클러스터링 균형을 사용하여 최적의 클러스터 개수를 결정하기 위한 효율적인 휴리스틱

An efficient heuristics for determining the optimal number of cluster using clustering balance

이상욱
목원대학교 정보통신공학과

Sangwook Lee
Division of Information and Communication,
Mokwon University

요약

데이터 클러스터링 분야에서 최적의 클러스터 개수를 추정하는 것은 매우 중요한 일이다. 그것은 클러스터링의 적합성을 판단할 기준을 정하고 그 적합성을 극대화 하는 최적의 클러스터의 개수를 찾는 것이다. 본 논문에서는 클러스터의 적합성을 판단할 기준으로써 클러스터링 균형을 사용하여 최적의 클러스터 개수를 찾기 위한 효율적인 휴리스틱 방법을 제안하였다. k-means 사용하여 가상 및 실제 데이터 셋에 적용한 결과, 제안한 알고리즘이 계산효율 측면에서 우수함을 확인할 수 있었다.

Abstract

Determining the optimal number of cluster is an important issue in research area of data clustering. It is choosing the cluster validity method and finding the cluster number where it optimizes the cluster validity. In this paper, an efficient heuristic for determining optimal number of cluster using clustering balance is proposed. The experimental results using k-means at artificial and real-life data set show that proposed algorithm is excellent in aspect of time efficiency.

Ⅰ. 서론

클러스터링이란 패턴 형태에 대한 사전정보 없이 분류하는 unsupervised grouping 기법이다. 이러한 unsupervised 기법에서는 라벨이 붙은 데이터를 이용할 수가 없다. 클러스터링의 목표는 알려지지 않은 데이터 구조의 라벨이 붙지 않은 데이터들을 몇 개의 의미 있는 그룹으로 분리시켜주는 것이다. 클러스터링은 데이터 마이닝 분야에서 중요한 기술로 사용되고 있다. 만약 데이터 마이닝이 방대한 양의 데이터로부터 의미있는 지식을 획득하는 것으로 생각한다면, 클러스터링을 통해서 전체 데이터들을 몇 개의 그룹으로 묶을 것인가를 결정하는 것도 유용한 정보가 될 것이다 [1].

클러스터링 과정에서, 그룹핑 결과의 좋고 나쁨은 몇 개의 그룹으로 묶느냐에 따라 크게 좌우되므로 적당한 수의 그룹을 결정하는 것은 매우 중요한 이슈이다. 너무 많은 그룹으로 묶으면 복잡한 결과를 야기하여 해석하고 분석하는 것이 어렵다. 반면에, 너무 적은 그룹으로 묶으면 정보 손실을 야기하여 마지막 결론을 잘못 내릴 수가 있다. 몇 개의 그룹으로 클러스터링을 하는 것이 최적인가를 결정하는데 있어서 클러스터 적합성에 대한 측정은 필요불가결하다. 그래서 최적의 클러스터링 측정을 공식화하기 위한 수많은 노력들이 오랜 과거부터 현재까지 시도되어왔다 [2, 3].

최근에 Jung은 [4] 클러스터의 유효성을 측정하기 위한 척도로서 '클러스터링 균형' (Clustering balance)을 정의하였다. 여기서 클러스터링 균형은 클러스터 내부 유사성 (inter-cluster similarity)이 최대

화 되고 클러스터 사이 유사성 (inter-cluster similarity)이 최소화 되는 시점에서 최소값을 나타내는 특성으로 알려져 있다. 특정 클러스터 상황에서의 클러스터링 균형은 클러스터 내부 오차 (intra-cluster error)와 클러스터 사이 오차 (inter-cluster error)의 가중된 합으로 계산될 수 있다. 문헌 [4, 5]에서, 클러스터링 균형은 어떤 특정한 클러스터링 알고리즘에 대해 클러스터링 유효성을 위한 척도로써 성공적으로 사용되었다. 그러나 그 문헌들에서는 하나의 특정한 가중치에 대한 클러스터링 균형을 사용하여 최적의 클러스터 개수를 찾기 위해 소모적인 탐색방법 (exhaustive search)을 사용하였다. 본 연구에서 우리는 클러스터링 균형을 사용하여 최적의 클러스터 개수를 결정하기 위해 비소모적인 방법인 반복적인 탐색 공간 축소 방법 (Iterative search space reduction method)을 제안한다.

Ⅱ. 배경

클러스터링의 기본적인 목적은 주어진 입력 데이터들을 클러스터 내부 유사성을 최대화하고 클러스터 사이 유사성을 최소화하도록 특정한 척도 공간에서 몇 개의 그룹으로 분류하는 것이다. 유사성을 측정하기 위해 사용되는 가장 유명한 척도 방법은 데이터 포인트들 간의 유클리디안 거리 (Euclidean distance)이고 가장 자주 사용되는 평가 함수는 제곱오차평가 (squared error criterion)이다. 이 토대에 따라, 클러스터 내부 오차의 합 (intra-cluster error sum) Λ와 클러스터 사이 오차의 합 (inter-cluster error sum) Γ는 다음과 같이 정의된다.

$$\Lambda = \sum_{j=1}^{k}\sum_{i=1}^{n_j} ||p_i^{(j)} - p_0^{(j)}||_2^2, \qquad (1)$$

$$\Gamma = \sum_{j=1}^{k} ||p_0^{(j)} - p_0||_2^2. \qquad (2)$$

여기서, $P=\{p_1, p_2, \ldots, p_n\}$는 데이터의 집합을 나타내고 n은 총 데이터의 개수를 나타낸다. $C_j = \{p_1^{(j)}, p_2^{(j)}, \ldots, p_{n_j}^{(j)}\}$는 클러스터링 알고리즘에 의해 함께 묶여진 데이터 아이템들을 나타낸다. 여기서 n_j는 클러스터 C_j의 데이터 개수를 나타낸다. $p_0^{(j)}$는 클러스터 j의 중심을 나타내고, $p_0^{(j)} = \frac{1}{n_j}\sum_{i=1}^{n_j} p_i^{(j)}$이며 중심들의 집합은 $C=\{p_0^{(1)}, p_0^{(2)}, \ldots, p_0^{(k)}\}$로 나타낼 수 있고, 데이터 전체 중심은 $p_0 = \frac{1}{n}\sum_{i=1}^{n} p_i$로 표현하였다.

집괴적 군집화 기법 (agglomerative clustering)에서, 클러스터 내부 오차는 클러스터링 과정에서 점차 증가하고 반대로 클러스터 사이 오차는 감소한다. 반면에 분할적 군집화 기법 (divisive clustering)에서는 위와는 반대의 성향을 보인다. 이를 토대로, Jung은 [4] 특정한 클러스터링 상황 χ에 대한 클러스터링 균형을 다음과 같이 정의하였다.

$$\text{Clustering balance, } \epsilon(\chi) = \alpha\Lambda + (1-\alpha)\Gamma, \qquad (3)$$

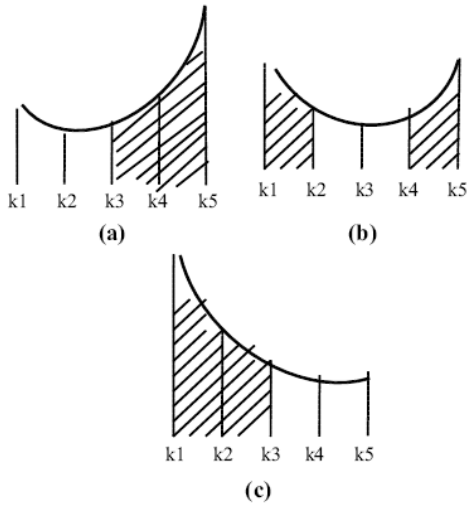
여기서 $0 \le \alpha \le 1$ 는 두 합의 관계를 나타내는 가중치이다.

Ⅲ. 반복적인 탐색 공간 축소

위의 유도과정으로부터, 우리는 어떠한 클러스터링 알고리즘을 사용하더라도 클러스터링 프로세스를 적당한 범위의 클러스터 개수에서 반복하면서 클러스터링 균형을 관찰하여 최소값에 이르는 점을 찾음으로써 최적의 클러스터 개수를 유추할 수 있다는 사실을 쉽게 발견할 수 있다. 많은 실험을 통해 우리는 클러스터링 균형이 클러스터 개수에 따라 대략적인 컨백스(convex) 형태를 가지고 있음을 알 수 있었다.

만약 탐색 공간이 대략적으로 컨백스 형태를 띠고 있고 찾아야 할 최소값을 포함하고 있다면, 탐색 공간을 4개의 구역 (5개의 점)으로 균등하게 나누고 그 점들 (그림 1 의 *k1*, *k2*, *k3*, *k4*, 그리고 *k5*)에서의 클러스터링 균형 값들을 비교함으로써 탐색 공간을 줄여나갈 수 있다. 점 x에서의 클러스터링 균형 값을 $Balance(x)$라 하자. 만약 $Balance(k2)$가 5개의 점 중에서 최소값이라면, 탐색 공간은 $[k1, k2]$로 줄일 수 있다. 같은 논리로, 만약 $Balance(k2)$,

$Balance(k3)$, $Balance(k4)$, 그리고 $Balance(k5)$가 5개의 점 중에서 최소값이라면, 탐색 공간은 각각 $[k1, k3]$, $[k2, k4]$, $[k3, k5]$, 그리고 $[k4, k5]$로 줄일 수 있다. 그러나 $Balance(k1)$과 $Balance(k5)$의 비교는 $Balance(k2)$과 $Balance(k4)$의 비교 결과가 그 내용을 포함하기 때문에 중복된다. 따라서 우리는 $Balance(k2)$, $Balance(k3)$, 그리고 $Balance(k4)$가 그들 중 최소값이 되는 그림 1 (a), (b), 그리고 (c)의 경우들만 고려하면 된다. 이를 토대로, 다음과 같은 반복적인 탐색 공간 축소를 수행함으로써 최적의 클러스터 개수를 찾기 위한 클러스터 프로세스의 반복 횟수를 줄일 수 있다.



▶▶ 그림 1. 최소값을 포함하고 있는 컨백스 그래프를 4개의 동일한 간격으로 나누었을 때의 3가지 경우

1. 탐색 공간을 5개의 점 *k1*, *k2*, *k3*, *k4*, 그리고 *k5*로 정확히 4등분 하고, 각 점에서 클러스터링 균형 값을 계산한다.

2. 그림 1 (a), (b), 그리고 (c)의 각 조건에 따라 빗금친 부분을 버림으로써 탐색 공간을 축소한다.

3. 1-2 과정을 원하는 탐색 공간만큼 축소 될 때까지 반복한다.

Algorithm 1은 반복적인 탐색 공간 축소 방법에 대한 슈도 코드 (pseudo-code)이다. 여기서 *int*는 의 소수점 이하를 버린 정수 값을 뜻한다. 반복적인 탐색 공간 축소를 실행하여 원하는 탐색 공간에 도달한 후, 줄여진 탐색 공간에서 클러스터링 균형의 전역 최소값을 발견하기 위하여 완전 탐색을 실행한다. 이 방법은 매우 간단하지만 뛰어난 계산 효율을 제공한다. 탐색 공간 축소를 실행할 때 마다 절반의 탐색 공간을 줄일 수 있으며, 단 두 번 (*k2*와 *k4* 점에서)의 클러스터링 균형 값을 구하기 위한 클러스터링 프로세스를 진행하면 된다. 따라서 제안하는 알고리즘은 n번의 클러스터링 프로세스 반복을 $\log_2 n$번으로 줄일 수 있을 것으로 기대한다.

Algorithm 1 *Iterative search space reduction*

```
begin
    k1 = 1;
    k3 = int(√n/2);
    k5 = int(√n);
    while (k5 − k1 > desired search space size) do
        k2 = int(k1 + (k5 − k1 + 1)/4);
        k4 = int(k1 + (k5 − k1 + 1) × 3/4) ;
        if (Balance(k2) < Balance(k3))
            k5 = k3 ;
            k3 = k2 ;
        else
            if (Balance(k3) < Balance(k4))
                k1 = k2 ;
                k5 = k4 ;
            else
                k1 = k3 ;
                k3 = k4 ;
    end while
end
```

Ⅳ. 실험

제안한 반복적인 탐색 공간 축소 방법의 효율성을 증명하기 위하여, 2개의 가상 클러스터 데이터 집합과 3개의 실제 데이터 집합을 사용하여 실험하였다. 클러스터링 알고리즘으로는 가장 유명한 k-means 알고리즘을 사용하였다.

1. 실험 데이터

1.1 가상 데이터 집합

2개의 가상 데이터 집합을 $Artificial\,1$과 $Artificial\,2$로 명칭하고, 다음과 같은 세부 사항으로 만들었다. 여기서 $N(\mu, \sigma)$는 평균 μ와 표준편차 σ를 나타내는 정규 변수를 뜻한다.

$Artificial\,1$: 데이터 집합은 클러스터의 중심들이 직선을 형성하는 2차원 정규 변수에 의해 만들어 졌다. 250개의 랜덤 변수들이 5개의 그룹으로 분류되어 있다.

5개의 그룹은 $N([0,0],[0.2,0.2])$, $N([-1,-1],[0.2,0.2])$, $N([1,1],[0.2,0.2])$, $N([2,2],[0.2,0.2])$, 그리고 $N([3,3],[0.2,0.2])$에 의해 만들어 졌으며 각 그룹들은 50개의 점들로 이루어져 있다.

Artificial 2: 데이터 집합은 클러스터의 중심들이 교차하는 형태의 2차원 정규 변수에 의해 만들어 졌다. 300개의 랜덤 변수들이 5개의 그룹으로 분류되어 있다. 그중 하나의 그룹은 $N([0,0],[0.2,0.2])$에 의해 만들어 졌으며 100개의 점들로 이루어져 있다. 나머지 그룹들은 $N([-2,0],[0.3,0.3])$, $N([2,0],[0.3,0.3])$, $N([0,2],[0.4,0.4])$, 그리고 $N([0,-2],[0.4,0.4])$에 의해 만들어 졌으며 각 그룹들은 50개의 점들로 이루어져 있다. 각 그룹 2차원 정규 변수의 표준편차는 다른 값을 가짐에 주의하라.

1.2. 실제 데이터 집합

4개의 실제 데이터 집합은 *Iris*, *Wine*, 그리고 *Yeast* 이다. 이 데이터 집합들은 UCI Machine Learning Repository [6]에서 참조하였다.

2. 실험 결과

고정된 k값에 대해 k-means 알고리즘을 10번 반복하였고, 그중 클러스터 내부 오차의 합이 가장 작은 클러스터 상황에서 클러스터링 균형 값을 계산하였다. k의 값은 $[2,n]$의 범위에서 변한다. 반복적인 탐색 공간 축소 방법에서, 탐색 공간 축소 후 원하는 탐색 공간의 크기를 10으로 정했다. 이것은 줄여진 공간의 크기가 10이하 일 때 전체 탐색을 한다는 의미이다. 이 변수는 컨백스 곡선의 거친 정도를 상쇄시킬 수 있는 중요한 역할을 담당하고 있다.

표 1은 클러스터링 균형에서 소모적인 방법과 제안하는 방법으로 구한 최적의 클러스터 개수를 비교하고 있다. 표 1의 비교 결과로부터, 본 논문에서 제안한 반복적인 탐색 공간 축소 방법은 클러스터링 균형 척도를 사용할 때 소모적인 방법으로 구한 결과와 동일한 결과를 제공함을 알 수 있다. 이 결과는 클러스터링 균형을 사용하여 최적의 클러스터 개수를 구하고자 할 때 정확도의 손실 없이 계산량을 줄일 수 있다는 것을 의미한다.

표 1. 클러스터링 균형의 척도를 사용하여 제안하는 방법과 소모적인 방법으로 발견한 최적의 클러스터 개수 비교

실험데이터	실제 클러스터 수	소모적인 방법	제안하는 방법
Artificial 1	5	5	5
Artificial 2	5	6	6
Iris	3	2	2
Wine	3	3	3
Yeast	10	12	12

V. 결론

본 연구는 클러스터링 균형을 사용하여 최적의 클러스터 개수를 결정하는데 있어서 반복적인 탐색 공간 축소 방법을 제안하였다. 2개의 가상 데이터 집합과 3개의 실제 데이터 집합에서 실험한 결과는 제안하는 방법이 정확도의 손실 없이 계산량을 현격히 줄여줌을 보여주었다.

■ 참 고 문 헌 ■

[1] T. Ishioka. An expansion of x-means for automatically determining the optimal number of clusters. In Proceedings of the Fourth IASTED International Conference, pages 91-96. Computational Intelligence, July 2005.[5] R. Xu and D. Wunsch. Survey of clustering. IEEE Trans. on Neural Networks, Vol. 16, No. 3, pp. 645-678, May 2005.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. ACM Computing Surveys, Vol. 31, No. 3, pp. 264-323, September 1999.

[3] A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs, NJ, 1988.

[4] Y. Jung, H. Park, D. Du, and B. L. Drake. A

decision criterion for the optimal number of clusters in hierarchical clustering. Journal of Global Optimization. Vol. 25, pp. 91-111. 2003.
[5] M. P. Jung, J. R. Broach, and C. A. Floudas. A novel clustering approach and prediction of optimal number of clusters: global optimum search with enhanced positioning. Journal of Global Optimization, Vol. 39, pp. 323-346, 2007.
[6] http://archive.ics.uci.edu/ml/