 활용하여 검색 시간을 절약할 수 있다.

반면, 기존에 이미 한 그룹으로 분류되어 있는 사용자의 흥미가 변하여 다른 그룹으로 분류되어야 할 경우, 사용자의 요청에 의해 수동적으로 분류되는 것이 아니고 자동적으로 변화를 모니터하여 재분류할 수 있다.

4. CUG 알고리즘에 의한 사용자 군집의 예

이 장에서는 연관 단어 지식 베이스를 만든 후 웹문서를 분류하며 이를 이용하여 사용자의 군집을 만드는 방법을 구체적으로 설명한다.

4.1 연관 단어 지식 베이스의 구축

본 논문의 실험을 위해 컴퓨터 분야의 웹문서를 8개의 클래스로 분류한다. 실험 대상의 웹문서를 8개의 클래스로 분류한 기준은 야후, 한미르, 알타비스타 등 기존의 정보검색 엔진에서 이용하는 분류 통계를 따른 것이다.

연관 단어 지식 베이스를 구성하기 위해서는 우선 웹문서를 수집하여야 한다. 이를 위하여, 웹문서 수집기를 사용하며, 수집기에 의해 사용자가 선정한 URL로부터 클래스별로 100개의 웹문서를 수집한다. 연관 단어 엔진은 수집된 웹문서로부터 <표 1>과 같은 형태로 명사를 추출한다.

〈표 1〉 연관 단어 엔진에 의해 추출된 명사의 예

클래스	훈련 URL(트랜잭션필드)	추출된 명사
게 임	http : //www.thegame.co.kr	게임,정고,인가,사용자,이벤트,참가,…
그래픽	http : //www.animatus.com	멀티미디어,출판사,컴퓨터,인테리어,활용.
뉴스와 미디어	http : //www.chollian.net /mynews	인터넷,날씨,방송,신문,환경,오염,…
반도체	http : //www.daejn-semi.co.kr	구축,설계,창업,기술,산업,메모리,...
보 안	http : //165.246.33.33	해킹,접근,발표,정보,활동,인공지능,…
인터넷	http : //www.gagaweb.co.kr	네트워크,컴퓨터,정보,교환,프로토콜,…
전자출판	http : //www.hiebook.com	서점,결제시스템,출판,기획,제작,내용,…
하드웨어	http : //www.savitmicro.com	메인보드,하드웨어,하드디스크,모니터,…

Apriori 알고리즘은 <표 1>과 같이 추출된 명사로부터 연관 단어를 마이닝한다. 그 결과는 <표 2>과 같은 형태로 나타난다. 이러한 자료로 구성된 연관 단어 지식 베이스는 평균 신뢰도 95.3과 평균 지지도 20.1를 나타내며, 총 231개의 연관 단어를 저장한다.

〈표 2〉 클래스별로 마이닝된 연관 단어

클래스	선행단어(Antecedent)	후행단어 (Consequent)	평균 신뢰도	평균 지지도	총연관 단어수
게 임	게임&구성&선수&경기&스포츠&참가	선 발	91.30%	20.1039%	18
그래픽	방법&중심&제작&사용	평 가	88.10%	21.4286%	27
뉴 스	뉴스&제공&홍보&속보	안 내	99.9%	20.2838%	30
반도체	시스템&사업&활용	컴퓨터	96.20%	20.3839%	22
보 안	세계&데이즌&인물&조작&수법	해 커	95.30%	21.7583%	25
인터넷	컨텐츠&사이트&관리	웹	94.90%	19.3838%	28
전자출판	입력&편집&출력&컬러	출 판	91.30%	18.2129%	29
하드웨어	보드&주변기기&슬롯	기 기	90.20%	21.2532%	30

4.2 Naïve Bayes 학습에 의한 가중치 부여

연관 단어 지식 베이스의 연관 단어에 가중치를 부여하기

위하여 각 클래스별로 30개의 웹문서를 추출한다. 연관 단어 엔진은 추출한 웹문서를 대상으로 연관 단어를 마이닝한다. Naïve Bayes 알고리즘은 마이닝한 연관 단어를 훈련 집합으로 설정하여 학습을 한다. 학습 과정은 누적 단계와 가중치 부여 단계로 나눈다. 누적 단계에서는 훈련 집합에 있는 연관 단어가 지식 베이스에 있을 때마다 숫자를 누적시킨다. 가중치 부여 단계에서는 누적 단계의 결과를 식 (3)에 적용하여 지식 베이스의 연관 단어에 가중치를 부여한다. 이러한 과정을 거친 후, 지식 베이스의 연관 단어에 가중치가 추가된다. <표 3>은 연관 단어 지식 베이스의 게임 클래스에 나타난 연관 단어에 가중치가 부여된 결과를 보인다.

<표 3> 가중치가 부여된 연관 단어 지식 베이스(게임클래스)

연 관 단 어	가 중 치
(1)게임&구성&선수&경기&스포츠&참가=>선발	0.09375
(2)국내&최신&기술&설치=>개발	0.012712
(3)게임&참가&인기&사용자&접속=>이벤트	0.100386
(4)운영&선발&경기&순위&규칙=>평가	0.016878
(5)게임&순위&이름=>스포츠	0.089494
(6)운영&스포츠&위원회&선수=>선발	0.100386
(7)게임&구성&선발&순위=>경기	0.086614
(8)게임&일정&선수&참가&운영=>스포츠	0.093023
(9)데이터&암호&통신망=>가입	0.008511
(10)게임&이용&문제=>규칙	0.085938
(11)그림&인기&서비스=>음악	0.008547
(12)그림&데이터&서비스=>엔진	0.017021
(13)데이터&프로그램=>음악	0.021186
(14)그림&데이터&프로그램=>사진	0.025316
(15)게임&설명&제공=>공략	0.096154
(16)게임&이용&기술=>개발	0.100386
(17)삭제&게임&개인전=>경고	0.100775
(18)게임&제공&일러스트=>설명	0.085603

4.3 Naïve Bayes 분류자에 의한 문서 분류

Naïve Bayes 분류자는 사용자가 검색해서 읽은 웹문서를 연관 단어 지식 베이스의 클래스의 하나로 분류한다. Naïve Bayes 분류자는 연관 단어 엔진을 사용하여 사용자가 검색해서 읽은 웹문서로부터 키워드를 추출한다. 이를 위하여 연관 단어 엔진은 신뢰도를 85보다 크게 지정하고, 지지도는 0으로 선택하여 연관 단어를 마이닝한다면 $\{aw_1, aw_2, \cdots, aw_{awnum}\}$형태의 효율적인 키워드를 추출할 수 있다. 신뢰도를 85보다 크게 지정한 이유는 (그림 1)에서 보였으며, 지지도를 0으로 지정하는 이유는 연관 단어 추출 대상이 다른 웹문서와는 관계없이 하나의 웹문서 만이기 때문이다. Naïve Bayes 분류자는 가중치가 부여된 연관 단어 지식 베이스와 키워드$\{aw_1, aw_2, \cdots, aw_{awnum}\}$를 수식 (4)에 대입하여 웹문서를 클래스별로 분류한다.

<표 4>는 {*이승준, 배찬우, 민유지, 정건이, 이용호*}의 사용자들이 검색하여 읽은 문서를 위와 같은 방법으로 클래스로 분류한 후 누적시킨 표이다.

<표 4> Naïve Bayes 분류자에 의해 분류된 문서

	Class1	Class2	Class3	Class4	Class5	class6	Class7	Class8
이승준	1		1		1		2	3
배찬우		2	1		1		2	2
민윤지	1	3		4		2		
정건이	1	2	1	3		1		
이용호	2		1		3		1	2

문서 관리자는 클래스로 분류된 문서와 <표 4>에 나타난 누적 내용을 비율로 전환한 뒤에 키워드 문서 데이터베이스에 저장한다. 비율로 전환하기 위해서는 <표 4>의 내용을 식 (1)에 대입해야 한다. <표 5>는 전환된 결과를 나타낸다.

<표 5> 전체 클래스에 대한 각 클래스의 비율

	Class1 Pos	Class2 Pos	Class3 Pos	class4 pos	class5 pos	class6 pos	Class7 Pos	class8 pos
이승준	0.1250	0.0000	0.1250	0.0000	0.1250	0.0000	0.2500	0.3750
배찬우	0.0000	0.2500	0.1250	0.0000	0.1250	0.0000	0.2500	0.2500
민윤지	0.1000	0.3000	0.0000	0.4000	0.0000	0.2000	0.0000	0.0000
정건이	0.1250	0.2500	0.1250	0.3750	0.0000	0.1250	0.0000	0.0000
이용호	0.2222	0.0000	0.1111	0.0000	0.3333	0.0000	0.1111	0.2222

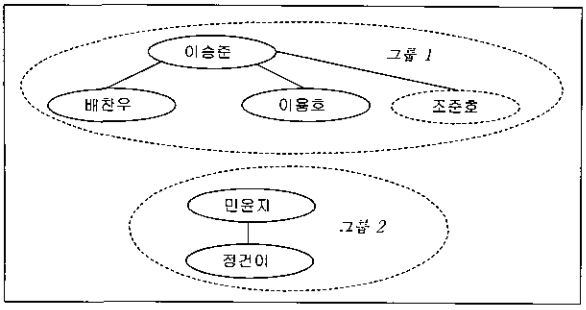
4.4 CUG 알고리즘을 사용한 사용자 군집

이 절에서는 CUG 알고리즘을 사용하여 사용자들의 군집을 만드는 과정을 보인다. 이러한 과정은 최초로 사용자의 군집을 만드는 경우와 새로운 사용자가 발견되어서 군집을 만드는 경우로 나눈다. 첫 번째 경우에서는 최초로 사용자의 군집을 만드는 과정을 보이기 위해 <표 5>를 이용한다. 먼저, <표 5>에 나타나는 사용자 간의 유사도를 계산하기 위해서 (그림 4)의 유사도 계산 알고리즘을 이용한다. <표 6>은 (그림 4)의 알고리즘에 의해 사용자 간의 유사도를 계산한 결과를 나타낸다.

<표 6> 사용자 간의 유사도

	이승준	*배찬우*	*민윤지*	*정건이*	*이용호*
이승준	1.0000	0.6667	0.0233	0.0667	0.6702
배찬우	0.6667	1.0000	0.1690	0.2000	0.4417
민윤지	0.0233	0.1690	1.0000	0.9130	0.0434
정건이	0.0667	0.2000	0.9130	1.0000	0.0941
이용호	0.6702	0.4417	0.0434	0.0941	1.0000

(그림 3)의 CUG 알고리즘에서는 사용자 군집을 만들기 위해서 유사도 임계값에 따라 군집을 만든다. 임계값이 0.5보다 클수록 정확도는 높아지나 재현율이 낮아지며, 0.5보다 작아질수록 재현율은 높아지나 정확도는 낮아진다. 따라서 이 논문에서는 유사도 임계값을 중간값인 0.5로 정한다. CUG 알고리즘은 이러한 임계값에 따라 {(*이승준, 배찬우*), (*이승준, 이용호*), (*민윤지, 정건이*)}의 3쌍을 대상으로 군집을 한다. 그 결과, (그림 6)에서 *그룹1*{*이승준, 배찬우, 이용호*}과 *그룹2*{*민윤지, 정건이*}의 두 그룹으로 군집이 생성되었음을 보인다.



(그림 6) CUG 알고리즘에 의한 군집의 예

이러한 과정으로 생성된 군집은 수시로 변한다. 수시로 변하는 이유는 새로운 사용자가 추가된 경우 군집 내용이 변하기 때문이며, 또한 기존에 한 그룹으로 군집된 사용자일지라도 그 사용자의 흥미는 쉽게 변할 수 있기 때문이다. 새로운 사용자가 추가된 경우 군집 내용이 변하는 과정을 보이기 위해 <표 7>을 이용한다. <표 7>은 새로운 사용자{조준호}가 검색하여 읽은 문서를 Naïve Bayes 분류자에 의해 클래스별로 분류한 내용을 나타낸다.

〈표 7〉 웹문서의 클래스별 분류

사용자	Class1	Class2	Class3	Class4	class5	Class6	Class7	Class8
조준호	3		2	1	1		1	1

CUG 알고리즘은 사용자{조준호}와 (그림 6)의 그룹1과 그룹2에 속한 사용자들과의 유사도를 (그림 4)의 함수를 사용하여 계산한다. 계산 결과는 <표 8>에 나타난다.

〈표 8〉 사용자{조준호}와 그룹1과 그룹2에 속한 사용자와의 유사도

조준호	사용자1	사용자2	사용자3	평균
그룹1	0.4975	0.3934	0.6364	0.5091
그룹2	0.1800	0.3186		0.2493

<표 8>에서와 같이 사용자{조준호}와 그룹1의 사용자들과의 유사도를 계산한 결과의 평균은 0.5091이며, 그룹2의 사용자들과의 유사도 계산 결과의 평균은 0.2493이다. 따라서 사용자{조준호}는 평균 유사도가 이 논문에서 선택한 유사도 임계값0.5보다 큰 그룹1로 군집된다. 만약, 새로운 사용자{조준호}의 평균 유사도가 그룹 1과 그룹2에서 임계값보다 작은 유사도를 나타낸다면 (그림 3)의 CUG알고리즘은 기존에 존재하지 않았던 새로운 그룹인 그룹3을 만든다. 그 결과, 새로운 사용자{조준호}는 그룹3으로 군집된다. 그룹1로 군집된 사용자{이승준}이 새로운 문서를 검색하여 읽은 경우, 먼저 그룹1에 있는 사용자와의 유사도를 계산한다. 그 결과, 평균 유사도가 0.5보다 같거나 크다면 사용자{이승준}은 검색하기 전과 마찬가지로 그룹1로 군집되며, 0.5보다 작다면 새로운 사용자와 같이 모든 그룹을 순회하며 적합한 그룹으로 군집된다.

5. 성능 평가

본 논문에서 설계한 CUG 알고리즘의 성능을 평가하기 위해 K-means 군집 방법과 Gibbs 샘플링 방법과의 결과를 비교하였다. 이를 위한 목적으로 컴퓨터공학과 대학원생 190명을 대상으로 10일간 협력적 여과 시스템에 적용하였다. 그 결과, 군집되는 그룹과 사용자가 점차 늘어나 최종적으로 21개의 그룹으로 전체 189명의 사용자가 각 그룹으로 군집되었다. 성능을 평가하기 위해, 각 그룹으로 군집된 사용자를 대상으로 <표 9>와 같은 분할표를 작성한다[13].

〈표 9〉 2 x 2 - 분할표

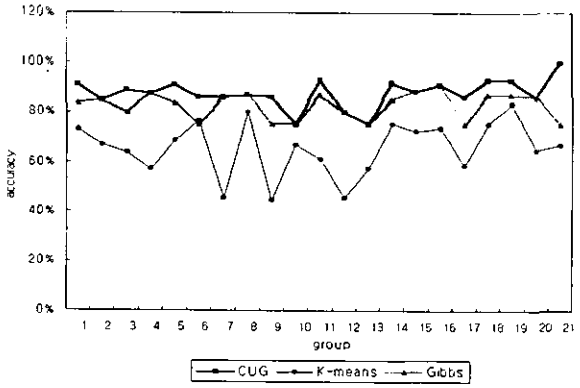
분류A \ 분류B		분할 B (알고리즘2로 생성된 분류)	
		YES	NO
분할 A (알고리즘1이 만든 분류)	YES	a	b
	NO	c	d

분류의 측정은 식 (8)의 F-measure 측정식을 이용한다. 식 (8)에서 P는 정확도, R은 재현율을 의미하는데, F-measure의 값이 클수록 분류가 우수함을 의미한다. 여기서, β는 정확도에 대한 재현율의 상대적인 가중치를 나타내는 수치로 1.0일 경우 정확도와 재현율의 가중치가 같다.

$$F\_measure=\frac{(\beta^2+1)PR}{\beta^2P+R} \quad P=\frac{a}{a+b}100\% \quad R=\frac{a}{a+c}100\% \quad (8)$$

본 실험에서는 β의 값을 1.0로 설정하여 그룹별로 분류 결과를 분석해 보았으며, 또한 β의 값을 0.5에서 1.4로 변화시키면서 F-measure의 결과 차이를 살펴보았다. <표 10>은 정확도와 재현율을 식 (8)에 대입하여 분석한 결과를 나타낸다.

(그림 7)과 (그림 8)은 <표 10>의 결과를 바탕으로 한 정확도와 F-measure의 성능 곡선을 나타낸다. (그림 7)에서 CUG 알고리즘은 K-means군집 방법보다는 22.06%, Gibbs 샘플링 방법보다는 5.09%의 높은 정확도를 나타낸다.

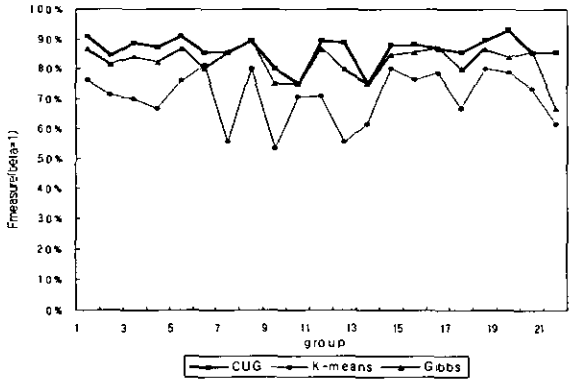


(그림 7) CUG 알고리즘, K-means군집 방법, Gibbs 샘플링 방법의 사용자 군집 정확도

〈표 10〉 CUG와 K-means, Gibbs와의 성능 비교표

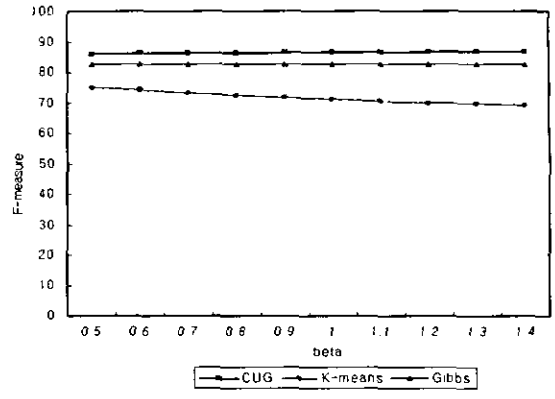
	CUG			K-means			Gibbs		
그룹	재현율(%)	정확도(%)	F-measure(%)	재현율(%)	정확도(%)	F-measure(%)	재현율(%)	정확도(%)	F-measure(%)
1	90.91	90.91	90.91	80.00	72.73	76.19	90.91	83.33	86.96
2	84.62	84.62	84.62	76.92	66.67	71.43	78.57	84.62	81.48
3	88.89	88.89	88.89	77.78	63.64	70.00	88.89	80.00	84.21
4	87.50	87.50	87.50	80.00	57.14	66.67	77.78	87.50	82.35
5	90.91	90.91	90.91	84.62	68.75	75.86	90.91	83.33	86.96
6	85.71	85.71	85.71	86.67	76.47	81.25	85.71	75.00	80.00
7	85.71	85.71	85.71	71.43	45.45	55.56	85.71	85.71	85.71
8	92.86	86.67	89.66	80.00	80.00	80.00	92.86	86.67	89.66
9	75.00	85.71	80.00	66.67	44.44	53.33	75.00	75.00	75.00
10	75.00	75.00	75.00	75.00	66.67	70.59	75.00	75.00	75.00
11	86.67	92.86	89.66	84.62	61.11	70.97	86.67	86.67	86.67
12	100.0	80.00	88.89	71.43	45.45	55.56	80.00	80.00	80.00
13	75.00	75.00	75.00	66.67	57.14	61.54	75.00	75.00	75.00
14	84.62	91.67	88.00	85.71	75.00	80.00	84.62	84.62	84.62
15	88.24	88.24	88.24	81.25	72.22	76.47	83.33	88.24	85.71
16	83.33	90.91	86.96	84.62	73.33	78.57	83.33	90.91	86.96
17	85.71	85.71	85.71	77.78	58.33	66.67	85.71	75.00	80.00
18	86.67	92.86	89.66	85.71	75.00	80.00	86.67	86.67	86.67
19	92.86	92.86	92.86	75.00	83.33	78.95	81.25	86.67	83.87
20	85.71	85.71	85.71	84.62	64.71	73.33	85.71	85.71	85.71
21	75.00	100.0	85.71	57.14	66.67	61.54	60.00	75.00	66.67
평균	85.76	87.50	86.44	77.79	65.44	70.69	82.55	82.41	82.34

(그림 8)에서 β=1.0일 경우, CUG 알고리즘은 K-means군집 방법보다는 15.75%, Gibbs샘플링 방법보다는 4.1% 향상된 F-measure의 결과를 나타낸다.



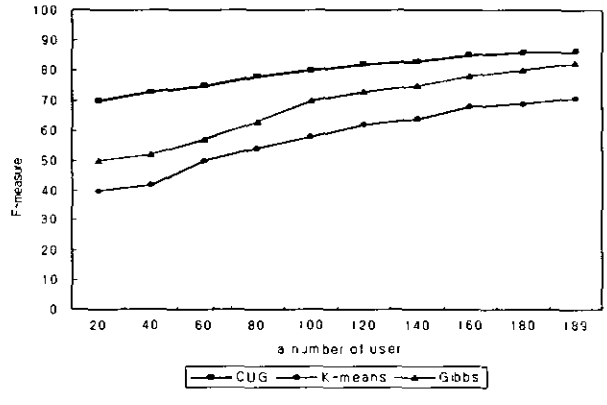
(그림 8) F-measure에 의한 그룹별 사용자 군집 성능 평가

(그림 9)는 β값을 0.5에서 1.4로 변화시킴에 따른 F-measure의 성능 분석을 나타낸다. K-means군집 방법은 β값이 커짐에 따라 하향곡선을 보이므로 정확도보다는 재현율에 좀더 높은 성능을 보임을 알 수 있다. 그러나 CUG알고리즘과 Gibbs샘플링 방법은 β의 값이 커짐에도 일정한 F-measure결과를 보인다. 따라서 CUG알고리즘과 Gibbs샘플링 방법은 정확도뿐만 아니라 재현율에서도 높은 성능을 나타냄을 알 수 있다. 평균적으로, CUG알고리즘은 K-means 군집 방법보다는 14.87%, Gibbs샘플링 방법보다는 4.06%의 높은 성능 차이를 보였다.



(그림 9) β의 변화에 따른 F-measure에 의한 그룹별 사용자 군집 성능 평가

(그림 10)은 사용자 수의 증가에 따른 F-measure의 성능 변화를 나타낸다. 세 가지 방법 모두 사용자 수가 증가함에 따라 점차 F-measure에 의한 성능이 점차 향상됨을 보인다. 특히, CUG알고리즘은 사용자의 수가 작은 경우에도 높은 성능을 나타낸다. 그러나 K-means군집 방법과 Gibbs 샘플링 방법은 사용자 수가 작은 경우 낮은 성능을 나타낸다.



(그림 10) 사용자수의 변화에 따른 F-measure에 의한 그룹별 사용자 군집 성능 평가

6. 결 론

본 논문에서는 협력적 여과 시스템의 사용자 군집을 위한 CUG알고리즘을 제안하였다. CUG 알고리즘은 연관 단어 지식 베이스를 구축한 후, 이를 대상으로 Naïve Bayes 학습을 함으로써 가중치를 부여한다. Naïve Bayes 알고리즘은 가중치가 부여된 연관 단어 지식 베이스를 이용하여 사용자가 검색하여 읽은 웹문서를 분류한다. CUG 알고리즘은 분류된 웹문서를 기반으로 하여 사용자 군집을 만든다. 제안된 CUG 알고리즘은 새로운 사용자를 짧은 시간에

적합한 그룹으로 군집시키며, 사용자의 흥미가 변하여 재군집이 필요할 경우 스스로 감지하여 적합한 그룹으로 재군집한다.

본 논문에서 설계한 CUG 알고리즘의 성능을 평가하기 위하여 기존의 K-means 방법과 Gibbs샘플링 방법에 의한 군집과 비교하여 분석하였다. 그 결과, CUG알고리즘은 K-means 군집 방법보다는 14.87%, Gibbs샘플링 방법보다는 4.06%의 높은 성능 차이를 보였다. 또한, K-means 군집 방법이나 Gibbs 샘플링 방법은 사용자가 적은 환경에서 낮은 성능을 나타냈으나 CUG알고리즘은 비교적 높은 성능을 나타냄을 보였다.

향후, 협력적 여과 시스템을 위해 사용된 Naïve Bayes 알고리즘보다 보다 효율적인 문서 분류 방법을 사용한다면, 사용자 군집의 성능을 향상시킬 수 있을 것이다.

참 고 문 헌

[1] R. Agrawal and T. Imielinski and A. Swami, "Mining association rules between sets of items in large databases," Proceedings of the 1993 ACM SIGMOD Conference, Washington DC, USA, 1993.

[2] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994.

[3] R. Agrawal and R. Srikant, "Mining Sequential Patterns," Proc. of the Intl Conference on Data Engineering (ICDE), Taipei, Taiwan, 1995.

[4] K. Alsabti and S. Ranka and V. Singh, "An Efficient K-Means Clustering Algorithm," http : //www.cise.ufl.edu/, ranka/, 1997.

[5] G. Casella and E. I. George, "Explaining the Gibbs Sampler," The American Statistician, pp.167-174, 1992.

[6] H. Chen, Y. Chung, M. Ramsey, C. Yang, P. Ma, J. Yen, "Intelligent Spider for Internet Searching," Proceedings of the 30th Annual Hawaii International Conference on System Sciences-Vol.IV, pp.178-188, 1997.

[7] D. W. Cheung and B. Kao and J. Lee, "Discovering User Access Patterns on the World-Wide Web," PAKDD-97, Singapore, 1997.

[8] W. W. Cohen and W. Fan, "Web-collaborative filtering : Recommending music by crawling the Web," Computer Networks-The International Journal of Computer & Telecommunications Networking, Vol.33 No.1-6, 2000.

[9] Cognitive Science Laboratory, Princeton University, "WordNet-a Lexical Database for English," http : //www.cogsci.princeton.edu/~wn/.

[10] G. Fischer and C. Stevens, "Information Access in Complex, Poorly Structured Information Spaces," Proceedings CHI'91 Human Factors in Computing Systems, pp.63-70, 1991.

[11] K. Funakoshi, T. Ohguro, "A content-based collaborative recommender system with detailed use of evaluations," Proceedings of the 4th International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies Vol.1, 2000.

[12] N. Good and J. B. Shafer and J. A. Konstan, and A. Borchers and B. Sarwar and J. Herlocker and J. Riedl, "Combining Collaborative Filtering with Personal Agents for Better Recommendations," Proceedings of the Sixteenth National Conference on Artificial Intelligence, 1999.

[13] V. Hatzivassiloglou and K. McKeown, "Towards the automatic identification of adjectival scales : Clustering adjectives according to meaning.," Proceedings of the 31st Annual Meeting of the ACL, pp.172-182, 1993.

[14] G. J. McLachlan and T. Krishnan, The EM Algorithm and Extensions, New York : John Wiley and Sons, 1997.

[15] A. Kohrs and B. Merialdo, "USING CATEGORY-BASED COLLABORATIVE FILTERING IN THE ACTIVE WEBMUSEUM," Proceedings of the IEEE International Conference on Multimedia and Expo-Vol.1, 2000.

[16] T. Michael, *Maching Learning*, McGraw-Hill, pp.154-200, 1997.

[17] B. Smyth and P. Cotter, "A personalized television listings service-Mixing the collaborative recommendation approach with content-based filtering seems to bring out the best in both methods," Communications of the ACM, Vol.43 No.8, pp.107-111, 2000.

[18] Ko. SuJeong and Lee. JungHyen, "Optimized Association Word Knowledge Base Construction Using Apriori-Genetic Algorithm," Proceedings of the International Symposium on Adaptive Systems, 2001.

[19] D. B. Terry, "7 Steps to a Better Mail System," Proceedings IFIP International Symposium on Message Handling System and Application Layer Communication Protocols, 1990.

[20] L. H. Ungar and D. P. Foster, "Clustering Methods for Collaborative Filtering," AAAI Workshop on Recommendation Systems, 1998.

[21] P. C. Wong and P. Whitney and J. Thomas, "Visualizing Association Rules for Text Mining," Proceedings of the 1999 IEEE Symposium on Information Visualization, pp.120-123, 1999.

[22] T. W. Yan ,et. al, "From User Access Patterns to Dynamic Hypertext Linking," Proceedings of the Fifth International World-Wide Web Conference, Paris, France, 1996.

[23] O. Zamir and O. Etzioni, "Web Document Clustering : A Feasibility Demonstration," SIGIR'98, pp.46-54, 1998.

[24] 박영규, 김진수, 김태용, 이정현, "연관 웹 문서 분류와 사용자 브라우징 패턴을 이용한 동적 링킹 시스템", 한국정보처리학회 추계 학술 발표 논문집, 2000.

고 수 정

e-mail : sujung@nlsun.inha.ac.kr

1990년 인하대학교 전자계산학과 졸업 (학사)

1997년 인하대학교 교육대학원 전자계산교육 (교육학석사)

2000년 인하대학교 대학원 박사과정 수료

관심분야 : 데이터마이닝, 정보검색, 기계학습

임 기 욱

e-mail : rim@omega.sunmoon.ac.kr

1977년 인하대학교 공과대학 전자공학과 졸업

1987년 한양대학교 전자계산학 석사

1994년 인하대학교 전자계산학 박사

1977년~1983년 한국전자기술연구소 선임 연구원

1983년~1988년 한국전자통신연구소 시스템소프트웨어 연구실장

1988년~1989년 미 캘리포니아주립대학(Irvine)방문 연구원

1989년~1997년 한국전자통신연구원 시스템연구부장 주전산기 (타이컴) III, IV개발 사업책임자

1997년~2000년 정보통신연구진흥원 정보기술 전문의원

2000년~현재 선문대학교 교수

관심분야 : 실시간 데이터베이스시스템, 운영체제, 컴퓨터구조

이 정 현

E-mail : jhlee@inha.ac.kr

1977년 인하대하교 전자공학과 졸업

1980년 인하대학교 대학원 전자공학과 (공학석사)

1988년 인하대학교 대학원 전자공학과 (공학박사)

1979년~1981년 한국전자기술연구소 시스템 연구원

1984년~1989년 경기대학교 전자계산학과 교수

1989년~현재 인하대학교 전자계산공학과 교수

관심분야 : 자연언어처리, HCI, 정보검색, 음성인식, 음성합성, 계산기 구조