경계변수 값의 동적인 변경을 이용한 점층적 클러스터링 알고리즘

(Incremental Clustering Algorithm by Modulating Vigilance Parameter Dynamically)

신 광 철 † 한 상 용 ††
(Kwangcheol Shin) (Sangyong Han)

요 약 본 논문은 점층적으로 대규모 문서 분류를 할 수 있는 새로운 클러스터링 알고리즘에 대한 것으로, 고차원의 대규모 문서 집합에 대한 클러스터링을 수행하는 spherical k-means (SKM) 알고리즘과 점층적인 방식으로 클러스터링을 수행하는 퍼지(fuzzy) ART(adaptive resonance theory) 신경망의 특징을 이용하였다. 즉, SKM의 벡터 공간 모델과 개념벡터를 토대로 퍼지 ART의 경계변수의 개념을 결합한 것이다. 제시하는 알고리즘은 점층적 클러스터링의 지원과 함께 최적의 클러스터 수를 자동으로 결정할 뿐 아니라 이상치(outlier)와 노이즈(noise)에 의한 overfitting의 문제도 해결하였다. 또한 생성된 클러스터들의 질을 평가할 수 있는 응집도를 측정하는 목적 함수의 값에 있어서도 CLASSIC3 데이타 집합으로 실험한 결과 기존의 SKM에 비해 평균 8.04%의 향상된 응집도를 나타냈다.

키워드 : spherical k-means, 벡터 공간 모델, 개념벡터, 퍼지 ART, 경계 변수

Abstract This study is purported for suggesting a new clustering algorithm that enables incremental categorization of numerous documents. The suggested algorithm adopts the natures of the spherical k-means algorithm, which clusters a mass amount of high-dimensional documents, and the fuzzy ART(adaptive resonance theory) neural network, which performs clustering incrementally. In short, the suggested algorithm is a combination of the spherical k-means vector space model and concept vector and fuzzy ART vigilance parameter. The new algorithm not only supports incremental clustering and automatically sets the appropriate number of clusters, but also solves the current problems of overfitting caused by outlier and noise. Additionally, concerning the objective function value, which measures the cluster's coherence that is used to evaluate the quality of produced clusters, tests on the CLASSIC3 data set showed that the newly suggested algorithm works better than the spherical k-means by 8.04% in average.

Key words : spherical k-means, vector space model, concept vector, fuzzy ART, vigilance parameter

1. 서 론

지금까지 인류는 수많은 문서들을 만들고 저장해오면서 발전해왔다. 더욱이 인터넷이 생활화된 20세기말 이후로 사람들은 엄청난 수의 새로운 형태의 문서들을 만들고 사용하고 있다. 현재 공식적으로 3억 5천만 개 이상의 인터넷 웹 페이지가 있으며 계속해서 증가하는 추세이고, 또한 인트라넷과 각종 미디어 회사, 과학 기술 출판소 등은 계속해서 수많은 텍스트 데이타를 만들어 내고 있다. 사람들은 이러한 수많은 문서들을 효과적으로 탐색하고, 저장하기 위해 조직화, 구조화를 시도해 왔으며 수작업을 통한 구조화는 그 중 하나의 해결책이다. 실제로 널리 알려진 검색엔진인 Yahoo!에서는 인터넷 문서들을 수작업으로 구조화하고 있으며, 도서관에서도 자료를 구조화하는 데 있어 수작업을 많이 이용한다. 그러나 이러한 작업은 매우 많은 노동력과 시간을 요하는 것이기 때문에 대규모 회사나 단체가 아니면 엄두도 못 낼 일이다. 따라서 아직 분류되지 않은 문서집합들을 자동적으로 분류하는 클러스터링 기술은 대단히 필요하

† 학생회원 : 중앙대학교 컴퓨터공학부
kcshin@archi.cse.cau.ac.kr
†† 종신회원 : 중앙대학교 컴퓨터공학부 교수
hansy@cau.ac.kr
논문접수 : 2002년 8월 2일
심사완료 : 2003년 8월 12일

며, 현재까지 통계적 패턴 인식과 기계학습에서 폭넓게 연구되어 왔다[1,2].

이와 함께 최근에는 인터넷 검색 결과를 실시간으로 클러스터링 하는 방향으로 활발한 연구가 진행되고 있다. 이러한 연구가 이뤄지는 배경을 살펴보면, 전통적인 웹 정보 검색 엔진이 사용자의 검색 질의에 대한 결과를 질의와의 관련정도에 따라 순위가 매겨진 매우 긴 문서 목록을 사용자에게 제공하기 때문에 지금과 같이 정보가 넘쳐나는 시대에는 오히려 정보를 더욱 찾기 어렵게 하는 것이 될 수 있기 때문이다. 따라서 검색 엔진이 찾아준 일차 검색 결과를 토대로 사용자의 요구사항에 알맞게 정보를 가공하는 후처리 웹 문서 클러스터링이 차세대 정보 검색 서비스의 대안으로 떠오르고 있다[3-8].

본 논문에서는 고차원의 대규모 문서 집합에 대한 클러스터링 알고리즘인 spherical k-means[3]에서 사용한 벡터 공간 모델[4]과 개념 벡터[9]를 기반으로 실시간 웹 문서 클러스터링 알고리즘으로 알려진 퍼지(fuzzy) ART 신경망[11]에서 사용한 경계변수(vigilance parameter) 개념을 이용한 대규모 문서집합에 대한 점층적 클러스터링 기법을 제시하고 실험결과를 보여준다.

본 논문의 구성은 다음과 같다.

2장에서는 spherical k-means의 벡터 공간 모델과 개념벡터에 대해 살펴보고 3장에서는 본 논문이 제시하는 알고리즘을 설명한다. 4장에서는 실험결과를 분석하며, 마지막 5장에서 결론을 맺는다.

2. 벡터 공간 모델과 개념 벡터

이 장에서는 구조화되지 않은 문서를 벡터 공간 모델로 표현하는 방법과 개념벡터에 대해 살펴본다.

2.1 벡터 공간 모델

벡터 공간 모델[12]의 기본적인 아이디어는 각각의 문서를 가중치를 갖는 용어 빈도수의 벡터로 표현하는 것이다. 파싱(parsing)과 추출(extraction)의 전처리를 통해서 문서 i에 대한 용어 j의 빈도수 f_{ji}를 구하고, 전체 문헌 집합에서 용어 j를 포함하는 모든 문서의 수 d_j를 구한다[12]. 이와 같은 값을 이용하여, 문서 집합에 속한 n개의 문서에 d개의 서로 다른 용어가 있다고 할 때, i번째 문서 벡터 x의 j번째 원소를 다음 세 가지 항(term)의 곱으로 나타낼 수 있다.

$$x_{ji} = t_{ji} \cdot g_j \cdot s_i,\ 1 \le j \le d,\ 1 \le i \le n,$$
where x_i is a document vector

여기서 t_{ji}는 용어 가중치 성분으로서 그 값은 f_{ji}에 의해 결정되며, g_j는 전체 가중치 성분으로서 d_j에 의해 결정된다. s_i는 x_i에 대한 정규화 성분이다. 직관적으로 t_{ji}는 용어의 상대적인 중요도를 의미한다는 것과 g_j는 한 단어의 전체 문서 집합에서의 전반적인 중요도을 의미함을 알 수 있다. 이와 같은 가중치 계산의 목표는 다양한 문서 벡터간의 구분을 명확하게 하여 더 나은 분류 효과를 얻는 것에 있다[13].

위의 세 가지 성분을 선택하는 여러 가지 방안이 있으나[14], 본 논문에서는 대표적으로 많이 이용되는 정규화된 용어 빈도수(normalized term frequency)로 알려진 txn scheme을 이용한다.

이 계산법은 $t_{ji} = f_{ji},\ g_j = 1,\ s_i = \left(\sum_{j=1}^{m} (t_{ji} g_j)^2 \right)^{-1/2}$

로 하는 것이다. 주목할 점은 이러한 정규화가 $\|x_i\| = 1$을 의미한다는 것이다. 즉, 각 문서 벡터는 d차원 공간상의 단위 구(unit sphere)의 표면에 놓이게 되는 것을 의미하는 것이고, 이러한 정규화는 문서에 나타나는 용어의 방향성만을 유지하게 해주므로, 문서의 길이가 다르더라도 같은 주제를 다루는 문서들(즉, 유사한 용어들로 구성된 문서들)을 유사한 문서 벡터로 변환해 주는 효과가 있는 것이다[8].

2.2 개념 벡터

앞 절에서 설명한 벡터 공간 모델에 의해, 문서 집합을 이루는 n개의 문서를 $x_1, x_2, \cdots, x_n$의 벡터 집합으로 표현할 수 있다. 이때 두 문서 벡터 x_i와 x_j 사이의 코사인 유사도는 다음과 같은 두 벡터 사이의 내적(inner product)으로 간단히 구할 수 있다[10].

$$s(x_i, x_j) = x_i^T x_j = \|x_i\| \|x_j\| \cos(\theta(x_i, x_j)) = \cos(\theta(x_i, x_j))$$

여기서 두 벡터 사이의 각은 $0 \le \theta(x_i, x_j) \le \pi/2$이다.

n개의 문서 벡터들이 k개의 서로 다른 클러스터 $\pi_1, \pi_2, \cdots, \pi_k$로 나누어진다고 가정하면, $\{\pi_j\}_{j=1}^{k}$ 대해 π_j에 속한 문서들의 평균 벡터 혹은 중점 벡터는 다음과 같이 정의된다.

$$m_j = \frac{1}{n_j} \sum_{x \in \pi_j} x$$

여기서 n_j은 π_j에 속하는 문서벡터의 수이다. 이때 중점 벡터 m_j를 다음과 같이 단위 노름(norm)을 갖도록 정규화하면, 중점벡터의 방향성만을 갖는 개념 벡터(concept vector) c_j를 정의할 수 있다.

$$c_j = \frac{m_j}{\|m_j\|}$$

위와 같이 정의된 개념 벡터 c_j는 d차원의 공간상의 임의의 단위 벡터 z에 대해 Cauchy-Schwarz 부등식이 성립한다.

$$\sum_{x \in \pi_j} x^T z \le \sum_{x \in \pi_j} x^T c_j \quad (1)$$

위의 식 1에 의해 개념 벡터 c_j는 클러스터 π_j에 속해 있는 모든 문서 벡터에 대해 가장 근접한 코사인 유사도를 갖는 벡터임을 알 수 있다.

2.3 목적 함수

우리는 앞 절에서 설명한 식 1을 통해 클러스터 π_j에 대해 클러스터의 응집도(coherence) 혹은 클러스터링의 질(quality)을 다음의 수식을 통해 측정할 수 있다[1].

$$\sum_{x \in \pi_j} x^T c_j \quad (2)$$

만약 하나의 클러스터에 있는 모든 문서의 벡터가 동일하다면 해당 클러스터의 평균 응집도는 1이라는 최고의 값이 될 것이다. 이것은 또한 하나의 클러스터에 있는 문서벡터들이 매우 넓게 퍼져있다면 평균 응집도는 0에 가까운 값이 되는 것을 의미한다.

우리는 어떤 주어진 클러스터 $\{\pi_j\}_{j=1}^{k}$에 대해 다음의 목적 함수를 통해 전체 클러스터의 응집도를 측정할 수 있다.

$$Q(\{\pi_j\}_{j=1}^{k}) = \sum_{j=1}^{k} \sum_{x \in \pi_j} x^T c_j \quad (3)$$

위의 목적 함수는 k개의 모든 클러스터의 응집도의 합이라는 직관적으로 알 수 있다. 우리는 위의 목적 함수 값을 최대화하는 것을 목표로 실험을 진행했다.

3. 경계변수의 동적인 변경을 통한 진보된 클러스터링 알고리즘

이번 장에서는 spherical k-means(SKM)[9]의 벡터 공간 모델과 개념벡터를 토대로 ART (adaptive resonance theory)[11]에서 도입한 경계변수의 개념을 이용하여 새로운 클러스터링 알고리즘을 제안한다.

ART는 비감독 학습(unsupervised learning)에 의해 입력 패턴을 클러스터링하기 때문에, 사전에 학습 데이타를 통한 훈련 없이 새로운 입력 패턴을 학습할 수 있다. 또한 경계 변수(vigilance parameter) 값에 따라 클러스터링의 분류 결과를 조정할 수 있다. 즉, 경계 변수의 값을 크게 주면 좀 더 세분화되고 구체적인 클러스터를 얻을 수 있는 장점을 가진다. 그러나 ART 신경망을 통해 클러스터링을 시도할 경우 ART가 경쟁학습의 Winner-Take-All(WTA) 전략이기 때문에 SKM에 비해 낮은 목적 함수의 값을 갖게 되며, 또한 입력 데이타 집단에 포함되어 있는 이상치(outlier)나 노이즈(noise)에 대해 각각 클러스터를 생성해주기 때문에 이로 인한 overfitting의 문제점 등을 갖고 있다.

ART에는 이진 벡터를 클러스터링하는 ART1과 아날로그 벡터를 클러스터링하는 ART2가 있으며, 퍼지 ART는 ART1의 교집합 연산을 퍼지 집합 이론의 *min* 연산으로 대치함으로서 ART1이 아날로그 벡터에 대하여 학습할 수 있도록 한 것이다.

우리는 기존의 SKM와 퍼지 ART 신경망을 결합하는 새로운 알고리즘을 제안한다.

제안하는 알고리즘에서는 벡터 공간모델을 이용하여 개념벡터를 이용한 코사인 유사도를 바탕으로 클러스터링을 수행함과 동시에 각 클러스터마다 고유한 경계변수를 주어 클러스터별로 경계변수의 동적인 변경을 허용하였다. 이를 통해 클러스터링의 효율을 나타내는 목적 함수의 값을 극대화시키고 또한 모든 입력 패턴을 처리한 후에 이상치나 노이즈에 의해 생성된 클러스터를 해제함을 통해 ART에서 발생하는 overfitting의 문제점을 해결하였다.

그리고 제안하는 알고리즘은 각 클러스터마다 고유한 경계변수를 갖도록 하였기 때문에 새로 생성되는 클러스터의 경계 변수 값을 일관성 있게 설정하기 위해 전체 경계변수 ρ_g를 사용하였으며 또한 각 클러스터 별로 경계변수를 조절하기 위해 클러스터 π_j에 대해 경계변수 ρ_j를 사용한다.

다음은 제시하는 알고리즘에 대한 세부적인 설명이다.

초기화 : 클러스터 개수 k는 1로 초기화하고, 입력 패턴(문서 벡터)들을 단위 L_2 노름(norm)을 갖도록 정규화한다. 또한 처음 입력 패턴으로 하나의 클러스터를 형성하고, 전체 경계변수 $\rho_g^{(0)}$와 첫 번째 클러스터의 경계변수 $\rho_1^{(0)}$를 0.5 이상의 높은 값으로 정한다.

$$w_1^{(0)} = x_1 \ , \ \rho_g^{(0)} \in [0.5, 1] \ , \ \rho_1^{(0)} = \rho_g^{(0)}$$

입력 패턴 x_i와 클러스터 π_j의 매칭 정도를 코사인 유사도에 의해 측정하기 때문에, 초기에 경계변수 값을 높게 설정함으로서 초기에 많은 수의 클러스터가 생성되도록 하는 효과가 있다.

활성화 함수(Activation Function : AF) : 입력 패턴과 가중치 벡터 사이의 매칭 정도를 측정하는 활성화 함수는 다음과 같이 두 벡터 사이의 코사인 유사도로 계산한다.

$$AF(w_j^{(t)}, x_i) = \cos(\theta(w_j^{(t)}, x_i)) = x_i \cdot \frac{w_i^{(t)}}{\|w_j^{(t)}\|} = x_i^T \cdot c_j^{(t)} \quad (4)$$

이 때 가중치 벡터 $w_j^{(t)}$는 클러스터 π_j의 입력 패턴들의 합이다:

$$w_j^{(t)} = \sum_{x \in \pi_j} x$$

여기서 우리는 퍼지 ART와 달리 매칭 함수(Matching Function)를 따로 계산하지 않고 활성화 함수로 매칭 함수를 대치한다. 식 (3-1)의 활성화 함수는 매칭 함수의 기능을 포함하고 있기 때문에 둘을 하나로 통일함으로서 매칭 함수를 위한 부가적인 계산을 줄일 수 있다.[8]

Resonance 클러스터의 선택, 경계변수 조정 : 활성화 함수가 매칭 함수를 대신하므로 다음의 식에 의해 resonance 클러스터를 찾는다.

$$AF(w_{j*}^{(t)}, x_i) \geq \rho_{j*}^{(t)}\ ,\ j* = \arg\max_{j=1,\cdots,k}\{AF(w_j^{(t)}, x_i)\} \quad (5)$$

즉, 입력 패턴과 가장 유사한 클러스터의 가중치 벡터가 해당 클러스터의 경계 변수 조건을 만족하는가를 검사한다.

위의 식 5를 만족하지 않으면, 그 다음 유사한 클러스터를 찾아서 역시 식 5에 따라 평가를 한다. 만약 모든 클러스터에 대해 경계변수 조건을 만족하는 클러스터가 나오지 않으면 새로운 클러스터를 형성하고 해당 입력 패턴을 할당한다. 이 때 새로 생성되는 클러스터 π_j의 경계변수 $\rho_j^{(t)}$의 값은 $\rho_g^{(t)}$가 되도록 한다. 이는 새로운 클러스터의 경계변수 값이 현재의 전체적인 경계변수 값을 따를 수 있도록 하여 새로 생성되는 클러스터의 경계변수가 너무 높아 입력 패턴이 할당되지 않아 고립되는 현상을 방지한다.

새로운 클러스터를 형성하면서 모든 클러스터의 경계변수와 전체 경계변수의 값을 아래 식에 따라 동일한 수치만큼 떨어뜨려 계속해서 클러스터가 생성되는 확률을 낮추어 잠재적인 overfitting의 문제점을 해결하고 입력 패턴들이 기존의 클러스터에 수렴되도록 유도한다. 여기서 제어 변수의 값을 생성된 전체 클러스터 수(n)로 나누어줌으로서 클러스터가 생성되는 확률을 조정한다.

$$\begin{aligned}&\rho_j^{(t+1)} = \rho_j^{(t)} - \zeta/n\ ,\ (j=1,\cdots,k),\\ &\rho_g^{(t+1)} = \rho_g^{(t)} - \zeta/n\ ,\ \zeta \in [0, 0.02]\ ,\\ &\text{where } \zeta \text{ is a control parameter and } n\\ &\textit{is} \text{ the number of cluster}\end{aligned} \quad (6)$$

가중치 갱신, 경계변수 조정 : 선택된 클러스터 π_{j*}에 대해 식 (3-2)가 만족하면, 입력 패턴을 π_{j*}에 할당하고 다음의 식에 의해 π_{j*}의 가중치 벡터와 개념벡터를 조정한다.

$$w_{j*}^{(t+1)} = w_{j*}^{(t)} + x_i\ \ ,\ \ c_{j*}^{(t+1)} = \frac{w_{j*}^{(t+1)}}{\|w_{j*}^{(t+1)}\|} \quad (7)$$

또한, 해당 클러스터의 경계변수의 값을 다음의 식에 의해 변경한다.

$$\begin{aligned}&\rho_{j*}^{(t+1)} = \rho_{j*}^{(t)} + \delta\ ,\ \delta \in [0, 0.002]\\ &\textit{where } \delta \text{ is a control parameter}\end{aligned} \quad (8)$$

여기서 클러스터 π_{j*}의 경계변수 ρ_{j*}의 값을 조정하는 것은 하나의 클러스터에 입력 패턴이 치중되는 것을 막기 위함이다. 즉, 클러스터에 속하는 입력 패턴의 수가 늘어나면 해당 클러스터 가중치 벡터의 원소 중 0이 아닌 값(nonzero)이 늘어나게 되고 벡터의 내적을 이용하는 코사인 유사도 측정시에 다른 클러스터에 비해 상대적으로 높은 유사도가 나오게 한다. 이는 하나의 클러스터에 입력 패턴이 급격히 몰리는 현상을 발생시키며 결과적으로 전체 목적 함수의 값을 떨어뜨리게 된다. 따라서 입력 패턴을 할당받은 클러스터의 경계변수 값을 높임으로 이러한 현상을 방지한다.

클러스터의 병합, 경계변수 조정 : 식 (3-3)에 의해 클러스터의 경계변수의 값이 작아지면, 임의의 클러스터 π_i와 π_j 사이에 아래 식 (3-6)이 성립하는 것이 나온다.

$$AF(c_i^{(t)}, c_j^{(t)}) \geq \max(\rho_i^{(t)}, \rho_j^{(t)}) \quad (9)$$

입력 패턴이 클러스터에 할당됨에 따라 주기적으로 모든 클러스터에 대해 식 (3-6)이 성립하는가를 점검하여 그 중에서 가장 높은 유사도를 갖는 클러스터 간에 병합 처리를 한다. 즉, 하나의 클러스터를 해제하고 가지고 있던 모든 입력 패턴들을 다른 하나에 할당하고, 입력 패턴들이 할당됨에 따라 새로 형성되는 개념벡터와 클러스터에 속해있는 입력 패턴들에 대한 유사도를 다시 측정하여 유사도가 경계변수보다 작은 것들은 재배치 되도록 한다. 또한 클러스터가 병합되었기 때문에 식 (3-3)의 역으로 경계변수의 값을 높여준다.

$$\begin{aligned}\rho_j^{(t+1)} &= \rho_j^{(t)} + \zeta/n \quad (j=1,\cdots,k),\\ \rho_g^{(t+1)} &= \rho_g^{(t)} + \zeta/n\end{aligned} \quad (10)$$

군소 클러스터(minor cluster)의 해제 : 위의 과정을 모든 입력 패턴이 처리되기까지 적용한 후 군소 클러스터를 해제해 주어야한다. 군소 클러스터란 입력 패턴이 모두 처리됨에도 불구하고 다른 클러스터에 비해 현저하게 작은 입력 패턴을 할당받은 클러스터들을 일컫는 것으로 이러한 클러스터들은 이상치나 노이즈등에 의해 잘못 생성된 경우로서 불필요하게 클러스터의 수를 늘리는 overfitting의 원인이 된다. 따라서 이러한 군소 클러스터들을 해제하여 주요 클러스터에 배치해줌으로 불필요한 클러스터를 제거하고 또한 최종적으로 생성되는 클러스터 수를 조절한다. 실제로 우리의 실험에서는 각 클러스터 할당된 입력패턴 수를 가지고 클러스터 수를 조절하였다.

표 1은 전체 알고리즘의 개요를 나타낸 것이다.

4. 실험 결과

본 논문에서 제안된 방법의 유용성을 입증하기 위해 대표적인 데이타 집합인 CLASSIC3에 대해 실험을 수행하였다. 먼저 문서의 벡터화를 위해서 문서로부터 빠르게 벡터를 생성해주는 MC[15] 프로그램을 사용하였고, 생성된 벡터는 0이 아닌 값만을 저장하는 CCS (Compressed Column Storage)[16] 포맷을 사용하여 저장 공간과 코사인 유사도 측정시 계산량을 줄였다.

CLASSIC3는 잘 알려진 세 개의 문서 데이타 집합 MEDLINE, CISI, CRANFIELD (ftp://ftp.cs.cornell.edu/pub/smart)에서 추출된 3893개의 문서로 이루어져 있다. MEDLINE은 1033개의 의학 논문의 요약문으로 구성되어 있고, CISI는 1460개의 정보 검색에 대한 논문 요약문으로 구성되어 있고, CRANFIELD는 1400개

표 1 제안하는 방법의 알고리즘

1. 초기화(처음 데이타를 가지고 하나의 클러스터 형성하고 경계변수 값 설정)
$$w_1^{(0)} = x_1 \ , \ \rho_g^{(0)} \in [0.5, 1] \ , \ \rho_1^{(0)} = \rho_g^{(0)}$$
2. 입력 패턴을 받아들인다. 더 이상 받아들일 입력 패턴이 없으면 step 6으로 간다.
3. 받아들인 입력 패턴과 기존의 클러스터와의 유사도를 측정하여 유사도가 해당 클러스터의 경계변수보다 큰 것이 있는지를 검사한다.
$$AF(w_j^{(t)}, x_i) = \cos(\theta(w_j^{(t)}, x_i)) = x_i \cdot \frac{w_i^{(t)}}{\|w_j^{(t)}\|} = x_i^T \cdot c_j^{(t)}$$
4. 만약 유사도가 해당 클러스터의 경계 변수보다 크면
   4.1 해당 클러스터에 입력 패턴을 할당하고, 클러스터의 가중치와 개념벡터를 수정하고 해당 클러스터의 경계변수를 일정 수치만큼 높인다.
$$\rho_{j*}^{(t+1)} = \rho_{j*}^{(t)} + \delta \ , \ \delta \in [0, 0.002]$$
   4.2 step 2로 간다.
5. 만약 3에서 계산한 유사도보다 작은 경계변수를 갖는 클러스터가 없으면
   5.1 입력을 바탕으로 새로운 클러스터를 생성하고 전체경계 변수를 이용하여 새로 생성된 경계변수 값을 설정하고 전체 경계변수 값과 모든 클러스터의 경계변수 값을 전체 클러스터 수(n)에 반비례하는 일정 수치만큼 낮춘다.
$$\rho_j^{(t+1)} = \rho_j^{(t)} - \zeta/n \ , \ (j = 1, \cdots, k) \ ,$$
$$\rho_g^{(t+1)} = \rho_g^{(t)} - \zeta/n \ , \ \zeta \in [0, 0.02]$$
   5.2 조건식에 따라 서로 병합할 수 있는 클러스터가 있는가를 주기적으로 검사하고 조건을 만족하는 클러스터 중에서 가장 높은 유사도를 갖는 클러스터간에 병합하고 해당 클러스터의 경계변수 값을 전체 클러스터 수에 반비례하는 일정수치만큼 높인다.
$$AF(c_i^{(t)}, c_j^{(t)}) \geq \max(\rho_i^{(t)}, \rho_j^{(t)})$$
$$\rho_j^{(t+1)} = \rho_j^{(t)} + \zeta/n \quad (j = 1, \cdots, k) \ ,$$
$$\rho_g^{(t+1)} = \rho_g^{(t)} + \zeta/n$$
   이때, 병합된 클러스터의 각 입력 패턴 중 변경된 개념 벡터에 대해 유사도 값이 경계 변수보다 작은 것은 재배치한다.
   5.3 step 2로 간다.
6. 일정한 입력 패턴 수를 확보하지 못한 클러스터는 해제하고 해당 입력 패턴은 가장 유사도가 높은 기존 클러스터에 재배치한다

의 항공 시스템 논문의 요약으로 구성되어 있다.

먼저 3893개의 데이타를 MC 프로그램을 이용하여 벡터화한다. 이때, 불용어와 0.2%이하의 저 빈도수(low-frequency) 용어와, 15%이상의 고 빈도수(high-frequency)를 갖는 용어를 제거한다[1]. 그 결과 4262개의 용어가 남게 되며, 따라서 각 문서 벡터는 4262차원을 갖는 고차원 벡터가 된다. 그리고 txn scheme를 이용하여 3893개의 문서벡터를 만든다. 각 문서 벡터는 4262차원의 고차원 벡터이나 평균적으로 각 문서 벡터는 약 40개정도의 nonzero 원소만을 갖고 있는 매우 sparse한 벡터이다.

표 2와 그림 1은 제안하는 방법과 기존의 SKM와의 실험결과를 비교한 것이다. 사용한 파라미터의 값들은 경험적인 수치를 이용한 것이며 700개의 입력을 받아들일 때마다 최고 유사도를 갖는 클러스터끼리 병합해주었다. 여기서 첫 번째 열의 값은 클러스터의 수이고, 두 번째 열의 값은 SKM알고리즘의 목적 함수(식 (3))

표 2 CLASSIC3 실험결과
($\rho_1^{(0)}$=0.55, δ=0.0015, ζ=0.017)

Number of Cluster	SKM	Proposed Method	Rate
1	685.12	782.08	14.15%
2	851.27	1052.54	23.64%
3	922.61	1101.91	19.43%
4	1016.1	1097.27	7.99%
5	1036.31	1203.24	16.11%
6	1092.54	1167.79	6.89%
7	1133.87	1178.12	3.90%
8	1152.07	1225.84	6.40%
9	1184.61	1195.68	0.93%
10	1207.34	1236.77	2.44%
11	1237.14	1238.29	0.09%
12	1251.33	1261.96	0.85%
13	1269.52	1290.61	1.66%
14	1288.61	1280.74	-0.61%
15	1290.75	1282.11	-0.67%

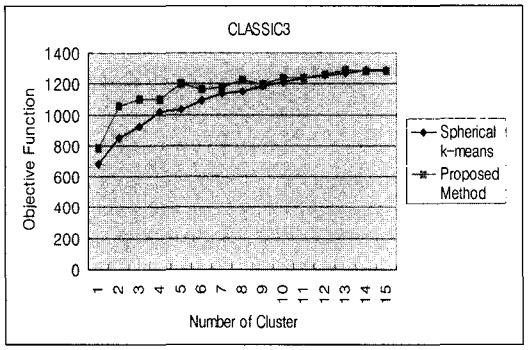

그림 1 CLASSIC3에 대한 목적 함수 값의 비교

표 3 MEDLINE에 대한 실험 결과

($\rho_1^{(0)}$=0.55, δ=0.0032, ζ=0.017)

Number of Cluster	Spherical k-means	Proposed Method	Rate
1	192.462	222.802	15.76%
2	228.125	266.545	16.84%
3	246.967	276.091	11.79%
4	265.68	290.132	9.20%
5	279.385	305.266	9.26%
6	294.136	316.446	7.58%
7	300.415	317.861	5.81%
8	307.889	332.984	8.15%
9	313.126	328.865	5.03%
10	328.759	335.093	1.93%
11	332.232	338.989	2.03%
12	342.056	343.481	0.42%
14	345.111	344.901	-0.06%

표 4 CISI에 대한 실험 결과

($\rho_1^{(0)}$=0.55, δ=0.004, ζ=0.017)

Number of Cluster	Spherical k-means	Proposed Method	rate
1	369.344	426.233	15.40%
2	432.328	499.074	15.44%
3	466.895	522.671	11.95%
4	493.167	521.021	5.65%
5	507.636	528.354	4.08%
6	522.301	537.025	2.82%
7	533.893	541.612	1.45%
8	544.024	547.713	0.68%
9	552.167	549.038	0.57%
10	561.833	552.356	-1.69%
13	576.799	547.425	-5.09%

값, 세 번째 열의 값은 제안하는 방법의 목적 함수 값이며, 마지막 열의 값은 SKM에 비해 제안하는 방법의 향상 정도를 백분율로 나타낸 것이다. 여기서 유의할 사항은 클러스터 수를 조절하기 위해서 클러스터 당 최소

표 5 CRANFIELD에 대한 실험 결과

($\rho_1^{(0)}$=0.55, δ=0.004, ζ=0.017)

Number of Cluster	Spherical k-means	Proposed Method	Rate
1	398.997	440.598	10.43%
2	445.017	485.056	9.00%
3	480.419	508.706	5.89%
4	504.503	520.962	3.26%
5	523.401	548.285	4.75%
6	541.232	550.432	1.70%
7	553.268	566.549	2.40%
8	563.997	565.895	0.34%
9	571.668	570.251	-0.25%
10	583.526	580.179	-0.57%
11	586.995	573.311	-2.33%
12	589.608	576.151	-2.28%
16	619.332	585.84	-5.41%

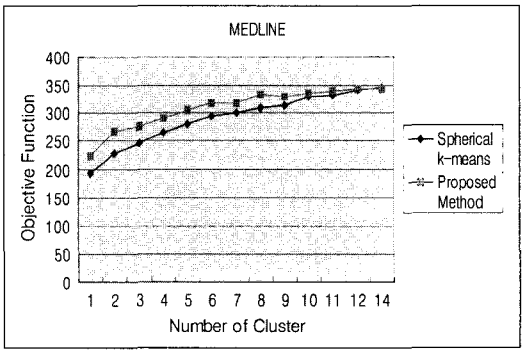

그림 2 MEDLINE에 대한 목적함수 값의 비교

데이타 수를 이용하였는데 최소 데이타 수가 160개 이하로 떨어지면서 제안하는 방법의 목적 함수 값이 SKM에 비해 더 낮아지는 것을 볼 수 있었다. 이러한 현상은 이상치나 노이즈에 의해 생성된 클러스터들이 제거되지 않고 남아서 불필요한 클러스터를 형성하고 있기 때문이다. 그러나 클러스터당 최소 데이타 수가 160을 넘게 되면 제안하는 알고리즘이 SKM에 비해 평균 8.04%의 향상된 응집도를 보여주고 있다.

다음으로 CLASSIC3 데이타 집합을 이루는 각 문서 데이타 집합인 MEDLINE, CISI, CRANFIELD에 대해 각각 실험을 하였다. 이때 불용어와 0.5%이하의 저 빈도 용어와, 15%이상의 고 빈도 용어를 제거한 후 txn scheme를 이용하여 문서벡터를 만들었으며 각 데이타 집합에 대한 용어의 수, 즉 차원은 MEDLINE이 2142, CISI가 1846, CRANFIELD가 1938가 된다. 표 3~5와 그림 2~6은 실험결과를 보여주며, 표와 함께 실험을 위해 사용한 파라미터 값들을 표시하였다. 실험 결과가 보

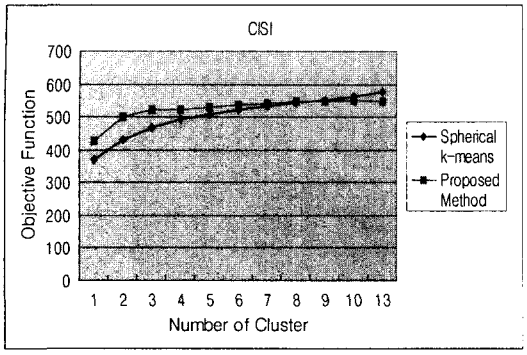

그림 3 CISI에 대한 목적함수 값의 비교

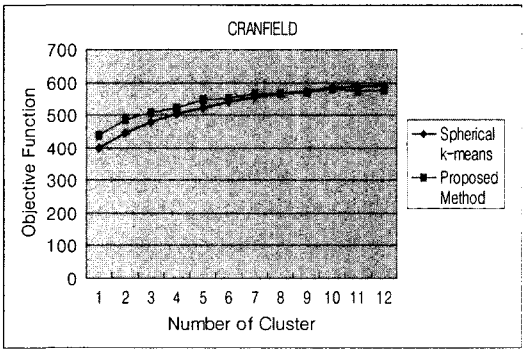

그림 4 CRAN에 대한 목적함수 값의 비교

여주는 것처럼, 제안하는 방법은 점층적이며 자동으로 클러스터 수를 생성해줌과 동시에 실제 분류 성능을 보여주는 목적 함수 값도 SKM에 비해 더 높은 값을 나타냄을 알 수 있다.

5. 결 론

본 논문에서 제시한 알고리즘은 SKM의 공간 벡터 모델과 개념벡터를 토대로 퍼지 ART의 경계 변수의 동적인 변경을 통한 새로운 클러스터링 알고리즘으로서, 기존의 SKM에서 할 수 없었던 적합한 클러스터 수를 자동으로 결정하며 점층적인 방법으로 클러스터링을 수행하는 장점을 지닌 것과 동시에 군소 클러스터의 해제를 통해 이상치나 노이즈에 의해 발생되는 문제점을 해결하였으며 생성된 클러스터의 응집도를 측정하는 목적 함수 값에서도 SKM에 비해 우수한 성능을 지녔음을 실험을 통해 확인해 보았다.

이러한 결과를 토대로 클러스터링의 강도를 조절하는 파라미터 값들을 지능적으로 선택해 주는 연구를 통해 알고리즘의 완성도를 높이려 한다.

참 고 문 헌

[1] Duda R. O. and Hart P. E., *"Pattern Classification and Scene Analysis,"* Wiley, 1973.

[2] Mitchell T., *"Machine Learning,"* McGraw Hill, 1997.

[3] Zamir O. and Etzioni O., *"Web Document Clustering : A Feasibility Demonstration," Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR '98)*, pp.46-54, 1988.

[4] Zamir O. and Etzioni O., *"Grouper : A Dynamic Clustering Interface to Web Search Results," Computer Networks Journal*, Vol.31, pp.1361-1374, 1999.

[5] Modha D. S. and Spangler W. S., *"Clustering Hypertext with Applications to Web Searching," Proceedings of ACM Hypertext Conference*, 2000.

[6] Leouski A. and Croft W. B., *"An Evaluation of Techniques for Clustering Search Results," Technical Report IR-76*, University of Massachusetts at Amherst, 1996.

[7] Hearst M. A. and Pedersen J. O., *"Reexamining the Cluster Hypothesis : Scatter/Gather on Retrieval Results," Proceedings of ACM SIGIR '96*, pp.76-84, 1996.

[8] 임영희, "후처리 웹 문서 클러스터링 알고리즘", 정보처리학회 논문지, 제9-B권, 제1호, pp7-16, 2002.

[9] Dhillon I. S. and Modha, D. S. *"Concept Decomposition for Large Sparse Text Data using Clustering," Technical Report* RJ 10147(9502), IBM Almaden Research Center, 1999.

[10] Salton G. and. McGill M. J., *"Introduction to Modern Retrieval."* McGraw-Hill Book Company, 1983.

[11] Carpenter G. A., Grossberg S. and Rosen D. B., *"Fuzzy ART : An Adaptive Resonance Algorithm for Rapid, Stable Classification of Analog Patterns," Proceedings of 1991 International Conference Neural Networks*, Vol.II, pp.411-416, 1991.

[12] Frakes W. B. and Baeza-Yates R., *"Information Retrieval : Data Structures and Algorithms,"* Prentince Hall, Englewood Cliffs, New Jersey, 1992.

[13] Salton G., and Buckley C., *"Term-weighting approaches in automatic text retrieval," Information Processing & Management*, 4(5):513:523, 1988.

[14] Kolda T. G. and O'Leary D. P., *"A Semi-Discrete Matrix Decomposition for Latent Semantic Indexing in Information Retrieval," ACM Transactions on Information Systems*, 16, 322-346. 1998.

[15] Dhillon I. S., Fan J., and Guan Y., *"Efficient Clustering of Very Large Document Collections"* Data Mining for Scientific and Engineering

Applications, Kluwer Academic Publishers, 2001. available at http://www.cs.utexas.edu/users/jfan/dm/.
[16] Available at http://www.cs.utexas.edu/users/inderjit/Resources/sparse_matrices.

신 광 철
1996년 중앙대학교 전자계산학과 졸업(공학사). 1998년 중앙대학교 컴퓨터 공학과 공학 석사. 2001년~현재 중앙대학교 컴퓨터 공학부 박사과정. 관심분야는 MPEG-21, 정보 검색, 웹 마이닝

한 상 용
1975년 서울대학교 공과대학 졸업(공학사). 1977년~1978년 KIST 연구원. 1984년 미네소타 대학 컴퓨터공학과 공학 박사. 1984년~1995년 미국 IBM 연구소 책임 연구원. 1995년~현재 중앙대학교 컴퓨터 공학부 교수. 관심분야는 MPEG-21, 웹 서비스, 시맨틱 웹, 정보 검색