군집의 효율향상을 위한 휴리스틱 알고리즘

이 석 환* · 박 승 헌*
*인하대학교 산업공학과

Heuristic algorithm to raise efficiency in clustering

Seog-Hwan Lee* · Seung-Hun Park*
*Department of Industrial Engineering, Inha University

Abstract

In this study, we developed a heuristic algorithm to get better efficiency of clustering than conventional algorithms. Conventional clustering algorithm had lower efficiency of clustering as there were no solid method for selecting initial center of cluster and as they had difficulty in search solution for clustering. EMC(Expanded Moving Center) heuristic algorithm was suggested to clear the problem of low efficiency in clustering. We developed algorithm to select initial center of cluster and search solution systematically in clustering.

Experiments of clustering are performed to evaluate performance of EMC heuristic algorithm. Squared-error of EMC heuristic algorithm showed better performance for real case study and improved greatly with increase of cluster number than the other ones.

Keywords : Clustering, EMC Heuristic

1. 서 론

데이터마이닝(data mining)에서 군집(clustering)은 주어진 객체들을 서로 유사성이 있도록 몇 개의 군으로 나누는 과정이다. 군집들은 여러 분야에서 활용되고 있다. 예를 들면, 마케팅에 있어서 군집은 서로 다른 고객집단을 구분하는 고객 세분화 또는 각 군집의 특징을 바탕으로 고객의 구매 패턴을 예측하여 그룹별 혹은 개인별로 차별화된 마케팅을 지원하는 고객관계관리(Customer Relationship Management, CRM) 등에서 사용되고 있다. 최근에는 멀티미디어와 같은 데이터들을 압축하거나 패턴 인식 · 분류 등을 위한 방안으로도 연구되고 있어서 군집의 중요성은 크다고 할 수 있다[1][2][4][5].

k-menas 알고리즘은 거리 기반 군집방법 중에서 가장 먼저 개발된 방법으로 우수한 성능이 입증되어 타 알고리즘의 기반 알고리즘으로 사용되고 있다[1][2].

k-means 알고리즘은 초기에 군집의 중심을 랜덤으로 배치하여 그 중심을 이동시키면서 최적 군집의 중심을 탐색한다. 군집의 중심 이동은 군집 간에 객체가 이동해야만 발생하며 이동하는 객체의 수가 증가할수록 그 이동 폭이 커진다. 군집 간에 이동하는 객체는 거의 대부분 군집 간 경계선 주위에 위치해 있다. 따라서 이 경계선 길이의 합이 길면 길수록 군집 간에 이동하는 객체의 수가 증가하여 군집의 중심 이동 폭이 커지게 되므로 해의 탐색영역이 넓어진다. 예를 들어 객체들이 2차원 영역에 분포되어 있는 경우, 군집 간 경계선 길이의 합은 군집의 중심을 좌표의 원점으로부터 대각선 방향의 일직선상에 배열할 경우 더욱 길어진다(<그림 1>의 b참조).

그러나 군집의 중심을 랜덤으로 배치하면 그 중심들은 일직선으로 배열되기 어렵다. 따라서 군집 간 경계선 길이의 합이 짧아져서 군집의 중심 이동 폭이 작아지기 때문에 해의 탐색영역이 좁아진다.

† 이 논문은 인하대학교 교내 연구비 지원에 의해 연구되었음.
† 교신저자: 이석환, 인하대학교 산업공학과 생산관리 연구실
Tel: 02-2610-0419, E-mail: seoghwan@inha.ac.kr
2009년 7월 13일 접수; 2009년 8월 31일 수정본 접수; 2009년 8월 31일 게재확정

이와 같은 이유로 초기에 군집의 중심이 최적 군집의 중심과 멀리 떨어져 배치될 경우, 군집의 중심 이동이 작기 때문에 최적 군집의 중심을 찾는 확률이 감소하게 된다. 따라서 최적 군집의 중심을 찾는 확률을 증가시키려면 군집의 중심 이동 폭이 커지도록 군집의 중심을 대각선 방향의 일직선상에 배치하여 해의 탐색영역을 넓혀야 한다. 최근에는 군집의 효율을 향상시키기 위해서 휴리스틱을 적용한 연구가 진행되고 있다[5][9][10]. 그 중에서 타부 탐색 알고리즘은 초기해를 생성하고 정해진 반복횟수 내에서 이웃해-초기해를 사용하여 생성한 해-를 찾아내어 목적함수인 제곱오차의 값을 비교하는 방법이다. 그러나 타부 탐색 알고리즘은 초기해의 우수한 정도, 이웃해의 탐색방법, 탐색한 해를 기록하는 타부리스트의 크기에 따라서 해를 탐색하는 영역이 변하기 때문에 군집의 해도 변하게 된다. 특히 이웃해의 탐색은 랜덤으로 수행하기 때문에 개선된 해(초기해로부터)를 발견할 수도 있지만 해를 체계적으로 탐색할 수 없어서 많은 경우에 개선된 해를 발견하지 못할 확률이 크다. 따라서 이웃해를 랜덤으로 탐색하는 방법이 아닌 최적해에 근사한 해를 체계적으로 탐색할 수 있는 새로운 방법이 필요하다.

본 연구에서는 앞에서 설명한 두 가지 문제점을 해결하여 군집의 효율을 향상시킬 수 있는 휴리스틱 알고리즘을 다음과 같이 제안한다. 첫째, 해의 탐색영역을 넓히기 위해서 초기 군집의 중심 배치방법을 개선한다. 둘째, 이웃해의 탐색이 아닌 최적해에 근사한 해를 체계적으로 탐색할 수 있는 방법을 제시한다. 그 다음으로 군집의 계산효율을 향상시키기 위해 군집에 소요되는 계산량을 줄일 수 있는 방법을 제시한다. 마지막으로 본 연구에서 제안한 휴리스틱 알고리즘을 실제 데이터에 적용하고 그 결과를 k-means와 타부 탐색 알고리즘의 군집결과와 비교하여 그 효율성을 확인한다.

2. 중심이동확장(Expanded Moving Center) 휴리스틱 알고리즘

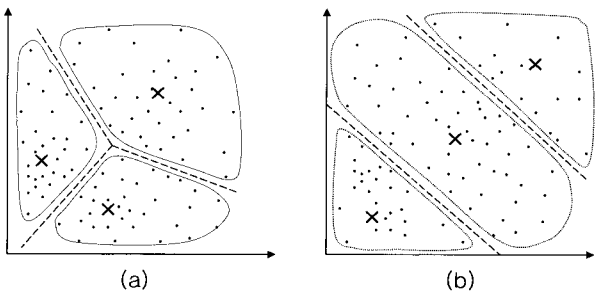
여기에서는 본 연구의 EMC(중심이동확장) 휴리스틱 알고리즘에 대해서 구체적으로 설명한다. 서론에서 언급한 것과 같이 해의 탐색영역을 넓히기 위해서는 초기 군집 중심의 배치방법을 개선하고 최적해에 근사한 해를 체계적으로 탐색할 수 있는 방법을 제시하며 군집에 소요되는 계산량을 줄여야 한다. 이 문제들의 해결을 위해서 2.1에서는 해의 탐색영역을 넓히기 위한 방법으로 초기 군집의 중심 등간격 배치에 대해 설명한다. 2.2에서는 최적해에 근사한 해를 체계적으로 탐색할 수 있는 방법으로 군집의 중심 간 간격 설정에 대해서 설명하고 2.3에서는 군집에 소요되는 계산량을 줄일 수 있는 방법으로 군집의 계산효율 향상에 대해서 설명한다. 마지막으로 2.4에서는 EMC 휴리스틱 알고리즘이 실제로 해를 탐색하는 과정을 설명한다.

2.1 초기 군집의 중심 등간격 배치

군집에서 해의 탐색영역을 넓히기 위해서는 군집의 중심 이동 폭을 크게 해야 한다. 군집의 중심 이동 폭은 군집 간에 이동하는 객체의 수에 따라서 달라진다.

일반적으로 군집 간에 이동하는 객체의 수가 많아지면 군집의 중심 이동 폭이 커진다. 따라서 군집의 중심을 배치할 때 군집 간에 이동하는 객체의 수가 증가되도록 배치하면 군집의 중심 이동 폭이 커져서 해의 탐색영역이 넓어지기 때문에 개선된 해를 탐색하는 확률이 증가한다. 군집 간에 이동하는 객체는 주로 군집 간 경계선 주위에 위치한다. 이 경계선이 길어지면 그 만큼 군집 간에 이동하는 객체의 수도 증가한다. <그림 1>은 군집 간 경계선을 표시한 것으로 점들은 객체이고 ×는 군집의 중심이다. 객체들은 자신과 가장 가까운 군집의 중심에 속하게 된다. 따라서 군집 간 경계선은 두 군집의 중심 사이에서 직선으로 생성된다.

여기서 모든 군집의 중심이 <그림 1>의 b와 같이 원점으로부터 대각선 방향의 일직선상에 배치될 경우 군집 간 경계선의 시작과 끝 점은 다른 경계선과 마주치지 않기 때문에 객체들이 분포해있는 영역의 끝까지 확장될 수 있다. 따라서 모든 경계선을 합한 길이는 <그림 1>의 a보다 b의 경우가 길어진다. 이 때 경계선 간의 간격이 동일한 크기로 설정되지 않으면 특정 군집의 중심이 타 군집의 중심에 비해서 이동을 제한받는 경우가 발생한다. 이것은 개선된 해를 찾는 확률을 감소시킬 수 있다. 따라서 군집의 중심을 원점으로부터 대각선 방향의 일직선상에 배치하되 등간격으로 배치하면 모든 군집의 중심이 비슷한 폭으로 이동할 수 있게 되어 개선된 해를 찾는 확률이 증가할 수 있다.



<그림 1> 군집 간 경계

경계선 길이의 합은 군집의 중심을 랜덤으로 배치한 경우보다 등간격으로 배치한 경우가 길어지는데 그 이유는 식 (1)~(5)로 설명할 수 있다. 식 (1)은 군집의 중심을 랜덤으로 배치한 경우로 각 군집 간의 경계가 수직과 수평인 특수한 경우로 한정하였다. 식 (2)와 (3)은 군집의 중심을 등간격으로 배치한 경우로 식 (2)는 군집 수 k가 홀수인 경우이고 식 (3)은 짝수인 경우이다.

식 (1), (2), (3)은 객체들이 분포한 영역이 정사각형이라는 가정을 전제로 한다. 식 (4)는 군집 수 k가 홀수인 경우 등간격 배치와 랜덤 배치의 차이를 계산한 식이다. 식 (4)는 k가 3일 경우 $0.39x$가 되므로 경계선 길이의 합은 등간격 배치가 랜덤 배치보다 최소 0.39 이상 커진다. 식 (5)는 군집 수 k가 짝수인 경우 등간격 배치와 랜덤 배치의 차이를 계산한 식이다. 식 (5)는 k가 2일 경우 $0.41x$가 되므로 경계선 길이의 합은 등간격 배치가 랜덤 배치보다 최소 0.41 이상 커진다.

특히 식 (4)와 (5)에서 차이는 군집 수 k의 증가에 비례하여 커진다. 즉, 군집의 중심을 대각선 방향의 일직선상에 등간격으로 배치하면 항상 랜덤 배치보다 경계선 길이의 합이 길다. 따라서 군집의 중심 이동 폭이 커지고 이로 인하여 해의 탐색영역이 넓어져 개선된 해를 탐색하는 확률이 증가한다.

$$d_s = \frac{k}{2}x \cdots\cdots\cdots\cdots\cdots\cdots (1)$$

$$k\text{가 홀수}: d_s = \frac{\sum_{i=1}^{(k-1)/2} 4i}{k}\sqrt{2}x \cdots\cdots\cdots (2)$$

$$k\text{가 짝수}: d_s = \frac{\sqrt{2}}{2}kx \cdots\cdots\cdots\cdots (3)$$

$$k\text{가 홀수}: \frac{\sum_{i=1}^{(k-1)/2} 4i}{k}\sqrt{2}x - \frac{k}{2}x \qquad \cdot (4)$$

$$= \frac{x(8\sqrt{2}\sum_{i=1}^{(k-1)/2} i - k^2)}{2k} \geq 0.39x$$

여기서 $k \geq 3$

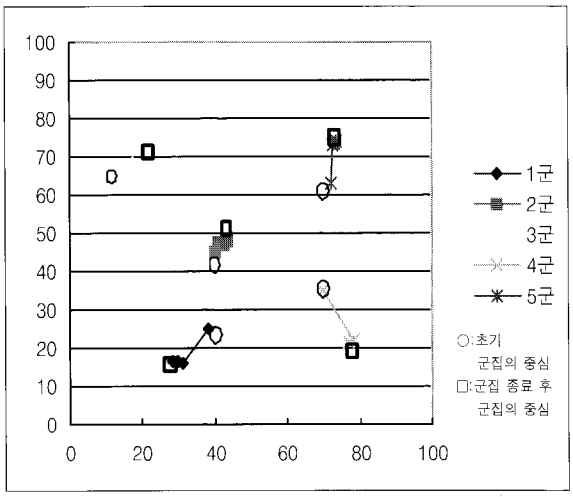
$$k\text{가 짝수}: \frac{k}{2}\sqrt{2}x - \frac{k}{2}x \qquad \cdots\cdots (5)$$

$$= \frac{(\sqrt{2}-1)}{2}kx \geq 0.41x$$

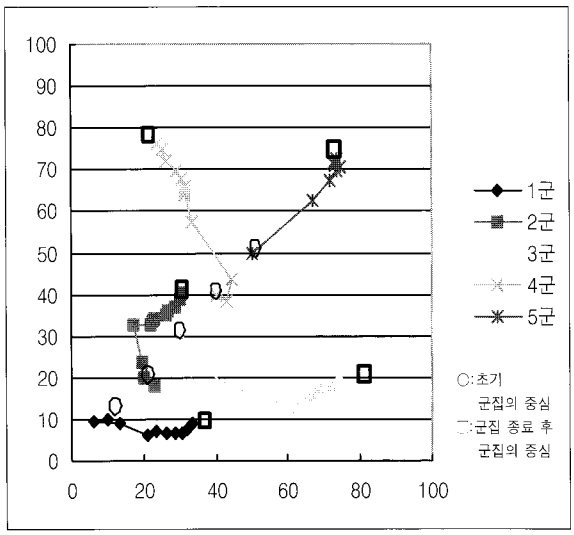
여기서 $k \geq 2$

k: 군집의 수
x: 한 변의 길이

<그림 2>는 군집의 중심을 랜덤으로 배치하여 5개의 군집으로 분류한 결과이고 <그림 3>은 군집의 중심을 등간격으로 배치하여 5개의 군집으로 분류한 결과이다. <그림 2>와 <그림 3>에서 원은 초기에 군집의 중심이고 사각은 군집이 종료된 후에 이동한 군집의 중심을 나타낸다. <그림 2>는 초기에 군집의 중심을 랜덤으로 배치하여 군집을 수행한 경우로 군집의 중심 이동 폭이 매우 작다. 반면 <그림 3>과 같이 초기에 군집의 중심을 등간격으로 배치하여 군집을 수행한 경우는 군집 간 경계선의 길이가 길어져서 군집의 중심 이동 폭이 크다. 그 결과 <그림 3>의 등간격 배치는 <그림 2>의 랜덤 배치보다 더 많은 해를 탐색할 수 있어서 개선된 해를 찾는 확률이 증가한다.



<그림 2> 초기 군집의 중심 랜덤 배치



<그림 3> 초기 군집의 중심 등간격 배치 (간격 10)

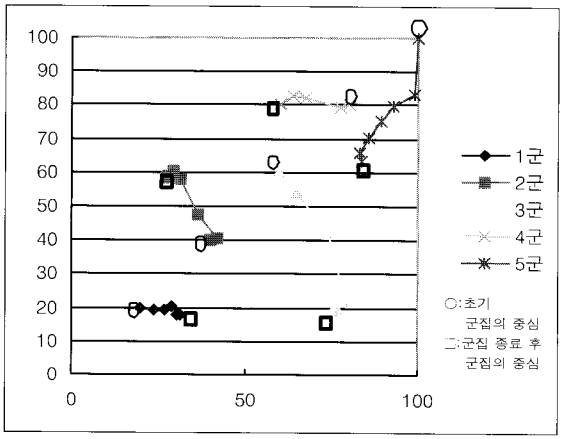
2.2 군집의 중심 간 간격 설정

초기에 군집의 중심을 대각선 방향의 일직선상에 등간격으로 배치하면 개선된 해를 찾는 확률이 증가한다.

여기서 최적해에 근사한 해를 찾기 위해서는 해를 체계적으로 탐색하는 방법이 필요하다. 그 방법은 군집의 중심 간 간격 크기를 일정한 범위 내에서 변경해나가면서 탐색영역을 다르게 하여 해를 탐색하는 것이다.

군집의 중심을 등간격으로 배치하면 <그림 1>의 b와 같이 해의 탐색영역이 몇 개의 구간으로 나누어진다. 이 때 간격의 크기를 변경하면 구간의 영역이 변하게 되어 해의 탐색영역도 변한다. 동시에 군집 간 경계선이 변하여 군집 간 이동객체가 변하게 되고 이에 따라서 다양한 해를 찾게 되어 최적해에 근사한 해를 찾을 수 있는 확률이 증가한다. 예를 들어 <그림 4>는 초기에 군집의 중심 간 간격이 20이고 <그림 3>은 10이다. 동일한 객체를 사용하여 군집한 결과이지만 초기에 군집의 중심 간 간격이 달라졌기 때문에 서로 다른 해를 탐색하게 되었다. 본 연구에서는 최적해에 근사한 해를 찾을 수 있는 확률이 증가되도록 초기 군집의 중심 간 간격의 크기를 일정 범위 내에서 균일하게 변경해나가면서 탐색영역을 다르게 하여 해를 탐색한다.

초기 군집의 중심 간 간격은 다음과 같은 식으로 계산한다. 식 (6)은 초기 군집의 중심 간 간격을 계산하는 식이고 식 (9)는 초기 군집의 중심 간 간격에 대한 범위를 제한하는 식으로 증가단위는 정수이다. 식 (6)의 D_s는 초기 군집의 중심 간 간격이다. D_s의 계산은 각 차원의 최대값과 최소값의 차이를 l값으로 나누어 구한다. 즉, l값이 감소하면 D_s는 넓어지고 반대로 l값이 증가하면 D_s는 좁아진다. 따라서 이 l값의 범위를 설정하면 D_s의 범위를 정해진 범위 내에서 균일하게 변경할 수 있다.



<그림 4> 초기 군집의 중심 등간격 배치 (간격 20)

l값은 군집 수 k보다 작아질 수 없는데 그 이유는 군집 수 k보다 작은 l값을 사용하게 되면 군집의 중심은 객체들이 분포해있는 범위 밖으로 벗어나기 때문이다.

따라서 l의 최소값은 군집 수와 동일한 k가 되며 식 (7)에 식 (8)을 대입하여 구할 수 있다. 식 (8)의 우변은 초기 군집의 중심 중에서 h차원의 원점으로부터 가장 멀리 떨어진 군집의 중심을 의미한다. l의 최대값은 $2k$로 정했다. 그 이유는 l값이 $2k$이상이 될 경우 D_s의 값이 크게 감소하여 오히려 군집 간 경계선의 합이 짧아지기 때문이다. l값의 범위는 식 (9)와 같다.

$$D_s = \frac{\max(O_{s_i}) - \min(O_{s_j})}{l} \quad \cdots\cdots (6)$$

$$l = \frac{\max(O_{s_i}) - \min(O_{s_j})}{D_s} \quad \cdots\cdots (7)$$

$$\max(O_{s_i}) = \min(O_{s_j}) + kD_s \quad \cdots\cdots (8)$$

$$k \le l \le 2k \quad \cdots\cdots (9)$$

s : 차원 $(s = 1, 2, .., h)$
D: 간격
O: 객체
l: 등분 개수
k: 군집 수

초기 군집의 중심 간 간격은 <그림 4>의 예로써 설명할 수 있다. <그림 4>에서 객체의 분포는 생략되어 있지만 각 차원에서 객체가 갖는 최소값은 0이고 최대값은 100이다. 군집의 수 k는 5개 이므로 l의 최소값은 5가 된다. 초기 군집의 중심 간 간격은 객체가 갖는 최대값과 최소값의 차이를 l로 나누어야 하므로 100/5=20이 된다. 이 값은 해당 차원의 최소값에 더해져서 첫 번째 군집의 중심이 된다. 두 번째 군집의 중심은 앞에서 구해진 초기 군집의 중심 간 간격을 이용하여 등간격으로 배치된다. 이 과정은 모든 군집의 중심이 배치될 때 까지 반복되고 배치가 완료되면 군집을 수행한다. 군집이 완료되면 l은 6이 되어 군집의 중심 간 간격은 16.7이 된다. 이 간격을 이용하여 모든 군집의 중심을 재배치하고 다시 군집을 수행한다. 이와 같이 간격의 크기를 일정한 범위 내에서 균일하게 변경하면 동일한 객체들에 대해서 찾아낸 군집의 해는 항상 같게 된다. 따라서 군집의 중심 간 간격의 크기를 일정한 범위 내에서 균일하게 변경해나가면서 탐색영역을 다르게 하여 해를 탐색하면 다양한 해를 찾을 수 있기

때문에 최적해에 근사한 해를 발견할 수 있다. 또한 동일한 객체들에 대해서 찾아낸 군집의 해는 타부 탐색과는 다르게 항상 동일한 해를 가지게 된다.

2.3 군집의 계산효율 향상 및 EMC 휴리스틱 알고리즘

군집의 효율을 향상시키기 위해서는 군집에 소요되는 계산량도 줄여야 한다. 군집에서 계산은 주로 군집 간에 객체들의 이동여부를 판단하기 위해서 행해진다.

군집 간에 이동하는 객체는 거의 대부분 군집 간 경계선 주위에 위치해 있다. 따라서 경계선 주위에 위치한 객체에 대해서만 이동여부를 판단하게 되면 계산량을 크게 줄일 수 있다. EMC 휴리스틱 알고리즘은 <그림 5>와 같다.

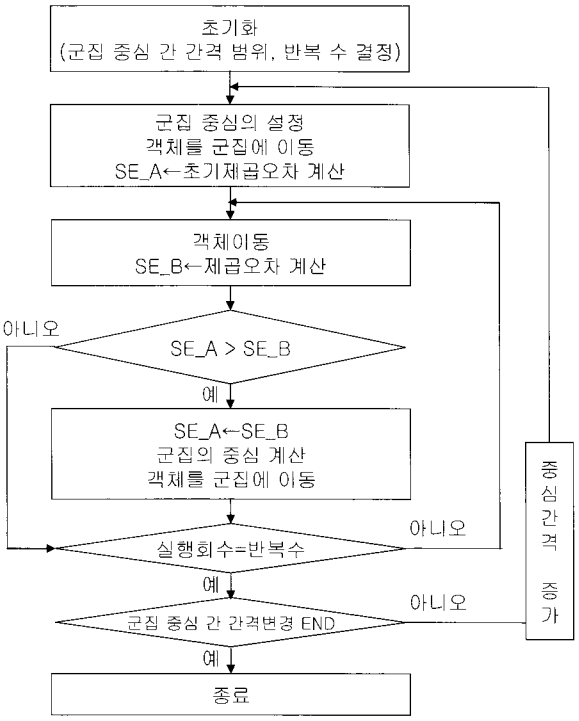
<그림 5>에서 객체를 군집에 이동시키는 과정은 초기에 설정한 반복수 만큼 실행된다. 이 과정은 거리계산과 거리비교의 두 가지 부분으로 구성되어있다. 먼저 거리계산을 수행한다. 거리계산은 각 군집의 중심과 모든 객체 간의 거리를 계산하는 것이다. 그 다음으로 거리비교를 수행한다. 거리비교는 거리계산에서 구해진 객체들의 거리를 비교하여 이 중에서 가장 짧은 거리의 객체를 해당 군집으로 이동시키는 것이다. 이 때 각 군집의 경계선 주위에 있는 객체들은 타 군집으로 이동할 가능성이 높다. 따라서 각 군집의 경계선 주위에 있는 객체에 대해서만 거리비교를 하면 거리비교 횟수를 줄일 수 있다. 특히 반복수가 많을 경우는 거리비교 횟수를 크게 줄일 수 있다.

객체를 선정하는 기준은 평균제곱오차(Mean Squared -error, MSE)이다. 식 (10)은 평균제곱오차를 계산하는 식으로 제곱오차를 모든 객체의 수로 나눈 값이다. 즉, 거리비교는 평균제곱오차보다 긴 거리를 갖는 객체에 대해서만 수행한다.

$$MSE = \frac{\sum_{i=1}^{k}\sum_{p\in C_i}|p-m_i|^2}{N} \cdots\cdots (10)$$

k : 군집의 수
C_i : 군집 i
p : 군집 내 객체의 위치
m_i : 군집 C_i의 평균
N : 객체의 수

식 (11)은 모든 객체에 대해서 거리비교를 했을 때 소요되는 거리비교 횟수이다. 기존 방법에서는 모든 객체에 대해서 거리를 비교하였기 때문에 군집의 수와 객체의 수 그리고 반복 횟수가 많을 경우 거리비교 횟수가 크게 증가한다.



<그림 5> EMC 휴리스틱 알고리즘

그러나 본 연구에서는 각 군집의 중심으로부터 멀리 떨어진 일부 객체에 대해서만 거리비교를 하기 때문에 식 (11)의 N값을 줄일 수 있어서 거리비교 횟수를 크게 줄일 수 있다. 특히 반복 횟수가 많아질 경우 군집에 소요되는 계산량을 크게 줄일 수 있다.

$$\text{거리비교횟수} = kRN \cdots\cdots (11)$$

k : 군집의 수
R : 반복횟수
N : 객체의 수

<그림 6>은 지금까지 설명한 EMC 휴리스틱 알고리즘을 상세하게 나타낸 것이다.

2.4 알고리즘 실행 예

여기에서는 EMC 휴리스틱 알고리즘이 군집 간 중심간격을 변경해나가면서 해를 탐색하는 과정을 설명한다. 사용한 데이터는 난수발생에 의한 데이터로 마이크로소프트오피스 엑셀의 난수 발생기능을 이용하여 x, y 값으로 구성되는 객체를 100개 생성하였다. <그림 7>은 객체들을 x, y 좌표 상에 표시한 것으로 객체가 갖는 x, y의 최소값과 최대값은 0과 100이다.

군집의 수행은 군집 수를 5개, 객체이동의 반복수를 20으로 하였다. 군집의 중심을 배치하기 위해서는 먼저

식(9)를 사용하여 l값의 범위를 정해야 한다. 군집 수 k가 5이기 때문에 l값의 범위는 5부터 10까지이다. l값의 범위가 정해지면 식 (6)을 사용하여 1회 차 군집의 중심 간 간격을 구한다. l의 최소값 5를 식 (6)에 대입하면 1회 차 군집의 중심 간 간격은 100/5=20이 된다.

이 간격을 사용하여 <그림 8>과 같이 군집의 중심을 등간격으로 배치하고 객체들을 이동시킨다. <그림 8>에서 원 모양은 군집의 중심이다. 객체들의 이동은 20회의 반복을 거치게 되고 이 과정에서 가장 작은 제곱오차를 갖는 해가 1회 차 군집에서 가장 개선된 해로 저장된다.

다음은 l값을 1만큼 증가시키고 2회 차 군집의 중심 간 간격을 구한다. 2회 차 군집의 중심 간 간격은 l값이 6이 되므로 100/6=16.67이 된다. <그림 9>는 초기에 군집의 중심 간 간격을 16.67로 설정하여 형성한 2회 차 군집에서 가장 개선된 해이다. 3회 차 군집의 중심 간 간격도 앞의 방법과 동일하게 계산하여 군집의 중심을 배치하고 군집을 수행한다. 이와 동일한 방법으로 마지막 11회 차 군집까지 수행한다. 1회 차부터 11회 차까지 군집이 종료되면 11개의 해가 생성되는데 이 중에서 제곱오차가 가장 작은 해를 군집의 최종 해로 선택한다.

<그림 10>은 알고리즘의 모든 과정이 종료된 후에 찾아진 군집의 최종 해로 군집의 중심 간 간격이 14.29(l=7) 일 때 찾은 군집의 해이다. EMC 휴리스틱 알고리즘으로 찾은 군집의 해가 k-means, 타부 탐색 알고리즘으로 찾은 군집의 해보다 개선되었는지 확인하기 위해서 각 알고리즘의 제곱오차를 서로 비교하였다. 그 결과 <그림 11>과 같이 EMC 휴리스틱 알고리즘의 제곱오차는 1599.47, k-means 알고리즘은 1647.59, 타부 탐색 알고리즘은 1643.52 로 EMC 휴리스틱 알고리즘은 두 알고리즘에 비해서 개선된 해를 찾을 수 있었다.

3. 군집 실험결과 및 분석

이 실험에서는 EMC 휴리스틱 알고리즘이 타 알고리즘보다 우수한 군집의 해를 발견할 수 있는지 확인한다.

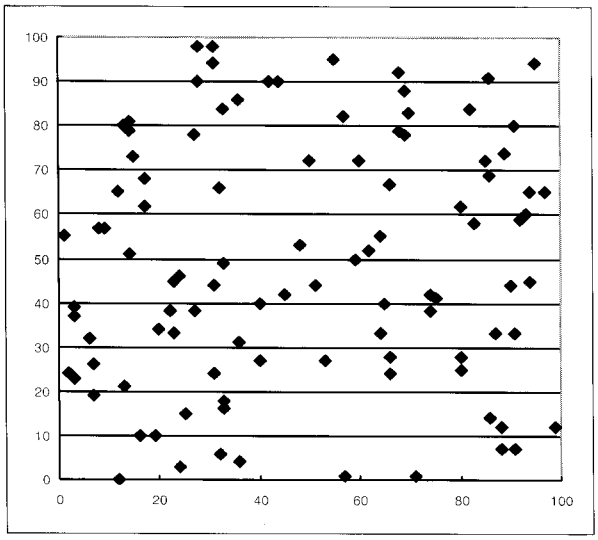
성능비교의 대상으로 선택한 알고리즘은 k-means와 타부 탐색 알고리즘이다.

```
main() //군집 메인

for ii_int = li_int_from to li_int_to //군집의 중심 간 간격 범위설정
        center() //초기 군집의 중심 설정
        cluster() //객체를 군집에 할당
        if 초기해의 제곱오차 > 개선해의 제곱오차 then
                초기해의 제곱오차 = 개선해의 제곱오차
        end if
next
```

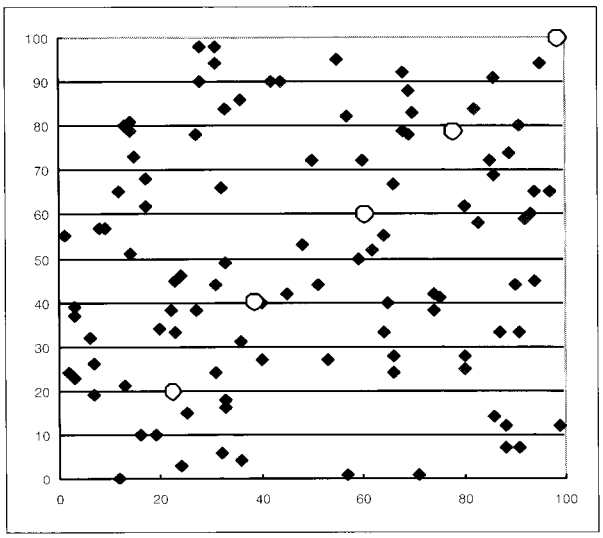
```
center() //초기 군집의 중심 설정

select min(x_val) into :ld_min_x_val from cluster_mst; //x차원의 최소값
select min(y_val) into :ld_min_y_val from cluster_mst; //y차원의 최소값

select max(x_val) - min(x_val) into :ld_x_val from cluster_mst; //x차원의 최대값과 최소값의 차이
select max(y_val) - min(y_val) into :ld_y_val from cluster_mst; //y차원의 최대값과 최소값의 차이

ld_x_val = ld_x_val / ii_int //x차원의 간격
ld_y_val = ld_y_val / ii_int //y차원의 간격

for i=1 to 군집의 개수
        dw_1.setitem(i, "x_cval", (ld_min_x_val + ld_x_val*i))
        dw_1.setitem(i, "y_cval", (ld_min_y_val + ld_y_val*i))
next
```

```
cluster() //객체를 군집에 할당

ld_old_mean_square=square_error() //객체 이동 전 제곱오차 계산
for i = 1 to li_iter //설정한 반복수 만큼 실행
        Declare sel_object Cursor For
                select ID, CLUST, x_VAL, y_VAL from cluster_mst
                where DISTANCE > (select avg(DISTANCE) from cluster_mst)
                order by distance asc; //평균거리 이상의 객체만을 거리비교 객체로 선택
        Open sel_object;
                Do
                Fetch max_dist into :ls_ID, :li_CLUST, :ld_x_VAL, :ld_y_VAL; //한 개의 객체 read
                        ld_DISTANCE = 거리비교 객체가 군집에서 이탈할 경우의 제곱오차
                        Declare sel_clust Cursor For
                                select distinct(CLUST), x_cval, y_cval from cluster_mst
                                where clust <> :li_clust group by clust, x_cval, y_cval
                                order by clust asc; //거리비교 객체를 제외한 군집의 중심
                        Open sel_clust;
                                Do
                                Fetch sel_clust into :li_comp_CLUST, :ld_x_center, :ld_y_center;
                                //군집의 중심을 read
                                        ld_new_distance = 거리비교 객체가
                                                          군집에 포함될 경우의 제곱오차
                                        if (ld_DISTANCE > ld_new_distance) then
                                                ld_DISTANCE = ld_new_distance
                                                update cluster_mst set clust =
                                                :li_comp_CLUST where ID = :ls_ID;
                                                //제곱오차가 적어지는 군집에 객체를 이동
                                        end if
                                Loop Until sqlca.sqlcode = 100;
                Loop Until sqlca.sqlcode = 100;
        ld_new_mean_square=square_error() //객체 이동 후 제곱오차 계산
        if ld_old_mean_square > ld_new_mean_square then
                refine_cluster() //1) 군집별 중심계산 2) 군집의 중심과 가까운 객체를 재 군집
        end if
next
```

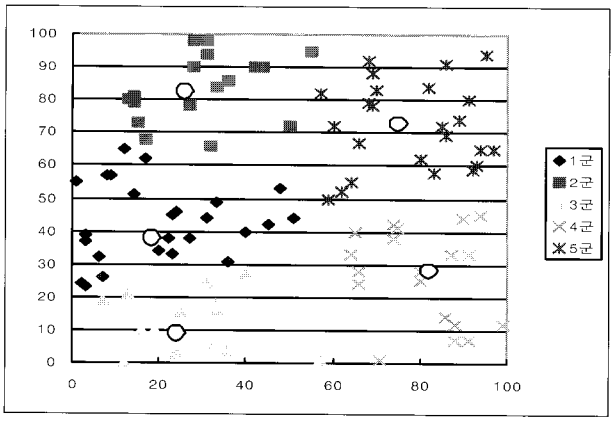
<그림 6> EMC 휴리스틱 알고리즘 상세



<그림 7> 객체들의 분포



<그림 8> 군집의 중심배치



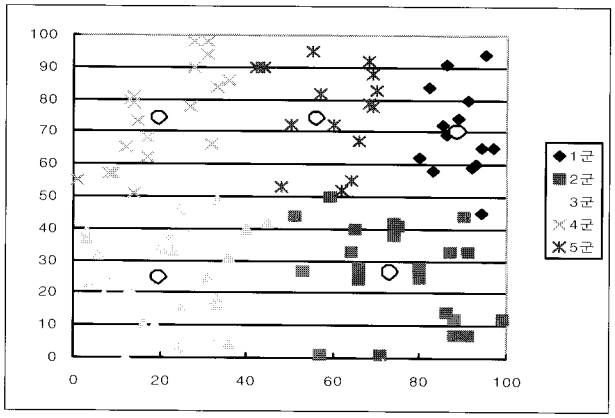
<그림 9> 초기에 군집의 중심 간 간격을 16.67로 하여 찾은 해 (제곱오차=1612.22)

성능의 비교는 각 알고리즘에 대해 군집의 수를 5부터 15까지 변화시키면서 제곱오차의 차이를 비교한다.

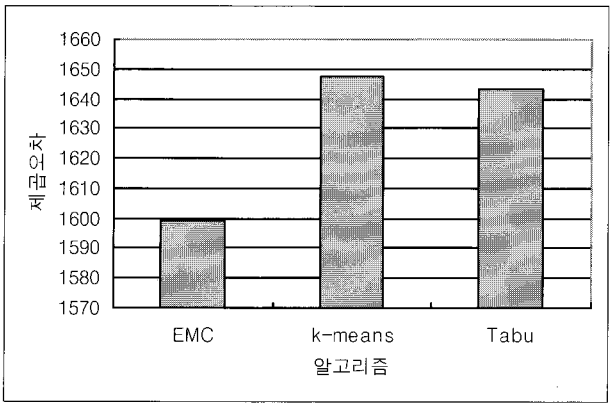
실험에 사용하는 데이터는 MovieLens[7] 데이터로 영화에 대한 고객들의 선호도를 조사한 실제 데이터이다.

영화는 총 1682편이며 선호도 정보를 제공한 고객은 총 943명이다. 선호도 등급은 1부터 5까지의 척도로 구성되어 있는데 선호도 등급이 5인 경우는 해당 영화에 대한 고객의 선호도가 높은 것이다. k-means 알고리즘은 Clementine 8.1 패키지의 K-평균[3][8]을 사용하였고 타부 탐색 알고리즘과 EMC 휴리스틱 알고리즘은 PowerBuilder 6.0으로 구현하였다. 그리고 객체들을 저장하기 위한 데이터베이스는 Oracle 7을 사용하였다.

<표 1>은 실험을 위해 준비한 실제 데이터의 일부분으로 랜덤으로 선택한 10편의 영화에 대해서 943명의 선호도를 기입한 표이다. 성능비교의 대상인 타부 탐색 알고리즘의 초기설정은 이웃해 수 5개, 타부 리스트의 크기 3개 그리고 반복수 30회로 정하였다. EMC 휴리스틱 알고리즘의 초기설정은 초기 군집의 중심 간 간격 범위를 $k \le l \le 2k$, 반복수는 30회로 정하였다.



<그림 10> 초기에 군집의 중심 간 간격을 14.29로 하여 찾은 해 (제곱오차=1599.47)



<그림 11> 알고리즘의 제곱오차 비교

<표 1> 실제 데이터의 구성

고객 ID \ 영화 ID	69	168	257	269	748	301	11	479	508	147
1	0	0	3	5	0	3	0	0	3	0
2	0	0	0	3	0	0	0	0	0	0
3	4	5	0	0	0	0	4	4	2	3
4	4	0	0	0	2	2	0	5	2	0
:	0	0	0	0	4	0	0	0	0	0
943	4	4	0	3	0	4	0	4	2	0

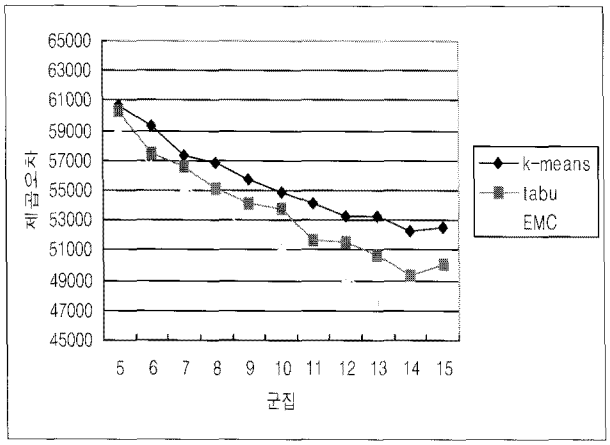
<표 2>는 실제 데이터로 군집을 수행한 결과로 음영부분은 세 가지 알고리즘 중에서 제곱오차가 가장 작은 것을 표시한 것이다. <표 2>와 같이 모든 군집 수에서 EMC 휴리스틱 알고리즘의 제곱오차는 타 알고리즘에 비해서 우수하다. 이것은 EMC 휴리스틱 알고리즘이 군집 수의 변화에 무관하게 타 알고리즘에 비해서 우수한 군집의 해를 발견할 수 있다는 것을 나타낸다.

<표 2> 실제 데이터의 군집 알고리즘 별 제곱오차

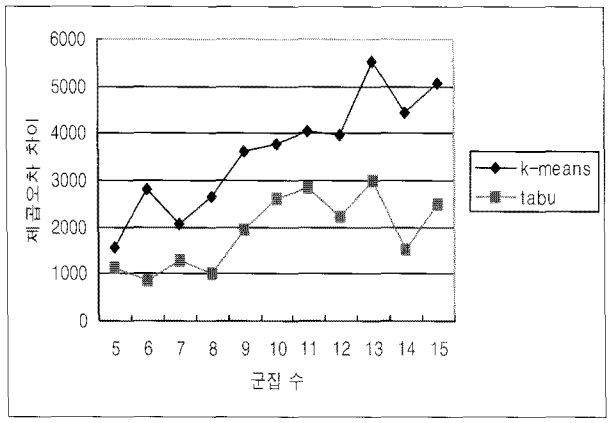
군집 수	k-means	tabu	EMC
5	60662.40	60251.81	59119.84
6	59371.09	57438.22	56578.62
7	57347.81	56575.43	55298.86
8	56814.05	55143.93	54146.97
9	55800.46	54116.13	52173.15
10	54917.95	53762.57	51147.05
11	54085.46	52848.93	50021.24
12	53271.32	51505.08	49284.51
13	53219.90	50693.43	47680.88
14	52243.98	49323.83	47795.54
15	52571.80	50010.97	47521.90

<그림 12>는 군집 수의 변화에 따른 각 알고리즘의 제곱오차 변화를 그래프로 나타낸 것이다. <그림 12>에서 군집의 수가 14, 15의 경우 EMC 휴리스틱 알고리즘의 제곱오차는 더 이상 개선이 되지 않았다. 이것은 이 실험에서 사용한 실제 데이터의 경우 최대 군집의 수가 13개일 때 군집의 효율이 가장 높다(제곱오차가 최소)는 것을 알 수 있다.

<그림 13>은 EMC 휴리스틱 알고리즘의 제곱오차와 k-means, 타부 탐색 알고리즘의 제곱오차 차이를 나타낸 것으로 EMC 휴리스틱 알고리즘의 제곱오차를 0으로 하였을 때 k-means, 타부 탐색 알고리즘의 제곱오차를 나타낸 것이다. 군집의 수가 많아지면서 EMC 휴리스틱 알고리즘의 제곱오차와 k-means 알고리즘의 제곱오차 차이는 점차적으로 커지고 있다. 또한 EMC 휴리스틱 알고리즘의 제곱오차와 타부 탐색 알고리즘의 제곱오차 차이도 군집의 수가 13개까지는 점차적으로 커지고 있다. 이 결과는 EMC 휴리스틱 알고리즘이 군집의 수가 많을 경우에 특히 타 알고리즘에 비해서 제곱오차의 개선효과가 크다는 것을 나타낸다.



<그림 12> 실제 데이터의 군집 알고리즘 별 제곱오차



<그림 13> EMC 휴리스틱 알고리즘과 타 알고리즘의 제곱오차 간 차이

4. 결 론

본 연구는 k-means와 타부 탐색 등 기존의 군집방법에 비해서 군집의 효율을 향상시킬 수 있는 EMC 휴리스틱 알고리즘을 제안하였으며 그 성과는 다음과 같다. 첫째, 초기 군집의 중심 배치방법을 개선하였다. 이 방법은 군집의 중심 이동 폭을 크게 하기 때문에 개선된 해를 찾는 확률을 증가시킨다. 둘째, 군집의 중심 간 간격을 변경해나가면서 탐색영역을 다르게 하여 최적해에 근사한 해를 체계적으로 탐색할 수 있도록 하였다. 이 방법은 모든 영역에 대해서 해를 체계적으로 탐색할 수 있기 때문에 최적해에 근사한 해를 찾을 수 있는 확률을 증가시킨다. 그리고 군집의 중심 간 간격 범위설정을 정형화하였기 때문에 타부 탐색과는 다르게 군집의 해는 일관성을 갖는다. 그 다음으로 군집의 계산효율을 향상시키기 위해 일부 객체들만으로 군집에 필요한 거리비교를 하였다. 이 방법은 반복수가 많은 경우 계산량을 크게 줄일 수 있다.

EMC 휴리스틱 알고리즘의 성능은 실험을 통해 그 우수성을 확인하였다(<표 2>, <그림 12> 참조). 실제 데이터를 사용한 실험에서 군집 수의 변화에 따른 EMC 휴리스틱 알고리즘의 성능을 실험하였고 그 결과 모든 군집 수에서 EMC 휴리스틱 알고리즘의 제곱오차는 타 알고리즘의 제곱오차에 비해 항상 개선된 결과를 나타내었다. 특히 군집의 수가 많아질수록 타 알고리즘 보다 제곱오차가 많이 개선되었다. 한편으로 군집 수의 변화는 EMC 휴리스틱 알고리즘의 성능에 영향을 주지 않는다는 것도 확인하였다.

본 연구에서 제안한 EMC 휴리스틱 알고리즘은 타 알고리즘 보다 개선된 해를 찾을 수 있기 때문에 군집을 사용하는 추천 시스템에 적용할 경우 추천 시스템의 성능을 향상시킬 수 있다. 향후 연구과제는 추천 시스템에 본 연구에서 제안한 EMC 휴리스틱 알고리즘을 적용하여 추천 시스템의 성능을 향상시키는 것이다.

5. 참 고 문 헌

[1] 강창완, 강현철, 데이터마이닝, 사이플러스, (2007)
[2] 박우창, 승현우, 용환승, 최기현, 데이터마이닝 개념 및 기법, 자유아카데미, (2004)
[3] 허준, 정규상, 허수희, 최희경, Clementine 7 매뉴얼, 데이터 솔루션, (2003)
[4] Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R. and Wu, A.Y., An efficient k-means

clustering algorithm: analysis and implementation, Pattern Analysis and Machine Intelligence, 24 (2002) : 881 - 892

[5] K.S. Al-sultan, A tabu search approach to the clustering problem, Pattern Recognition, 28 (1995) : 1443 - 1451

[6] Ordonez, C., Integrating K-means clustering with a relational DBMS using SQL, Knowledge and Data Engineering, 18 (2006) : 188 - 201

[7] Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, Grouplens : An Open Architecture for Collaborative Filtering of Netnews, Proceedings of the ACM Conf. on Computer Supported Cooperative Work (1994), 175-186

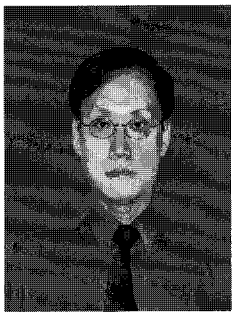
[8] SPSS, Clementine 8.0 user's guide, SPSS, (2003)

[9] Temel Oncan, Jean-Francois Cordeau and Gilbert Laporte, A tabu search heuristic for the generalized minimum spanning tree problem, European Journal of Operational Research, 191 (2008) : 306 - 319

[10] Yongguo Liu, Zhang Yi, Hong Wua, Mao Ye and Kefei Chen, A tabu search approach for the minimum sum-of-squares clustering problem, Information Sciences, 178 (2008) : 2680 - 2704

저 자 소 개

이 석 환



인하대학교 산업공학과에서 공학사 및 공학석사를 취득 하였고 박사과정을 수료하였다. 주요 관심 분야는 데이터 마이닝이다.

주소: 인천광역시 남구 용현동 253, 인하대학교 산업공학과

박 승 헌

인하대학교 금속공학과에서 공학사, 일본 Keio대학 관리공학과에서 공학석사 및 공학박사를 취득하였다. 현재 인하대학교 산업공학과 교수로 재직 중이다. 주요 관심 분야는 FMS와 각종 생산 시스템의 설계 및 운영, 인터넷 마케팅과 데이터 마이닝 등이다.

주소: 인천광역시 남구 용현동 253, 인하대학교 산업공학과