2002년 한국정보처리학회 춘계학술발표논문집 제9권 제1호

LSI를 이용한 가중치 변화에 따른 클러스터링 결과 분석

고지현*, 오형진**, 박순철*
*전북대학교 정보통신공학과
**전북대학교 컴퓨터공학과
e-mail : pludy@internet.chonbuk.ac.kr, hyungjin@duan.chonbuk.ac.kr
scpark@moak.chonbuk.ac.kr

The Analysis of Clustering Result with Weight Change using LSI

Ji-Hyun Goh*, Hyung-Jin Oh**, Soon-Cheol Park*
*Dept. of Information and Communication, Chon-Buk University
**Dept. of Computer Engineering, Chon-Buk University

요 약

정보검색시스템에서 가장 중요한 것은 사용자의 요구에 부합하는 결과를 도출하는 것이다. 이를 위하여 사용자의 질의와 연관된 모든 문서들을 추출하게 되는데, 이 많은 결과 문서들 중에서 사용자가 원하는 문서는 소수이고, 원하는 문서를 찾는 것도 쉽지 않다. 따라서 적절한 결과 문서 도출을 위하여 연관된 문서들끼리 그룹화 시키는 클러스터링 방법이 많이 이용된다. 본 논문에서는 클러스터링에 영향을 끼치는 요소 중 문서별 색인어의 가중치가 클러스터링에 끼치는 영향을 알아보았다. 이를 위해 가중치의 변화에 따른 클러스터링 된 결과를 LSI 를 이용하여 도식화하고 그 결과를 분석하였다.

1. 서론

최근 웹 문서 양이 기하 급수적으로 증가하면서 정보검색엔진의 성능평가에 대한 논의가 대두되고 있다. 문서의 양 뿐만 아니라 작성된 문서의 종류도 다양해서 사용자의 요구에 적합한 결과 문서를 도출해 내기가 어렵다. 대부분의 정보검색엔진은 사용자의 질의를 분석하여 사용자의 의도와는 상관없이 질의에 따른 모든 문서를 찾아내는 방법을 사용한다. 많은 양의 결과 문서들을 사용자에게 보여주고, 사용자가 스스로 적합한 문서를 찾아내게 한다. 각 검색엔진 별로 문서의 순위화를 이용하고 있지만 적합한 문서를 찾아내는 것은 결국 사용자의 몫이다. 검색된 모든 문서들 중에서 사용자가 적절한 결과 문서를 추출하기란 쉽지 않다. 오히려 적합한 결과 문서를 찾지 못하는 경우가 더 많다. 그래서 사용자가 검색된 많은 문서들 중에서 결과 문서를 더 빨리 찾을 수 있도록 클러스터링 방법을 이용한다. 이 방법은 유사한 문서끼리 그룹화 시키는 방법으로 자동문서분류나 데이터 마이닝 분야에서 많이 이용하고 있다. 이를 이용하여 결과 문서들을 클러스터링 하면 비슷한 문서별로 그룹화가 되어 있기 때문에 사용자는 한 눈에 검색결과를 볼 수 있고, 자신이 생각하는 개념과의 일치 여부에 따라 원하는 결과 문서를 찾을 수 있다.

본 논문에서는 클러스터링의 성능을 높이기 위해서 클러스터링에 영향을 끼치는 여러 요소들 중 문서별 색인어의 가중치 부분을 이용하여 가중치 변화에 따른 문서들의 클러스터된 결과를 분석한다. 클러스터된 결과 분석 방법은 각 문서들을 벡터화 하고 이차원 공간으로 사상 시킨 도식화를 통해 가중치 변화에 따른 클러스터링 결과를 비교한다. 2 장에서는 구현한 클러스터링 알고리즘에 대하여 설명하고 3 장에서는 가중치 부여 방법과 그 종류에 대해 설명한다. 4 장에서는 결과 분석을 위해 사용한 LSI 에 대해 알아보고 5

장은 실험한 가중치 변화에 따른 클러스터링 결과를 비교, 분석하고 끝으로 6 장에서는 결론을 맺는다.

2. K-Means 알고리즘을 이용한 클러스터링

클러스터링 부분을 위한 실험 데이터는 요약된 문서들을 이용하였다. 데이터는 각 문서 당 색인어, 단어 빈도수(tf), 전체 문서에서의 단어 빈도수(df)들로 구성되어 있고 물론 문서와 색인어 모두 유일한 값을 갖는다. 구현된 클러스터링 부분은 크게 두 부분으로 나뉜다. 가중치를 부여하는 부분과 클러스터링 부분으로, 가중치 부여는 3 장에서 설명할 것이다. 클러스터링 부분은 일반적으로 많이 사용하고 있는 K-Means 알고리즘을 이용하였다. K-Means 알고리즘은 다음과 같다.

---

1. K 값(클러스터의 개수)을 정한다.
2. K 개의 초기 중심값(proto-centroid)를 정한다.
3. 각 문서 (d_i) 들과 중심값 (c_j) 사이의 거리를 구한다.

$$dist(\vec{d_i}, \vec{c_j}) = \sqrt{\sum_{k=1}(d_{ki} - c_{kj})^2} \quad : \text{Euclidean Distance}$$

$(i = 1, 2, \cdots, n \quad n$: 전체 문서의 개수
$j = 1, 2, \cdots, k \quad k$: 중심값(centroid)의 개수
= 클러스터의 개수)

4. 가장 짧은 거리의 문서를 각 중심값의 클러스터에 할당한다.

- $\arg\min_{i=1,\cdots,n \; j=1,\cdots,k} dist(\vec{d_i}, \vec{c_j})$
- $d_{i \; i=1,\cdots,n} \in G_{c_j} \quad if \; dist(\vec{d_i}, \vec{c_j}) < dist(\vec{d_i}, \vec{c_l})$

$(for \; all \; l = 1, 2, \cdots, k \quad l \neq j)$

5. 새로운 중심값을 계산한다.

$$\vec{c_j} = \frac{1}{|\vec{c_j}|} \sum_{l=1}^{|c_j|} \vec{d_l}$$

6. 이전의 중심값과 새로운 중심값를 비교하여 벡터간 차이가 거의 없을 때까지 반복한다.

$$If \; \max \delta(\vec{c_j^{old}}, \vec{c_j^{new}}) < \theta \; then \; return$$
$$else \; goto \; 3$$

---

3. 가중치 부여

모든 색인어가 문서 내용을 설명하는데 똑같이 유용하지 않으며, 어떤 색인어는 다른 것보다 더 모호한 경우가 있다. 모든 문서에 나타나는 단어는 색인어로서는 무용하다. 이는 어떤 문서가 사용자가 관심있는 것인지에 영향을 끼치지 못하기 때문이다. 반면, 단지 몇 개의 문서에만 출현한 단어는 사용자가 관심있어 하는 문서들을 추출해 내는데 유용하다. 따라서 문서 내용을 설명하는데 같이 사용된 단어라 할지라도 다양한 비중을 가지고 있으며 문서의 각 색인어에 가중치를 부여하는 효과를 가져온다.

m 개의 색인어와 n 개 문서로 구성된 모든 문서 집합(collection)은 $m \times n$ 행렬로서 표현하고, 이를 A 라 하자. A 를 다음과 같이 정의한다.

$$A = (a_{ij})$$

여기서 행렬 A 의 각 원소 a_{ij} 의 값은 j 번째 문서에서의 i 번째 색인어의 가중치로 표현한다. 각 원소 a_{ij} 는 다음과 같이 정의한다.

$$a_{ij} = l_{ij} g_i d_j$$

l_{ij} 는 j 번째 문서에서의 i 번째 색인어의 로컬 가중치(local weight)이고, g_i 는 전체 문서집합(collection)에서 i 번째 색인어의 글로벌 가중치(global weight), d_j 는 문서의 정규화(normalization) 요소이다.

다음은 로컬 가중치를 구하는 공식들을 도표화한 것이다. 이는 각 문서에 따른 색인어의 가중치를 부여하는 방법으로 문서 내에서의 색인어의 중요도를 나타낸다.

Symbol	Name	Formula
b	Binary	$\chi(f_{ij})$
l	Logarithmic	$\log(1 + f_{ij})$
n	Augmented normalized Term frequency	$(\chi(f_{ij}) + (f_{ij} / \max_k f_{kj})) / 2$
t	Term frequency	f_{ij}

(표 1) Formulas for local term weights (l_{ij})

(표 1) 에서 Binary 는 다음과 같이 정의된다.

$$\chi(r) = \begin{cases} 1 & if \; r > 0, \\ 0 & if \; r = 0, \end{cases}$$

다음 표는 글로벌 가중치를 구하는 공식이다. 전체 문서에서의 색인어의 가중치를 나타내는 방법으로, 이것은 문서 전체를 기준으로 한다는 점에서 로컬 가중치 부여 방법과 다르다.

Symbol	Name	Formula
x	None	1
e	Entropy	$1 + (\sum_j p_{ij} \log(p_{ij}) / \log n)$
f	Inverse document frequency (IDF)	$\log(n / \sum_j \chi(f_{ij}))$
g	Gfldf	$(\sum_j f_{ij}) / \sum_j \chi(f_{ij}))$
n	Normal	$1 / \sqrt{\sum_j f_{ij}^2}$
p	Probabilistic Inverse	$\log((n - \sum_j \chi(f_{ij})) / \sum_j \chi(f_{ij}))$

(표 2) Formulas for global term weights (g_i)

(표 2)의 Entropy 에서 $p_{ij} = \frac{f_{ij}}{\sum_j f_{ij}}$ 이다.

(표 3)은 문서 정규화를 위한 공식이다. 이는 문서들의 길이를 조절하는 방법으로, 보통 각 문서들의 벡터 길이를 1로 정규화 한다.

Symbol	Name	Formula
x	None	1
c	Cosine	$(\sum_i (g_i l_{ij})^2)^{-1/2}$

(표 3) Formulas for document normalization (d_j)

4. LSI (Latent Semantic Indexing)

LSI 는 문서의 내용이 서술된 색인어 보다는 그 안에 표현된 개념에 기반한다는 점에 착안하여 제안된 모델이다. 이것은 문서들이 같은 색인어로 구성되어 있지 않더라도 연관성을 나타낼 수 있다. 어떤 문서가 다른 문서와 개념을 공유한다면 유사한 문서라 할 수 있다. 이 모델의 요점은 문서들을 저차원 벡터 공간으로 사상 시키는데 있다.

행렬 $m \times n$ 으로 나타내는 전체 문서 집합(collection) A 는 각 원소의 값으로 가중치를 갖는다고 하자. 이 때 A 를 SVD(Singular Value Decomposition)로 분해한다.

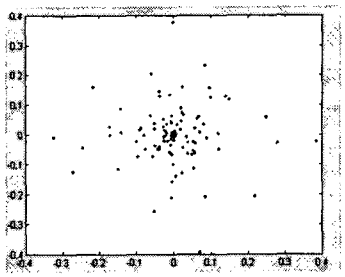
$$A = U \Sigma V^T$$

여기서 U 는 단어간 상관 행렬(association matrix)로부터 얻은 $m \times m$ 고유 벡터 행렬이고, V 는 문서간 상관 행렬로부터 얻은 $n \times n$ 고유 벡터 행렬이다. Σ 는 단일 값을 갖는 $m \times n$ 대각 행렬이다. U 를 이용하여 단어들은 m 차원, V 를 이용하여 문서들은 n 차원으로 사상 시킬 수 있다. 동일한 차원으로 단어 벡터와 문서 벡터를 사상 시킨다면 단어와 단어의 관계, 단어와 문서간의 관계, 문서와 문서와의 관계를 알 수 있다.

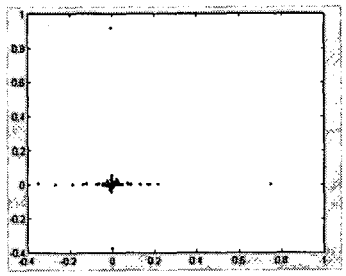
본 논문에서는 가중치 변화에 따른 클러스터링의 결과를 분석하기 위한 도식화 과정에 이 모델을 적용하였다. 문서별 색인어에 따른 가중치로 구성된 부분과 클러스터링 결과로 나타난 클러스터 중심값(centroid) 부분을 행렬식으로 표현하였다. 7507 개의 색인어와 102 개의 문서, 10 개의 중심값에 따른 7507×112 행렬식을 가지고 SVD 로 분해하였다. 문서들간의 관계를 위해서 V 부분만을 이용했다. 이차원으로 도식화하기 위해 V 부분에서 첫 번째와 두 번째 열(column)만을 택하여 x 축과 y 축에 나타냈다.

5. 결과 분석

아래 그림은 3 장에서 알아본 가중치의 변화에 따른 클러스터링 결과를 LSI 를 이용하여 나타낸 것이다. 점(·) 으로 표시된 부분은 문서들의 위치를 나타내고, (∘)으로 표시된 부분은 클러스터링의 중심값(centroid)의 위치를 나타낸다. 이차평면으로 표현하기 위해 1, 2 열(column)을 제외한 나머지 부분 행렬은 무시했기 때문에 유사한 문서들은 거의 동일한 구역에서 점들이 분포하고 있다. 어떤 경우엔 서로 다른 문서의 벡터값이 동일함을 보였다. 이로 인해 점들의 수가 문서와 중심값들의 수와 일치하지 않게 된다.

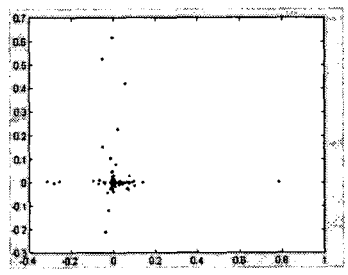


(그림 1) Weight = b (Binary)

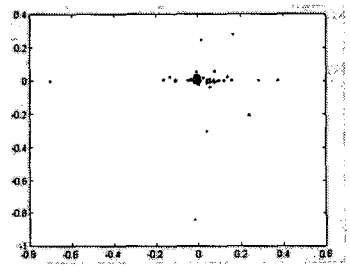


(그림 2) Weight = b * f

(그림 1)과 (그림 2)에서, 색인어 존재 여부만을 가지고 가중치를 부여한 (그림 1)의 경우는 문서들의 위치가 (그림 2)에 비해 산재되어 있음을 볼 수 있다. 클러스터의 중심값의 위치는 (그림 1), (그림 2) 모두 비슷하지만 역문헌빈도수(idf)를 적용한 (그림 2)의 경우 동일한 구역에 분포하는 점들이 많았다. 이는 글로벌 가중치 부여가 클러스터링의 효과에 큰 영향을 끼친다는 것을 보여 준다.

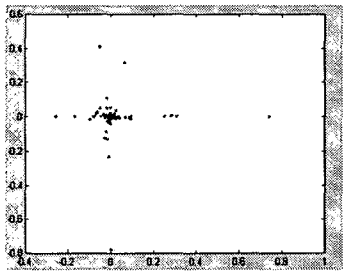


(그림 3) Weight = t/(t+2) * f



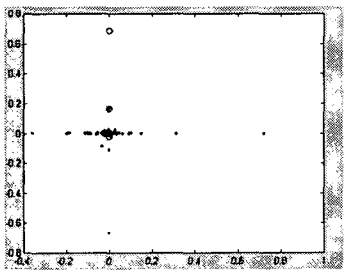
(그림 4) Weight = l * f

(그림 3)과 (그림 4)는 앞의 경우(그림 1), (그림, 2)보다 클러스터링이 더 잘 이루어졌음을 볼 수 있다. 한 곳에 집중적인 분포를 보이고 있고, (그림 2)의 경우 분포의 집중도는 높지만 로컬 가중치 값이 0 아니면 1 이라서 값의 변화가 적기 때문에 x 축으로만 길게 분포된 반면 (그림 3, 4)는 x 축과 y 축의 고른 분포를 보이고 있다. (그림 3)과 (그림 4)의 전체적인 분포도는 비슷한 양상을 보이고 있지만, 중심부분의 점들의 분포를 비교해 보면 (그림 3)보다는 logarithm 을 적용한 (그림 4)에서 비슷한 벡터 값을 가지는 문서가 더 많이 나타남을 알 수 있다. 이는 (그림 4)의 경우가 클러스터링 성능이 더 높음을 보여준다.

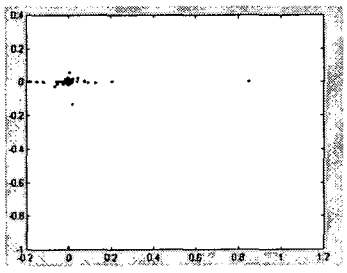


(그림 5) Weight = t/(t+2) * p

(그림 5)는 문서 몇 개를 제외하고는 (그림 3)과 비슷한 분포를 보이고 있다. 이는 가중치 부여에 영향을 끼치는 요소로서 역문헌빈도수(idf)와 역확률(Probabilistic Inverse)의 차이가 크지 않다는 것을 보여준다. 하지만 (그림 5)의 경우가 약간은 분포의 집중도가 높다.



(그림 6) Weight = t



(그림 7) Weight = t * f

(그림 6)은 가장 많이 사용하는 가중치 부여 방법인 단어빈도수(tf)를 이용한 것이다. (그림 7)과 비슷한 양상을 보이고 있지만 가장 큰 차이점은 중심값들의 위치이다. (그림 7)의 경우 중심값들의 위치가 한 부분에 집중적으로 분포하는 반면 (그림 6)은 (그림 7)에 비해 산재되어 있다. 이는 글로벌 가중치인 역문헌빈도수(idf)의 영향 때문이다. (그림 6)의 상단 부분에 나타나는 중심값의 위치는 글로벌 가중치의 영향 뿐만 아니라 이차 평면으로의 도식화를 위한 벡터 값의 축소로 인한 결과이다. (그림 7)이 (그림 2)와 비슷한 양상을 보이고 있는데 이 또한 벡터 값의 축소로 나타나는 현상이다. 하지만 사실 단순 이진기법을 이용한 (그림 2) 보다는 문헌빈도수(tf)를 이용한 (그림 7)의 경우가 클러스터링의 효과가 좋다. 이는 색인어에 고유한 값을 부여할수록 각각 고유한 벡터값을 가지기 때문이다.

6. 결론

본 논문에서는 클러스터링에 있어서 색인어 가중치가 끼치는 영향과 가중치 변화에 따른 클러스터링의 결과를 분석하기 위해 LSI 모델을 이용하였다. 이를 위해 가중치 부여 방법과 LSI 모델에 대해서 알아보았다. 클러스터링을 하기 위해 일반적인 K-Means 알고리즘을 이용하였고 가중치 부여 방법에 따른 클러스터링된 결과를 도식화하여 비교하기 위해 LSI 를 이용하여 이차 평면으로 표현하였다. 이차 평면으로 도식화 하기 위한 벡터값들의 축소로 인해 각 문서들의 벡터값이 동일한 경우도 발생했지만, 상대적으로 유사하지 않은 문서들은 벡터값이 다르기 때문에 다른 분포도를 보였다. 분석 결과 동일한 값이나 비슷한 값을 부여한 가중치 보다는 복잡하게 적용되는 가중치 부여에 있어서 클러스터링의 효과를 알아볼 수 있었다. 특히 logarithm 을 적용한 가중치와 글로벌 가중치 부여가 훨씬 더 효과적임을 알 수 있었다. 하지만 가장 적절한 가중치 부여에 대한 논의와 문서의 정규화를 적용한 가중치에 따른 클러스터링 방법에 대한 연구가 더 필요할 것이다. 또한 클러스터링에 영향을 끼치는 다른 요소에 관한 연구도 요구된다.

7. 참고문헌

[1] Michael W. Berry, Murray Browne, "Understanding Search Engines", Univ. of Tennessee

[2] Ricardo Baeza-Yates, Berthier Ribeiro-NetoRoger, "Modern Information Retrieval", Addison Wesley, 1999

[3] Khaled Alsabti, Sanjay Ranka, Vineet Singh, "An Efficient K-Means Clustering Algorithm", IIPS 11th Intrnational Parallel Processing Symposium, 1998

[4] William B. Frakes, Richard Baeza-Yates, "Information Retrieval", Prentice Hall, 1992

[5] Tapas Kanungo, David M.Mount, Nathan S. Netanyahu, Christine Piatko, "The Analysis of a Simple K-Means Clustering Algorithm", ACM, 2000

[6] Michael W. Berry, Susan T. Dumais, Todd A. Letsche, "Computational Methods for intelligent Information Access", ACM, 1995