(AI / 빅데이터 / 블록체인) 문제

|  |  |  |  |
| --- | --- | --- | --- |
| 지역 | 대전 | 조 | 원투쓰리 조 |
| 조원 | 이현경, 유승주, 임현철, 오상민, 정상영 | | |
| 문제 1 | | | |
| Sparse Matrix에 대해 설명하고, 문제점과 해결방법에 대해 말하세요. | | | |
| 출제 키워드 | | Sparse Matrix | |
| 참고 데이터  (사이트, 서적 등) | | - <https://gday2code.tistory.com/34>  - <https://real2won0.tistory.com/8> | |
| 해설 | | | |
| * Sparse matrix(희소행렬) : 행렬 안에 든 원소의 수가 희소한 행렬      * 문제점 : 희소 행렬을 2차원 배열이라는 자료구조에 저장하게 된다면, 희소 행렬의 특성 상 메모리를 상당 부분 낭비하게 되는 결과를 가져온다. * 해결방법 : 2차원 행렬은 a[max\_rows][max\_cols]로 표현 할 수 있다. 희소 행렬의 원소를 저장하기 위해 struct를 사용하고, 전체 행렬을 나타내기 위해 구조체 배열을 사용한다. 구조체를 이용해서 배열을 만들어 주면 간단하게 1차원 배열로도 값을 표현할 수 있다. | | | |

(AI / 빅데이터 / 블록체인) 문제

|  |  |  |  |
| --- | --- | --- | --- |
| 지역 | 대전 | 조 | 원투쓰리 조 |
| 조원 | 이현경, 유승주, 임현철, 오상민, 정상영 | | |
| 문제 1 | | | |
| hierarchy clustering에 대해 설명하세요. | | | |
| 출제 키워드 | | hierarchy clustering | |
| 참고 데이터  (사이트, 서적 등) | | - <https://bcho.tistory.com/1204> | |
| 해설 | | | |
| * Hierarchical clustering(계층적 군집 분석)이란 비슷한 군집끼리 묶어 가면서 최종적으로는 하나의 케이스가 될 때까지 묶는 클러스터링 알고리즘이다.   군집간의 거리를 기반으로 클러스터링을 하는 알고리즘이며, K-mean clustering과 다르게 군집의 수를 미리 정해주지 않아도 된다.   * Ex. “진돗개, 셰퍼드, 요크셔테이어, 푸들, 물소, 젖소” 를 계층적 군집 분석을 하게 되면  1. 중형견, 소형견, 소와 같은 군집으로 3개의 군집으로 묶일 수 있다.      1. 이를 한번 더 군집화하게 되면 [진돗개, 셰퍼드]와 [요크셔테리어, 푸들] 군집은 하나의 군집(개)로 묶일 수 있다.      1. 한번 더 군집화를 하게 되면 전체가 한군집(동물)으로 묶이게 된다.     이렇게 단계별로 계층을 따라가면서 군집을 하는 것을 계층적 군집 분석이라고 할 수 있다. | | | |

(AI / 빅데이터 / 블록체인) 문제

|  |  |  |  |
| --- | --- | --- | --- |
| 지역 | 대전 | 조 | 원투쓰리 |
| 조원 | 이현경, 유승주, 임현철, 오상민, 정상영 | | |
| 문제 1 | | | |
| python pickle 모듈의 기능을 설명하고 pickle모델 함수인 dump, load에 대해 설명하시오. | | | |
| 출제 키워드 | | Python pickle | |
| 참고 데이터  (사이트, 서적 등) | | https://blog.naver.com<https://wikidocs.net/83>/221272342263 | |
| 해설 | | | |
| **pickle** 모듈은 파이썬 객체를 파일로 저장하고 로딩하는 기능을 제공합니다. (한 개의 파일에 한 개이상의 객체를 저장할 수 있다.)   * - dump(object,file,protocol): 파이썬 객체를 저장할 때 쓰는 함수   import pickle list = [1,2,3,4,5] with open('list.pkl', 'wb') as f: pickle.dump(l,f)   * - load(file): 파이썬 객체형태로 저장된 파일을 읽어오는 함수   with open('list.pkl',''rb') as f: list2 = pickle.load(f) print(list2) //[1,2,3,4,5] print(list==list2) // true | | | |

(AI / 빅데이터 / 블록체인) 문제

|  |  |  |  |
| --- | --- | --- | --- |
| 지역 | 대전 | 조 | 원투쓰리 |
| 조원 | 이현경, 유승주, 임현철, 오상민, 정상영 | | |
| 문제 1 | | | |
| K-means Clustering에 대하여 정의하고 시간 복잡도에 대해 고민해 보세요 | | | |
| 출제 키워드 | | K-means Clustering | |
| 참고 데이터  (사이트, 서적 등) | | <https://blog.naver.com/bsw2428/221400954483>  <https://untitledtblog.tistory.com/132> | |
| 해설 | | | |
| **정의**  k-평균 군집화 알고리즘은 분리형 군집화 알고리즘 가운데 하나로, 주어진 데이터를 k개의 클러스터로 묶는 알고리즘이다. 각 군집은 하나의 중심(centroid)를 가진다. 각 개체는 가장 가까운 중심에 할당되며, 같은 중심에 할당된 개체들이 모여 하나의 군집을 형성한다.  **시간 복잡도**  K-means clustering의 최적 해를 찾는 것은 NP-hard 문제에 속한다. 따라서, k-means clustering에서는 Lloyd 알고리즘과 같은 휴리스틱 (heuristic) 기법을 주로 이용한다. 데이터 벡터의 수가 N, 차원이 D인 데이터에 대하여 Lloyd 알고리즘을 이용하는 k-means clustering의 시간 복잡도는 O(NKDI)가 된다. 그러나 보통 K는 N보다 매우 작은 값으로 설정되고, assignment step과 update step의 반복 횟수인 I 또한 N보다 매우 작은 값으로 설정되기 때문에 k-means clustering은 데이터의 수에 대하여 선형 시간복잡도를 갖는다. 이러한 특성 때문에 k-means clustering은 대규모 데이터를 처리하는 데 많이 이용되고 있다. | | | |

(AI / 빅데이터 / 블록체인) 문제

|  |  |  |  |
| --- | --- | --- | --- |
| 지역 | 대전 | 조 | 원투쓰리! 조 |
| 조원 | 이현경, 유승주, 임현철, 오상민, 정상영 | | |
| 문제 1 | | | |
| 클러스터링이란 데이터 포인트의 그룹화와 관련된 알고리즘이다. 이중 대표적인 알고리즘은 3가지 이상 기술하고 각각의 장단점에 대하여 간단히 설명하여라. | | | |
| 출제 키워드 | | **Clustering** | |
| 참고 데이터  (사이트, 서적 등) | | <https://www.nextobe.com/single-post/2018/02/26/%EB%8D%B0%EC%9D%B4%ED%84%B0-%EA%B3%BC%ED%95%99%EC%9E%90%EA%B0%80-%EC%95%8C%EC%95%84%EC%95%BC-%ED%95%A0-5%EA%B0%80%EC%A7%80-%ED%81%B4%EB%9F%AC%EC%8A%A4%ED%84%B0%EB%A7%81-%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98> | |
| 해설 | | | |
| 1. **K-Means Clustering**   K-Means는 우리가 실제로 수행하는 모든 작업이 포인트와 그룹 중앙 사이의 거리를 계산하므로 매우 빠르다는 이점이 있습니다. 매우 적은 계산량으로 선형 복잡도 O ( n )을 갖습니다.  반면 K-Means에는 몇 가지 단점이 있습니다. 첫째, 얼마나 많은 그룹 / 클래스가 있는지 선택해야합니다. 이것은 항상 사소한 것은 아니며 클러스터링 알고리즘을 사용하면 이상적으로 데이터에서 통찰력을 얻는 것이므로 우리를 위해 알고리즘을 파악하는 것이 이상적입니다. K-Means는 또한 클러스터 중심의 무작위 선택으로 시작되므로 알고리즘을 실행할 때 마다 다른 클러스터링 결과를 산출 할 수 있습니다. 따라서 결과가 반복적이지 않고 일관성이 부족할 수 있습니다. 다른 클러스터 방법은 보다 일관성이 있습니다.  K-Medians는 그룹의 중앙 벡터를 사용하는 평균을 사용하여 그룹 중심점을 다시 계산하는 K-Means와 관련된 또 다른 클러스터링 알고리즘입니다. 이 방법은 (Median을 사용하기 때문에) 이상치에 덜 민감하지만 Median 벡터를 계산할 때 각 반복에서 정렬이 필요하므로 더 큰 데이터세트의 경우 훨씬 느립니다.   1. **Mean-Shift Clustering**   Mean shift 클러스터링은 데이터 포인트의 밀집된 영역을 찾기 위해 시도하는 슬라이딩 윈도우 기반 알고리즘입니다. 이는 중심점에 대한 후보를 슬라이딩 윈도우 내의 포인트의 평균으로 업데이트하여 작동하는 각 그룹 / 클래스의 중심점을 찾는 것이 목표라는 것을 의미하는 centroid 기반 알고리즘입니다. 이러한 후보 윈도우는 후 처리 단계에서 필터링되어 거의 중복을 제거하여 최종 세트의 중심점과 해당 그룹을 형성합니다.   K-means 클러스터링과는 대조적으로, 평균 이동이 자동으로 이를 감지하여, 클러스터 수를 선택할 필요가 없습니다. 그것은 엄청난 이점입니다. 클러스터 센터가 최대 밀도 포인트로 수렴한다는 사실은 자연스러운 데이터 중심의 의미에서 이해하기 쉽고 잘 맞는 것처럼 매우 바람직합니다. 단점은 윈도우 크기 / 반경 "r"을 선택하는 것이 중요하지 않을 수 있다는 것입니다. 3. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) DBSCAN은 mean-shift와 비슷한 밀도 기반의 클러스터된 알고리즘이지만 두드러진 장점이 있습니다.  첫째로, 그것은 클러스터 집합을 전혀 필요로하지 않습니다. 또한 데이터 포인트가 매우 다르더라도 이상 값을 단순히 클러스터에 버리는 K-Means와는 달리 이상 치를 노이즈로 식별합니다. 또한 임의로 크기가 정해지고 임의로 모양이 지정된 클러스터를 매우 잘 찾을 수 있습니다.  DBSCAN의 가장 큰 단점은 클러스터의 밀도가 다양 할 때 다른 클러스터와 마찬가지로 잘 수행되지 않는다는 것입니다. 이는 밀도가 변할 때 neighborhood point를 식별하기위한 거리 임계 값 ε 및 minPoints의 설정이 클러스터마다 다양하기 때문입니다. 이 단점은 거리 임계치 ε가 다시 추정하기가 어려워지기 때문에 매우 고차원적인 데이터에서도 발생합니다. 4.  Gaussian Mixture Models (GMM)을 사용한 Expectation-Maximization (EM) Clustering GMM(Gaussian Mixture Models)은 K-Means보다 더 많은 유연성을 제공합니다. GMM을 사용하여 데이터 포인트가 가우스 분포라고 가정합니다. 이는 평균을 사용하여 원형이라고 말하는 것보다 덜 제한적인 가정입니다. 그렇게하면 클러스터의 모양을 나타내는 두가지 파라미터가 있습니다 : 평균 및 표준 편차입니다. 2차원에서 예를 들자면, 이것은 클러스터가 모든 종류의 타원형을 취할 수 있음을 의미합니다 (x와 y 방향 모두에 표  GMM 사용에는 실제로 두가지 주요 이점이 있습니다. 첫째, GMM은 K-Means보다 cluster covariance(공분산) 측면에서 훨씬 유연 합니다 . 표준 편차 파라미터로 인해 클러스터는 원으로 제한되기보다는 타원 모양을 취할 수 있습니다. K-Means는 실제로 모든 차원에서 각 클러스터의 공분산이 0에 가까워지는 GMM의 특별한 경우입니다. 둘째, GMM은 확률을 사용하기 때문에 데이터 포인트 당 여러 개의 클러스터를 가질 수 있습니다. 따라서 데이터 포인트가 두 개의 겹치는 클러스터의 중간에있는 경우 클래스 1에 X- 퍼센트, 클래스 2에 Y- 퍼센트로 속한다고 말하는 것으로 클래스를 정의 할 수 있습니다. 즉, GMM은 mixed membership을 지원 합니다.준 편차가 있으므로). 따라서 각 가우스 분포는 단일 클러스터에 할당됩니다.  **5. Agglomerative Hierarchical Clustering**  Hierarchical 클러스터링 알고리즘은 실제로 하향식 또는 상향식의 두가지 범주로 나뉩니다. 상향식 알고리즘은 각 데이터 포인트를 처음에는 단일 클러스터로 취급한 다음 모든 클러스터가 모든 데이터 포인트를 포함하는 단일 클러스터로 병합될 때까지 클러스터 쌍 을 연속적으로 병합 (또는 집적 )합니다. 따라서 상향식 계층적 클러스터링은 hierarchical agglomerative clustering 또는 HAC 라고 합니다 . 이 클러스터 계층 구조는 트리 (또는 dendrogram)로 표시됩니다. 트리의 루트는 모든 샘플을 수집하는 고유한 클러스터이며 하나의 샘플만 가진 클러스터입니다. 알고리즘 단계로 이동하기 전에 아래 그래픽을보고 그림을 확인하십시오.  Hierarchical 클러스터링에서는 클러스터 수를 지정할 필요가 없으며 트리를 구축 할 때 가장 적합한 클러스터 수를 선택할 수도 있습니다. 또한 이 알고리즘은 거리 메트릭의 선택에 민감하지 않습니다. 이들 모두는 동등하게 잘 작동하는 경향이 있는 반면 다른 클러스터링 알고리즘에서는 거리 메트릭의 선택이 중요합니다. Hierarchical 클러스터링 방법의 특히 좋은 사용 사례는 기본 데이터가 계층 구조를 가지고 있고 계층 구조를 복구하려는 경우입니다. 다른 클러스터링 알고리즘은 이를 수행 할 수 없습니다. Hierarchical 클러스터링의 이러한 이점은 K-Means와 GMM의 선형 복잡성과 달리 O (n³) 의 시간 복잡성을 가지므로 효율성이 낮아지는 단점이 있습니다. | | | |

(AI / 빅데이터 / 블록체인) 문제

|  |  |  |  |
| --- | --- | --- | --- |
| 지역 | 대전 | 조 | 원투쓰리! 조 |
| 조원 | 이현경, 유승주, 임현철, 오상민, 정상영 | | |

|  |  |
| --- | --- |
| 문제 2 | |
| EM clustering에 취약점에 대하여 간단히 설명하고, 이를 대체할 수 있는 방법을 기술하여라. | |
| 출제 키워드 | **EM** **Clustering** |
| 참고 데이터  (사이트, 서적 등) | <https://ko.wikipedia.org/wiki/%EA%B8%B0%EB%8C%93%EA%B0%92_%EC%B5%9C%EB%8C%80%ED%99%94_%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98> |
| 해설 | |
| EM은 극값으로만 수렴하고 일반적으로 수렴율에 대한 범위가 없다. 고차원에서 잘 작동하지 않고 극값의 수가 다수가 존재할 가능성이 있다. 따라서 특히 고차원 상에서 다른 학습 방법을 사용할 필요가 있다. 이 때 모멘트-기반 접근 방식이나 스펙트럼 기법이 더 좋은 효과를 보장한다. 모멘트-기반 접근은[]](https://ko.wikipedia.org/wiki/%EA%B8%B0%EB%8C%93%EA%B0%92_%EC%B5%9C%EB%8C%80%ED%99%94_%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98#cite_note-anandkumar2014tensor-21) 확률 모형의 매개변수를 학습하기 위한 것으로 최근 들어 사용이 많아지기 시작했는데, 이는 EM이 극값만을 보장해주는 것에 반해 모멘트-기반 접근은 최대, 최솟값을 보장하기 때문이다. 혼합 모형, 은닉 마르코프 모델, 공동체 모델 등으로부터 학습을 보장하는 알고리즘들이 유도될 수 있다. 이 스펙트럼 방법들은 잘못된 극값을 찾아내지 않고, 모수의 참값을 특정 제약 조건 하에서 일관적으로 도출시킨다. | |

(AI / 빅데이터 / 블록체인) 문제

|  |  |  |  |
| --- | --- | --- | --- |
| 지역 | 대전 | 조 | 원 투 쓰리! |
| 조원 | 이현경, 유승주, 임현철, 오상민, 정상영 | | |
| 문제 1 | | | |
| 다음 전치행렬의 수식은 무엇이랑 같을까요?.  ,, | | | |
| 출제 키워드 | | Transpose Matrix | |
| 참고 데이터  (사이트, 서적 등) | | Jeffrey Chasnov 교수님 유튜브 강좌  <https://www.youtube.com/results?search_query=lu+decomposition> | |
| 해설 | | | |
| 캡처 | | | |
|  | | | |

\

(AI / 빅데이터 / 블록체인) 문제

|  |  |  |  |
| --- | --- | --- | --- |
| 지역 | 대전 | 조 | 원 투 쓰리! |
| 조원 | 이현경, 유승주, 임현철, 오상민, 정상영 | | |
| 문제 1 | | | |
| Matrix factorization은 LU분해라고도 해요. LU 분해는 수치 해석 분야에서 하삼각행렬과 상삼각행렬의 곱으로 표현하는 것이다. L과 U는 각각 Lower와 Upper를 의미해요 그럼 다음 A 행렬을 LU 분해했을 때 L이 먼지 계산해보세요. U는 알려드릴게요.  A = = LU , L을 맞춰보세요. U = | | | |
| 출제 키워드 | | Matrix Factorization | |
| 참고 데이터  (사이트, 서적 등) | | Jeffrey Chasnov 교수님 유튜브 강좌  <https://www.youtube.com/results?search_query=lu+decomposition> | |
| 해설 | | | |
| C:\Users\multicampus\AppData\Local\Microsoft\Windows\INetCache\Content.Word\W2.7-1.png | | | |
|  | | | |