



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0008461  
(43) 공개일자 2015년01월22일

(51) 국제특허분류(Int. Cl.)  
G06F 17/30 (2006.01)  
(21) 출원번호 10-2014-7034418  
(22) 출원일자(국제) 2013년05월09일  
심사청구일자 없음  
(85) 번역문제출일자 2014년12월08일  
(86) 국제출원번호 PCT/US2013/040248  
(87) 국제공개번호 WO 2013/169968  
국제공개일자 2013년11월14일  
(30) 우선권주장  
61/644,519 2012년05월09일 미국(US)

(71) 출원인  
툼슨 라이센싱  
프랑스 92130 이씨레물리노 잔 다르크 뒤편 1-5  
(72) 발명자  
이오애니디스 에프스트라티오스  
미국 94105 캘리포니아주 샌프란시스코 넘버307  
뉴 몽고메리 스트리트 99  
마쭈리, 로렐  
프랑스 에프-92420 보크레송 비스 뒤편 제르멘느 루  
이즈 5  
(74) 대리인  
양영준, 전경석, 백만기

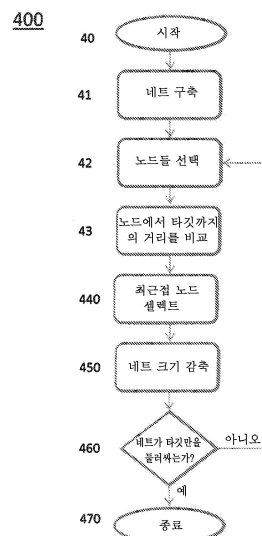
전체 청구항 수 : 총 24 항

(54) 발명의 명칭 비교 기반 활성 검색/학습

(57) 요약

방법이 비교를 통한 콘텐츠 검색을 수행하기 위해 제공되며, 여기서 사용자가 2개의 후보 오브젝트를 제시받으며 또한 사용자의 의도한 타깃 오브젝트에 더 가까운 것을 드러낸다. 개시된 원리들은 소수의 비교로 사용자의 타깃을 찾기 위한 활성 전략들을 제공한다. 잡음 없는 사용자 피드백을 위한 소위 랭크 넷 전략이 기술된다. 한정된 배가 상수를 가진 타깃 분포에 대해, 랭크 넷트는 타깃 분포의 및 따라서 최적 조건의 엔트로피에 가까운 소정 회수의 단계들로 타깃을 발견한다. 잡음성 사용자 피드백의 경우가 또한 고려된다. 해당 맥락에서 랭크 넷트들의 변형이 또한 기술되는데, 이에 대해 최적 조건의 천천히 증가하는 함수(이중 대수적) 내의 성능 한도들이 발견된다. 영화 데이터집합들에 대한 수치적 평가들은 랭크 넷트가 더 작은 계산 비용을 초래하면서 일반화된 이전 검색의 검색 효율성과 일치한다는 것을 보여준다.

대표도 - 도4



## 특허청구의 범위

### 청구항 1

데이터베이스 내의 타깃을 검색하기 위한 방법으로서,  
적어도 하나의 타깃을 둘러싸는 크기를 갖는 노드들의 네트를 구축하는 단계;  
상기 네트 내의 노드들의 집합을 선택(choose)하는 단계;  
타깃으로부터 상기 노드들의 집합 내의 각각의 노드까지의 거리를 비교하는 단계;  
상기 비교 단계에 따라서 상기 타깃에 최근접한, 상기 노드들의 집합 내의 노드를 선택(select)하는 단계;  
상기 선택 단계에 따라서 상기 타깃을 여전히 둘러싸는 크기로 상기 네트를 감축하는 단계;  
상기 네트의 크기가 상기 타깃만을 둘러싸도록 충분히 작아질 때까지 상기 선택, 비교, 선택, 및 감축 단계들을 반복하는 단계  
를 포함하는, 데이터베이스 내의 타깃 검색 방법.

### 청구항 2

제1항에 있어서, 상기 감축 단계는 상기 네트를 감축시켜서 상기 네트가 상기 타깃에 최근접한 상기 노드상에 중심을 두고 또한 상기 네트가 상기 타깃에 이르는 상기 최근접 노드의 거리보다 크지 않은 반경을 갖도록 하는, 데이터베이스 내의 타깃 검색 방법.

### 청구항 3

제2항에 있어서, 상기 네트는 보로노이 셀(Voronoi cell)에 의해 정의되는, 데이터베이스 내의 타깃 검색 방법.

### 청구항 4

제3항에 있어서, 상기 보로노이 셀은 노드들의 거리들에 관한 순서 정보를 이용하여 계산되는 모자이크 분할들(tessellations)을 갖는, 데이터베이스 내의 타깃 검색 방법.

### 청구항 5

제1항에 있어서, 상기 거리들의 비교는 유클리드 거리(Euclidean distance)를 이용하는, 데이터베이스 내의 타깃 검색 방법.

### 청구항 6

제1항에 있어서, 상기 반복 단계는 적어도 2회 반복으로 실행되는, 데이터베이스 내의 타깃 검색 방법.

### 청구항 7

데이터베이스 내의 콘텐츠를 검색하기 위한 컴퓨터로서,  
적어도 하나의 타깃을 둘러싸는 크기를 갖는 노드들의 네트를 구축하기 위한 수단;  
상기 네트 내의 노드들의 집합을 선택하기 위한 수단;  
타깃으로부터 상기 노드들의 집합 내의 각각의 노드까지의 거리를 비교하는 비교기 수단;  
상기 비교기 수단에 응답하여 상기 타깃에 최근접한, 상기 노드들의 집합 내의 노드를 선택하기 위한 수단;  
  
상기 선택 수단에 응답하여 상기 타깃을 여전히 둘러싸는 크기로 상기 네트를 감축하기 위한 수단; 및  
상기 선택 수단, 상기 비교기 수단, 상기 선택 수단, 및 상기 감축 수단이 상기 네트의 크기가 상기 타깃만을 둘러싸도록 충분히 작아질 때까지 자신들의 작동들을 반복하도록 야기하기 위한 제어 수단

을 포함하는 컴퓨터.

#### 청구항 8

제7항에 있어서, 상기 네트의 크기를 감축하기 위한 수단은 상기 타깃에 최근접한 상기 노드상에 중심을 두도록 상기 네트를 감축시키고, 상기 네트는 상기 타깃에 이르는 상기 최근접 노드의 거리보다 크지 않은 반경을 갖는 장치.

#### 청구항 9

제8항에 있어서, 상기 네트는 보로노이 셀에 의해 정의되는 장치.

#### 청구항 10

제9항에 있어서, 상기 보로노이 셀은 노드들의 거리들에 관한 순서 정보만을 이용하여 계산되는 모자이크 분할들을 갖는 장치.

#### 청구항 11

제7항에 있어서, 상기 비교기 수단은 유클리드 거리를 이용하는 장치.

#### 청구항 12

제7항에 있어서, 상기 제어 회로는 작동들의 반복이 적어도 2회 반복으로 실행되도록 야기하는 장치.

#### 청구항 13

데이터베이스 내의 타깃을 검색하기 위한 방법으로서,

적어도 하나의 타깃을 둘러싸는 크기를 갖는 노드들의 네트를 구축하는 단계;

상기 네트 내의 적어도 하나의 쌍의 노드들을 선택하는 단계;

소정 회수의 반복들 동안, 타깃으로부터 상기 각각의 적어도 하나의 쌍의 노드들 내의 각각의 노드까지의 거리를 비교하는 단계;

상기 비교 단계에 따라서 상기 타깃에 최근접한, 상기 각각의 적어도 하나의 쌍 내의 노드를 선택하는 단계;

상기 선택 단계에 응답하여 상기 타깃을 여전히 둘러싸는 크기로 상기 네트를 감축하는 단계;

상기 네트의 크기가 상기 타깃만을 둘러싸도록 충분히 작아질 때까지 상기 선택, 비교, 선택, 및 감축 단계들을 반복하는 단계

를 포함하는, 데이터베이스 내의 타깃 검색 방법.

#### 청구항 14

제13항에 있어서, 상기 감축 단계는 상기 네트를 감축시켜서 상기 네트가 상기 타깃에 최근접한 상기 노드상에 중심을 두고 또한 상기 네트가 상기 타깃에 이르는 상기 최근접 노드의 거리보다 크지 않은 반경을 갖도록 하는, 데이터베이스 내의 타깃 검색 방법.

#### 청구항 15

제14항에 있어서, 상기 네트는 보로노이 셀에 의해 정의되는, 데이터베이스 내의 타깃 검색 방법.

#### 청구항 16

제15항에 있어서, 상기 보로노이 셀은 노드들의 거리들에 관한 순서 정보를 이용하여 계산되는 모자이크 분할들을 갖는, 데이터베이스 내의 타깃 검색 방법.

#### 청구항 17

제13항에 있어서, 상기 거리들의 비교는 유클리드 거리를 이용하는, 데이터베이스 내의 타깃 검색 방법.

#### 청구항 18

제13항에 있어서, 상기 반복 단계는 적어도 2회 반복으로 실행되는, 데이터베이스 내의 타깃 검색 방법.

#### 청구항 19

데이터베이스 내의 콘텐츠를 검색하기 위한 컴퓨터로서,

적어도 타깃을 둘러싸는 크기를 갖는 노드들의 네트를 구축하기 위한 수단;

상기 네트 내의 적어도 하나의 쌍의 노드들을 선택하기 위한 수단;

복수 회를 반복하는 동안, 타깃으로부터 상기 적어도 하나의 쌍의 노드들 내의 각 노드까지의 거리를 비교하는 비교기 수단;

상기 비교기 수단에 응답하여 상기 타깃에 최근접한, 상기 적어도 하나의 쌍의 노드들 내의 노드를 선택하기 위한 수단;

상기 선택 수단에 응답하여 상기 타깃을 여전히 둘러싸는 크기로 상기 네트의 크기를 감축하기 위한 수단; 및

상기 선택 수단, 상기 비교기 수단, 상기 선택 수단, 및 상기 감축 수단이 상기 네트의 크기가 상기 타깃만을 둘러싸도록 충분히 작아질 때까지 자신들의 작동들을 반복하도록 야기하기 위한 제어 수단

을 포함하는 컴퓨터.

#### 청구항 20

제7항에 있어서, 상기 네트를 감축하기 위한 수단은 상기 타깃에 최근접한 상기 노드상에 중심을 두도록 상기 네트를 감축시키고, 상기 네트는 상기 타깃에 이르는 상기 최근접 노드의 거리보다 크지 않은 반경을 갖는 장치.

#### 청구항 21

제8항에 있어서, 상기 네트는 보로노이 셀에 의해 정의되는 장치.

#### 청구항 22

제9항에 있어서, 상기 보로노이 셀은 노드들의 거리들에 관한 순서 정보만을 이용하여 계산되는 모자이크 분할들을 갖는 장치.

#### 청구항 23

제7항에 있어서, 상기 비교기 수단은 유클리드 거리를 이용하는 장치.

#### 청구항 24

제7항에 있어서, 상기 제어 수단은 작동들의 반복이 적어도 2회 반복으로 실행되도록 야기하는 장치.

### 명세서

#### 기술분야

[0001] 본 출원은 2012년 5월 9일에 출원된 미국 가출원 일련 번호 제61/644,519호의 이익을 청구하는데, 그 개시 내용 전체가 여기에 참조에 의해 포함된다.

[0002] 본 발명의 원리는 비교 기반 활성 검색 및 학습(comparison based active searching and learning)과 관련된다.

#### 배경기술

[0003] 비교를 통한 콘텐츠 검색은 사용자가 하기의 반복 형태로 대규모 데이터베이스에서 타깃 오브젝트의 위치를 파악하는 방법이다. 각각의 단계에서, 데이터베이스는 사용자에게 2개의 오브젝트를 제시하고, 사용자는 그 쌍 중에서 자신이 염두에 두고 있는 타깃에 가장 근접한 오브젝트를 선택(select)한다. 다음 반복에서, 데이터

베이스는 사용자의 이전의 선택들에 기초하여 새로운 오브젝트 쌍을 제시한다. 이 과정은, 사용자의 응답들에 기초하여, 데이터베이스가 사용자가 염두에 두고 있는 타깃을 고유하게 식별할 수 있을 때까지 계속된다.

[0004]

탐색적 검색으로도 알려진 이런 종류의 대화형 네비게이션은 수많은 실세계 응용들을 갖는다. 한 예는 Flickr 또는 Picasa와 같은, 자유로운 환경에서 사진 찍힌 사람들의 사진 데이터베이스를 통해 네비게이팅하는 것이다. 자동화된 방법들은 그와 같은 사진들로부터 의미 있는 특징들을 추출하는 데 실패할 수 있다. 더욱이, 대다수의 실제 사례들에서, (SIFT(Scale-Invariant Feature Transform) 특징들과 같은) 유사한 하위 레벨 기술자(descriptor)들을 제시하는 이미지들은 매우 상이한 의미론적 콘텐츠 및 상위 레벨 기술들을 가질 수 있고, 그러므로 사용자들에 의해 다르게 지각될 수 있다. 다른 한편, 특정 사람에 대한 인간 검색은 사진들의 리스트로부터 그가 염두에 두고 있는 사람과 가장 유사한 대상을 쉽게 선택할 수 있다.

[0005]

집합  $N$ 에 의해 표현되고 또한 상이한 오브젝트들 사이의 "거리" 또는 "비유사도"를 포착해 내는 거리 메트릭(distance metric)  $d$ 를 부여받은 오브젝트들의 데이터베이스를 고려하자. 특정 오브젝트  $t \in N$ 가 주어지면, "비교 오러클(comparison oracle)"은 하기 종류의 질문들에 대답할 수 있는 오러클이다:

[0006]

" $N$ 에서의 두 개의 오브젝트  $x$ 와  $y$  간에서, 어느 것이 메트릭  $d$  하에서  $t$ 에 최근접하는가?"

[0007]

형식적으로, 인간 사용자의 거동은 그러한 비교 오러클에 의해 모델링될 수 있다. 특히, 집합  $N$ 에 의해 표현되고 거리 메트릭  $d$ 를 부여받은 오브젝트들의 데이터베이스가 사진이라고 가정하자.

[0008]

비교들을 통한 대화형 콘텐츠 검색의 목적은 오러클/인간에 제출되기 위한 오브젝트들의 제안 쌍들의 시퀀스를 찾아내어 가능한 한 소수의 질의로 타깃 오브젝트를 식별하도록 이끄는 것이다.

[0009]

비교를 통한 콘텐츠 검색은 최근접 이웃 검색(nearest neighbor search: NNS)의 특별한 경우이고, 메트릭 공간에 임베딩되는 오브젝트들에 대해 NNS 문제를 고려하는 작업의 확대로서 볼 수 있다. 이 임베딩은 작은 본래적 차원을 갖는 것으로 또한 가정되는데, 이 가정은 실 세계에서 지지되는 것이다. 특히, 종래 기술 접근법은 메트릭 공간들을 배가하는 데 있어서 NNS를 지원하기 위한 결정론적 데이터 구조인 네비게이팅 네트(navigating net)들을 도입한다. 비슷한 기술이 특정의 구 패키징 속성을 충족시키는 공간에 임베딩된 오브젝트들에 대해 고려되는 한편, 기타 작업은 성장 제한된 메트릭들에 의존하였다; 모든 앞서의 가정들은 여기서 고려되는 배가 상수(doubling constant)와의 연계성을 갖는다. 앞서 언급한 종래 기술 접근법들 모두에서, 타깃 오브젝트들에 걸친 요구는 균일한 것으로 가정된다.

## 발명의 내용

### 해결하려는 과제

[0010]

비교 오러클로의 접근을 가진 NNS가 여러 종래 작업들에 도입되었다. 이러한 작업들의 상당한 장점은 오브젝트들이 사전에(a-priori) 메트릭 공간에 임베딩된다는 가정이 제거되는 것이다; 오브젝트들 간의 유사도가 거리 메트릭에 의해 포착되는 것을 요구하는 것이 아니라, 이러한 종래 작업들은 임의의 2개의 오브젝트가 비교 오러클에 의해 임의의 타깃에 대한 자신들의 유사도 관점에서 랭크(rank)될 수 있다는 점을 가정할 뿐이다. 그럼에도 불구하고, 이러한 작업들도 균일 요구를 가정하는데, 본 발명의 원리는 비교에 의한 검색을 이질성(heterogeneity)까지 확대한 것으로 볼 수 있다. 이러한 점에서, 또 다른 종래 접근법도 이질적 요구 분포를 가정한다. 그러나, 메트릭 공간이 존재하고 또한 검색 알고리즘이 이것을 인식한다는 가정 하에서는, 본 발명의 원리를 이용하면 평균 검색 비용의 관점에서 더 나은 결과가 제공된다. 앞서 언급한 접근법에서의 주요 문제점은 이 접근법이 무기억적(memoryless)이라는 것인데, 즉 이런 접근법은 이전 비교들을 활용하지 않는 반면에, 본 발명의 해결책에서 이 문제는 E-네트 데이터 구조를 배치함으로써 해결된다.

### 과제의 해결 수단

[0011]

종래 기술에서의 이런 및 기타 결점들 및 단점들이 비교 기반 활성 검색을 위한 방법을 지향하는 본 발명의 원리들에 의해 해결된다.

[0012]

본 발명의 원리의 양태에 따라, 데이터베이스 내의 콘텐츠를 검색하기 위한 여러 방법 및 여러 장치가 제공된다. 제1 방법은 적어도 타깃을 둘러싸는(encompass) 크기를 갖는 노드들의 네트를 먼저 구축하고, 네트 내의 노드들의 집합을 선택하고, 및 타깃으로부터 노드들의 집합 내의 각각의 노드까지의 거리를 비교함으로써 데이터베이스 내의 타깃을 검색하기 위한 단계들로 구성된다. 방법은 비교 단계에 따라서 타깃에 최근접한, 노

드들의 집합 내의 한 노드를 선택하는 단계 및 선택 단계에 응답하여 타깃을 여전히 둘러싸는 크기로 네트워크의 크기를 감축하는 단계를 더 포함한다. 방법은 네트워크의 크기가 타깃만을 둘러싸도록 충분히 작아질 때까지 선택하고, 비교하고, 선택하고, 감축하는 단계들을 반복하는 단계를 더 포함한다.

[0013]

본 발명의 원리들의 또 다른 양태에 따라, 제1 장치가 제공된다. 장치는 적어도 타깃을 둘러싸는 크기를 갖는 네트워크를 구축하기 위한 수단 및 네트워크 내의 노드들의 집합을 선택하기 위한 수단으로 구성된다. 장치는 타깃으로부터 노드들의 집합 내의 각각의 노드까지의 거리를 비교하는 비교기 수단 및 비교기 수단에 따라서 타깃에 최근접한, 노드들의 집합 내의 한 노드를 찾아내는 선택 수단을 더 포함한다. 장치는 선택 수단에 응답하여 타깃을 여전히 둘러싸는 크기로 네트워크의 크기를 감축시키기 위한 회로, 및 선택 수단, 비교기 수단, 선택 수단, 및 감축 수단이 네트워크의 크기가 타깃만을 둘러싸도록 충분히 작아질 때까지 자신들의 작동들을 반복하도록 야기하기 위한 제어 수단을 더 포함한다.

[0014]

본 발명의 원리의 또 다른 양태에 따라, 제2 방법이 제공된다. 방법은 적어도 타깃을 둘러싸는 크기를 갖는 네트워크를 구축하는 단계 및 네트워크 내의 노드들의 적어도 하나의 쌍을 선택하는 단계로 구성된다. 방법은 소정 회수의 반복들 동안 타깃으로부터 노드들의 적어도 하나의 쌍의 각각 내의 각각의 노드까지의 거리를 비교하는 단계, 및 비교 단계에 따라서 타깃에 최근접한 적어도 하나의 쌍의 각각 내의 한 노드를 선택하는 단계를 더 포함한다. 방법은 선택 단계에 응답하여 타깃을 여전히 둘러싸는 크기로 네트워크의 크기를 감축하는 단계, 및 네트워크의 크기가 타깃만을 둘러싸도록 충분히 작아질 때까지 선택, 비교, 선택, 및 감축 단계들을 반복하는 단계를 더 포함한다.

[0015]

본 발명의 원리들의 또 다른 양태에 따라, 제2 장치가 제공된다. 장치는 적어도 타깃을 둘러싸는 크기를 갖는 노드들의 네트워크를 구축하기 위한 수단 및 네트워크 내의 노드들의 적어도 하나의 쌍을 선택하기 위한 수단으로 구성된다. 장치는 소정 회수의 반복들 동안 타깃으로부터 노드들의 적어도 하나의 쌍 내의 각각의 노드까지의 거리를 비교하는 비교기 수단, 및 비교기 수단에 응답하여 타깃에 최근접한, 노드들의 적어도 하나의 쌍 내의 한 노드를 선택하기 위한 수단을 더 포함한다. 장치는 선택 수단에 응답하여 타깃을 여전히 둘러싸는 크기로 네트워크의 크기를 감축시키기 위한 수단, 및 선택 수단, 비교기 수단, 선택 수단, 및 감축 수단이 네트워크의 크기가 타깃만을 둘러싸도록 충분히 작아질 때까지 자신들의 작동들을 반복하도록 야기하기 위한 제어 수단을 더 포함한다.

[0016]

본 발명의 원리들의 이런 및 기타 양태들, 특징들 및 장점들은 첨부된 도면들과 연계하여 읽어보게 될 예시적 실시예들의 하기 상세한 설명으로부터 명백하게 될 것이다.

### 도면의 간단한 설명

[0017]

도 1은 (a) 각각의 샘플 데이터집합에 대해 구축되는 랭크 네트워크 트리 계층 구조의 크기뿐만 아니라 크기, 차원의 표, (b) 기대되는 질의 복잡도, 및 (c) 기대되는 계산 복잡도를 도시한다.

도 2는 데이터집합 크기의 함수로서의 5개 알고리즘의 (a) 질의 복잡도와 (b) 계산 복잡도, 및 (c) 불완전한 오일러 하에서의  $n$ 의 함수로서의 질의 복잡도를 도시한다.

도 3은 본 발명의 원리에 의해 구현되는 예시적 알고리즘들을 도시한다.

도 4는 본 발명의 원리 하에서의 방법의 제1 실시예를 도시한다.

도 5는 본 발명의 원리 하에서의 장치의 제1 실시예를 도시한다.

도 6은 본 발명의 원리 하에서의 방법의 제2 실시예를 도시한다.

도 7은 본 발명의 원리 하에서의 장치의 제1 실시예를 도시한다.

### 발명을 실시하기 위한 구체적인 내용

[0018]

본 발명의 원리는 비교 기반 활성 검색을 위한 방법 및 장치를 지향한다. 방법은 이전 국면의 결과들을 이용하는 반복된 비교 국면들이 있기 때문에 "활성 검색"이라고 불린다. 방법은 (예를 들어, 오브젝트들, 사진들, 영화들, 아티클들, 기타 등등인) 오브젝트들의 데이터베이스를 통해 네비게이팅하고, 2개의 오브젝트 중 어느 것이 타깃(예를 들어, 사진, 영화 또는 아티클, 기타 등등)에 최근접한 것인지를 결정하는 비교 오러클에게 오브젝트들의 쌍들을 제공한다. 다음 반복에서, 데이터베이스는 사용자의 이전의 선택들에 기초하여 오브젝트들의 새로운 쌍을 제시한다. 이 과정은, 사용자의 응답들에 기초하여 데이터베이스가 사용자가 염두에 두고 있는

타깃을 고유하게 식별할 수 있을 때까지 계속된다. 각각의 국면에서, 오브젝트들의 소규모 리스트가 비교를 위해 제시된다. 리스트 중에서의 하나의 오브젝트가 타깃에 최근접한 오브젝트로서 선택되고; 이후 새로운 오브젝트 리스트가 이전의 선택들에 기초하여 제시된다. 이 과정은 타깃이 제시된 리스트에 포함될 때까지 계속되고, 이 시점에서 타깃은 발견되고 검색은 종료된다.

[0019] 여기서 기술된 접근법은 이질적 요구의 시나리오 하에서의 문제를 고려하며, 여기서 타깃 오브젝트  $t \in N$ 은 확률분포  $\mu$ 로부터 샘플링된다. 이 설정에서, 비교를 통한 대화형 콘텐츠 검색은 고전적 "스무 고개 게임" 문제와 강한 관계성을 갖는다. 특히, 멤버십 오러클은 하기 형태의 질의들에 응답할 수 있는 오러클이다:

[0020] "부집합  $A \subseteq N$  이 주어지면,  $t$ 가  $A$ 에 속하는가?"

[0021] 타깃  $t$ 를 찾기 위해, 평균적으로 적어도  $H(\mu)$  개의 질의를 멤버십 오러클에게 제출할 필요가 있다는 것이 주지되어 있는데, 여기서  $H(\mu)$  는  $\mu$  의 엔트로피이다. 더욱이, 평균적으로  $H(\mu) + 1$  개의 질의만으로 타깃을 찾아내는 알고리즘(허프만 부호화)이 존재한다.

[0022] 비교를 통한 콘텐츠 검색은 데이터베이스  $N$ 이 메트릭  $d$ 를 부여받다고 가정할 시에 상기 설정에서 이탈한다. 멤버십 오러클은, 거리 메트릭  $d$ 가 알려지면 비교 질의들이 멤버십 질의들을 통하여 시뮬레이팅될 수 있으므로 비교 오러클보다 더 강하다. 다른 한편, 멤버십 오러클은 실제로 구현하기에 더 어렵다:  $A$ 가 간결한 방식으로 표현될 수 없다면, 사용자는  $|A|$ 의 선형 시간에 멤버십 질의에 응답할 것이다. 이는 응답들이 일정 시간에 주어질 수 있는 비교 오러클과 대조된다. 간단히 말하면, 비교를 통한 검색에 대해 여기서 다루지는 문제는 (a) 구현하기에 더 쉬운 오러클에 대해 및 (b) 데이터베이스의 구조에 대한 추가적 가정, 즉 데이터베이스에 거리 메트릭이 부여되는 가정 하에서, 고전적 설정에 대한 유사한 성능 한도들을 구한다.

[0023] 직관적으로, 비교를 통한 오브젝트 검색의 성능은 타깃 분포의 엔트로피뿐만 아니라 메트릭  $d$ 에 의해 기술되는 것과 같은 타깃 집합  $N$ 의 토폴로지도 의존한다. 특히, 비교 오러클을 이용하여 타깃의 위치를 파악하기 위해서는,  $\Omega(cH(\mu))$  개의 질의가 기대치(expectation)에서 필요하다는 것이 확립되었는데, 여기서  $c$ 는 메트릭  $d$ 의 이른바 배가 상수이다. 더욱이, 발명자들은 기대치에서,  $O(c^3 H \log(1/\mu^*))$  개의 질의로 타깃의 위치를 파악하는 방법을 이전에 제공했는데, 여기서  $\mu^* = \min_{x \in N} \mu(x)$  이다. 본 발명의 원리 하에서, 이전 한도에 대한 개선은 기대치에서,  $O(c^5 H(\mu))$  개의 질의로 타깃의 위치를 파악하는 방법을 이용하여 달성된다.

[0024] 비교를 통한 검색

[0025] 오브젝트들 사이의 "비유사성"을 포착하는 거리 메트릭  $d$ 가 부여된, 크기  $n := |N|$ 의 오브젝트들의 큰 유한집합  $N$ 을 고려하자. 사용자는 프라이어(prior) 분포  $\mu$ 로부터 타깃  $t \in N$ 을 선택한다. 본 발명의 원리의 목표는 가능한 한 소수의 질의로  $t$ 를 발견하기 위한 목적으로 사용자에게 오브젝트들의 쌍으로 질의하는 대화형 방법을 설계하는 것이다.

[0026] 비교 오러클은, 두 개의 오브젝트  $x, y$  및 타깃  $t$ 가 주어지면,  $t$ 에 최근접한 오브젝트를 리턴하는 오러클이다. 더 형식적으로는,

[0027] 
$$O_{\text{비교}}(x, y, t) = \begin{cases} x & \text{if } d(x, t) \leq d(y, t) \\ y & \text{if } d(x, t) > d(y, t) \\ \text{error} & \text{if } d(x, t) = d(y, t) \end{cases}$$

[0028] 메트릭  $d$ 가 존재한다고 가정하기는 하였지만, 거리들의 시야는 오브젝트들 간의 순서 관계성(order



relationship)들을 관찰하는 것에만 한정된다. 더 정밀하게는, 비교 오러클을 통하여 획득될 수 있는 정보에 대한 접근만이 있다. 오브젝트  $z$ 가 주어지면, 비교 오러클  $O_z$ 는 질의로서 순서 쌍  $(x, y) \in N^2$ 을 수신하고, 질문 "z가 y보다 x에 더 가까운가?"에 대답하는데, 즉

### 수학식 1

$$O_z(x, y) = \begin{cases} +1 & \text{if } d(x, z) < d(y, z), \\ -1 & \text{if } d(x, z) \geq d(y, z) \end{cases}$$

[0029]

[0030]

알려지지 않은 타깃  $t$ 를 결정하기 위해 기술되는 여기서의 방법은 비교 오러클  $O_t$ , 즉 사용자에게 질의들을 제출한다. 사용자가 오브젝트들을  $t$ 로부터의 이들의 거리를 기준으로 순서화할 수 있지만, 이러한 거리들의 정확한 값들을 개시할(또는 심지어 알) 필요는 없다는 점을 실효적으로 가정한다.

[0031]

다음에, 오러클이 항상 정확한 대답들을 제공한다고 가정한다; 이 후에, 이 가정은 확률  $\epsilon < 0.5$ 에 놓여 있는 불완전한 오러클(faulty oracle)을 고려함으로써 완화된다.

[0032]

본 발명의 원리의 주안점은  $O_t$ 에게 제출하는 것으로서 거리 메트릭  $d$ 의 지식을 요구하지 않는 질의를 결정하는 것이다. 제시된 방법들은 (a) 분포  $\mu$  및 (b) 모든  $z \in N$  마다에 대해 매핑  $O_z: N^2 \rightarrow \{-1, +1\}$ 의 값들의 사전 지식에만 의존한다. 이것은 거리 메트릭  $d$ 가 존재하지만, 이것이 직접적으로 관찰될 수 없다는 가정과 부합한다.

[0033]

프라이어(prior)  $\mu$ 는 오브젝트들이 과거에 타깃이 되었던 빈도로서 경험적으로 추정될 수 있다. 순서 관계성들은  $\Theta(n^2 \log n)$ 개의 질의들을 비교 오러클에게 제출하고, 및  $\Theta(n^2)$  공간을 요청함으로써 오프라인으로 계산될 수 있다: 각각의 가능한 타깃  $z \in N$ 에 대해,  $N$  내의 오브젝트들은  $O_z$ 에게의  $\Theta(n \log n)$ 개의 질의에 의해  $z$ 로부터의 자신들의 거리를 기준으로 소팅될 수 있다.

[0034]

이 소팅의 결과는 (a) 그 엘리먼트들이  $z$ 로부터 동일 거리에 있는 오브젝트들의 집합들인 링크된 리스트에, 및 (b) 모든 엘리먼트  $y$ 마다를 소팅된 리스트에서의 그 랭크와 연관시키는 해시 맵에 저장된다. 그러므로  $O_z(x, y)$ 가  $x$  및  $y$ 의 상대적 랭크들을  $z$ 로부터의 자신들의 거리에 대하여 비교함으로써  $O(1)$ 회에 회수될 수 있다는 것을 유의하라.

[0035]

본 발명의 원리의 초점은 다음에 제출할  $N^2$ 에서의 질의에 대한 관정이 오러클의 이전 대답들에 의해 결정되는 적응 알고리즘들에 있다. 방법의 성능은 2개의 메트릭을 통해 측정될 수 있다. 제1 메트릭은 방법이 타깃을 결정하기 위해 오러클에게 제출할 필요가 있는 질의들의 기대된 개수에 의해 결정되는, 방법의 질의 복잡도(query complexity)이다. 제2 메트릭은 각각의 단계에서 오러클에 제출할 질의를 결정하기 위한 시간 복잡도에 의해 결정되는, 방법의 계산 복잡도이다.

[0036]

하한

[0037]

$\mu$ 의 엔트로피가  $H(\mu) = \sum_{x \in \text{supp}(\mu)} \mu(x) \log(1/\mu(x))$ 로서 정의된다는 것을 상기하는데, 여기서  $\text{supp}(\mu)$ 는  $\mu$ 의 서포트(support)이다. 오브젝트  $x \in N$ 가 주어지면,  $B_x(r) = \{y \in N : d(x, y) \leq r\}$ 가  $x$  주위의 반경  $r \geq 0$ 의 닫힌 볼(closed ball)이라고 하자. 집합  $A \subseteq N$ 가 주어지면,  $\mu(A) = \sum_{x \in A} \mu(x)$ 라고 하자. 임의의  $x \in \text{supp}(\mu)$



및 임의의  $R \geq 0$ 에 대해, 분포  $\mu$ 의 배가 상수  $c(\mu)$ 가  $\mu(B_x(2R)) \leq c \mu(B_x(R))$ 인 최소  $c > 0$ 가 된다.

[0038]

배가 상수는 거리  $d$ 에 의해 결정되는 대로의 데이터집합의 기본 차원에 대한 자연적 연결을 갖는다. 엔트로피 및 배가 상수 둘 모두는 또한 비교를 통한 콘텐츠 검색에 본래적으로 연결된다. 타깃  $t$ 의 위치를 파악하기 위

한 임의의 적응적 메커니즘이 기대치에서 오러클  $Q$ 에게 적어도  $\Omega(c(\mu)H(\mu))$ 개의 질의를 제출해야만 한다는 것

이 보여졌다. 더욱이, 이전 작업들은  $O(c^3 H(\mu)H_{\max}(\mu))$ 개의 질의로 타깃을 결정하기 위한 알고리즘을 기술했는

데, 여기서  $H_{\max}(\mu) = \max_{x \in \text{supp}(\mu)} \log(1/\mu(x))$ 이다.

[0039]

활성 학습

[0040]

비교를 통한 검색은 활성 학습의 특별한 경우로 볼 수 있다. 활성 학습에서, 가설 공간(hypothesis space)  $H$ 는 질의 공간이라고 불리는, 유한집합  $Q$ 상에서 정의되는 이진 값 함수들의 집합이다. 각각의 가설  $h \in H$ 는 모든 질의  $q \in Q$ 마다  $\{-1, +1\}$ 로부터 라벨을 생성한다. 타깃 가설  $h^*$ 는 몇몇 프라이어  $\mu$ 에 따라  $H$ 로부터 샘플링된다; 질의  $q$ 를 묻는 것은  $h^*(q)$ 의 값을 밝히는 것과 다름없으며, 그에 의해 가능한 후보 가설들을 제한한다. 목표는 가능한 한 소수의 질의를 물음으로써, 적응적 방식으로  $h^*$ 를 고유하게 결정하는 것이다.

[0041]

본 발명의 원리에 대해, 가설 공간  $H$ 는 오브젝트들의 집합  $N$ 이고, 질의 공간  $Q$ 는 순서 쌍들의 집합  $N^2$ 이다.  $\mu$ 로부터 샘플링되는 타깃 가설은 다름 아닌  $t$ 이다. 각각의 가설/오브젝트  $z \in N$ 는 사전에 알려진 것으로 가정되는 매핑  $O_z: N^2 \rightarrow \{-1, +1\}$ 에 의해 고유하게 식별된다.

[0042]

일반적 활성 학습 설정에서 참 가설을 결정하기 위한 주지된 알고리즘은 소위 일반화된 이진 검색(generalized binary search: GBS) 또는 스플리팅 알고리즘(splitting algorithm)이다. 버전 공간  $V \subseteq H$ 을 지금까지 관찰된 질의 응답들과 일치하는 가능한 가설들의 집합이라고 정의하자. 각각의 단계에서, GBS는  $|\sum_{h \in V} \mu(h)h(q)|$ 를 최소화하는 질의  $q \in Q$ 를 선택한다. 다르게 말하면, GBS는 현재 버전 공간을 대략 동등한(확률) 규모(mass)의 2개의 집합으로 분리하는 질의를 선택한다; 이것은 기대치에서 가능한 대로 버전 공간의 규모에서의 최대 축소로 이어지고, 따라서 GBS는 탐욕적 질의 선택 정책으로 볼 수 있다.

[0043]

GBS의 질의 복잡도에 대한 한도는 하기 정리에 의해 주어진다:

[0044]

정리 1. GBS는  $OPT$ 가 임의의 적응적 정책에 의해 만들어지는 질의들의 최소 기대 회수라면, 가설  $h^* \in N$ 를 식별하기 위해 기대치에서 최대로는  $OPT \cdot (H_{\max}(\mu) + 1)$ 개의 질의를 한다.

[0045]

비교를 통한 검색에서의 GBS

[0046]

본 발명의 원리에 대해, 버전 공간  $V$ 는 지금까지 주어진 오러클 대답들과 일치하는  $z \in N$ 에서의 모든 가능한 오브젝트들을 포함한다. 다시 말하면, 지금까지 오러클에게 제출된 모든 질의들  $(x, y)$ 에 대해  $O_z(x, y) = O_t(x, y)$ 이면,  $z \in V$ 이다. 그러므로 다음 질의를 선택하는 것은

## 수학식 2

$$f(x, y) = \left| \sum_{z \in V} \mu(z) O_z(x, y) \right|.$$

[0047]

- [0048] 를 최소화하는 쌍  $(x, y) \in N^2$  를 찾아내는 것과 다름없다.
- [0049] 시뮬레이션들은 GBS의 질의 복잡도가 실제상 뛰어나다는 것을 보여준다. 이것은 이 상한이 비교를 통한 검색의 특정 맥락에서 잠재적으로 향상될 수 있다는 점을 시사한다.
- [0050] 그럼에도 불구하고, GBS의 계산 복잡도는, 이것이  $N^2$ 에서의 모든 쌍들에 걸쳐서  $f(x, y)$ 를 최소화하는 것을 요구하므로, 질의당  $\Theta(n^2 |V|)$  작동이다. 큰 집합들  $N$ 에 대해, 이것은 참으로 터무니없는 것일 수 있다. 이것은 그 계산 복잡도가  $O(1)$ 이고 그 질의 복잡도가 최적인 것으로부터  $O(c^5(\mu))$  인자 내에 포함되는 새로운 알고리즘 RANKNETSEARCH를 제안하도록 동기 부여한다.
- [0051] 효율적 적응 알고리즘
- [0052] 본 발명의 원리를 이용하는 방법은 NNS(Nearest Neighbor Search)의 맥락 하에서 이전에 도입된 구조인  $\mathcal{E}$ -네트들에 의해 고취된다. 주요 전제는 거의 중첩을 갖지 않는 볼들로 구성되는, 네트를 가진 버전 공간(즉, 현재 유효한 가설/가능한 타깃들)을 커버하는 것이다. 각각의 볼의 중심을 타깃에게의 그들의 거리에 대하여 비교함으로써, 방법은 타깃이 속하는 볼을 식별할 수 있다. 검색은 버전 공간을 이 볼에 국한시키고 또한 이 과정을 반복함으로써 진행되고, 더 미세한 네트로 이 볼을 커버하게 된다. 직면한 주요 도전은, 표준 NNS와는 대조적으로 기본 거리 메트릭(underlying distance metric)에 대한 어떤 접근도 없다는 것이다. 또한,  $\mathcal{E}$ -네트들에 의해 만들어진 비교 횟수에 대한 한도들은 최악 경우이다(즉, 프라이어-프리(prior-free): 이 방법을 이용하는 구축은 기대치에서 경계들을 제공하기 위해 프라이어  $\mu$ 를 고려한다.
- [0053] 랭크 네트들
- [0054] 상기 문제들을 해결하기 위해, 본 방법은 이 설정에서  $\mathcal{E}$ -네트들의 역할을 할 랭크 네트들(rank nets)의 개념을 도입한다. 몇몇  $x \in N$ 에 대해, 볼  $E = B_x(R) \subseteq N$ 를 고려하자. 임의의  $y \in E$ 에 대해,

### 수학식 3

$$d_y(\rho, E) = \inf\{r : \mu(B_y(r)) \geq \rho\mu(E)\}$$

[0055]

- [0056] 이,  $\rho\mu(E)$ 를 넘는 규모를 유지하는  $y$  주위의 최소 볼의 반경이라고 정의한다. 이 정의를 이용하여,  $\rho$ -랭크 네트를 다음과 같이 정의한다.

- [0057] 정의 1. 어떤  $\rho < 1$ 에 대해,  $E = B_x(R) \subseteq N$ 의  $\rho$  랭크 네트는, 임의의 두 개의 구별되는  $y, y' \in R$ 에 대해

### 수학식 4

$$d(y, y') > \min\{d_y(\rho, E), d_{y'}(\rho, E)\}.$$

[0058]

- [0059] 가 되도록 포인트들  $R \subseteq E$ 의 최대 모음이다.

- [0060] 임의의  $y \in R$ 에 대해, 보로노이 셀(Voronoi cell)
- [0061]  $V_y = \{z \in B : d(y, z) \leq d(y', z), \forall y' \in R, y' \neq y\}$ .
- [0062] 을 고려하자.
- [0063] 또한, 보로노이 셀  $V_y$ 의 반경  $r_y$ 를  $r_y = \inf\{r : V_y \subseteq B_y(r)\}$ 라고 정의한다.
- [0064] 여기 목적들을 위해 중요한 것으로는, 랭크 넷 및 이것이 정의하는 보로노이 모자이크 분할(tessellation)은 순서 정보만을 이용하여 둘 모두 계산될 수 있다는 점이다:
- [0065] 보조정리 1.  $E$ 의  $p$ -랭크 넷  $R$ 은  $O(|E|(\log |E| + |R|))$  단계로 구축될 수 있고,  $R$  주위의 보로노이 셀들을 에워싸는 볼들  $B_y(r_y) \subseteq E$ 은 오직 (a)  $\mu$  및 (b) 모든  $z \in E$  마다의 매핑들  $O_z : \mathbb{N}^2 \rightarrow \{-1, +1\}$ 을 이용하여  $O(|E||R|)$  단계로 구축될 수 있다.
- [0066] 이 결과에 의해, 논의의 초점은  $p$ 의 셀렉트가 넷의 크기뿐만 아니라 그 주위의 보로노이 볼들의 규모에 어떻게 영향을 미치느냐가 된다. 다음 보조정리는  $|R|$ 의 한도를 정한다.
- [0067] 보조정리 2. 넷  $R$ 의 크기는 최대로는  $c^3/p$ 이다.
- [0068] 하기 보조정리는 넷에서의 보로노이 볼들의 규모를 결정한다.
- [0069] 보조정리 3.  $r_y > 0$ 이라면,  $\mu(B_y(r_y)) \leq c^3 p \mu(E)$ 이다.
- [0070] 보조정리 3은 반경 0의 보로노이 볼들의 규모의 한도를 정하지 않는다는 것을 유의하라. 이 보조정리는 사실상 반드시 고 확률 오브젝트들  $y$ (이것에 대해  $\mu(y) > c^3 p \mu(E)$ )가  $R$ 에 포함되고 대응하는 볼들  $B_y(r_y)$ 가 단독자(singleton)들이라는 것을 내포한다.
- [0071] 랭크 넷 데이터 구조 및 알고리즘
- [0072] 랭크 넷들은 알고리즘 1에서 기술된 바와 같은 비교 오러클  $\mathcal{O}$ 를 이용하여 타깃  $t$ 를 식별하는데 사용될 수 있다. 초기에,  $N$ 을 커버하는 넷  $R$ 이 구축된다; 노드들  $y \in R$ 은  $t$ 로부터의 이들의 거리에 대하여 비교되고, 타깃에 최근접한 것이 결정되는데,  $y^*$ 라고 칭한다. 이것이  $|R|^{-1}$ 개의 질의를 오러클에게 제출하는 것을 요구한다는 것을 주의하라. 버전 공간  $V$ (가능한 가설들의 집합)는 그러므로 보로노이 셀  $V_{y^*}$ 이고 또한 볼  $B_{y^*}(r_{y^*})$ 의 부집합이다. 방법은 이후  $B_{y^*}(r_{y^*})$ 에게 검색을 한정하고 상기 과정을 반복함으로써 진행된다. 항상, 버전 공간은 넷에 의해 커버되는 현재 볼에 포함된다는 사실에 유의하라. 이 과정은, 이 볼이 구축에 의해 타깃을 포함해야만 하는 단독자가 될 때 종결된다.
- [0073] 상기 방법에서의 한가지 질문은 어떻게  $p$ 를 셀렉트하는가이다: 보조정리 3에 의해, 작은 값들은 하나의 레벨로부터 다음 레벨까지의 보로노이 볼들의 규모에서의 급격한 감소로 이어지고, 따라서 더 적은 반복들로 타깃에 도달한다. 다른 한편, 보조정리 2에 의해, 작은 값들은 또한 더 큰 넷들을 내포하여서, 각각의 반복에서 오러클에 대한 더 많은 질의들로 이어진다. 여기의 방법은 알고리즘 2의 의사 코드에 의해 표시된 것처럼, 반복적 형태로  $p$ 를 셀렉트한다. 방법은 결과로 생기는 넷의 모든 비-단독자 보로노이 볼들  $B_{y^*}(r_{y^*})$ 가  $0.5 \mu(E)$ 에 의해 한정되는 규모를 가질 때까지 반복적으로  $p$ 를 이등분한다. 이 셀렉트는 RANKNETSEARCH의 대응하는 질

의 및 계산 복잡도에 대한 하기 한도들로 이어진다:

[0074] 정리 2. RANKNETSEARCH는 기대치에서, 비교 오러클에게  $4c^6(1+H(\mu))$  개의 질의를 함으로써 타깃의 위치를 파악한다. 다음 차례로 어느 질의를 제출한 것인지를 결정하는 비용은  $O(n(\log n + c^6)\log c)$  이다.

[0075]  $\Omega(cH(\mu))$  의 질의 복잡도에 대한 하한에 비추어 보아, 본 방법, RANKNETSEARCH는 질의 복잡도의 관점에서 최적 알고리즘의  $O(c^5)$  인자 내에 있고, 그러므로 상수  $c$ 에 대해 순서 최적이다. 더욱이, 질의당 계산 복잡도는 GBS 알고리즘의 세제곱 비용과는 대조적으로,  $O(n(\log n + c^6))$  이다. 이것은 GBS와 비교하여 계산 복잡도의 대폭적 감축으로 이어진다.

[0076] 상기 계산 비용이 사실상 할부상환(amortization)을 통하여  $O(1)$  로 감소될 수 있다는 것을 유의하라. 특히, RANKNETSEARCH가 추종하는 가능한 경로들이 계층 구조를 정의하고, 그에 의해 모든 오브젝트 각각이 그 보로노이 볼을 커버하는 오브젝트들에게 부모(parent)로서 구실한다는 것을 아는 것은 용이하다. 이 트리는 미리 구축될 수 있고, 검색은 이 트리에 걸친 강하로서 구현될 수 있다.

[0077] 잡음성 비교 오러클

[0078] 이제, 임의의 주어진 질의  $O(x, y, t)$  에의 대답이 확률  $1-p_{x,y,t}$  로 정확하고, 그외의 경우에는 잘못된 대답이고, 이것이 구별되는 질의들에 대해 독립적인 잡음성 오러클들을 고려하자. 이어지는 부분에서도, 오류 확률들  $p_{x,y,t}$  가 1/2로부터 떨어져서 한정된다고 가정하는데, 즉 모든  $(x, y, t)$ 에 대해  $p_{x,y,t} \leq p_e$  가 되도록  $p_e < 1/2$  가 존재한다.

[0079] 이런 맥락에서, 본 발명의 원리의 또 다른 실시예는 질의 복잡도가 그에 대해 한정되는 이전 알고리즘의 수정을 제안한다. 이 절차는 앞서와 같이 구축되는 랭크 네트 계층 구조에 여전히 의존한다. 그러나, 이 실시예는 랭크 네트의 잘못된 엘리먼트가 계층 구조를 따라 아래로 1 레벨 이동했을 때 선택된 확률을 한정하기 위해 각각의 라운드에서 반복들을 이용한다.

[0080] 특정하게는, 주어진 레벨  $\ell$  및 랭크 네트 크기  $m$ 에 대해,

## 수학식 5

$$R_{\ell, \beta}(\ell, m) := \frac{2 \log((\ell + \ell_0)^\beta \lceil \log_2(m) \rceil)}{(1 - p_e)^2}.$$

[0081]

[0082] 에 의해 반복 인자  $R_{\ell, \beta}(\ell, m)$  를 정의하는데, 여기서  $\beta > 1$  및  $\ell_0$ 는 두 개의 설계 파라미터이다.

[0083] 수정된 알고리즘은 이후 상위 레벨 ( $\ell = 0$ )에서 시작하여, 계층 구조를 따라 아래로 진행된다. 기초 단계는, 레벨  $\ell$  에 있을 때, 대응하는 랭크 네트에서의 노드들의 집합  $A$ 를 가지며, 다음과 같이 진행된다. 토너먼트가

초기에 쌍을 이룬 랭크 네트 멤버들 중에 조직된다. 경쟁하는 멤버들의 쌍들은  $R_{\ell, \beta}(\ell, |A|)$  회 비교된다. 최대 게임 회수를 승리하는 주어진 쌍으로부터의 "플레이어"는 제1 국면으로 움직이는데, 여기서 이 플레이어는 제1 라운드의 또 다른 승자와 다시 쌍을 이룰 것이고, 단 하나의 플레이어가 남을 때까지 그렇게 계속된다. 반

복들  $R$ 의 회수가 레벨  $\ell$ 에 따라 대수적으로(logarithmically)만 증가한다는 것을 주목하라.

[0084]

질의 복잡도 및 정확한 타겟 식별의 대응 확률에 대한 한도들은 하기를 레버리지(leverage)함으로써 도출될 것이다:

[0085]

보조정리 4. 고정 타겟  $t$  및 오류 확률에 대한 상한  $p_e$ 를 가진 잡음성 오라클이 주어지면, 반복들  $R_{\ell, \beta}(\ell, |A|)$ 을 가진 집합  $A$ 의 엘리먼트들 중의 토너먼트는 적어도  $1 - (\ell + \ell_0)^{-\beta}$  확률로 타겟  $t$ 에 최근접한 집합  $A$ 에서의 엘리먼트를 리턴한다.

[0086]

이것은 단순화를 위해 어떤 타이(tie)들도 없다는 것, 즉  $t$ 에 최근접한  $A$ 에서 유일한 포인트가 있다고 가정함으로써 입증될 수 있다. 타이들을 가진 경우도 유사하게 추론될 수 있다. 먼저, 질의들  $O(x, y, t)$ 를  $R$ 회 반복할 시에  $x$  및  $y$  중에서 비교의 대부분을 승리하는 것이  $t$ 에 최근접한 것이 아닌 확률  $p(R)$ 을 한정한다. 오류 확률에 대한 상한  $p_e$  때문에, (타이들의 가능성을 무시하면)

[0087]

$$p(R) \leq \Pr(\text{Bin}(R, p_e) \geq R/2)$$

[0088]

을 얻는다.

[0089]

아주마-호에프딩(Azuma-Hoeffding) 부등식은 상기 부등식의 우변이  $\exp(-R(1/2 - p_e)^2/2)$ 보다 크지 않은 것을 보장한다. 수학적식 5에 의해 반복들  $R$ 의 회수를 대체할 시에, 오류의 대응 확률은

[0090]

$$p(R_{\ell, \beta}(\ell, |A|)) \leq (\ell + \ell_0)^{-\beta} \frac{1}{\lceil \log_2(|A|) \rceil}.$$

[0091]

에 의해 상한이 지어진다는 것을 발견한다.

[0092]

이제 게임들이  $t$ 에 최근접한  $A$  내의 엘리먼트에 의해 플레이되는 것을 고려하자. 최대로는  $\lceil \log_2(|A|) \rceil$ 의 그런 게임들이 있다. 연합 한도(union bound)에 의해, 최근접 엘리먼트가 이러한 게임들 중 임의의 하나에서 지는 확률은 이론적으로는  $(\ell + \ell_0)^{-\beta}$ 보다 작지 않다.

[0093]

참고 1. 무잡음 오라클로 타겟  $t$ 에 최근접한 오브젝트를 찾아내기 위해, 명확하게는  $O(|A|)$ 의 질의 회수가 필요하다. 제안된 알고리즘은 최대로는 인자  $2 R_{\ell, \beta}(\ell, |A|)$ 만큼 더 많은 비교들을 함으로써 높은 확률로 동일 목적을 달성한다.

[0094]

이런 맥락에서, 방금 제안된 알고리즘은 다음을 검증한다:

[0095]

정리 3. 반복들 및 토너먼트들을 가진 알고리즘은 적어도  $1 - \sum_{\ell \geq \ell_0} \ell^{-\beta} \ln O(\sum_{\ell \in \mathcal{N}} p_{\ell} \log \frac{1}{p_{\ell}} \log \log \frac{1}{p_{\ell}})$  질의 확률로 정확한 타겟을 출력한다.

[0096]

참고 2.  $\beta > 1$  및 충분히 큰  $\ell_0$ 을 선택함으로써 오류 확률이 임의로 작게 될 수 있다는 것을 유의하라. 또한, 균등 분포  $p_i \approx 1/n$ 에 대해, 차수  $H(\mu) = \log(n)$ 항 이외에도 여분 인자  $\log \log(n)$ 을 또한 유의하라.

[0097] 이것은, 연합 한도와 이전 보조정리에 의해 임의의 타깃  $t \in N$ 에 대해 조건부로  $Pr(success|T=t) \geq 1 - \sum_{k=0}^{\infty} (1-t^k)$  이므로 증명될 수 있다. 타깃이  $T=t$ 라고 주어진 경우에 비교 회수는 최대로는

$$\sum_{t=0}^{\lfloor \log_2(1/p_t) \rfloor} 2N(R_{t,p}(t, W_t)) = O\left(\log \frac{1}{p_t} \log \log \frac{1}{p_t}\right),$$

[0098]

[0099] 인데, 여기서 0항은 배가 상수  $c$ , 오류 확률  $p_e$  및 설계 파라미터들  $\ell_0$  및  $\beta$ 에만 의존한다. 질의들의 기대되는 회수에 대한 한도는  $t \in N$ 에 걸쳐서 평균화함으로써 따라온다.

[0100] 도 1(a)는 각각의 데이터집합에 대해 구축된 크기, 차원(특징들의 수)뿐만 아니라, 랭크 네트 트리 계층 구조의 크기의 표를 보여준다. 도 1(b)는 각각의 데이터집합에 적용되는 5개 알고리즘의, 검색당 기대되는 질의 복잡도를 보여준다. RANKNET 및 T-RANKNET는 동일한 질의 복잡도를 갖기 때문에, 하나만 도시된다. 도 1(c)는, 각각의 데이터집합에 적용되는 5개 알고리즘의, 검색당 기대되는 계산 복잡도를 보여준다. MEMORYLESS 및 T-RANKNET에 대해 이 기대된 계산 복잡도는 질의 복잡도와 동등하다.

[0101] 평가

[0102] 본 발명의 원리 하에서의 제안된 방법인, RANKNETSEARCH는 6개의 공개적으로 이용 가능한 데이터집합에 걸쳐서 평가될 수 있다; *iris*, *abalone*, *ad*, *faces*, *swiss roll(isomap)*, 및 *netflix(netflix)*. 후자의 두 개는 *swiss roll*로부터 1000개의 무작위로 선택된 데이터 포인트들, 및 *netflix*에서의 1000개의 최대로 등급 매겨진 영화들을 취하여 하위 샘플링될 수 있다.

[0103] 이러한 데이터집합들은 유클리드 공간  $\mathbf{R}^d$ 에 매핑된다(범주형 변수들이 표준 방식으로 이전 값들에 매핑된다); 차원들  $d$ 는 도 1(a)의 표에 보여진다. *netflix*에 대해, 영화들은 SVD를 통한 사용자/영화 등급 매김 행렬의 낮은 랭크 근사를 획득함으로써 50 차원 벡터들에 매핑되었다. 그러면,  $\ell_2$ 를 오브젝트들 간의 거리 메트릭으로 이용하여,  $\alpha=0.4$ 를 가진 멍승 법칙(power law) 프라이어로부터 타깃들을 선택한다.

[0104] RankNetSearch의 두 가지 구현의 성능: 알고리즘 1에서와 같이 랭크 네트가 온라인으로 결정되는 하나의 구현이 평가되었고, 다른 하나는 랭크 네트들의 전체 계층 구조가 미리 계산되고 또한 트리로서 저장되는 경우인데 T-RANKNETSEARCH로 표기된다. 양 알고리즘들은 오라클에게 정확하게 동일 질의들을 제안하고, 그래서 동일 질의 복잡도를 갖는다; 그러나, T-RANKNETSEARCH는 질의당  $O(1)$  계산 복잡도만을 갖는다. 각각의 데이터집합에 대해 T-RANKNETSEARCH에 의해 미리 계산되는 트리들의 크기들은 도 1(a)의 표에 도시된다.

[0105] 이러한 알고리즘들은 (a) 한가지 종래 기술 방법에 의해 제안되는 무기억 정책 및 (b) GBS에 기초한 두 가지 휴리스틱(heuristic)과 비교될 것이다. 질의당 GBS의  $\Theta(n^3)$  계산 비용은 이것을 여기서 고려되는 데이터집합들 상에서 처리하기 어렵게 만든다.

[0106] GBS와 같이, 고속 GBS에 대한 F-GBS라고 칭해지는 제1 휴리스틱은 수학적 2를 최소화하는 질의를 선택한다. 그러나, 이것은 현재 버전 공간  $V$ 에서의 오브젝트들의 쌍들에게 질의들을 국한함으로써 그렇게 한다. 이것은

$\Theta(n^2|V|)$ 이 아니라  $\Theta(|V|^3)$ 로 질의당 계산 비용을 감축시킨다. 물론, 이것은 초기 질의들에 대해 여전히  $\Theta(n^3)$ 이다. 소한(sparse) GBS에 대한 S-GBS라고 칭해지는 제2 휴리스틱은 다음의 방식으로 랭크 네트들을 활용한다. 첫째로, 랭크 네트 계층 구조는 T-RANKNETSEACH에서와 같이, 데이터집합상에 구축된다. 그러면, 수학적 2를 최소화함에 있어서, 질의들은 동일 네트에 출현하는 오브젝트들의 쌍들 간의 질의들에만 국한된다. 직



관적으로, S-GBS는 오브젝트들의 "양호한"(즉, 공정한) 분할이 그와 같은 쌍들 중에 발견될 수 있다고 가정한다.

[0107] 질의 대 계산 복잡도

[0108] 검색당 질의들의 평균 회수로 표현되는, 상이한 알고리즘들의 계산 복잡도가 도 1(b)에 도시된다. F-GBS에 대해서도 S-GBS에 대해서도 어떤 알려진 보장들도 없기는 하지만, 양 알고리즘들은 모든 데이터집합들에 걸친 질의 복잡도의 관점에서 우수하여, 기대치에서 약 10개의 질의 내에 타깃을 찾아낸다. GBS가 이러한 알고리즘들의 어느 쪽에서든 마찬가지로 수행할 것이므로, 이것들은 이것이 정리 1에 의해 예측된 것처럼 더 잘 수행할 것이라는 점을 시사한다. RANKNETSEARCH의 질의 복잡도는 2 내지 10배 더 높은 질의 복잡도에 있다; 그 영향은 c 배가 상수에 대한 랭크 네트 크기의 종속성을 통하여 예상된 것처럼, 고차원 데이터집합들에 대해 더 크다. 최종적으로, MEMORYLESS는 다른 모든 알고리즘들과 비교하여 열등하게 수행한다.

[0109] 도 1에 도시된 바와 같이, 상기 순서는 검색당 수행되는 작동들의 모음 회수로서 측정되는 계산 복잡도에 대하여 완전히 역전된다. 하나의 알고리즘에서 다음 알고리즘까지의 차이들은 크기의 50 내지 100 차수의 범위를 갖는다. F-GBS는 몇몇 데이터집합들에 대해 기대치에서  $10^9$  작동에 가까운 것을 요구한다; 대조적으로, RankNetSearch는 100과 1000 작동 사이의 범위를 갖는다.

[0110] 확장성 및 강건성

[0111] 상기 알고리즘들이 데이터집합 크기에 대해 어떻게 확장성을 갖는지를 연구하기 위해, 알고리즘들은  $R^3$ 에서 무작위로 균등하게 배치된 오브젝트들을 포함하는 합성 데이터집합에 대해 평가될 수 있다. 5개 알고리즘의 질의 및 계산 복잡도가 도 2(a) 및 도 2(b)에 도시된다. 도 2는 데이터집합 크기의 함수로서 (a) 질의 및 (b) 5개 알고리즘의 계산 복잡도를 보여준다. 데이터집합은 반경 1의  $\mathbb{R}^3$  볼로부터 무작위로 균등하게 선택된다. 도 2(c)는 불완전한 오일러 하에서 n의 함수로서의 질의 복잡도를 보여준다.

[0112] 똑같은 불일치들이 도 1에 표시된 알고리즘들 간에 존재한다. 로그 n으로 환산한 선형 성장은 모든 방법들에 대해 엔트로피  $H(\mu)$ 에 대하여 복잡도의 양 측도들 간의 선형 관계성을 내포한다. 도 2(b)는 강건한 RANKNETSEARCH 알고리즘의 질의 복잡도의 플롯을 보여준다.

[0113] 본 발명의 원리를 이용하여 데이터베이스 내의 타깃을 검색하기 위한 제1 방법(400)의 일 실시예가 도 4에 도시된다. 시작 블록(401)은 제어를 기능 블록(410)에게 넘긴다. 기능 블록(410)은 타깃을 둘러싸는 크기를 갖는 노드들의 네트를 구축한다. 기능 블록(410)은 제어를 기능 블록(420)에게 넘겨 주고, 이것은 네트 내로부터의 노드들의 집합을 선택한다. 블록(420)에 이어서, 제어는 기능 블록(430)에게 넘겨지고, 이것은 타깃으로부터 노드들의 집합 내의 각각의 노드까지의 거리들을 비교한다. 제어는 기능 블록(430)으로부터 기능 블록(440)으로 넘겨지고, 이것은 기능 블록(430)의 비교에 따라서 타깃에 최근접한 노드의 선택을 실행한다. 제어는 기능 블록(440)으로부터 기능 블록(450)으로 넘겨지고, 이것은 기능 블록(440) 동안 발생하는 선택에 따라서 타깃을 여전히 둘러싸는 크기로 네트를 감축시킨다. 제어는 기능 블록(450)으로부터 제어 블록(460)으로 넘겨지고, 이것은 네트의 크기가 타깃만을 둘러싸도록 충분히 작아질 때까지 기능 블록들(420, 430, 440, 및 450)의 반복을 야기한다. 네트가 타깃만을 둘러쌀 때, 방법은 정지한다.

[0114] 본 발명의 원리를 이용하여 데이터베이스 내의 타깃을 검색하기 위한 제1 장치의 일 실시예가 도 5에 도시되고, 일반적으로 참조 번호(500)에 의해 표시된다. 장치는 독립형 하드웨어로 구현되거나, 컴퓨터에 의해 실행될 수 있다. 장치는 적어도 타깃을 둘러싸는 크기를 갖는 노드들의 네트를 구축하기 위한 수단(510)을 포함한다. 수단(510)의 출력은 네트 내의 노드들의 집합을 선택하기 위한 수단(520)의 입력과 신호 통신 상태에 있다. 선택 수단(520)의 출력은 타깃으로부터 노드들의 집합 내의 각각의 노드까지의 거리들을 비교하는 비교기 수단(530)의 입력과 신호 통신 상태에 있다. 비교기 수단(530)의 출력은 선택 수단(540)의 입력과 신호 통신 상태에 있는데, 이 선택 수단(540)의 출력은 선택 수단(540)에 응답하여 타깃에 최근접한, 노드들의 집합 내의 노드를 선택한다. 선택 수단(540)의 출력은 선택 수단(540)에 응답하여 타깃을 여전히 둘러싸는 크기로 네트를 감축시키기 위한 수단(550)과 신호 통신 상태에 있다. 감축 수단(550)의 출력은 제어 수단(560)과 신호 통신 상태에 있다. 제어 수단(560)은 선택 수단(520), 비교기 수단(530), 선택 수단(540), 및 감축 수단(550)이 네트의 크기가 타깃만을 둘러싸기에 충분히 작아질 때까지 자신들의 작동들을 반복하도록 야기할 것이다.



- [0115] 본 발명의 원리를 이용하여 데이터베이스 내의 타깃을 검색하기 위한 제2 방법(600)의 실시예가 도 6에 도시된다. 시작 블록(601)은 제어를 기능 블록(610)에게 넘겨준다. 기능 블록(610)은 타깃을 둘러싸는 크기를 갖는 노드들의 네트를 구축한다. 기능 블록(610)은 제어를 기능 블록(620)에게 넘겨 주고, 이것은 네트 내로부터의 노드들의 적어도 하나의 쌍을 선택한다. 블록(620)에 이어서, 제어는 기능 블록(630)에게 넘겨지고, 이것은 소정 회수의 반복들 동안, 타깃으로부터 각각의 적어도 하나의 쌍 노드들 내의 각 노드까지의 거리들을 비교한다. 제어는 기능 블록(630)으로부터 기능 블록(640)에게 넘겨지고, 이것은 소정 횟수의 반복 도중에 걸쳐서, 기능 블록(630)의 비교에 따라서 타깃에 최근접한, 노드들의 각각의 적어도 하나의 쌍 내의 한 노드의 셀렉트를 수행한다. 제어는 기능 블록(640)으로부터 기능 블록(650)에게 넘겨지고, 이것은 기능 블록(640) 동안에 발생하는 셀렉트에 따라서 타깃을 여전히 둘러싸는 크기로 네트를 감축시킨다. 제어는 기능 블록(650)으로부터 제어 블록(660)에게 넘겨지고, 이것은 네트의 크기가 타깃만을 둘러싸기에 충분히 작아질 때까지 기능 블록들(620, 630, 640, 및 650)의 반복을 야기한다. 네트가 타깃만을 둘러쌀 때, 방법은 정지한다.
- [0116] 본 발명의 원리를 이용하여 데이터베이스 내의 타깃을 검색하기 위한 제2 장치의 실시예가 도 7에 도시되고, 일반적으로 참조 번호(700)에 의해 표시된다. 장치는 독립형 하드웨어로 구현되거나, 컴퓨터에 의해 실행될 수 있다. 장치는 적어도 타깃을 둘러싸는 크기를 갖는 노드들의 네트를 구축하기 위한 수단(710)을 포함한다. 수단(710)의 출력은 네트 내의 노드들의 적어도 하나의 쌍을 선택하기 위한 수단(720)의 입력과 신호 통신 상태에 있다. 선택 수단(720)의 출력은 소정 회수의 반복들 동안, 타깃으로부터 노드들의 적어도 하나의 쌍 내의 각각의 노드까지의 거리들을 비교하는 비교기 수단(730)의 입력과 신호 통신 상태에 있다. 비교기 수단(730)의 출력은 셀렉트 수단(740)의 입력과 신호 통신 상태에 있고, 이것은 비교기 수단(730)에 응답하여 타깃에 최근접한, 노드들의 적어도 하나의 쌍 내의 노드를 셀렉트한다. 셀렉트 수단(740)의 출력은 셀렉트 수단(750)에 응답하여 타깃을 여전히 둘러싸는 크기로 네트를 감축시키기 위한 수단(750)과 신호 통신 상태에 있다. 감축 수단(750)의 출력은 제어 수단(760)과 신호 통신 상태에 있다. 제어 수단(760)은 선택 수단(720), 비교기 수단(730), 셀렉트 수단(740), 및 감축 수단(750)이 네트의 크기가 타깃만을 둘러싸기에 충분히 작아질 때까지 자신들의 작동들을 반복하도록 야기한다.
- [0117] 본 발명의 현재 양호한 실시예들의 특정한 특징들 및 양태들을 갖는 하나 이상의 구현들이 제공되었다. 그러나, 기술된 구현들의 특징들 및 양태들은 또한 다른 구현들에 적용될 수 있다. 예를 들어, 이러한 구현들 및 특징들은 다른 비디오 장치들 또는 시스템들의 맥락 하에서 이용될 수 있다. 이 구현들 및 특징들은 표준에서만 사용될 필요는 없다.
- [0118] 본 발명의 원리들의 "일 실시예" 또는 "실시예" 또는 "일 구현" 또는 "구현"뿐만 아니라 이들의 다른 변형들에 대한 본 명세서에서의 언급은, 실시예와 관련하여 설명한 특정한 특징, 구조, 특성 등이 본 발명의 원리들의 적어도 하나의 실시예에 포함된다는 것을 의미한다. 명세서 전체에 걸쳐서 다양한 위치들에서 출현하는 구인 "일 실시예에서" 또는 "실시예에서" 또는 "일 구현에서" 또는 "구현에서" 뿐만 아니라 임의의 다른 변형들은 반드시 모두 동일 실시예를 참조하는 것은 아니다.
- [0119] 본 명세서에서 기술된 구현들은, 예를 들어, 방법 또는 프로세스, 장치, 소프트웨어 프로그램, 데이터스트림, 또는 신호로 구현될 수 있다. 비록 단일 형태 구현의 맥락에서 논의될지라도(예컨대, 방법으로서만 논의될지라도), 논의된 특징들의 구현은 다른 형태들(예컨대, 장치 또는 컴퓨터 소프트웨어 프로그램)로 구현될 수 있다. 장치는, 예컨대 적절한 하드웨어, 소프트웨어, 및 펌웨어로 구현될 수 있다. 방법은, 예컨대 컴퓨터, 마이크로 프로세서, 집적회로, 또는 프로그램 가능 논리 디바이스를 포함하는 처리 디바이스들을 일반적으로 지칭하는, 예컨대 프로세서와 같은 장치에서 구현될 수 있다. 프로세서들은 또한 예컨대 컴퓨터들, 셀폰들, 휴대용/개인용 디지털 지원기기("PDA")들, 및 최종 사용자들 간의 정보 통신을 용이하게 하는 다른 디바이스들과 같은 통신 디바이스들을 포함한다.
- [0120] 여기에 기술된 다양한 프로세스 및 특징들의 구현들은 다양한 상이한 장비 또는 애플리케이션으로 구체화될 수 있다. 이러한 장비의 예들은 웹 서버, 랩톱, 개인용 컴퓨터, 셀폰, PDA, 및 다른 통신 디바이스들을 포함한다. 명확하게도, 장비는 모바일일 수 있으며 심지어 모바일 차량에 설치될 수 있다.
- [0121] 부가적으로, 방법들은 프로세서에 의하여 수행되는 명령어들에 의하여 구현될 수 있으며, 이러한 명령어들(및/또는 구현에 의하여 산출되는 데이터 값들)은 예컨대 집적회로, 소프트웨어 캐리어, 또는 예컨대 하드 디스크, 콤팩트 디스크, RAM, 또는 ROM과 같은 다른 저장 디바이스와 같은, 프로세서 판독가능 매체상에 저장될 수 있다. 명령어들은 프로세서 판독가능 매체상에 유형적으로 구체화되는 애플리케이션 프로그램을 형성할 수 있다. 명령어들은 예컨대 하드웨어, 펌웨어, 소프트웨어 또는 이들의 조합일 수 있다. 명령어들은 예컨대 운영

체제, 개별 애플리케이션, 또는 이들의 조합으로 발견될 수 있다. 따라서, 프로세서는, 예컨대 프로세스를 수행하도록 구성되는 디바이스 및 프로세스를 수행하기 위한 명령어들을 가진(저장 디바이스와 같은) 프로세서 판독가능 매체를 포함하는 디바이스 둘 모두로서 특징지어질 수 있다. 게다가, 프로세서 판독가능 매체는 구현에 의하여 산출되는 데이터 값들을, 명령어들을 대신하여 또는 이들에 부가하여 저장할 수 있다.

[0122]

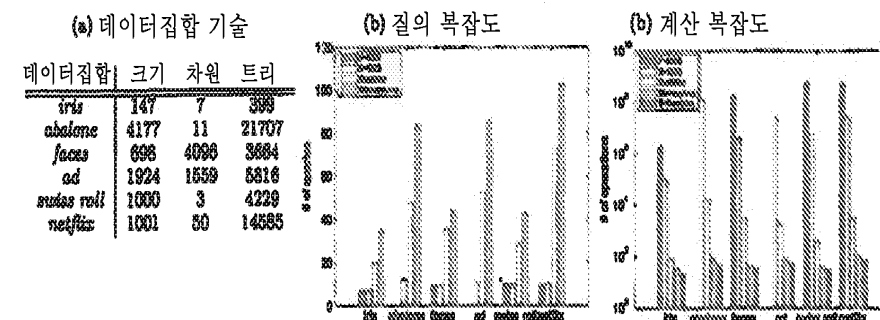
당업자에게 명백한 바와 같이, 구현들은 여기에 기술한 접근법들 중 모두 또는 일부를 사용할 수 있다. 구현들은, 예를 들어, 기술된 실시예들 중 하나에 의해 산출되는 데이터, 또는 방법을 수행하기 위한 명령어들을 포함할 수 있다.

[0123]

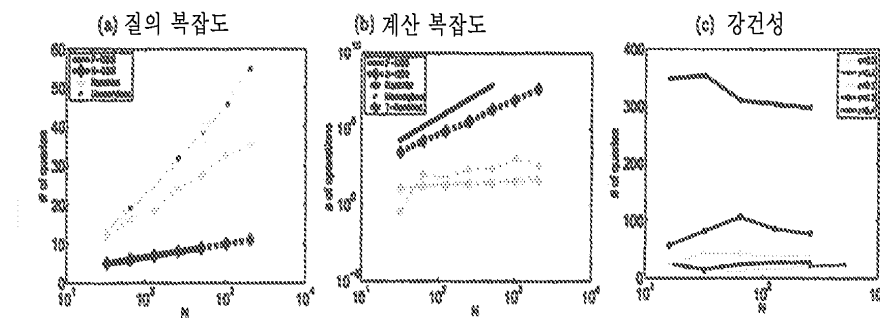
다수의 구현이 설명되었다. 그렇기는 하지만, 다양한 수정이 행해질 수 있다는 것을 이해할 것이다. 예를 들어, 상이한 구현들의 요소들이 조합되거나, 보완되거나, 수정되거나, 제거되어 기타 구현들을 산출할 수 있다. 또한, 당업자는, 기타 구조들 및 프로세스들이 개시된 것을 대체할 수 있고, 그 결과로 생기는 구현들은 개시된 구현들과 적어도 실질적으로 동일한 결과(들)를 달성하도록 적어도 실질적으로 동일한 기능(들)을 적어도 실질적으로 동일한 방식(들)으로 수행할 것임을 이해할 것이다. 따라서, 이들 및 기타 구현들은 본 개시에 의해 상정되고 또한 이들 원리들의 범위 내에 있다.

## 도면

### 도면1



### 도면2



도면3

랭크 넷들을 가진 비교 기반 학습

Algorithm 1 RANKNETSEARCH( $O_i$ )

Input: Oracle  $O_i$

Output: Target  $i$

```

1: Let  $E \leftarrow \mathcal{N}$ ; select arbitrary  $x \in E$ 
2: repeat
3:    $(\mathcal{R}, \{B_y(r_y)\}_{y \in \mathcal{R}}) \leftarrow \text{RANKNET}(x, E)$ 
4:   Find  $y^*$ , the object in  $\mathcal{R}$  closest to  $i$ , using  $O_i$ .
5:   Let  $E \leftarrow B_{y^*}(r_{y^*})$  and  $x \leftarrow y^*$ ;
6: until  $E$  is a singleton
7: return  $y$ 
    
```

Algorithm 2 RANKNET( $x, E$ )

Input: Root object  $x$ , Ball  $E = B_x(R)$

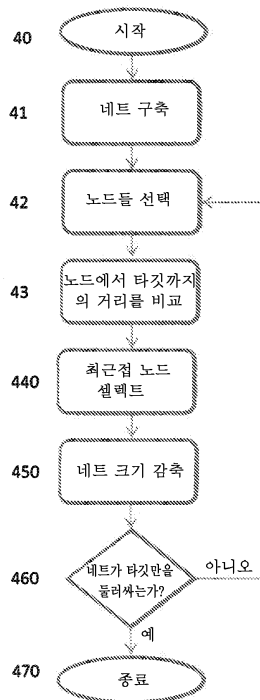
Output:  $\rho$ -rank net  $\mathcal{R}$ , Voronoi balls  $\{B_y(r_y)\}_{y \in \mathcal{R}}$

```

1:  $\rho \leftarrow 1$ 
2: repeat
3:    $\rho \leftarrow \rho/2$ ; construct a  $\rho$ -net  $\mathcal{R}$  of  $E$ 
4:    $\forall y \in \mathcal{R}$ , construct ball  $B_y(r_y)$ 
5:   Let  $I \leftarrow \{y \in \mathcal{R} : |B_y(r_y)| > 1\}$ 
6: until  $I = \emptyset$  or  $\max_{y \in I} \mu(B_y(r_y)) \leq 0.5\mu(E)$ 
7: return  $(\mathcal{R}, \{B_y(r_y)\}_{y \in \mathcal{R}})$ 
    
```

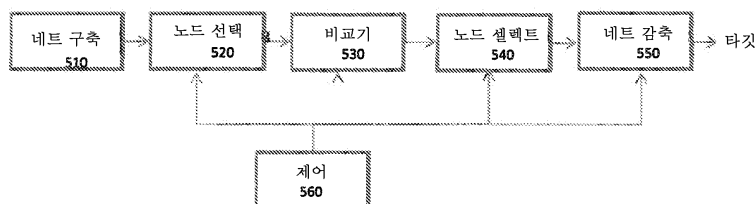
도면4

400



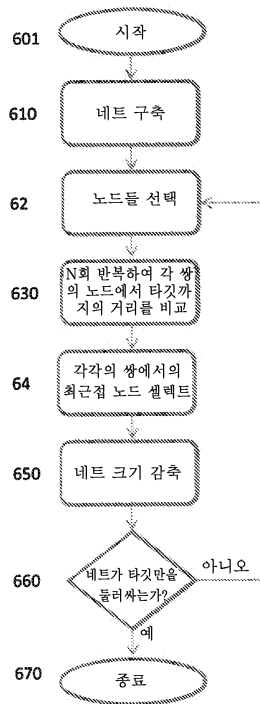
도면5

500



도면6

600



도면7

700

