J. Soc. Korea Ind. Syst. Eng Vol. 40, No. 2 : 92-98, June 2017
https://doi.org/10.11627/jkise.2017.40.2.092
ISSN : 2005-0461(print)
ISSN : 2287-7975(online)

Hybrid Simulated Annealing for Data Clustering

Sung-Soo Kim† · Jun-Young Baek · Beom-Soo Kang

Department of System & Management Engineering, Kangwon National University

데이터 클러스터링을 위한 혼합 시뮬레이티드 어닐링

김성수† · 백준영 · 강범수

강원대학교 시스템경영공학과

Data clustering determines a group of patterns using similarity measure in a dataset and is one of the most important and difficult technique in data mining. Clustering can be formally considered as a particular kind of NP-hard grouping problem. K-means algorithm which is popular and efficient, is sensitive for initialization and has the possibility to be stuck in local optimum because of hill climbing clustering method. This method is also not computationally feasible in practice, especially for large datasets and large number of clusters. Therefore, we need a robust and efficient clustering algorithm to find the global optimum (not local optimum) especially when much data is collected from many IoT (Internet of Things) devices in these days. The objective of this paper is to propose new Hybrid Simulated Annealing (HSA) which is combined simulated annealing with K-means for non-hierarchical clustering of big data. Simulated annealing (SA) is useful for diversified search in large search space and K-means is useful for converged search in predetermined search space. Our proposed method can balance the intensification and diversification to find the global optimal solution in big data clustering. The performance of HSA is validated using Iris, Wine, Glass, and Vowel UCI machine learning repository datasets comparing to previous studies by experiment and analysis. Our proposed KSAK (K-means+SA+K-means) and SAK (SA+K-means) are better than KSA(K-means+SA), SA, and K-means in our simulations. Our method has significantly improved accuracy and efficiency to find the global optimal data clustering solution for complex, real time, and costly data mining process.

Keywords : Data Clustering, Hybrid Simulated Annealing, K-means

1. 연구의 배경 및 목적

최근 다양한 사물인터넷(Internet of Things, IoT) 기기로부터 엄청난 양의 데이터들이 수집되고 수집된 빅데이터를 더 효율적으로 분석하는 방법의 연구와 개발의 필요성이 대두되고 있다. 기존 데이터 분석 기능 중 데이터 클러스터링은 비계층적(partitioning) 방법과 계층적(hierarchical) 방법으로 나눌 수 있고[8], 이 중 비계층적 방법은 철강 산업 등 다양한 분야에 적용될 수 있다[11]. 비계층적 데이터클러스터링을 하기 위해 보편적으로 사용되는 K-means 방법은 언덕 오르기(hill climbing) 방식의 탐색을 하기 때문에 초기 해에 따라 민감하고 탐색 해의 편차가 매우 크며 지역해에 빠질 가능성이 높다는 문제점이 있다[12]. 특히, 최근의 빅데이터 분석을 위해서는 기존 데이터 클러스터링 방법의 한계를 극복하고 안정적이고 빠르게 전역해를 탐색할 수 있는 빅데이터 클러스터링 방법의 개발이 절실하다. 또한, 데이터 클러스터링은 클러스터의 수가 3 이상이 될 경우 가능 해의 수가 기하급수적으로 증가하여 해 탐색 공간이 엄청나게 증가하고 복잡도가 매우 커져 NP-hard 문제가 된다[5]. 이와 같이 복잡도가 매우 큰 데이터

Received 24 April 2017; Finally Revised 16 June 2017; Accepted 19 June 2017
† Corresponding Author : kimss@kangwon.ac.kr

클러스터링 그룹핑 문제를 해결하기 위해 휴리스틱알고리즘을 적용해야할 필요성이 주장 되었다[2]. 이와 관련된 기존 연구들은 다음과 같다.

Selim[10]은 클러스터링 문제에 시뮬레이티드 어닐링(simulated annealing, SA)을 제안하였고, Sun[13]과 Perim[9]은 K-means 또는 개선된 K-means로 구한 해를 SA의 초기값으로 하는 클러스터링 방법을 제안하였다. Gungor[1]은 K-harmonic means 클러스터링 문제를 해결하기 위해 SA를 적용하였다. Krishna[4]는 유전자 알고리즘(Genetic algorithm, GA)의 교배과정 대신에 K-means 알고리즘을 적용한 Genetic K-means 방법을 제안하였고 수렴속도를 개선하였다. Maulik[6]는 GA 방법을 제안하고 기존 방법의 한계를 극복하고 전역해를 탐색할 수 있음을 검증하였다. Kumar[5]는 초기해를 임의적으로 선택하는 대신에 K-means로 선택하여 사용한 인공벌군집(artificial bee colony)을 제안하였다. 이와 같이 데이터분석 분야의 데이터 클러스터링을 위해 연구가 계속해서 진행되고 있고 특히, 빅데이터를 효율적으로 데이터 클러스터링할 수 있는 방법의 개발이 절실하다.

본 논문의 목적은 데이터 분석 분야의 비계층적 클러스터링 방법의 성능을 개선하기 위해 K-means의 장점과 SA의 장점을 효율적으로 혼합한 시뮬레이티드 어닐링(Hybrid Simulated Annealing, HSA) 데이터 클러스터링 4가지 방법들을 비교 분석하는 것이다. 제 2장에서는 데이터분석 분야의 데이터 클러스터링 문제의 수학적 모델 정립을 설명하였다. 제 3장에서는 K-means와 SA의 장점을 혼합한 HSA 데이터 클러스터링 방법을 설명하였다. 제 4장에서는 HSA 방법의 성능을 검증하기 위해 실제 UCI데이터를 사용하여 K-means, 임의로 초기값을 선택하는 SA[10]와 초기값을 K-means로 구한 SA 즉 KSA[9]와 본 논문에서 새롭게 시도한 SAK(SA 적용 후 K-means 적용), KSAK(초기값을 K-means로 구하고 SA 적용 후 다시 K-means 적용) 방법의 실험결과를 검증하고 비교 분석 하였다.

2. 데이터 클러스터링 문제

데이터 클러스터링 문제(즉, n개의 데이터를 K개의 그룹으로 클러스터링 하는 문제)를 수리적으로 정립화 할 수 있다[4, 6, 9, 10, 13, 14]. 데이터 집합 $X = \{x_1, x_2, \cdots, x_n\}$는 데이터 $i(x_i)$로 구성된다($i = 1, 2, \cdots, n$). 각각의 x_i는 d차원(특징, attribute)으로 구성 되는데 $x_{ij} = [x_{i1}, \cdots, x_{id}]$는 데이터 i의 특징데이터 j의 값을 표현한 것이다. 또한, $k(k = 1, 2, \cdots, K)$개의 클러스터 서브 집합 $C = \{C_1, C_2, \cdots, C_K\}$로 서로 겹치지 않는 클러스터로 구성된다. 각 클러스터 집합은 적어도 한 개의 데이터가 존재한다. 본 논문의 목적은 각 클러스터 내에서 평균과 소속된 데이터 사이의 유클리드 거리의 총합을 나타내는 식 (1)을 최소화하는 것이고 식 (2)~식 (6)으로 나타낼 수 있다.

$$\text{Minimize} \quad \sum_{k=1}^{K} S(k) \tag{1}$$

$$\text{s.t} \quad w_{ik} = \begin{cases} 1 & \text{데이터 } i\text{가 클러스터 } k\text{에 포함된 경우} \\ 0 & \text{그렇지 않을 경우} \end{cases} \tag{2}$$

$$\sum_{k=1}^{K} w_{ik} = 1,\ i = 1, 2, \cdots, n \qquad w_{ik} \in \{0, 1\} \tag{3}$$

$$\sum_{i=1}^{n} w_{ik} \geq 1,\ k = 1, 2, \cdots, K \tag{4}$$

$$C_{ij} = \frac{\sum_{i=1}^{n} w_{ik} x_{ij}}{\sum_{i=1}^{n} w_{ik}} \tag{5}$$

$$S(k) = \sum_{i=1}^{n} w_{ij} \sqrt{\sum_{j=1}^{d} (x_{ij} - c_{kj})^2} \tag{6}$$

만약 데이터 $i(x_i)$가 클러스터 k에 포함되었을 경우 w_{ik}를 1로 표시하고 그렇지 않을 경우 0으로 표시하여 식 (2)와 같이 정의할 수 있다. 데이터 클러스터링 해 표현 매트릭스를 $W = \{w_{ik}\}$로 나타낼 수 있고 x_i가 하나의 클러스터 k에 포함되는 여부에 따라 식 (3)과 같이 표현할 수 있다. 식 (4)는 클러스터 k에 적어도 하나 이상의 데이터 $i(x_i)$가 포함되어 있는 것을 표현하였다. 식 (5)의 C_{kj}는 $C_k = (C_{k1}, C_{k2}, \cdots, C_{kd})$의 클러스터 k에서 특징(feature) 데이터 j의 평균값을 나타낸다. 식 (6)은 클러스터 k의 유클리드 거리의 합을 나타낸 것이다.

3. 혼합 시뮬레이티드 어닐링 데이터 클러스터링

본 장에서는 데이터 클러스터링을 위해 어떻게 효율적으로 시뮬레이티드 어닐링(Simulated Annealing, SA)과 K-means를 혼합하여 적용할 수 있는지를 설명하였다. 제 3.1절에서 K-means와 SA의 일반적인 메커니즘을 서술하였고, 제 3.2절에서 본 논문에서 제안하는 혼합 시뮬레이티드 어닐링(Hybrid Simulated Annealing, HSA)을 설명하였다.

3.1 K-means와 시뮬레이티드 어닐링

K-means는 초기해를 임의적으로 선택한다. 각 클러스터에 소속된 데이터들의 평균을 구하고 각 클러스터의 평균과 소속 데이터 간의 거리의 총합을 기준으로 현재의

해를 평가한다. 모든 데이터들을 각 클러스터의 평균을 기준으로 재할당하여 새로운 클러스터링을 구하고 해당 클러스터 해를 다시 평가한다. 이러한 과정들을 더 좋은 클러스터링 결과가 나오지 않을 때까지 반복하여 클러스터링 해를 탐색한다. K-means는 효율적이지만 초기해에 민감하고 지역해를 탐색할 가능성이 높다[12].

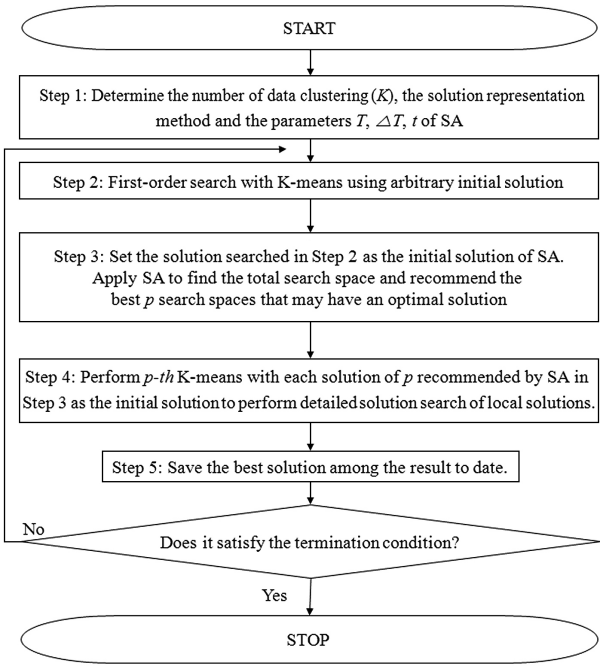
시뮬레이티드 어닐링 SA는 일반적으로 해 탐색 공간이 크고 멀티모달 함수에서 최적해를 탐색할 수 있는 알고리즘이다[3, 7]. 즉, SA는 현재 해를 바탕으로 일정한 규칙을 통해 새로운 이웃 해를 생성 하면서 비교해 나가는 과정을 이용하여 최적값에 접근한다. 또한, SA는 단순 비교 외에 '확률치를 이용하여 새로운 이웃해 채택여부판단' 단계에서 SA의 온도 파라미터를 이용하여 다양한 새로운 해를 선택하게 함으로써 지역해에 빠지는 것을 방지하면서 해들을 탐색한다. 즉, 새로운 해가 더 좋은 해가 아닐 때에도 확률적으로 해를 받아들여 다양한 해 탐색을 통하여 전역해 탐색을 시도한다. Selim[10]은 데이터 클러스터링 문제에 SA를 제안하였다.

이와 같이 SA는 전체 탐색 공간을 다양하게 탐색하여 최적해가 존재할 가능성이 높은 해 탐색 공간을 찾아 낼 수 있다. K-means는 지역 탐색 공간 내에서 빠른 시간 내에 효율적으로 탐색하여 해를 찾아낼 수 있다. 두 방법의 장점을 공유하고 단점을 보완하는 새로운 방법의 개발이 필요하다.

3.2 혼합 시뮬레이티드 어닐링 데이터 클러스터링

HSA는 SA를 수행하고 K-means를 적용하는 방법(SA+K-means, SAK), K-means로 초기값을 구하고 SA를 수행하는 방법(K-means+SA, KSA), K-means로 초기값을 구하고 SA를 수행한 후 다시 K-means를 수행하는 방법(K-means+SA+K-means, KSAK)들이다. Selim[10]이 제안한 SA만을 적용한 방법, Perim[9]이 제안한 KSA 방법들과 비교하기 위해 본 논문에서 차별화 하여 제안하는 SAK와 KSAK를 중점적으로 설명하고자 한다. KSAK는 K-means로 탐색한 좋은 해를 초기해(SAK는 임의의 초기해 사용)로 하는 SA를 적용하여 다양한 해 탐색 후 가능성 있는 해들을 추천하고 이 해들을 초기값으로 하여 다시 한 번 K-means로 해를 수렴시켜 더 좋은 해를 탐색하는 방법이다. 본 논문에서 제안하는 방법의 SA의 역할은 매우 큰 탐색 공간에서 다양한 해 탐색 능력을 활용하여 p개의 가능성 있는 해를 추천하는 것이고 지역탐색 공간과 초기해가 정해지면 지역 탐색능력이 뛰어나고 효율적인 K-means의 능력을 활용하는 것이다.

<Figure 1>의 단계 1에서 클러스터링 수, 해 표현과 SA 파라미터를 설정 한다. 단계 2에서 K-means로 1차 해를 생성한다(SAK는 단계 2 생략). 단계 3에서 다양한 해 탐색을 위한 SA를 적용한다. 단계 4에서 SA로 추천된 탐색 공간에서 빠르고 효율적인 해 수렴을 위해 K-means를 적용한다. SA와 K-means의 장점을 공유하고 단점을 보완할 수 있도록 균형을 맞추어 혼합 사용하였고 다음과 같이 단계별로 설명한다.



<Figure 1> KSAK(Step 2 is omitted in SAK) Flowchart

[데이터 클러스터링을 위한 KSAK 방법 : SAK는 단계 2가 생략됨]

단계 1 : [클러스터링 수, 해 표현, SA파라미터 설정]

데이터 클러스터링의 수(K)와 해 표현을 결정한다. 예를 들어, 분석할 데이터의 수가 150개이고 K가 3일 경우 해 표현 방법은 2차원 매트릭스 형태인 3×150으로 표현하였다. 첫 번째 데이터가 첫 번째 클러스터에 할당되었을 경우 이진수 (1, 0, 0)으로 표현하고 나머지 데이터도 이와 같은 방법으로 표현할 수 있다. SA의 다양한 탐색을 위해 해를 받아들일 확률을 설정하는 역할을 하는 T의 크기, 다양한 탐색의 확률을 낮추고 수렴적 탐색을 증가시키기 위해 T를 감소시키는 역할을 하는 $\triangle T$의 크기, 각 데이터의 이웃 해를 탐색하는 횟수의 역할을 하는 t를 결정한다. 이 파라미터는 데이터의 형태에 따라 제 4장의 실험데이터 Iris, Wine, Glass는 (T = 1~10, $\triangle T$ = 0.01~1, t = 1,000~3,000), Vowel은 (T = 50~100, $\triangle T$ = 10~20, t = 5,000~10,000), Cloud는 (T = 10~30, $\triangle T$ = 0.001~1, t = 100~500) 선택 범위에서 여러 번의 실험을 통하여 적절하게 설정되었다.

단계 2 : [K-means로 해 생성 : SAK에서는 생략]

n개의 데이터를 K개의 클러스터 중 랜덤으로 각 클러스터에 할당하여 K-means의 초기해로 설정한다. 각 클러스터의 평균을 구하고 각 클러스터 내의 데이터와 클러스터 평균과의 거리를 구하여 해를 평가한다. 모든 데이터에서 각 클러스터의 평균과의 거리가 가장 가까운 클러스터의 소속으로 각 데이터를 재할당하여 새로운 클러스터를 구성한다. 새롭게 클러스터링 하여 데이터 군집화한 평가값이 기존 평가값보다 좋고 각 클러스터의 평균값이 변하지 않을 때까지 반복하여 K-means의 해를 탐색한다.

단계 3 : [지역 공간 탐색 추천을 위한 SA 적용]

단계 2에서 K-means로 얻은 해를 초기해로 SA를 적용하여 전체 탐색공간에서 best p개의 해를 탐색한다. 초기해(현재해)의 데이터군집 평가값 $f(e)$와 새롭게 구한 이웃해의 평가값 $f(n)$을 제 2장의 식 (1)로 계산한다. 더 좋은 해가 생성이 되면 해를 새로 업데이트 한다. 만약, 좋지 않은 해가 생성되었을 경우 최소화 문제인 경우 해를 받아들이는 함수$[exp[-(f(n)-f(e))/T]$에 따라 확률적으로 좋지 않은 해도 다양한 해 탐색을 위해 받아들일 수 있다. 초기에는 다양한 탐색을 추구하고 탐색 과정이 진행되면서 다양한 해 탐색 확률을 낮추고 수렴적 해 탐색 확률을 높이기 위해 T를 $\triangle T$만큼 감소시키고 종료조건을 만족할 때까지 현재까지 탐색한 가장 좋은 p개의 해(지역탐색 공간)를 도출하여 단계 4의 더 좋은 해 탐색 K-means 적용을 위해 추천한다.

단계 4 : [추천된 지역탐색 공간에서 K-means 적용]

단계 3의 SA가 추천한 p개의 해 각각을 초기해로 하여 K-means를 p회 실행하여 지역 탐색 공간들을 세밀하게 최적해 탐색을 한다.

단계 5 : [좋은 해 저장과 종료 조건 확인]

현재까지 나온 해 집합에서 가장 좋은 해를 저장한다. 종료 조건(일정 세대를 진행하여도 해의 개선이 되지 않을 때 종료)을 만족하면 종료하고 그렇지 않으면 단계 2부터 반복한다.

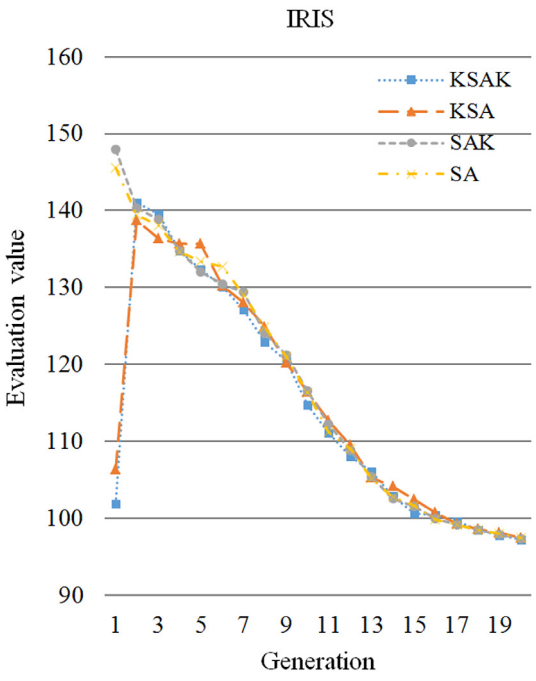
4. 실험결과 및 분석

본 장에서의 실험은 윈도우10 프로세서 : Intel(R) Core TM i5-4590 CPU @ 3.30GHz 3.30 GHz 메모리(RAM) : 4GB, 64비트 운영 체제, x64 기반 프로세서 운영체제, Visual C++ 환경에서 실험하였다. 본 논문에서 제안한 HSA(SAK, KSAK) 데이터 클러스터링 방법의 성능을 검증하기 위해서 기존 연구에서 널리 사용한 Iris, Wine, Glass, Vowel, Cloud 데이터(UCI machine learning repository[15~19])를 사용하여 각 20회 실험 분석하였다.

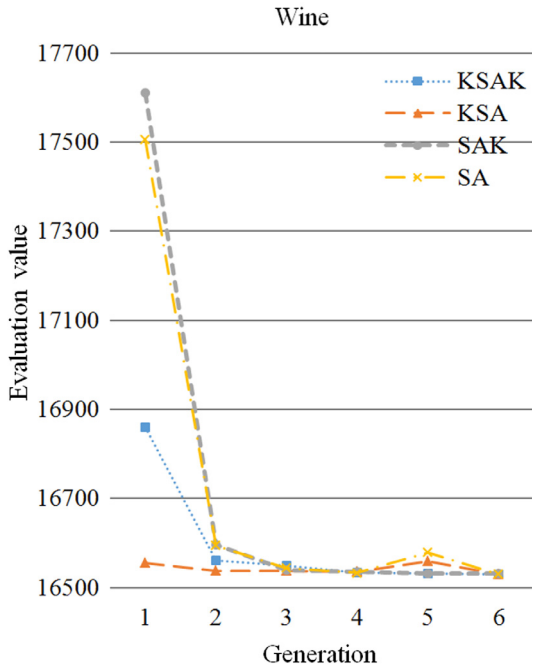
<Table 1>은 실험 분석 데이터의 클러스터 수, 각 데이터의 특징 수, 데이터 수를 나타낸 것이다. 실험 시 사용한 HSA의 파라미터 값 T(초기 온도), $\triangle T$(온도 감소치), t(고정된 온도에서 이웃해 탐색 횟수)는 평가값의 크기, 데이터 수, 특징 수, 클러스터 수를 고려하여 다양한 해를 충분히 탐색할 수 있도록 실험을 통해 설정하였다. 각 데이터 파라미터는 Iris(T = 1, $\triangle T$ = 0.05, t = 1,500), Wine(T = 6, $\triangle T$ = 1, t = 1,780), Glass(T = 1, $\triangle T$ = 0.03, t = 2,140), Vowel(T = 100, $\triangle T$ = 10, t = 8,710), Cloud(T = 15, $\triangle T$ = 0.025, t = 250)와 같다.

<Table 1> Data for Experiment(UCI Machine Learning Repository)-[15~19]

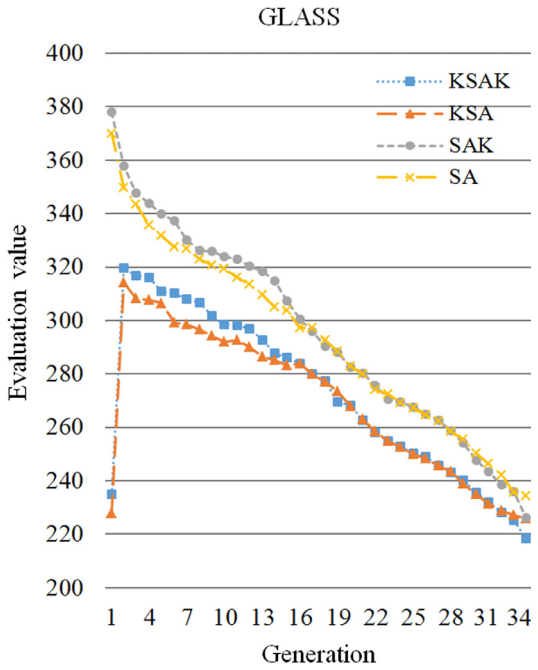
Name of dataset	No. of classes	No. of features	No. of data
Iris	3	4	150
Wine	3	13	178
Glass	6	9	214
Vowel	6	3	871
Cloud	10	10	1,024



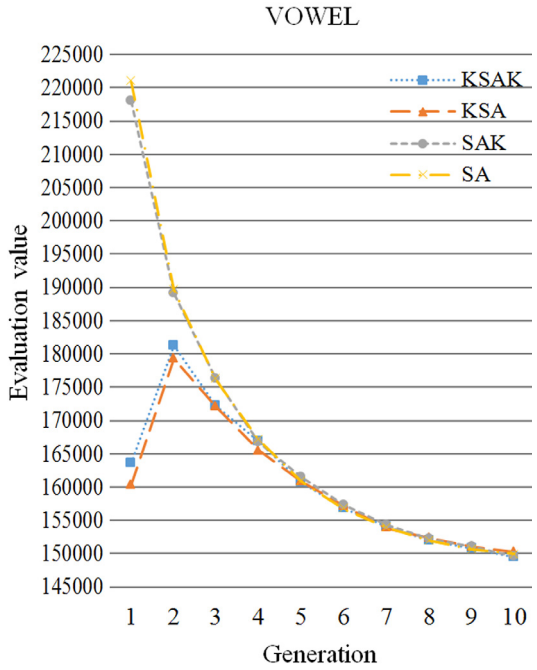
<Figure 2(A)> Trend of Convergence for Data Iris



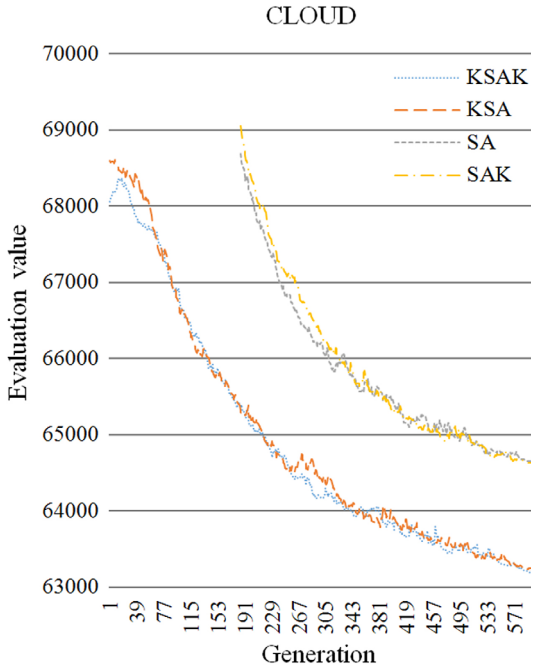
<Figure 2(B)> Trend of Convergence for Data Wine



<Figure 2(C)> Trend of Convergence for Data Glass



<Figure 2(D)> Trend of Convergence for Data Vowel



<Figure 2(E)> Trend of Convergence for Data Cloud

<Figure 2(A)~<Figure 2(E)>는 <Table 1>의 5가지 데이터에 대하여 4가지 데이터 클러스터링 방법(SA, KSA, SAK, KSAK)을 적용했을 때 수렴 경향을 나타낸 것이다. X축은 세대수를 나타내는데, 세대란 SA의 T값이 고정되어 있는 상태이다. T가 ΔT만큼 감소했을 때 세대가 바뀐다. 평가값(Y축)은 본 논문 제 2장의 식 (1)을 사용하여 유클리드 거리가 계산되는데 해당 세대에서 얻을 수 있는 데이터 클러스터링 해 평가값을 구하여 실험 20회에 대한 평균값을 나타낸다.

초기값을 임의적으로 선택한 SA와 SAK는 초기해의 평가값이 좋지 않으나 세대가 진행되면서 해의 평가값이 수렴하는 경향을 보였다. 이와 다르게 임의적으로 초기값

을 선택하지 않고 K-means를 활용하여 해를 구한 후 이 해를 SA의 초기값으로 사용한 KSA와 KSAK는 해의 평가값이 상대적으로 좋은 초기해로 시작하지만 SA의 다양한 해 탐색 시도(SA의 적용 초기에는 다양한 해 탐색을 위하여 초기 파라미터 T가 상대적으로 크기 때문에 좋지 않은 평가값을 선택할 확률이 큼)로 평가값이 잠시 수렴하지 못하다가 다시 정상적으로 수렴하는 경향이 나타났다. 초기에 K-means를 적용한 클러스터링 방법이 전반적으로 유리한 조건에서 해를 탐색하는데 특히, Glass 데이터 경우 초기값을 K-means로 적용한 경우 다른 방법과 비교하여 해 평가값의 표준편차가 낮아 안정적인 해 탐색이 가능하였다. 특히, <Figure 2(E)>의 Cloud 데이터 경우 초기값을 랜덤으로 선택한 SA, SAK와 K-means로 선택한 KSA, KSAK의 차이가 심하여 수렴 경향을 표시하기 위해SA, SAK의 초기값 부분을 표시하지 않고 수렴 경향을 비교하였다. KSA와 KSAK 방법에서 SA의 역할은 전체 해 공간을 다양하게 해 탐색 후 몇 개의 해를 추천하는 것이며 K-means의 역할은 추천된 각각의 해를 사용하여 효율적으로 해를 탐색하는 것이다.

<Table 2>는 K-means와 SA, KSA, SAK, KSAK 실험결과를 비교한 것이다. 각 방법에 대하여 20회의 실험을 통하여 평균, 편차, 가장 좋은값을 비교하였다. <Table 2>의 결과에 따르면 K-means는 임의로 선택한 초기값에 따라 해의 평가값의 차이가 커서 평가값의 표준편차가 매우 크다. Iris, Wine, Glass 데이터의 경우 Perim[9]이 제안한 KSA는 K-means와 SA[10]만을 적용한 경우보다 해의 평가값의 평균과 표준편차가 향상되었다. Vowel 데이터의 경우 SA보다 KSA 방법이 해 평가값의 평균과 표준편차가 좋지 못하였다.

본 논문에서 제안한 SAK와 KSAK는 5가지 데이터를 데이터 클러스터링 했을 때, K-means, SA, KSA보다 해 평가값의 평균, 표준편차, 가장 좋은 해가 모두 우수하였다. 이 방법들은 SA로 Best p개의 추천된 탐색 공간에서 K-means의 조화로운 해 탐색을 통하여 더 안정적이고 효율적인 탐색이 가능하였다. 상대적으로 탐색 공간이 작은 Iris와 Wine 데이터로는 KSA, SAK, KSAK 방법간의 차별성이 크지 않으나, 상대적으로 해 탐색 공간과 복잡도가 높은 Glass와 Cloud 데이터로 실험한 결과 KSAK 방법이 가장 우수하였고 Vowel 데이터인 경우 SAK 방법이 가장 우수하였다. 본 논문에서 제안한 SAK와 KSAK는 상대적으로 더 큰 사이즈의 데이터를 분석할 때 성능(평가값)이 더 우수하였다. 계산 시간 측면에서 4가지 클러스터링 방법의 성능을 비교해 볼 때 데이터의 수와 복잡도가 가장 높은 Cloud 경우 SA는 74.8719 sec, KSA는 73.43905 sec, SAK는 72.15625 sec, KSAK는 73.20595 sec로 측정되어 차이가 크지 않은 것으로 분석되었다.

<Table 2> Comparative Study of K-means, SA, KSA, SAK, KSAK

		K-means	SA[10]	KSA[9]	SAK	KSAK
IRIS	Mean	102.0028	97.41501	97.68034	97.27234	97.23119
	S.D.	11.37883	0.210537	0.906196	0.053436	0.003109
	Best	97.3259	97.2221	97.2221	97.2322	97.2221
WINE	Mean	16934.61	16564.47	16530.5	16530.5	16530.5
	S.D.	1651.633	151.8961	0	0	0
	Best	16555.7	16530.5	16530.5	16530.5	16530.5
GLASS	Mean	225.2024	231.3193	223.1558	222.0516	217.8665
	S.D.	10.66858	14.56847	2.489383	10.55073	1.294846
	Best	215.678	221.69	214.727	218.476	214.669
VOWEL	Mean	159251.1	149685.3	150412.1	149430.6	149758.8
	S.D.	9794.832	283.408	880.1715	81.15448	533.7285
	Best	149384	149407	149405	149375	149380
CLOUD	Mean	67055.389	64638.702	63214.06	64338.549	63132.689
	S.D.	648.09	755.63	406.5902	737.73	417.618
	Best	66194.641	62889.885	62937.95	62834.697	62856.856

5. 결 론

데이터분석을 위한 군집방법의 하나인 K-means는 현재까지도 널리 사용되는 효율적인 알고리즘이지만 초기값에 매우 민감하여 탐색한 데이터 클러스터링 결과가 지역해에 머물 가능성이 높다. 이러한 문제점은 과거 수집된 데이터의 양이 상대적으로 적었을 때 보다는 최근처럼 엄청난 양의 데이터를 분석하고자할 때에는 더 큰 문제가 될 수 있다. 이런 문제점을 극복하기 위해 본 논문에서는 안정적인 전역해를 탐색할 수 있는 새로운 혼합 시뮬레이티드 어닐링 데이터 클러스터링 방법인 SAK와 KSAK를 제안하였다. KSAK는 K-means로 해를 구한 후 이 해를 초기값으로 한 SA로 해 탐색 후 몇 개의 해를 추천하면 추천된 각각의 해를 K-means로 다시 가장 좋은 해를 탐색하는 방법이다. SAK는 KSAK에서 초기값을 K-means로 먼저 탐색하는 부분을 생략한 방법이다.

혼합 시뮬레이티드 어닐링 데이터 클러스터링 방법은 전체 해 공간을 대상으로 다양한 해 탐색을 통하여 최적해가 존재할 가능성이 높은 가장 좋은 p개의 탐색 공간의 해들을 추천한다. 추천 받은 p개의 탐색공간의 해들을 K-means의 초기해로 각각 사용하여 해당 지역 탐색 공간에서 해를 탐색하여 가장 좋은 데이터 클러스터링 해를 찾아낸다. 즉, HSA는 전역탐색과 지역탐색의 효율적인 혼합방법인 것이다. UCI 데이터를 활용하여 본 논

문에서 새롭게 제안하는 방법인 SAK와 KSAK의 성능이 데이터의 크기와 복잡도가 커질수록(Glass, Vowel, Cloud가 Iris와 Wine보다 상대적으로 큼) 우수하고 효과적임을 검증하였다.

Acknowledgements

This study is supported by 2016 Research Grant from Kangwon National University(Grant No. 520160235). This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP)(No.R0250-17-1002, Education Project for SW Convergence Business Data Analysis).

References

[1] Gungor, Z. and Unler, A., K-harmonic means data clustering with simulated annealing heuristic, *Applied Mathematics and Computation*, 2007, Vol. 184, No. 2, pp. 199-209.

[2] Hruschka, E.R. and Campello, R.J., A survey of evolutionary algorithms for clustering, *IEEE Transactions on systems, man, and cybernetics-Part C(Applications and reviews)*, 2009, Vol. 39, No. 2, pp. 133-155.

[3] Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P., Optimization by simulated annealing, *Science*, 1983, Vol. 220, No. 4598, pp. 671-680.

[4] Krishna, K. and Murty, M.N., Genetic K-means algorithm, *IEEE Transactions on Systems, Man, and Cybernetics, Part B(Cybernetics)*, 2002, Vol. 29, No. 3, pp. 433-439.

[5] Kumar, Y. and Sahoo, G., A two-step artificial bee colony algorithm for clustering, *Neural computing & Applications*, 2017, Vol. 28, No. 3, pp. 537-551.

[6] Maulik, U. and Bandyopadhyay, S., Genetic algorithm-based clustering technique, *Pattern Recognition*, 2000, Vol. 33, No. 9, pp. 1455-1465.

[7] Michalewicz, Z., Genetic Algorithms+Data Structures = Evolution Programs, *Springer Verlag Berlin Heidelberg*, New York, 1992.

[8] Oh, S.J. and Kim, J.Y., Clustering Algorithm for Sequences of Categorical Values, *Journal of the Society of Korea Industrial and Systems Engineering*, 2003, Vol. 26, No. 1, pp. 17-21.

[9] Perim, G.T., Wandekokem, E.D., and Varejao, F.M., K-Means Initialization Methods for Improving Clustering by Simulated Annealing, *11th Ibero-American Conference on AI*, 2008, Lisbon, Vol. 5290, pp. 133-142.

[10] Selim, S.Z. and Alsultan, K., A simulated annealing algorithm for the clustering problem, *Pattern Recognition*, 1991, Vol. 24, No. 10, pp. 1003-1008.

[11] Seo, M.K. and Yun, W.Y., Clustering-based Monitoring and Fault detection in Hot Strip Roughing Mill, *Journal of the Korean Society for Quality Management*, 2017, Vol. 45, No. 1, pp. 25-38.

[12] Singh, S.S. and Chauhan, N.C., K-means v/s K-medoids : A Comparative Study, *National Conference on Recent Trends in Engineering & Technology*, 2011.

[13] Sun, L.X., Xie, Y.L., Song, X.H., Wang, J.H., and Yu, R.Q., Cluster analysis by simulated annealing, *Computers & Chemistry*, 1994, Vol. 18, Issue. 2, pp. 103-108.

[14] Sun, L.X., Xu, F., Liang, Y.Z., Xie, Y.L., and Yu, R.Q., Cluster analysis by the K-means algorithm and simulated annealing, *Chemometrics and intelligent Laboratory Systems*, 1994, Vol. 25, No. 1, pp. 51-60.

[15] *UCI machine learning repository Cloud datasets*, https://archive.ics.uci.edu/ml/datasets/cloud.

[16] *UCI machine learning repository Glass datasets*, https://archive.ics.uci.edu/ml/datasets/glass.

[17] *UCI machine learning repository Iris datasets*, https://archive.ics.uci.edu/ml/datasets/iris.

[18] *UCI machine learning repository Vowel datasets*, https://archive.ics.uci.edu/ml/datasets/vowel.

[19] *UCI machine learning repository Wine datasets*, https://archive.ics.uci.edu/ml/datasets/wine.

ORCID

Sung-Soo Kim | http://orcid.org/0000-0002-8765-1193
Jun-Young Baek | http://orcid.org/0000-0003-4418-1643
Beom-Soo Kang | http://orcid.org/0000-0003-0507-3658