균등 격자를 이용한 공간 클러스터링 기법의 성능 평가

문상호
부산외국어대학교 컴퓨터공학부

Performance Evaluation of Spatial Clustering Method using Regular Grid

Sang-Ho Moon
Division of Computer Engineering, Pusan University of Foreign Studies
E-mail : shmoon87@pufs.ac.kr

요 약

본 논문에서는 기존 연구에서 제시된 균등 격자를 이용한 공간 클러스터링 기법의 효율성을 검증하기 위한 성능 평가를 수행한다. 세부적으로 다양한 분포 형태를 가지는 실험데이타들을 대상으로 먼저 객체 수의 변화에 따른 수행 시간을 비교한다. 그리고 동일한 실험데이타를 대상으로 임계값의 변화에 따른 실험 평가를 수행한다. 또한, 각 실험 결과에 대하여 전체 수행 시간을 기준으로 클러스터 생성 알고리즘과 클러스터 합병 알고리즘에 대한 상대적인 비교를 평가한다.

ABSTRACT

In this paper, experimental tests are performed to evaluate the efficiency of spatial clustering method using regular grid that is proposed in our recent research. In details, we estimate the execution time for finding clusters varying spatial objects on sample data sets with various distributions and perform experimental tests varying threshold value on a data set. We also compare the running time of cluster generating algorithm with that of cluster merging algorithm per each test.

키워드
공간 클러스터링, 균등 격자, 공간데이타 마이닝, 성능 평가

Ⅰ. 서 론

공간 데이터 마이닝을 위하여 제시된 기존 공간 클러스터링 기법들[1, 2, 3, 4]은 대부분이 객체들의 거리 계산을 기반으로 하므로 데이타 양이 많아질수록 비용이 커진다. 또한, 메모리 상주 데이타를 대상으로 하므로 대용량의 데이타인 경우에 효율이 떨어진다. 이러한 문제점을 해결하기 위하여 기존 연구에서 균등 격자(regular grid)를 이용한 공간 클러스터링 기법을 제시하였다[5, 6]. 이 기법에서는 방대한 양의 공간데이타를 대상으로 효율적인 클러스터링을 위하여 계산 비용 감소에 중점을 둔다. 세부적으로 객체간의 거리 계산을 대체하여 균등 격자의 관련성을 이용하여 클러스터링을 수행한다.

균등 격자를 이용한 공간 클러스터링 기법은 세부적으로 클러스터 생성 알고리즘과 클러스터 합병 알고리즘으로 구성된다. 먼저 클러스터 생성 알고리즘은 셀 관련성을 이용하여 후보 클러스터들을 생성한다. 그리고 클러스터 합병 알고리즘은 전단계에서 생성된 후보 클러스터들에 대하여 합병 가능 여부를 판단하여 합병한 후에 최종 클러스터들을 생성한다.

본 논문에서는 기존 연구에서 제시된 균등 격자를 이용한 공간 클러스터링 기법의 효율성을 검증하기 위한 성능 평가를 수행한다. 기본적으로 클러스터 생성 알고리즘은 균등 격자를 구성하는 셀 수에 영향을 받고, 클러스터 합병 알고리즘은 공간객체의 수에 영향을 받는다. 따라서 공간 클러스터링 기법의 성능 평가를 위하여 공간객체의 수, 전체 셀의 수 등에 따른 다양한 실험들이 수행되어야 한다. 이를 위하여, 본 논문에서는 세부적으로 다양한 분포 형태를 가지는 실험데이타들을 대상으로

먼저 객체 수의 변화에 따른 수행 시간을 비교한다. 그리고 동일한 실험데이타를 대상으로 임계값의 변화에 따른 실험 평가를 수행한다. 또한, 각 실험 결과에 대하여 전체 수행 시간을 기준으로 클러스터 생성 알고리즘과 클러스터 합병 알고리즘에 대한 상대적인 비중을 평가한다.

II. 성능 평가

2.1 실험데이타

먼저 본 논문의 성능 평가를 위한 실험 데이타는 각 축의 데이터 분포를 기준으로 생성하였으며, 실험의 공정성을 위하여 [7]의 Scholl Benchmark 데이타 집합으로부터 생성된 MBR 데이타의 좌표점을 객체의 위치로 하는 데이타를 생성하였다. 즉, 클러스터링을 위해 사용한 데이타는 점 객체를 대상으로 하였으며, 실험데이타에 대한 특성은 표 1과 같다. 여기서 DS1, DS2, DS3과 DS4는 데이터 분포를 다양하게 하였으며, 또한 데이터 객체 수는 차이가 난다. 전체적인 실험은 Pentium4 2GHz, 메모리 512MB와 Windows XP 운영체제를 탑재한 PC상에서 수행하였다.

표 1. 실험데이타의 특성

실험 데이타	X축범위	Y축범위	X축 분포	Y축 분포	객체수
DS1	0-10,000	0-10,000	gaussian	exponential	4000
DS2	0-10,000	0-10,000	exponential	gaussian	8000
DS3	0-10,000	0-10,000	gaussian	gaussian	12000
DS4	0-10,000	0-10,000	gaussian	gaussian	16000

2.2 실험데이타에 대한 성능 평가

첫 번째 실험은 앞에서 기술한 실험데이타를 대상으로 공간 클러스터링 기법을 적용하였다. 실험에 앞서, 격자구조를 생성하기 위하여 각 실험데이타에 대하여 임계값을 500과 1000으로 설정하였다. 임계값에 따라 생성된 격자구조의 크기(전체 셀 수)는 표 2에서 보여준다. 동일한 실험데이타에 대하여 기본적으로 임계값이 작아질수록 전체 셀의 수는 증가하게 된다. 그리고 임계값 500은 1000에 비하여 격자구조의 가로축과 세로축에 대하여 각각 2배의 차이가 나므로, 전체 셀의 수는 4배가 증가하게 된다.

표 2. 임계값에 따른 격자구조 크기

임계값 / 데이타	1000		500	
	가로*세로	전체 셀 수	가로*세로	전체 셀 수
DS1	53*53	2809	107*106	11342
DS2	108*53	5724	216*106	22896
DS3	108*107	11556	216*215	46440
DS4	108*107	11556	216*215	46440

각 실험데이타에 대하여 격자구조를 생성한 후에 공간 클러스터링 알고리즘을 적용하여 수행한 실험 결과는 그림 1과 같다. DS1에서 DS4로 갈수록 수행시간이 많이 걸리는 이유는 객체 수와 전체 셀 수가 많기 때문이다. 그리고 DS3과 DS4는 전체 셀 수는 동일하지만 수행시간은 DS4가 더 걸린다. 이것은 비록 셀 수는 같지만 공간객체 수에서 D4가 더 많기 때문에 수행시간이 많이 걸린다. 세부적으로 살펴보면, 후보 클러스터의 객체들을 대상으로 수행되는 클러스터 합병 알고리즘의 수행시간에서 차이가 남을 알 수 있다.

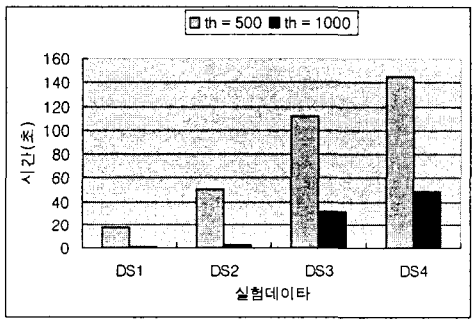

그림 1. 각 실험데이타에 대한 수행시간 비교

전체 수행시간을 기준으로 클러스터 생성 알고리즘과 클러스터 합병 알고리즘의 비중을 비교하면, 각 알고리즘의 세부적인 비교 분석이 가능하다. 이를 위하여, 본 논문에서는 각 실험데이타를 대상으로 임계값을 500으로 설정한 후에 클러스터링 알고리즘을 적용한 전체 수행 시간에서 클러스터 생성 알고리즘이 차지하는 비중을 그림 2에서 보여준다. 이 결과를 분석하면, 수행시간 관점에서 전체적으로 클러스터 생성 알고리즘이 차지하는 비중은 0,012% 이하로 매우 적은 편이다. 따라서 클러스터 생성 알고리즘은 클러스터 합병 알고리즘에 비하여 매우 빠름을 알 수 있다. 또한, DS3와 DS4가 DS1에 비하여 전체 셀 수가 많음에도 불구하고 비중이 적음을 알 수 있다. 이것은 비록 셀 수의 증가에 따른 클러스터 생성 알고리즘의수행시간이 증가하였지만, 객체 수의 증가에 따른 클러스터 합병 알고리즘의 수행시간이 더 증가하므로 상대적으로 비중이 줄어들었기 때문이다.

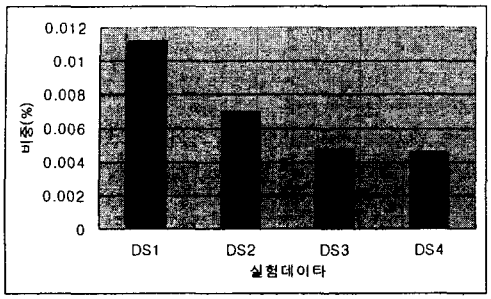

그림 2. 클러스터 생성 알고리즘의 비중 비교

2.3 임계값 변화에 따른 성능 평가

이번 실험에서는 객체의 수에 따라 임계값을 변화시키면서 실험을 수행하였으며, 여기서는 데이터 양이 많은 DS3과 DS4를 대상으로 하였다. 표 1에서 기술한 바와 같이, 공간객체의 수는 DS 3인 경우에는 12000개, DS 4인 경우에는 16000개를 가지고 있으며, 임계값은 3500, 3000, 2500, 2000 값으로 각각 변화를 주었다. 격자 구조는 임계값이 작을수록 전체적인 셀들의 수는 많아지게 된다. 임계값 변화에 따른 DS3과 DS4에 대한 전체적인 성능 평가에 대한 결과는 표 3, 표 4와 같다.

표 3. DS3에 대한 성능 평가

(단위 : 초)

임계값	GridX	GridY	셀 수	생성 시간	합병 시간	전체 시간
3500	30	30	900	0.000192	0.070219	0.070411
3000	36	35	1260	0.000278	0.127943	0.128221
2500	43	43	1849	0.000377	0.24502	0.245397
2000	54	53	2862	0.000549	0.516933	0.517482

표 4. DS4에 대한 성능 평가

(단위 : 초)

임계값	GridX	GridY	셀 수	생성 시간	합병 시간	전체 시간
3500	30	30	900	0.000284	0.135395	0.135679
3000	36	35	1260	0.000368	0.22654	0.226908
2500	43	43	1849	0.000504	0.45118	0.451684
2000	54	53	2862	0.000753	0.946074	0.946827

DS3과 DS 4를 대상으로 임계값 변화에 따라 증가된 전체 셀들을 대상으로 한 클러스터링 알고리즘의 수행 시간은 그림 3과 같다. DS4가 DS 3에 비하여 시간이 증가하는 이유는 객체 수가 많기 때문이다. 실험 결과를 분석해 보면, 클러스터 생성 알고리즘과 합병 알고리즘은 전체 셀의 수에 따라 비례하여 증가하며, 또한 객체들의 수에 따라 비례하여 증가함을 알 수 있다. 여기서 주목할 것은 생성 알고리즘이 합병 알고리즘에 비하여 상대적으로 시간이 적게 들며, DS3과 DS4의 생성 시간을 비교하더라도 큰 차이가 없다. 이것은 생성 알고리즘이 셀들을 대상으로 셀 관계 연산만 수행하기 때문이다. 반면에 합병 알고리즘은 실제 객체들에 대한 연산을 수행하므로 DS4가 DS3에 비하여 시간이 증가한다. 또한 셀들 내에 있는 객체들을 대상으로 하므로 전체 셀의 수에 비례하여 시간이 증가한다.

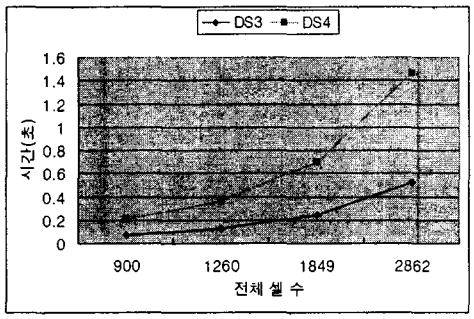

그림 2. 임계값 변화에 따른 수행시간 비교

2.4 종합 검토

성능 평가를 검토해보면 전체 알고리즘 수행에서 클러스터 합병 알고리즘의 수행 시간이 차지하는 비중이 매우 크다. 이것은 상대적으로 클러스터 생성 알고리즘의 수행 시간이 적게 걸린다는 것을 의미한다. 즉, 전체적인 클러스터링 과정에서 셀 연산을 통하여 시간을 많이 줄였음을 알 수 있다. 그리고 클러스터 합병 알고리즘에서 실제 거리 계산의 대상이 되는 객체들은 클러스터링 가능셀에 포함된 것만을 대상으로 하기 때문에, 계산 시간이 많이 빨라졌음을 알 수 있다. 이것은 그림 2의 실험 결과에서 임계값에 따라 결과는 다소 차이가 있지만, DS3은 12000개 객체들을 대상으로 전체적인 클러스터링 수행 시간이 약 0.52초 미만, DS4는 16000개 객체들을 대상으로 약 0.95초 미만이 걸린 것을 통하여 확인할 수 있다.

III. 결 론

본 논문에서는 기존 연구에서 제시한 균등 격자를 이용한 공간 클러스터링 기법의 효율성을 입증하기 위한 성능 평가를 수행하였다. 세부적으로 다양한 분포 형태와 다른 공간객체의 수를 가지는 실험데이타를 대상으로 평가를 수행하였으며, 또한 임계값의 변화에 따른 실험도 수행하였다. 실험 결과를 통하여 전체적으로 공간 클러스터링의 성능은 우수함을 알 수 있었고, 세부적으로 클러스터 생성 알고리즘이 클러스터 합병 알고리즘에 비하여 매우 빠르게 수행되었다. 앞으로 향후 연구에서 클러스터 합병 알고리즘에 대한 성능 개선이 필요하고, DBSCAN, CLARANS 등과 같은 기존의 클

러스터링 기법들과의 성능 평가를 통하여 우수성을 입증해야 한다.

참고 문헌

[1] Ng and J. Han, "Efficient and Effective Clustering Method for Spatial Data Mining", Int. Conf. on VLDB, pp.144~155, 1994.

[2] M. Ester, H.P. Kriegel, J. Sander, and X. Xu., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Int. Conf. on KDD, pp.226~231, 1996.

[3] W. Wang, J. Yang and R. Muntz, "STING: A Statistical Information Grid Approach to Spatial Data Mining", Int'l Conf. on VLDB, pp.186-195, 1997.

[4] 오병우,한기준, "H-SCAN: 지식 추출을 위한 해시-기반 공간 클러스터링 알고리즘", 한국정보과학회 논문지, 26권 7호, pp.857~869, 1999.

[5] 문상호, 이동규, 서영덕, "공간데이타 마이닝을 위한 효율적인 그리드 셀 기반 공간 클러스터링 알고리즘", 정보처리학회논문지, 10-D권, 4호, 2003.

[6] 문상호, "균등 격자를 이용한 공간 클러스터링 기법의 설계 및 구현", 한국해양정보통신학회 춘계 종합학술대회논문집, 2003.

[7] Spatial Join Benchmarking (http://www.enst.fr/~bdtest/sigbench/index.html)