온라인 문서 군집화에서 군집 수 결정 방법

지 태 창[†] · 이 현 진[††] · 이 일 병[†††]

요 약

군집화는 주어진 데이터를 분할하여 데이터 속에 숨겨져 있는 의미를 자동으로 발견하는 방법으로, 사람이 일일이 살펴보기 어려운 데이터를 분석해서 비슷한 성향을 가진 데이터들끼리 모은 여러 개의 군집들을 만들어 낸다. 온라인 문서 군집화는 검색 엔진을 통해 검색된 문서들을 대상으로 군집화를 실행하여 유사한 특성의 문서들을 묶어서 보여줌으로써 사용자의 검색 환경의 편의성을 증진시키는 것이 목적이다. 문서 군집화는 사람의 개입이 없이 자동으로 이루어져야 하고, 군집화 결과에 영향을 미치는 군집의 개수 선정도 자동으로 이루어져야 한다. 또한, 온라인 시스템에서는 빠른 응답 시간을 보장하는 것이 중요하다. 본 논문에서는 기하학적인 정보를 이용하여 군집의 수를 결정하는 방법을 제안한다. 제안하는 방법은 군집의 중심을 저차원 평면에 사상하는 것과 사상된 군집 중심의 거리 정보를 이용하여 군집들을 병합하는 두 단계로 이루어져 있다. 제안하는 방법을 실데이터에 적용하여 실험한 결과 군집화 성능이 향상되고, 처리 시간도 온라인 환경에 적합한 것을 확인 할 수 있었다.

키워드 : 온라인 문서, 문서 군집화, 군집 수 최적화, 군집 수 결정, K-Means 알고리즘, 다차원 척도법

Determining the number of Clusters in On-Line Document Clustering Algorithm

Tae-Chang Jee[†] · Hyunjin Lee[††] · Yillbyung Lee[†††]

ABSTRACT

Clustering is to divide given data and automatically find out the hidden meanings in the data. It analyzes data, which are difficult for people to check in detail, and then, makes several clusters consisting of data with similar characteristics. On-Line Document Clustering System, which makes a group of similar documents by use of results of the search engine, is aimed to increase the convenience of information retrieval area. Document clustering is automatically done without human interference, and the number of clusters, which affect the result of clustering, should be decided automatically too. Also, the one of the characteristics of an on-line system is guarantying fast response time. This paper proposed a method of determining the number of clusters automatically by geometrical information. The proposed method composed of two stages. In the first stage, centers of clusters are projected on the low-dimensional plane, and in the second stage, clusters are combined by use of distance of centers of clusters in the low-dimensional plane. As a result of experimenting this method with real data, it was found that clustering performance became better and the response time is suitable to on-line circumstance.

Key Words : On-Line Document, Document Clustering, Optimizing the Number of Clusters, Determining the Number of Clusters, K-Means Algorithm, Multi-Dimensional Scaling

1. 서 론

군집화는 주어진 데이터 또는 개체를 군집으로 분할하는 것으로, 기계학습의 무교사학습 (unsupervised learning)에 해당한다. 군집화는 유사한 데이터 개체들의 집합인 군집으로 데이터를 분할함으로써 데이터 속에 숨겨져 있는 의미 있는 정보를 자동으로 발견하는 것이다[7]. 문서 군집화(Document Clustering)의 목표는 해당 문서를 정확하게 다른 문서와 구분 하는 것이 아니라, 유사한 특징을 갖는 문서들을 함께 모아둠으로써 사용자가 심도 있는 분석을 할 수 있도록 보조하는 것이다.

군집화와 함께 잘 논의 되는 인공지능의 또 다른 지식 분류 방법인 분류(Classification)는 처음부터 목표값(target value)이 명확하기 때문에 출력 개수는 명확하게 된다. 하지만, 군집화는 그 특성상 입력대상에 대한 목표값이 없기 때문에 사용자가 임의로 목표 군집의 개수를 설정해야 한다.

※ 본 연구는 (주)웹스 및 산업자원부 연구과제로 이루어졌음.
† 정 회 원 : 연세대학교 컴퓨터과학과 박사과정
†† 정 회 원 : 한국싸이버대학교 컴퓨터정보통신학부 조교수
††† 정 회 원 : 연세대학교 컴퓨터과학과 교수
논문접수 : 2007년 2월 5일, 심사완료 : 2007년 10월 17일

하지만, 좋은 군집화 결과를 얻기 위해서는 최적의 군집 개수를 설정하는 것이 중요하다. 일반적으로는 군집화를 수행할 때 군집의 개수를 여러 번 다르게 설정하여 각각에 대해 군집화를 수행하고, 그 중 가장 군집화 결과가 좋은 것을 최종 군집 개수로 선택한다. 그러나, 이러한 방법은 일반 사용자들이 사용하기에는 어려움이 있다. 왜냐하면, 군집의 개수에 따라서 다른 군집화 결과가 나오고, 최적의 군집 개수는 사용자가 다양한 군집 개수를 실험해서 결정해야 하기 때문이다. 따라서, 자동으로 군집의 개수를 결정하는 방법이 필요하지만, 군집화 알고리즘에 대한 연구에 비해서는 그 연구는 미비하고, 좋은 연구성과가 미비한 것이 현실이다[15][18].

온라인 시스템에서 정보를 검색할 때 사용자들은 빠른 시간에 결과를 얻기를 원한다. 그렇기 때문에 온라인 문서 검색에 사용되는 문서 군집화 시스템도 군집화 시간이 오래 걸리면, 그 결과가 좋을 지라도 사용자에게 불편한 시스템이 된다. 따라서 이 시스템 상의 군집 수 최적화 방법은 빠른 시간 내에 군집 수를 최적화 할 수 있어야 한다.

본 논문에서는 온라인 문서 검색 시스템에 적용할 수 있는 최적화된 군집화 방법을 제안 한다. 이를 위하여 사용자의 반복적인 실험 없이, 자동적으로 군집의 수를 최적화 함으로서 최적의 군집을 형성하여 사용의 편리성과 성능 향상을 할 수 있는 방법을 제안한다. 온라인 문서 검색에 사용되기 위해서는 시간적인 요소가 중요한 요인이 되기 때문에 최적화를 수행하는데 있어서 시간상의 부담을 최소화 하는 것이 관건이다. 따라서 본 논문에서는 큰 군집 수에서 군집의 합병을 통해 개수를 축소하는 방법을 적용하여 최적화시 재군집화로 인한 시간적인 부담을 최소화 하도록 설계하였다. 군집 결정시에는 다차원 척도법에 의해 군집의 중심을 2차원 평면에 사상하고, 이 사상된 군집 중심들의 기하학적인 정보를 이용하여 군집들을 결합하여 군집 수를 최적화하는 방법을 제안하였다.

본 논문의 구성은 다음과 같다. 먼저, 2장에서는 군집화와 관련된 연구를 살펴보고, 3장에서는 군집 수 최적화를 자동적으로 수행할 수 있는 제안하는 군집 개수 결정 방법을 설명한다. 그리고 4장에서는 제안한 군집 개수 최적화 방법을 이용한 실험결과를 분석함으로써 그 유용성을 보인다. 마지막으로 5장의 결론에서는 연구의 의의와 향후 연구 과제를 제시한다.

2. 관련 연구

2.1 군집화 알고리즘

문서 군집화에는 군집화 알고리즘이 이용된다. 군집화는 데이터의 구조 등과 같은 사전 정보 없이 분석을 시작한다는 점에서 '자료의 탐험' 또는 '자료의 발굴' 과정으로 볼 수 있다. 그리고 양적으로 많은 데이터를 훨씬 작은 개수의 동질적 집단으로 군집화함으로써 최소의 정보손실을 통한 '자료의 집약' 내지 '자료의 단순화'의 과정으로 볼 수 있다. 군집화의 결과는 모집단의 구조적 특성에 관한 정보를 도출함으로써 '가설의 형성' 단계와도 연계된다[3][11][12].

대상들을 군집화하는 방법은 매우 다양하지만 모든 방법이 공통적으로 가지고 있는 기본전제는 군집 내의 객체들 간의 유사성을 극대화 하고, 군집간의 유사성은 극소화하는 것이다. 군집화 방법으로는 자기조직화 신경망(Self Organizing Map: SOM), 완전연결(Complete Linkage), K-Means 등이 있다[8].

자기조직화 신경망은 무교사학습 기반의 신경회로망으로써 그 성능은 좋지만, 같은 차원의 다른 문제에 대해 정해진 시간 내에 학습이 종료되리라는 보장을 할 수 없다. 완전연결은 계층적 기법(Hierarchical Clustering)으로 각 문서들 간의 거리를 비교하는 방법으로, 항상 수렴하고, 같은 차원의 다른 문제에 대해 항상 계산시간이 일정한 장점이 있지만, 문서-특징 벡터가 커지면 전체 계산시간이 기하급수적으로 증가하는 단점이 있다[6]. K-Means는 분할 기법(Partitional Clustering)으로 일정 시간 내에 계산이 종료할 수 있고, 문서-특징 벡터가 커져도 계산 시간이 완전연결 방법보다는 점진적으로 증가한다.

2.2 다차원 척도법

군집화의 결과는 한번에 알아보기 쉽지 않다. 원천 데이터가 2차원 또는 3차원으로 이루어져 있다면, 평면상에 나타내서 쉽게 결과를 알아 볼 수 있지만, 문서 데이터와 같이 특징 벡터의 차원이 큰 경우 평면에 나타내기 어려운 단점이 있다. 따라서, 고차원의 데이터를 저차원의 평면으로 사상하는 방법이 필요한데, 그 중의 하나가 다차원 척도법(MDS: Multi-Dimensional Scaling)이다[1][4].

다차원 척도법은 제품이나 브랜드, 객체 등의 대상들에 대한 유사성 평가자료를 이용하여 대상들이 어떤 차원에서 관찰되고, 이들 차원에서 각 대상은 어떤 위치를 차지하고 있는가를 그림으로 나타내주는 기법들의 총칭이다. 다차원 척도법은 다차원 공간상에서 자극 좌표 또는 가중치를 유도하기 위하여 유클리디안(Euclidean)과 가중치 유클리디안 모형을 이용한다. 다차원 척도법은 거리행렬상에서 분석되며 통계절차는 입력자료로써 기존의 거리행렬을 읽을 수 있으며 또는 원천 데이터로부터 거리행렬을 계산할 수도 있다. 다차원 척도법은 응답자의 어떤 대상에 대한 응답자들의 지각과 선호도에 관계되는 태도를 조사하는데 사용될 수 있는 분석기법으로써 시장세분화, 제품수명주기, 판매업체평가, 광고매체 선택 등을 결정하는 데에 주로 이용되고 있다[1].

2.3 군집 개수 설정에 관한 연구

최근 군집화 방법에 관한 연구들 중에서 최적의 군집화를 위해, 최적의 모델을 선택하는 방법을 적용하는 연구들이 등장하고 있다.

Yu(1998)는 베이지안 정보 기준(BIC:Baysian Information Criterion)을 사용하여 자동으로 군집의 수를 결정하는 방법을 연구하였다[22]. 1차원과 2차원 데이터에 대해 EM 군집

화 알고리즘을 적용하여 좋은 결과를 보였지만, 문서 군집화의 경우에는 베이지안 정보 기준 척도를 적용하기가 어려웠다.

Pelleg와 Moore(2002)는 K-means를 확장한 방법인 X-means를 제안하였는데, 이 방법에 군집의 수를 추정하는 기능을 추가하였다[18]. 여기서는 군집의 분리 여부를 판단할 때 베이지안 정보 기준을 사용하였다. 군집을 분리할 때 계산되는 정보 이득(Information Gain)이 군집을 유지할 때 계산되는 정보 이득보다 클 경우에 군집의 분리는 이루어지게 된다.

Liu와 Gong(2002)은 다음의 가정을 기반으로 모델 선택을 수행하는 방법을 제안하였다[15]. 이 방법에서 사용한 가정은 부정확한 솔루션 공간(solution space)에서 솔루션을 찾으면, 솔루션이 존재하지 않기 때문에 모델을 달리해서 수행한 군집화의 결과는 임의의 솔루션들을 제시한다는 것이다. 다른 관점에서 보면, 솔루션 공간에 단지 하나의 솔루션만 존재한다면 다양하게 수행한 군집화 결과는 유사한 결과를 보인다는 것이다. 이 가정을 적용한 방법은 군집의 수를 증가시키면서, 각 군집의 수에 대해 초기 군집의 위치를 달리해서 실험하여도 결과가 가장 일정한 군집의 수를 최적의 군집의 수로 선택하였다.

Salvador(2004)는 '군집의 수 vs. 군집 평가 척도' 그래프의 "무릎(knee)"을 발견하는 L 방법을 제안하였다[20]. 이 방법의 군집화와 분류(segment) 알고리즘은 군집과 분류의 수를 결정하는데 좋은 결과를 보였지만, 이 방법은 계층적 알고리즘에만 적용 가능하다.

Lu(2005)는 군집의 최적 수를 추정하는 진화(evolutionary) 알고리즘을 제안하였다[16]. 제안하는 진화 알고리즘은 새로운 엔트로피 (entropy) 기반 적합 함수(fitness function)와 군집을 분리하고, 결합하고, 제거하는 세 개의 새로운 유전 연산자 (operator)를 정의하였다. 이 방법을 사용하여 데이터 집합에서 최적의 군집 수를 추정 할 수 있었다.

Boutsinas(2006)은 z-window 군집화 알고리즘을 제안하였다[5]. 이 방법은 윈도잉 (windowing) 기법을 사용해서 군집의 수를 결정하는 것을 목표로 한다. 주 아이디어는 충분히 많은 수의 초기 윈도우를 설정하고, 알고리즘을 수행하면서 윈도우들을 적절하게 결합하는 것이다.

이 연구들은 모두 문서 군집화 알고리즘이 수행 중에 여러 모델을 생성하고, 그 중 가장 적합한 모델을 선택하여 최적의 모델을 찾는 방법들이다. 하지만 이러한 방법들은 실행 시간이 중요한 온라인 문서 군집화에서 적용하는 데는 문제가 있다. 따라서 본 논문에서는 군집화 알고리즘의 종류에 상관 없이 군집의 개수를 자동적으로 최적화 함으로써 최적의 모델을 생성하는 방법을 제안하였다.

2.4 평가 지표

군집화는 무교사학습 방법으로 교사학습 방법에서 사용하는 인식률이나 신뢰도 같은 성능 평가 척도를 사용할 수 없다. 군집화 결과를 평가할 수 있는 방법으로 여러 가지 연구가 진행되고 있는데, 그 중의 하나는 군집의 구조적인 형태로 평가하는 방법이다[9].

Cluster Compactness(Cmp)는 전체 입력 데이터의 분포와 비교하여 각 군집의 분포가 얼마나 잘 정렬되었느냐를 평가한다. Cmp는 다음과 같은 데이터의 분산에 기초한다.

$$v(X)=\sqrt{\frac{1}{N}\sum_{i=1}^{N}d^2\left(x_i,\bar{x}\right)} \tag{1}$$

여기서, $d\left(x_i,x_j\right)$는 벡터 x_i와 x_j 사이의 거리를 계산하는 척도이고, N은 입력 집합 X의 원소의 수이고, $\bar{x}=\frac{1}{N}\sum_i x_i$는 X의 평균이다. 작은 분산을 가진다는 것은 데이터의 동질성이 높다는 의미이다. Cmp는 다음과 같이 정의된다[9].

$$Cmp=\frac{1}{C}\sum_{i}^{C}\frac{v(c_i)}{v(X)} \tag{2}$$

여기서, C는 입력 집합에 의해 생성된 군집의 개수이고, $v(c_i)$는 군집 c_i의 분산이고, $v(X)$는 입력 데이터 X의 분산이다.

Cluster Separation(Sep)는 군집간에 얼마나 멀리 떨어져 있느냐를 평가하는 척도이다. Sep는 다음과 같이 정의된다[9].

$$Sep=\frac{1}{C(C-1)}\sum_{i=1}^{C}\sum_{j=1,j\neq i}^{C}\exp\left(-\frac{d^2\left(x_{c_i},x_{c_j}\right)}{2\sigma^2}\right) \tag{3}$$

여기서, σ는 가우시안 상수(Gaussian Constant)이고, C는 군집의 개수이고, x_{c_i}는 군집 c_i의 중심 좌표이고, $d()$는 군집화 시스템에 사용된 거리계산척도이며, $d\left(x_{c_i},x_{c_j}\right)$는 군집 c_i와 c_j의 중심 사이의 거리를 나타낸다.

Cmp와 Sep 모두 작은 값을 가질 수록 더 좋은 결과를 나타낸다고 알려져 있다[9]. 군집의 성능을 비교하는 척도가 Cmp와 Sep로 두 개이기 때문에 전체 성능을 비교하기는 어렵다. 따라서, 전체 군집 성능(Ocq)을 다음과 같이 계산한다[9].

$$Ocq(\beta)=\beta\cdot Cmp+(1-\beta)\cdot Sep \tag{4}$$

여기서, $\beta\in[0,1]$는 Cmp와 Sep의 가중치를 조정하는 역할을 한다. 이 전체 군집 성능이 작으면 더 좋은 성능을 보인다고 할 수 있다[9].

3. 제안하는 방법

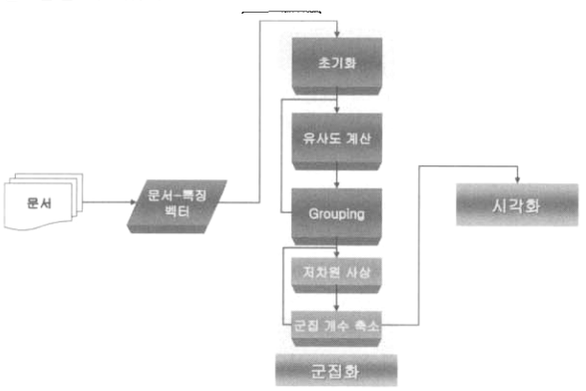
3.1 문서 군집화 시스템

제안하는 방법을 적용한 전체 문서 군집화 시스템의 구성도는(그림 1)과 같다.

우선, 검색 엔진을 통해서 웹 또는 데이터베이스로부터 문서들을 검색한다. 검색된 문서를 분석하여 문서에 있는 특징 단어들을 추출하여 문서-특징 벡터를 구성한다. 이 문서-특징 벡터를 입력으로 하여 K-Means 군집화 알고리즘을 적용하여 문서 군집화를 수행한다. 군집이 구성되면, 제안하는 군집 결정 알고리즘을 이용하여 군집의 개수를 최적화 한다. 마지막으로 문서들에 대한 시각화가 이루어진다.

본 논문에서는 전체 문서 군집화 시스템 중 자동 온라인 문서 군집화 시스템을 달성하기 위해 중요한 요소의 하나인 군집들의 수를 결정하는 방법에 대하여 제안한다.

제안하는 방법은 고차원에서 군집화가 이루어진 후 저차원에서 군집을 결합하는 방법으로, 고차원에서 군집을 결합하는 것보다 수행시간 측면에서 우수하고, 저차원에서 군집을 표현함으로써 시각화하기 쉬워, 시각적인 결과 해석의 이점을 얻을 수 있다.



(그림 1) 문서 군집화 시스템 구성도

3.2 문서-특징 벡터

문서-특징 벡터는 문서를 분석하여 문서에 나타나는 특징(키워드, 명사)을 추출하여 컴퓨터가 인식할 수 있는 형태로 변환한 것이다[2]. 문서-특징 벡터는 각 문서에서 특징을 추출하는 부분(<표 1>)과 추출된 특징을 문서와 결합하여 문서-특징 벡터로 만드는 부분(<표 2>)으로 구성된다.

<표 1> 문서 별 추출된 특징은 각 문서 별로 특징들을 추출하여 나열한 것으로 n은 문서의 개수, k는 한 문서에서 가장 많이 나온 특징의 개수, m은 전체 특징의 개수를 나타

〈표 1〉 문서 별 추출된 특징

	1	2	…	k
D_1	T_1	T_5	…	
D_2	T_3	T_9	…	T_m
…	…	…	…	
D_n	T_1	T_2	…	

〈표 2〉 문서-특징 벡터

	T_1	T_2	…	T_m
D_1	t_{11}	t_{12}	…	t_{1m}
D_2	t_{21}	t_{22}	…	t_{2m}
…	…	…	…	
D_n	t_{n1}	t_{n2}	…	t_{nm}

내며, $D_i(i=1,\cdots,n)$는 i번째 문서를 나타내며, $T_j(j=1,\cdots,m)$는 j번째 단어를 의미한다.

<표 2>는 <표 1>을 이용하여 문서-특징 벡터를 구성한 것으로 추출된 특징들에 대한 통계 정보를 이용하여 $n \times m$ 벡터를 구성한 것이다. 여기서, t_{ij} 는 i번째 문서에 나타난 j번째 특징의 발생 빈도수를 의미한다.

3.3 K-Means 군집화 알고리즘

K-Means는 n개의 입력 데이터를 K개의 군집으로 분할하는 방법이다[8][19]. 군집화에 영향을 미치는 요소는 초기 군집의 개수, 초기 군집 중심값과 유사성 계산 척도이다.

초기 군집의 개수는 임의의 수를 선택하는 방법이 일반적으로 사용된다. 본 논문에서는 임의의 수를 선택해도 항상 거의 일정한 결과 즉, 최종 군집의 개수는 거의 일정하다는 것을 보이기 위하여 군집의 개수를 3개에서 50개까지 증가시키면서 실험하였다.

초기 군집 중심값을 선택하는 방법은 임의로 K개를 선택하는 방법[10], 데이터 집합 순서로 K개를 선택하는 방법[8], 거리를 최적화 하는 방법[10]등이 있다. 임의로 값을 선택하는 방법은 초기값이 항상 같지 않기 때문에 항상 같은 결과를 보인다는 것을 보장할 수 없다. 순서대로 K개를 선택하는 방법은 항상 같은 결과를 보이지만, 초기에 비슷한 성향의 데이터가 모여 있는 경우 군집화 성능이 저하될 수 있다. 거리를 최적화 하는 방법은 초기값을 구하는 시간은 오래 걸리지만, 항상 같은 결과를 보이고, 군집화 성능도 좋은 결과를 보이게 된다. 본 논문에서는 거리를 최적화하는 방법 중 SCS (Simple Cluster Seeking)을 사용하였다[10]. SCS 수행 방법은 다음 <표 3>과 같다.

〈표 3〉 SCS 알고리즘

Step 1 : 첫 군집의 중심을 첫 입력 데이터로 초기화 ($C_1 = x_1$).

Step 2 : $j = 2,...,N$ 에 대하여, 모든 군집 중심 C_k 에 대해 $\| x_j - C_k \| > \rho$ 이면 x_j 를 새로운 군집 중심으로 설정, 여기서 ρ 는 임계치. K 개의 군집 중심이 초기화되면 종료.

Step 3 : 모든 입력 집합에 대하여 계산을 한 후 K 개보다 적은 군집 중심이 생성되면, ρ 를 감소하고, Step 1-2를 반복.

〈표 4〉 군집 유사성 척도

방법	수식	결과 품질	속도
유클리디안 거리	$d=\sqrt{\sum_{j=1}^{M}\left(x_{ij}-c_{kj}\right)^2}$	결과 편중	빠름
맨하탄 거리	$d=\left\lvert\sum_{j=1}^{M}\left(x_{ij}-c_{kj}\right)\right\rvert$	결과 편중	가장 빠름
피어슨 상관계수	$d=\frac{\sum_{j=1}^{M}x_{ij}c_{kj}-\frac{\sum_{j=1}^{M}x_{ij}\sum_{j=1}^{M}c_{kj}}{M}}{\sqrt{\left(\sum_{j=1}^{M}x_{ij}^2-\frac{\left(\sum_{j=1}^{M}x_{ij}\right)^2}{M}\right)\left(\sum_{j=1}^{M}c_{kj}^2-\frac{\left(\sum_{j=1}^{M}c_{kj}\right)^2}{M}\right)}}$	편중 없음	보통
코사인 상관계수	$d=\frac{\sum_{j=1}^{M}x_{ij}c_{kj}}{\sqrt{\sum_{j=1}^{M}x_{ij}^2\sum_{j=1}^{M}c_{kj}^2}}$	편중 없음	보통

여기서, C_k는 K번째 군집의 중심이고, N은 전체 데이터의 수이고, x_j는 j번째 데이터이다. SCS를 사용하면, 초기화에 시간이 걸리지만, 반복 횟수는 감소하여 전체 수행 시간은 감소하는 결과를 보였다.

유사성 계산 척도는 유클리디안 거리(Euclidean Distance), 멘하탄 거리 (Manhattan Distance), 피어슨 상관계수 (Pearson Correlation), 코사인 상관 계수 (Cosine Correlation) 등을 사용한다[8][17][19]. 각 유사성 계산 척도는 <표 4>와 같은 특징을 가지며, 본 논문에서는 코사인 상관 계수를 유사성 계산 척도로 사용했다.

<표 4>의 결과 품질은 군집화를 수행하고 난 후 각 군집에 소속된 문서들의 수를 살펴본 것이다. 결과 편중은 특정 군집에 전체 문서들이 소속되고, 절반 이상의 군집은 소수의 문서만 소속된 것이다. 편중 없음은 군집들에 소속된 문서들이 일정한 비율(5%~30%) 이내에 존재한다는 의미이고, 결과 편중이 없을 때 좋은 결과를 보인다고 할 수 있다. 군집의 개수 K는 분석 전에 설정하며, 입력된 문서 수에 따라 상대적으로 결정하였다.

본 논문에서 사용한 K-Means의 수행 방법[19]은 다음 <표 5>와 같다.

〈표 5〉 K-Means 알고리즘

Step 1:	K개의 군집 대표값 초기화.
Step 2:	각 입력 데이터를 가장 가까운 군집 중심에 할당.
Step 3:	군집에 속한 데이터의 평균으로 각 군집의 중심 재계산.
Step 4:	각 입력 데이터의 소속 군집이 변화가 없을 때까지 Step 2-3 반복.

3.4 다차원 척도법

본 논문에서 군집의 개수를 결정하는데 기하학적인 정보를 이용하였다. 다차원 데이터의 경우 특히, 문서 데이터와 같이 차원의 수가 수천 차원인 경우에 원 데이터에서 기하학적인 정보를 추출한다는 것은 거의 불가능한 일이다. 따라서 다차원 데이터를 저차원 데이터로 사상을 할 필요가 있는데, 이때 사용하는 알고리즘이 다차원 척도법이다. 본 논문에서 사용한 다차원 척도법의 알고리즘은 다음 <표 6>과 같다 [5].

〈표 6〉 다차원 척도법 알고리즘

Step 1: 점들을 p-차원 공간의 임의의 점에 할당한다.
Step 2: *Dhat* 매트릭스를 구성하기 위해 각 점의 쌍들의 거리를 계산한다.
Step 3: *Dhat* 매트릭스와 입력값인 D 매트릭스를 스트레스 함수(stress function)을 통해 비교한다. 이 값이 작을수록 두 매트릭스의 연관도가 크다는 것을 의미한다. 스트레스 함수 S는 다음과 같다.

$$S=\left[\frac{\sum_i\sum_j\left(d_{ij}-f\left(\delta_{ij}\right)\right)^2}{\sum_i\sum_j d_{ij}^2}\right]^{1/2}$$

Step 4: 스트레스를 최소화하는 방향으로 각 점의 좌표를 조정한다.
Step 5: 스트레스가 최소가 될 때까지 Step 2 - Step 4를 반복한다.

<표 6>의 수식에서, d_{ij}는 저차원상에서 계산된 거리를 의미하고, δ_{ij}는 입력 데이터 즉, 고차원상의 거리를 의미한다. $f\left(\delta_{ij}\right)$는 입력 데이터를 크기 순서로 변환하는 함수이다.

본 논문에서는 문서들을 시각적으로 평면에 표현하기 위한 전단계로 다차원 척도법을 이용하여 군집의 중심을 2차원 평면에 사상하였다. 문서 군집화가 완료되면, 처음에 결정한 숫자만큼의 군집 중심이 생기고, 군집 중심간의 거리를 계산할 수 있다. 이 군집 중심간의 거리에 다차원 척도법을 적용하여 2차원 평면에 사상한다. <표 7>은 12개의 군집 중심간의 거리를 계산한 것이다. 본 논문에서 거리를 비교하는 척도로 코사인 상관 계수를 사용했기 때문에 값은 0~1사이의 값이고, 0이 거리가 멀고, 1이 거리가 가까운 것을 의미한다. 이를 다차원 척도법에 적용하기 위해서는 0이 거리가 가깝고, 1이 거리가 멀어야 하기 때문에, 코사인의 역함수를 적용하였다. (그림 2)는 <표 7>의 값을 다차원 척도법으로 처리하여 2차원 평면상에 나타낸 것이다.

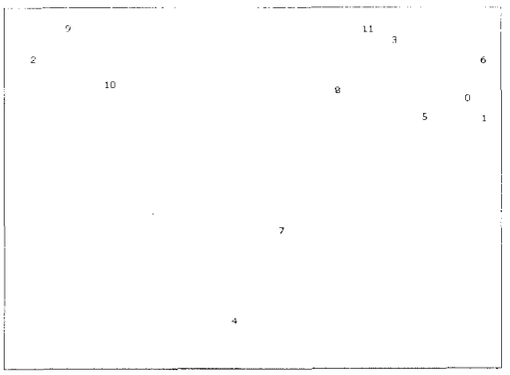
3.5 군집 개수 결정

본 논문에서 군집의 개수를 줄이는 방법은 2차원의 기하학적인 구조에 기반을 두고 있다. 다차원 척도법과 같이 차원을 축소하는 전통적인 기법은 적용하기 쉽고, 계산이 효율적이며, 다차원 공간상 데이터의 선형 부분공간(linear subspace)에 존재하는 진정한 구조를 발견하는 것을 보장한다[14][21]. 다차원 척도법은 다차원 공간상에서 계산된 데이터의 분산을 저차원 공간상에 가장 잘 유지시킬 수 있다 [14][21].

다차원 척도법을 이용하여 2차원 평면에 사상된 군집 중심을 살펴보면, 대부분의 군집 중심들은 다른 군집 중심과

〈표 7〉 군집 중심의 거리

	0	1	2	3	4	5	6	7	8	9	10	11
0	-	1.09	1.44	1.13	1.36	1.05	1.15	1.12	1.19	1.41	1.37	1.20
1	1.09	-	1.51	1.28	1.41	1.18	1.27	1.28	1.36	1.49	1.48	1.36
2	1.44	1.51	-	1.46	1.51	1.46	1.53	1.40	1.50	1.39	1.44	1.50
3	1.13	1.28	1.46	-	1.47	1.33	1.31	1.32	1.39	1.45	1.42	1.27
4	1.36	1.41	1.51	1.47	-	1.38	1.45	1.06	1.43	1.50	1.45	1.46
5	1.05	1.18	1.46	1.33	1.38	-	1.25	1.25	1.32	1.43	1.39	1.32
6	1.15	1.27	1.53	1.31	1.45	1.25	-	1.31	1.34	1.48	1.48	1.20
7	1.12	1.28	1.40	1.32	1.06	1.25	1.31	-	1.27	1.36	1.34	1.27
8	1.19	1.36	1.50	1.39	1.43	1.32	1.34	1.27	-	1.40	1.41	1.28
9	1.41	1.49	1.39	1.45	1.50	1.43	1.48	1.36	1.40	-	1.38	1.37
10	1.37	1.48	1.44	1.42	1.45	1.39	1.48	1.34	1.41	1.38	-	1.40
11	1.20	1.36	1.50	1.27	1.46	1.32	1.20	1.27	1.28	1.37	1.40	-



(그림 2) 다차원 척도법으로 군집 중심을 2차원 평면에 표시

적절한 거리를 두고 있지만,(그림 2)의 0-1, 3-11과 같이 일부 군집의 중심들은 상당히 가까운 거리에 인접해 있는 것을 확인할 수 있다. <표 7>의 원본 데이터를 보면, 군집 중심간의 거리는 1.05부터 1.53까지 분포되어 있다. 가까이 있는 군집을 결합할 때, 단순하게 가장 가까운 거리에 있는 군집들을 결합하게 되면, <표 7>에서는 거리가 1.06인 4-7을 결합하면 되지만, 실제로 이 두 군집이 결합될 만한 거리인지에 대한 기준이 없고, 다른 군집과의 관계를 고려해야 하기 때문에 <표 7>에서 가장 가까운 거리에 있는 군집들이 실제로 서로 합쳐질 수 있는지 확신할 수 없게 된다. 또한, 데이터가 달라지게 되면, 거리의 크기가 달라지게 되고, 이러한 경우에 어느 정도 거리의 군집들을 합쳐야 할 지 결정하기 어렵게 된다. 하지만, (그림 2) 와 같이 저차원 평면에 사상하면, 다른 군집과의 관계가 반영된 상태이기 때문에, 가장 가까운 거리에 있는 군집들을 서로 합칠 수 있고, 거리의 크기도 데이터가 바뀌어도 항상 일정하게 표현되기 때문에 군집을 결합하기 위한 기준을 결정할 수 있다.

본 논문에서 제안하는 알고리즘은 <표 8>과 같다.

〈표 8〉 제안하는 군집 개수 결정 알고리즘

Step 1: 역치 T를 계산한다.

$$T = \frac{1}{\left\langle \sqrt{NumC} \right\rangle} \times \alpha \qquad (5)$$

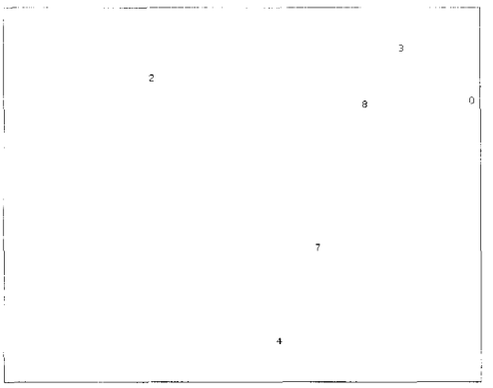
여기서, $NumC$는 군집의 개수이고, $\alpha \in [0,1]$는 군집을 합치는 거리의 한계치를 계산하기 위한 가중치이다.

Step 2: 3.3절의 다차원 척도법 알고리즘을 이용하여 2차원 평면상의 거리를 계산한다.

Step 3: 다차원 척도 결과를 이용하여 각 군집 간의 거리 매트릭스 $Dist$를 계산한다.

Step 4: $Dist$의 최소값이 T 보다 작으면, $Dist$의 최소값에 속한 두 군집을 병합한다.

Step 5: $Dist$의 최소값이 T 보다 클 때까지 2-4 단계를 반복한다.



(그림 3) 제안하는 군집 개수 결정 알고리즘을 적용하여 군집의 수가 6개로 줄어진 결과

12개의 군집 중심을 가지고 있는 (그림 2)에 대하여 <표 8>의 알고리즘을 적용한 결과는 (그림 3)과 같다. 군집 0에 군집 1, 군집 5, 군집 6이 결합되었고, 군집 2에 군집 9, 군집 10이 결합되었고, 군집 3에 군집 11이 결합되었다. $NumC$는 12이고, α는 0.8을 주었을 때, 6개의 군집으로 줄어든 것을 확인 할 수 있다.

4. 실험 결과

본 연구는 Pentium 4 2.8GHz CPU 시스템에서 C# 언어로 구현하였다. 사용된 컴파일러 버전은 .Net Framework 1.1이다. 군집화 알고리즘은 K-Means를 사용했다.

4.1 데이터 집합

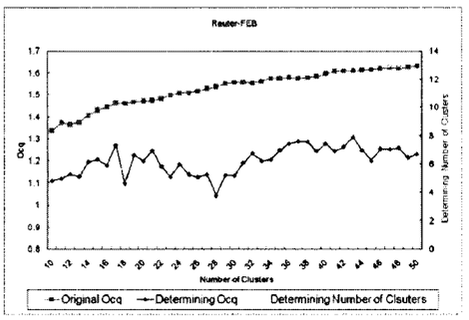
실험은 다양한 실제 데이터 집합을 사용하여 이루어졌다. 데이터 집합은 Reuter-21578 문서 집합(Reuters)[13]에서 다섯 개를 선택하여 사용하였다. <표 9>는 이 다섯 개의 데이터 집합에 대한 통계치의 요약이다.

〈표 9〉 실험에 사용된 데이터

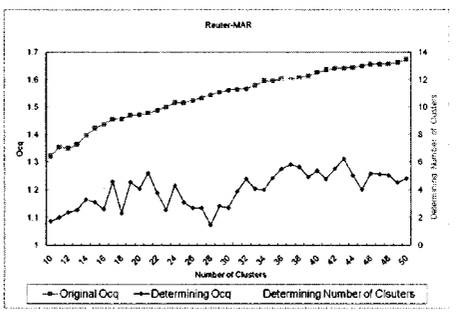
		Num. of instances	*Num. of features*	*Num. of clusters*
Reuter-21578 문서 집합	February	132	250	10
	March	6,030	4,411	10
	April	2,106	2,671	10
	June	988	1,740	10
	October	605	828	10

4.2 Reuter-21578 문서 집합 군집화 결과

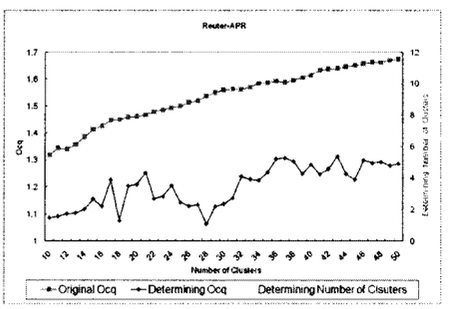
제안하는 방법의 문서에 대한 성능을 살펴보기 위하여 문서의 개수를 달리하여 Reuter 문서 집합을 이용하여 실험을 하였다. 군집의 수를 50까지 증가시키면서, 그에 따른 평가 지표(Ocq)의 변화와 제안하는 방법에 의한 군집 수의 변화, 평가 지표의 변화는 (그림 4)와 같다. 2.4 절에서 얘기한 바와 같이 Ocq는 작을수록 좋은 성능을 보이는 것이고, 다섯 개의 문서 집합 모두 더 작은 값 즉, 더 좋은 성능을 보이



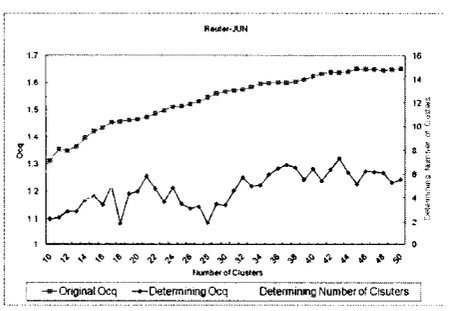
(a) February



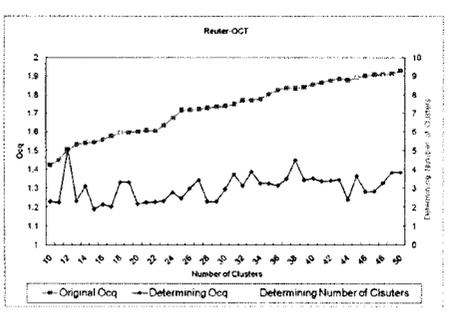
(b) March



(c) April



(d) June



(e) October

(그림 4) Reuter-21578 문서 집합 군집화 결과

는 것을 확인할 수 있다. (그림 4)의 (b), (c), (d)는 10% 정도의 성능 향상을 보였고, (a), (e)는 20% 정도의 성능 향상을 보였다.

제안하는 방법에 의한 군집 수 결정 결과를 살펴보면, (a) February는 처음 선택한 군집 수가 30개 이전에는 7개, 50개 이전에는 11개 정도로 결정되는 것을 확인할 수 있다. 군집 수를 10으로 선택했을 때의 Ocq가 1.33인데, 제안하는 방법에 의해 군집 수가 결정되면, 1.2 대의 Ocq를 보임으로써 좋은 성능을 보인다. (b) March를 살펴보면, 처음 선택한 군집 수가 30개 이전에는 7개, 50개 이전에는 12개 정도로 결정되는 것을 확인할 수 있고, Ocq는 10개를 선택했을 때의 1.3보다 항상 낮은 값을 보이고 있다. (c) April을 처음 선택한 군집 수가 25개 이전에는 결정되는 군집수가 불안정한 경향을 보이지만, 그 이후에는 9개 정도로 결정되고, Ocq는 군집 수 10개인 경우의 1.3 보다 작은 경향을 보이고 있다. (d) June을 살펴보면, 군집 수는 계단형으로 6에서 14까지 증가하고, Ocq는 10개인 경우의 1.3보다 낮은 경향을 보이고 있다. (e) October는 군집수가 4-9 사이에 존재하며, Ocq는 10개인 경우의 1.4보다 2번을 제외하고는 항상 낮은 값을 보이고 있다.

군집의 수를 증가시킬수록 Ocq 값은 증가하는 경향을 보이지만, 제안하는 방법에 의해 군집 수를 결정하면, Ocq 값은 안정적인 경향을 보이고 있다. 또한, 제안하는 방법에 의해 결정되는 군집 수는 6~12 사이의 안정적인 경향을 보이는 것을 확인할 수 있다.

4.3 군집화 결과 및 시간

<표 10>은 다섯 개의 데이터 집합에 대해 임의의 수를 선택한 후 성능 평가 지표인 *Cmp, Sep, Ocq* 의 변화량과 수행 시간을 기록한 실험 결과이다. 여기서, Optimal은 최적의 군집 개수로 제시되는 것으로 <표 9> 의 Num. Of Cluster를 따른다. Random은 임의로 선택한 수이고, Shrink는 Random에서 선택된 군집 개수가 본 논문에서 제안하는 알고리즘에 의해 결정된 군집 개수를 의미한다.

*Cmp*를 살펴 보면, 제안하는 방법으로 결정한 경우가 최적의 군집 개수나 임의의 수를 선택한 경우보다 February, March, April, June, October에 대해서는 더 좋은 결과를 보였다. *Sep*를 살펴 보면, *Cmp*와 마찬가지로, 제안하는 방법이 더 좋은 결과를 보였다. *Cmp*와 *Sep*가 비슷한 경향을 보이고 있기 때문에, *Ocq* 도 February, March, April, June, October의 경우에 더 좋은 결과를 보이고 있다.

수행 시간은 군집화 알고리즘인 K-Means의 학습 시간을 포함하여, Optimal과 Random은 3.4절의 다차원 척도법의 수행 시간을 측정하였고, Shrink는 3.5절에 설명되어 있는 제안하는 알고리즘의 수행 시간을 측정한 것이다. Optimal과 Random은 다차원 척도법이 1회 수행되고, Shrink는 다차원 척도법이 5회에서 20회 정도 더 수행되기 때문에 수행 시간이 늘어날 수 밖에 없고, 그 결과는 <표 10>의 Time에서 확인 할 수 있다. Random과 Shrink를 비

〈표 10〉 아홉 개 데이터 집합에 대한 실험 결과 (a=0.8, β=0.5)

	Num. of Clusters	*Cmp*	*Sep*	*Ocq*	Time (msec)
Reuter - February	Optimal (10)	1.73	0.94	1.33	88
	Random (5)	2.07	0.95	1.51	46
	Shrink (4)	*1.43*	*0.85*	*1.14*	46
Reuter - March	Optimal (10)	1.73	0.91	1.32	198,197
	Random (8)	2.10	0.97	1.54	291,733
	Shrink (5)	*1.42*	*0.84*	*1.13*	278,718
Reuter - April	Optimal (10)	1.73	0.91	1.32	38,166
	Random (7)	2.07	0.93	1.52	41,468
	Shrink (3)	*1.43*	*0.86*	*1.15*	43,796
Reuter - June	Optimal (10)	1.73	0.89	1.31	6,656
	Random (6)	2.12	0.96	1.54	12,390
	Shrink (3)	*1.28*	*0.87*	*1.08*	12,952
Reuter - October	Optimal (10)	1.92	0.93	1.42	2,197
	Random (6)	2.48	0.97	1.72	1,280
	Shrink (3)	*1.79*	*0.89*	*1.34*	1,358

교해 보면, Shrink의 소요 시간이 2% 정도 증가하였다. 이 경우 *Ocq*를 살펴보면, 25%의 감소, 즉 성능이 25% 증가했다는 것을 알 수 있다.

결국, 제안하는 군집 개수 결정 알고리즘은 문서 데이터의 경우 평가 지표상 성능이 증가하면서도 시간이 오래 걸리지 않기 때문에 온라인 문서 군집 시스템에 적합하다고 할 수 있다.

4.4 BIC를 이용한 방법과의 비교

BIC를 이용하여 군집의 수를 자동으로 결정하는 방법[22]과 비교한 결과는 <표 11>과 같다. BIC에 의한 방법은 초기 군집의 수를 2로 설정하고, 군집의 수를 하나씩 증가시키면서 BIC를 계산해서, BIC의 크기가 가장 큰 경우의 군집의 수를 선택하는 것이다. BIC에 의한 방법과 제안하는 방법에서 선택한 군집의 수는 거의 동일한 것을 알 수 있다. 하지만, *Ocq*와 수행 시간을 살펴보면, 각각 5%와 22% 정도 제안하는 방법이 BIC에 의한 방법보다 우수한 것을 확인할 수 있다.

〈표 11〉 제안하는 방법과 BIC 방법의 비교

		Num. of Clusters	*Ocq*	Time (msec)
Reuter - February	BIC	3	1.16	93
	제안하는 방법	4	1.14	46
Reuter - March	BIC	3	1.14	340,015
	제안하는 방법	5	1.13	278,718
Reuter - April	BIC	3	1.18	39,468
	제안하는 방법	4	1.15	43,796
Reuter - June	BIC	3	1.15	10,671
	제안하는 방법	3	1.08	12,952
Reuter - October	BIC	3	1.37	1,406
	제안하는 방법	3	1.34	1,358

5. 결 론

본 연구에서는 군집의 개수를 자동으로 최적화 하는 방법을 제안하였다. 군집화는 대량의 데이터에 대한 분석을 통해 사람이 알지 못했던 데이터에 대한 특성을 발견 할 수 있게 해준다. 하지만, 알지 못했던 데이터에 대한 분석이라는 한계로, 데이터를 몇 개의 군집으로 나누어야 정확한 결과를 얻을 수 있다는 것을 알 수 없다. 따라서, 임의의 개수로 군집의 개수를 결정할 수 밖에 없다. 군집화의 특성상 분류(Classification)할 수 없지만, 군집화 성능 척도로 더 좋은 성능을 보일 수 있도록 군집화를 수행해야 한다. 따라서, 본 연구에서는 군집의 개수를 자동으로 최적화 하는 방법을 제안하였다. 본 연구에서 제안하는 방법들을 실 데이터에 적용하여 본 결과 군집화 결과에 서 동등하거나 더 좋은 결과를 보이는 것을 확인할 수 있었다. 실행시간 측면에서 군집 수 최적화 시간이 추가되었지만 실제 시스템에 적용할 수 있을 정도의 증가를 보였다.

본 연구에서 제시하는 군집 수 최적화 방법은 군집의 합병에 의해 군집의 개수를 결정하는 방법이다. 그렇기 때문에 군집화 수행 후 재군집화 과정없이 수행할 수 있기 때문에 시간상의 이점을 얻을 수 있었다.

군집화 결과 향상을 위하여 큰 군집을 분리하는 방법을 추가해 볼 수 있다. 하지만 이러한 방법이 추가되면 재군집화 과정이 불가피 하며, 이러한 과정이 실행시간을 최소화 시켜야 하는 실용화 시스템에서 적용가능 할지는 앞으로 향후 연구해볼 과제이다.

참 고 문 헌

[1] 장익진, "다차원 척도 분석법", 연암사, 1998.

[2] 지태창, 이현진, 이일병, "차원축소를 통한 온라인 문서분류 시스템", 한국데이터마이닝학회 2005 추계학술대회, pp. 197-206, 2005.

[3] M. J. A. Berry and G. S. Linoff, "Data Mining Techniques for Marketing, Sales, and Customer Support", John Wiley & Sons, 1997.

[4] I. Borg, P. J. F. Groenen and S. P. Borgatti,, "Modern Multidimensional Scaling", Springer Verlag, 2005.

[5] B. Boutsinas, D. K. Tasoulis and M. N. Vrahatis, "Estimating the number of clusters using a windowing technique", Journal of Pattern Recognition an Image Analysis, Vol. 16, No. 2, April, pp. 143-154, 2006.

[6] D.R. Cutting, D.R. Karger, J. O. Pedersen and J. W. Tukey, "Scatter/Gather: a cluster-based approach to browsing large document collections", In Proc. of the 15th annual international ACM SIGIR, June, pp. 318-329, 1992.

[7] R. O. Duda, P. E. Hart and Da. G. Stork, "Pattern Classification (2nd Edition) ", Wiley-Interscience, Oct., 2000.

[8] E. Gose, R. Johnsonbugh and S. Jost, "Pattern Recognition and Image Analysis", Prentice Hall, 1996.

[9] J. He, A.H. Tan, C.L. Tan, and S.Y. Sung, "On quantitative evaluation of clustering systems", In Weili We, Hui Xiong, and Shashi Shekhar, editors, Information Retrieval and Clustering. Kluwer Academic Publishers, 2003.

[10] J. He, M. Lan, C.L. Tan, S.Y. Sung and H.B. Low, "Initialization of clusters refinement algorithms: a review and comparative study," International Joint Conference on Neural Networks 2004, pp. 25-29, 2004.

[11] A. K. Jain and R. C. Dubes, "Algorithms for Clustering Data", Prentice Hall, 1988.

[12] L. Kaufman and P. J. Rousseuw, "Finding Groups in Data an Introduction to Cluster Analysis", Wiley Series in Probability and Mathematical Statistics, 1990.

[13] D. D. Lewis, "Reuters-21578 text categorization test collection distribution 1.0", http://www.research.att.com/~lewis, 1999.

[14] C.G. Li, J. Guo, G. Chen, X.F. Nie and Z. Yang, "A Version of ISOMAP with Explicit Mapping", In Proc. of Fifth International Conference on Machine Learning and Cybernetics, Dalian, 13-16 Aug., pp.3201-3206, 2006.

[15] A. Liu and Y. Gong, "Document clustering with cluster refinement and model selection capabilities", In Proc. of ACM SIGIR 2002, Tampere, Finland, Aug, pp. 191-198, 2002.

[16] W. Lu and I. Traore, "Determining the optimal number of clusters using a new evolutionary algorithm", In Proc. Of the 17th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 05), Nov., 2 pp., 2005.

[17] H. Motulsky, "Intuitive Biostatistics", Oxford University Press, 1995.

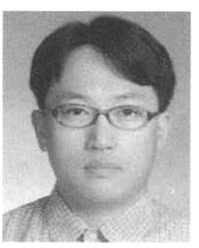
[18] D. Pelleg and A. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters", In Proc. of the Seventeenth International Conference on Machine Learning (ICML2000), June, pp. 727-734, 2000.

[19] E. Rasmussen, "Clustering algorithms", In W.B. Frakes and R. Baeza-Yates, eds. Information Retrieval . Prentice Hall, 1992.

[20] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms", In Proc. of the 16th IEEE International Conference on Tools with Artificial Intelligence, Nov., pp. 576-584, 2004.

[21] J. B. Tenenbaum, V. de Silva and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction", SCIENCE, Vol. 290, Dec., pp. 2319-2323, 2000.

[22] H. Yu, "Automatically Determining Number of Clusters", Information Retrieval (CMU CS11-741) Final Report, Apr., 5 pp., 1998.



지 태 창

e-mail: taecjee@dreamwiz.com

1997년 연세대학교 컴퓨터과학과 (학사)

1999년 연세대학교 컴퓨터과학과(공학석사)

1999년~현재 LG CNS 부책임연구원

2004년~현재 연세대학교 컴퓨터과학과 박사과정

관심분야 : 인공지능, 데이터마이닝, 패턴인식

이 현 진

e-mail : hjlee@mail.kcu.ac

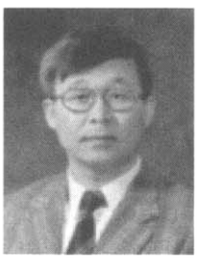
1996년 순천향대학교 전산학과(학사)

1998년 연세대학교 컴퓨터과학과 (공학석사)

2002년 연세대학교 컴퓨터과학과 (공학박사)

2003년~현재 한국싸이버대학교 컴퓨터정보통신학부 조교수

관심분야 : 신경회로망, 데이터마이닝, 이러닝



이 일 병

e-mail : yblee@csai.yonsei.ac.kr

1976년 연세대학교 전자공학과(학사)

1980년 University of Illinois 전산과학과 (공학석사)

1985년 University of Massachusetts 전산정보과학과(공학박사)

1985년~현재 연세대학교 컴퓨터과학과 교수

관심분야 : 신경회로망, 문서인식, Computer Vision, Data Mining, 필기체 문자 인식, Biometrics