

초고차원에서 거리기반의 공간 변환을 위한 유전자 알고리즘을 활용한 데이터 독립적인 빈티지 포인트 생성 기법

김상근⁰, 정성원
서강대학교 컴퓨터공학과

Data Independent Vantage Point Selection method for Distance-Based Space Transformation in Very High Dimensional Space using Genetic Algorithm

Sangkeun Kim⁰, Sungwon Jung
Department of Computer Science and Engineering, Sogang University

요약

빈티지 포인트에 의한 공간 변환은 고차원 데이터 공간을 유사 질의어(Similarity query) 처리에 적합한 거리 기반 벡터 공간으로 변환시킨다. 기존에 제안된 빈티지 포인트 선택 알고리즘들은 데이터 객체 중에서 빈티지 포인트를 선택하기 때문에 업데이트가 빈번한 동적 환경에 적용하기 어렵다. 이 문제를 해결하기 위하여 코너 포인트에서 빈티지 포인트를 선택하는 탐욕적 알고리즘 기반의 연구가 있었다. 그러나 이 방법은 차원 수에 따라 코너 포인트가 지수 함수적으로 증가하기 때문에 고차원 데이터를 다루기에 적합하지 않아 고차원 데이터에서도 빈티지 포인트를 생성할 수 있는 효과적인 휴리스틱 알고리즘이 필요하다. 따라서 본 논문에서는 메타 휴리스틱 기법 중에 하나인 유전자 알고리즘을 활용하여, 고차원 데이터에서도 적용 가능한 향상된 성능의 빈티지 포인트를 생성하는 기법을 제시한다.

1. 서 론

최근 유사 질의어를 이용한 응용분야에서 데이터베이스의 크기가 증가하고 데이터 객체가 갖는 속성 값이 다양해짐에 따라서 효과적인 유사 질의어 처리를 위한 기술 개발이 요구되고 있다.

빈티지 포인트는 고차원 데이터 공간을 유사 질의어 처리에 적합한 거리 기반 벡터 공간으로 공간 변환하는데 쓰인다. 기존의 연구에서 빈티지 포인트들은 데이터 객체 중에서 빈티지 포인트들을 선택하기 때문에, 업데이트가 빈번한 동적 환경에 적용하기 어렵고, 이를 해결하기 위하여 데이터 독립적인 방법으로 빈티지 포인트를 생성하는 연구가 있었다[1]. 해당 연구에서 데이터들은 닫힌 공간(Closed Space)에 존재한다고 가정되며, 각 차원의 좌표 축에 위치한 꼭지점들을 코너 포인트라 한다. 코너 포인트는 실제 데이터 객체들보다 더 효과적인 빈티지 포인트로 간주된다[1,2]. 코너 포인트를 활용한 빈티지 포인트의 선택에서 중요한 부분은 선택된 빈티지 포인트들이 공간상에서 얼마나 고르게 분포하고 있는가이다. 탐색의 대상이 되는 모든 코너 포인트들을 탐색하는 [1]의 방법으로는 차원수가 증가할 때, 코너 포인트의 수가 지수 함수적으로 증가하기 때문에 고차원 공간의 데이터를 다루기에는 적합하지 않다. 따라서 본 논문에서는 고차원 데이터에 적용 가능한 데이터 독립적인 빈티지 포인트 선택 기법으로 메타 휴리스틱 기법중에 하나인 유전자 알고리즘을 활용하여 효율적인 세대 교체를 통한

최적의 빈티지 포인트를 생성하여 이 문제를 해결하고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 이 문제를 유전자 알고리즘으로 해결할 수 있도록 모델링을 하고, 3장에서 효율적으로 세대 교체하는 방법을 제시한다. 4장에서는 제안하는 방법과 기존 연구의 성능을 비교 평가하고, 마지막 5장에서는 결론을 맺도록 한다.

2. 유전자 알고리즘을 위한 모델링

2.1 빈티지 포인트의 최적 분포

빈티지 포인트들이 공간상에서 얼마나 고르게 분포하고 있는지를 측정하기 위하여 빈티지 포인트들의 상호간 거리(pairwise distance)를 이용하여 편향성(skewness)을 측정한다. 만약 전체 공간의 크기에 비해 너무 작은 거리를 갖거나 너무 큰 거리를 갖는 빈티지 포인트 쌍이 존재한다면 이는 고르게 분포되어 있지 않다[1]. 따라서 우리는 공간상에서 발생할 수 있는 상호간 거리의 평균값인 $\sqrt{\frac{n}{2}}$ 을 기준으로 하여 최대한 편차가 적은 상호간 거리를 갖도록 빈티지 포인트를 찾아낸다.

정의 1. 매트릭 공간의 차원 수를 n 이라고 하고, 빈티지 포인트의 수를 m 이라고 하자. n 차원에서 m 개의 빈티지 포인트 집합 $V = \{v_1, \dots, v_m\}$ 이 정의되었다고 할 때, 모든 빈티지 포인트 간의 거리들의 $\sqrt{\frac{n}{2}}$ 에 대한 절대 편차

$$D = \sum_{i=1}^{m-1} \sum_{j=i+1}^m \left| \sqrt{\frac{n}{2}} - Euclidian_distance(v_i, v_j) \right|$$

가 최소화 되는 분포 상태를 빈티지 포인트의 최적 분포 상태라 정의한다.

2.2 유전자 알고리즘을 위한 용어 정의

최적 분포 상태의 빈티지 포인트를 생성하기 위해서 휴리스틱 기법중에 하나인 유전자 알고리즘을 활용한다. 그 전에 용어를 정의한다.

정의 2. n 차원에서 m 개의 빈티지 포인트 집합 $V = \{v_1, \dots, v_m\}$ 를 크기가 m 인 유전자 g 라 정의하고, k 개의 유전자 집합 $G = \{g_1, \dots, g_k\}$ 라고 정의한다. [그림 1]은 $k=4$, $n=6$, $m=6$ 일 때의 유전자 집합 G 를 나타낸다.

$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$
$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$

[그림 1] $k=4$, $n=6$, $m=6$ 일 때의 유전자 집합 G

[그림 1]에서는 4개의 행렬로 크기가 6인 4개의 유전자를 보여주고 있으며 각 행렬의 6개의 행은 6차원의 빈티지 포인트 6개를 나타낸다. 생성할 빈티지 포인트는 각 차원의 코너 포인트에서 선택하므로 0과 1로 구성되어 진다.

유전자 알고리즘을 수행하기에 앞서, 초기 유전자 집합 G 를 생성한다. 세대 교체를 진행할 때, 한번의 세대에서 Selection, Crossover, Mutation 세가지 연산이 수행되고, 원하는 성능의 최적 분포에 근접했을 때, 해당하는 세대에서 종료한다.

유전자의 성능을 평가할 수 있는 적합도 함수 $fit(x)$ 는 빈티지 포인트 집합이 최적 분포 상태에 가까울수록 우수한 해로 평가되므로 절대편차 D 의 역수로 다음 식 (1)과 같다.

$$fit(x) = \frac{1}{\sum_{i=1}^{m-1} \sum_{j=i+1}^m \left| \sqrt{\frac{n}{2}} - Euclidian_distance(v_i, v_j) \right|} \dots \dots (1)$$

3. 최적 분포를 위한 유전자 알고리즘

3.1 효율적인 세대 교체 방법

유전자 알고리즘의 성능을 향상시키기 위해서는 효율적인 세대 교체 방법이 필요하다. 따라서 Selection, Crossover, Mutation 세가지 연산의 방법을 제시한다.

3.1.1 Selection

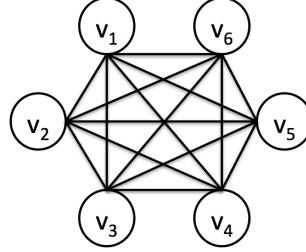
각 세대별 우수한 해는 반드시 생존할 수 있도록 하는 엘리트 보존전략을 적용하였고, 교차할 유전자가 남아있을 때까지 탐욕적 알고리즘 방법으로 항상 적합도가 높은 두 유전자를 선별한다.

3.1.2 Crossover

교자는 두 부모가 갖는 유전자를 조합하여 자손을 생성하는 과정으로 Selection된 두 유전자 g_1, g_2 의 적절한 교차점을 선정하는 것이 가장 중요하다. 하나의 유전자에서 적절한 교차점

을 찾는 방법은 유전자에 있는 빈티지 포인트들을 두 집합 S_1, S_2 로 나누는데, S_1, S_2 각각의 적합도가 가장 좋게 분할하는 것이다.

먼저 유전자 g_1 을 고려해보면 g_1 의 빈티지 포인트들을 점점으로 하고, 빈티지 포인트들간의 거리들의 $\sqrt{\frac{n}{2}}$ 에 대한 절대 편차를 가중치로 하는 complete graph를 만들 수 있다. 이 그래프를 가질 수 있는 최적의 적합도를 갖는 두 집합으로 분할하기 위해서 다음의 특성을 정의한다.



	v1	v2	v3	v4	v5	v6
v1	-	0.00	0.27	0.00	0.72	0.00
v2	0.00	-	0.00	0.27	0.00	0.72
v3	0.27	0.00	-	0.00	0.32	0.00
v4	0.00	0.27	0.00	-	0.00	0.32
v5	0.72	0.00	0.32	0.00	-	0.00
v6	0.00	0.72	0.00	0.32	0.00	-

[그림 2] [그림 1]의 유전자로 구성한 complete graph

정리 1. 유전자 g 가 나타내는 빈티지 포인트 집합을 두 집합 S_1, S_2 로 분할할 때, 유전자 g 로 구성한 Complete graph에서 maximum cut 으로 분할한 두 집합 S_1, S_2 의 적합도 평가 $fit(S_1) + fit(S_2)$ 가 가장 높다.

(증명) 유전자 g 에서 만든 두 집합을 $S = \{x_1, x_2, \dots, x_s\}$, $T = \{y_1, y_2, \dots, y_t\}$ 라고 하자. 먼저 함수 $f(x_i, y_j)$ 를 두 점 x_i 와 y_j 사이 거리의 $\sqrt{\frac{n}{2}}$ 에 대한 절대 편차로 정의한다.

$$f(x_i, y_j) = \left| \sqrt{\frac{n}{2}} - Euclidian_distance(x_i, y_j) \right|$$

또한, 집합 S 의 절대 편차를 D_s 라 하고 집합 T 의 절대 편차를 D_t 라 했을 때 나누기 전의 편차 D는 다음 식과 같다.

$$D = D_s + D_t + f(x_1, y_1) + f(x_1, y_2) + \dots + f(x_s, y_t)$$

편차 D는 일정하므로 $D_s + D_t$ 의 크기를 낮추기 위해서는 $f(x_1, y_1) + f(x_1, y_2) + \dots + f(x_s, y_t)$ 의 크기가 커야 한다. 따라서 maximum cut으로 분할한 두 집합 S_1, S_2 의 적합도 평가가 가장 높음을 알 수 있다.

이 특성을 이용해서 유전자 g_1 을 두 집합 S_1, S_2 로 분할하고, 유전자 g_2 를 두 집합 T_1, T_2 로 분할한다. 이제 분할 된 두 집합을 교차시켜서 새로운 유전자 g_3 과 g_4 를 만드는데, maximum cut을 해서 분할된 두 집합의 크기는 일정하지 않다. 따라서 S 집합과 T 집합의 크기가 다를 수 있으므로 항상 교차할 수는 없다. 이 때, 항상 교차할 수 있도록 원하는 집합의 크기로 분할하기 위하여 다음의 특성을 정리한다.

정리 2. 크기가 m 인 유전자 g 는 $m = |S| + |T|$ 를 만족하는 크기가 $|S|$ 와 $|T|$ 인 유전자 g_1 과 g_2 로 최대 $\min(|S| + |T|)$ 번 maximum cut를 적용하여 구할수 있다.

(증명) 유전자 g 의 총 빈티지 포인트의 개수를 V 라고 하고, 분할 할 집합의 크기를 S, T 라고 하자. 그래프를 구성하고

maximum cut을 유한번해서 분할 한다면, 최종적으로 더이상 분할 할 수 없는 상태의 집합은 하나의 빈티지 포인트를 가진 집합이다. 더이상 분할 할 수 없는 상태의 집합의 크기를 전부 더하면 총 빈티지 포인트의 개수 V 이고, 이는 S or T 보다 크므로 항상 원하는 크기의 집합으로 분할 할 수 있다.

따라서 Selection된 두 유전자 g_1, g_2 를 maximum cut을 이용하여 같은 크기의 집합으로 분할 한 다음 교차하여 다음 세대의 유전자를 만든다.

3.1.3 Mutation

돌연변이는 개체에 새로운 유전자를 추가하는 것으로, 한 개체에서 아주 작은 확률로 임의의 유전자를 변형하는 과정이다. 교차 후 한 유전자 안에서 같은 빈티지 포인트가 존재할 때 임의의 하나의 비트를 flip시켜주고, 아주작은 확률로 임의의 비트를 flip시키도록 적용하였다.

3.2 유전자 알고리즘의 의사 코드

3.1에서 제시한 효율적인 세대 교체의 방법을 적용한 유전자 알고리즘의 의사 코드는 [그림 3]과 같다. 먼저 초기 유전자 집합을 생성한다. 매 세대마다 적합도 평가를 우선순위로 하는 최소 힙을 구성한 뒤, Selection을 수행한다. 선별된 두 유전자를 이용하여 각각 완전 그래프를 만들고, [4]에서 제시된 maximum cut 알고리즘을 이용하여 분할한 다음 교차한다. 그리고 세대 교체가 모두 종료하면 매우 적은 확률로 mutation 을 수행한다.

```

Algorithm make vantage points using genetic algorithm
Input : number of dimension n and number of vantage points m
Output : return gene sets S
1. Initial gene sets S ← φ
2. Add new gene to S
   for i = 0 to INIT_SET_SIZE do
     S = S ∪ make_new_gene_set()
   end for
3. for i = 0 to MAX_GENERATION do
4.   min heap PQ ← φ based on function f
5.   Insert S to PQ
6.   Clear set S
7.   for 10% of INIT_SET_SIZE do
8.     let p as gene
9.     p = PQ.pop()
10.    insert p to S for next generation
11.  end for
12.  for not PQ.empty() do
13.    let p1,p2 as gene
14.    p1 = PQ.pop()
15.    p2 = PQ.pop()
16.    Create complete graph g1,g2 such that vertex is vantage points of
p1,p2 and weight is each  $\sqrt{\frac{n}{2} - \text{pairwise distance}}$ 
17.    V11,V12 is vertex set returned by MAX_CUT
18.    V11,V12 = MAX_CUT(g1)
19.    V21,V22 is vertex set returned by MAX_CUT
20.    V21,V22 = MAX_CUT(g2)
21.    V11,V12 and V21,V22 cross over each other.
22.    if f(p1) < f(p2) then
23.      insert p1 to S for next generation
24.      insert p2 to PQ
25.    else
26.      insert p2 to S for next generation
27.      insert p1 to PQ
28.    end if
29.  end for
30.  produces mutation in a very small probability.
31. end for
32. return S

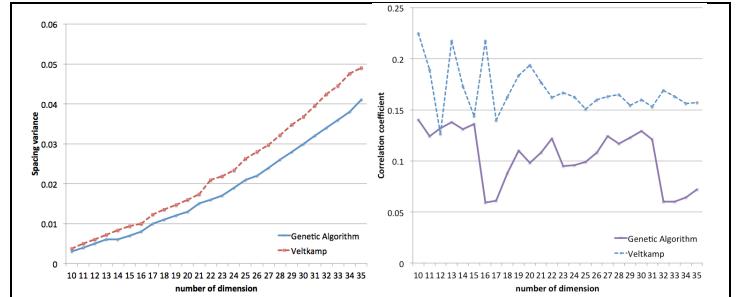
```

[그림 3] 빈티지 포인트 생성 유전자 알고리즘

4 성능 평가

이 장에서는 제안한 알고리즘의 성능을 평가하기 위하여

[1,3]의 correlation coefficient와 spacing variance를 기준으로 측정한다. Spacing variance는 공간 변환 후 거리의 편차를 나타내고, correlation coefficient는 데이터 객체들과 빈티지 포인트 쌍의 거리의 유사도를 나타내기 때문에, 모두 0에 가까울수록 더 좋은 분포를 나타냄을 알 수 있다. 비교는 균등 분포를 갖는 10만개의 데이터 셋으로 실험하였다.



[그림 4] 차원 수의 증가에 따른 m개의 빈티지 포인트의 Spacing variance와 Correlation coefficient

[그림 4]는 제안된 기법의 빈티지 포인트와 [3]의 데이터 기반 빈티지 포인트의 측정 결과이다. 실험 결과는 두 측정 모두 제안된 기법의 빈티지 포인트가 우수함을 볼 수 있다.

5 결론

본 논문에서는 효율적인 세대 교체를 통한 유전자 알고리즘을 활용하여 최적의 빈티지 포인트 선택 방법을 초고차원 데이터에 적용하는 방법을 제안하였으며, 실험을 통하여 우수한 성능을 나타냄을 보였다. 그러나 제안된 방법은 초기 모집단을 구성하는데 많은 부분 임의생성 방법으로 만들기 때문에 임의성이 강하고 maximum cut 자체가 NP-complete 문제이므로 complete graph를 구성할 때 정점, 즉 생성하고자 하는 빈티지 포인트의 개수가 많아질수록 교차시간이 크게 증가하므로 탐색시간이 길어진다. 이는 향후의 연구에서 보완해야 할 문제이다.

"본 연구는 미래창조과학부 및 정보통신산업진흥원의 서울어코드활성화지원 사업의 연구결과로 수행되었음" (NIPA-H1807-14-1011)

참고문헌

- [1] S. Pramanik, A. Watve, S. Jung, and C. Lim, "Database Independent Vantage Point Selection for Range Queries", Technical Report, MSU, 2014.
- [2] T. Bozkaya and M. Ozsoyoglu, "Distance-based indexing for high-dimensional metric spaces". In Proceedings of the 1997 ACM SIGMOD international conference on Management of data, SIGMOD '97, pages 357-368, New York, NY, USA, 1997. ACM.
- [3] R. H. Van Leuken and R. C. Veltkamp, "Selecting vantage objects for similarity indexing", ACM Trans. Multimedia Comput. Commun. Appl., 7:16:1-16:18, September 2011.
- [4] Michel X. Goemans and David P. Williamson, "Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming", Journal of the ACM., 42:6:1115-6:1145, Nov 1995.