|  |
| --- |
| **알고리즘 > 인공지능/기계학습** |

|  |  |  |  |
| --- | --- | --- | --- |
| **토픽** | **중토픽** | 키워드 특징 및 설명 | 타연관 토픽 |
| **인공지능/기계학습** | **특이점** |  |  |
| **엑소브레인(Exobrain)** | 엑소브레인(Exobrain) 소프트웨어 |  |
| **딥뷰(Deepview)** | 딥뷰(Deepview) |  |
| **전문가시스템** | 전문가시스템 기술, 구성요소, 추론 |  |
| **기계 학습 (Machine Learing)** | 유전자 알고리즘, 인공지능, 신경망분석, NLQA |  |
| **감독 학습** | 분류(classification), 은닉 마르코프 모델 (Hidden Markov model) 회귀 분석 (Regression), KNN, 신경망 (Neural network),  의사결정트리, 서포트 벡터 머신 (support vector machine) **(암기법: 클라큰신의)** |  |
| **무감독 학습** | 입력값에 대한 목표치가 주어지지 않는 기계학습 클러스터링(Clustering), 독립 성분 분석 (Independent Component Analysis) |  |
| **PCA(주성분분석)** | PCA(Principal Component Analysis) 알고리즘 고차원의 데이터를 저차원의 데이터로 환원시키는 기법이다. 서로 연관 가능성이 있는 고차원 공간의 표본들을 선형 연관성이 없는 저차원 공간(주성분)의 표본으로 변환하기 위해 직교 변환을 사용한다.  - 공분산 행렬, 고유 값 분해(Eigen Value Decomposition; EVD) | 데이터마이닝 |
| **튜링 테스트** |  |  |
| **서포트 벡터 머신 SVM(Support Vector  Machine)** | 기계 학습의 분야 중 하나로 패턴 인식, 자료 분석을 위한 지도 학습 모델이며, 주로 분류와 회귀 분석(하나나 그 이상의 독립변수의 종속변수에 대한 영향의 추정을 할 수 있는 통계기법, regression analysis) 을 위해 사용한다. 두 카테고리 중 어느 하나에 속한 데이터의 집합이 주어졌을 때, SVM 알고리즘은 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단하는 비확률적 이진 선형 분류 모델을 만든다.  - 초평면(Hyperplane), 분류, 회귀분석 - 지지도 이용, 벡터 이용, 기계학습 - 각 분류의 경계선에 가장 가까이 있는 벡터(그림에서 굵은 테두리를 가진 벡터)를 서포트 벡터라고 부름 - http://www.itdaily.kr/news/articleView.html?idxno=69618 | 빅데이터 분류 벡터: 변위(물체가 운동하고 있을 때 물체의 위치 변화), 크기, 방향을 동시에 나타내는 (물리)량 |
| **딥러닝** | 딥러닝 모델 |  |
| **인공지능** | 사례: 왓슨(IBM): 신뢰성 높은 빠른 답변, 구글 브레인 프로젝트: 인간 인지방식 자율학습 연구 |  |
| **베이즈 이론** |  |  |
| **퍼지 이론** | Fuzzy Theory |  |
| **유전자 알고리즘** | 유전자(Genetic) 알고리즘 |  |
| **퍼셉트론(Perceptron)** | 인간의 신경망과 유사하게 만든 입력층, 은닉층, 출력층으로 구성되어진 학습 능력을 가진 신경망 모델 입력층(In1~4), 출력층(Out), 은닉층(가중치, 합, 활성화 함수)로 구성, 비지도 학습을 적용 | **다중 계층 퍼셉트론** |
| **알파고(AlphaGo)** | 딥마인드가 2014년 구글에 인수되면서 본격적인 개발에 착수한 알파고는 완성단계가 아닌 프로토타입 단계로 2016년 현재 개발이 진행중이다.  구글은 알파고 알고리즘을 활용해 기후변화예측, 질병진단 및 건강관리, 무인자율주행차, 스마트폰 개인비서 등 미래의 핵심적 서비스 사업에 적용한다는 계획이다. |  |
| **몬테카를로 트리 탐색** | MCTS(Monte Carlo tree search)은 의사 결정을 위한 체험적 탐색 알고리즘 키워드: 선택 (Selection), 확장 (Expansion), 시뮬레이션 (Simulation), 역전달 (Backpropagation), 플레이아웃'(playout)  - 검색 공간에서 무작위 추출에 기초한 탐색 트리를 확장하는 데 중점을 둔다. 몬테카를로 트리 탐색을 게임에 적용하는 것은 많은 '플레이아웃'(playout)에 기초한다. | **몬테카를로 트리 탐색 (Monte Carlo tree search, MCTS)** |

**[기출문제]**

|  |  |  |
| --- | --- | --- |
| **No** | **기출 문제 및 출제 예상 리스트** | **비고** |
| 1 | 5. 데이터마이닝에서프로토타입 기반의 군집기법인 k-means 알고리즘을 설명하시오. | 관리101-4 |
| 2 | 8. K-means clustering algorithm | 컴시응 99-1 |
| 3 | 10. 데이터 마이닝 기술에서 연관규칙을 찾아주는 Apriori 알고리즘을 예를 들어 설명하시오. | 응용96-1 |
| **모의고사** | | |
| 1 | 5. K-Means 알고리즘은 데이터 마이닝의 대표적 분류기법으로 주어진 데이터를 특정 성질에 기초해서 k개 묶음으로 나누는 대표적 방법이다. 좌료 [x,y] 로 주어진 아래의 입력값과중심값에 대해 K-Means 알고리즘을 수행 후 변화된 중심값과입력값들의 군집을 구하시오.  - 입력값 : A = (1,3) B = (2,2) C = (2,3) D = (3,4) E = (5,6) F = (6,4)  - 중심값 : C1= (2,2) C2= (5,6) | 관리 35-4 |
| 2 | 1. 다음 데이터베이스에서 Apriori Algorithm을 사용하여 최소 지지도 50%이상, 최소 신뢰도 80% 이상인 연관규칙을 찾는 과정의 풀이를 제시하시오. | 관리26-3 |

**[Keyword]**

Data중심점(Centeroid), 거리 기반 분류기법, 유클리디안 거리 측정

군집(Cluster), 중심값(Centroids), 거리기반 분류, 군집갯수 사전 설정, 빠른 결과산출

,Training Set, Code-Vector, Euclidean 거리, 군집(Clustering), Infinite/Finite Training, 반복

**문) K-평균 알고리즘(k-means algorithm)**

답)

Ⅰ. K-평균 알고리즘의 개요

1. **K-평균 알고리즘(k-means algorithm)의 정의**

* 분석 대상 데이터를 특성 별 군집(cluster)으로 나누어 데이터를 분류하는 데이터 마이닝 기법

1. **K-평균 알고리즘의 특징**

|  |  |
| --- | --- |
| 규모가 큰 자료에 적용 용이 | -거리 행렬 불필요 |
| 단계별 군집 다를 수 있음 | -계층적 군집법과 달리 매 단계별 군집 할당이 다를 수 있음. |

Ⅱ. K-means 알고리즘 수행 절차 및 비교

1. **K-means 알고리즘 수행절차**

|  |  |
| --- | --- |
| **수행절차** | **개념도** |
| 1. k 개의 중심 값 임의 선정 |  |
| 2. 각각 데이터와 중심 값 거리 기준으로 소속 군집 결정 |  |
| 3. 분할된 군집에서 중심 값 기준으로 재결정 |  |
| 4. 재정의 된 중심 값 기준으로 재 분류  (군집 경계가 변경되지 않을 때까지 반복) |  |

1. 데이터 분류에 활용하는 데이터 마이닝 방법간 비교

|  |  |  |
| --- | --- | --- |
| **구분** | **의사결정 트리** | **k-means 알고리즘** |
| 특징 | -특정 관점상에서의 분류  -설명력이 우수함 | -특정 값을 이용한 grouping  -거리기반의 분류(군집화) |
| 고려사항 | -데이터의 치우침, 왜곡 심함 | -군집의 개수를 미리 설정 |

Ⅲ. K-means 알고리즘 적용 시 고려사항 및 전망

|  |  |
| --- | --- |
| 고려사항 | -초기 군집 수 결정에 따라 신뢰성 차이 발생  -왜곡된 데이터에 의한 결과의 영향 배제 필요 |
| 전망 | -분석 CRM(고객 세분화), 실시간 WEB 마이닝, BI 2.0(실시간 BI) 등에 활용 예상 |

**[기출문제 풀이]**

|  |  |
| --- | --- |
| **4** | **K-means 알고리즘** |
| 문제 | **데이터마이닝에서프로토타입 기반의 군집기법인 K-means 알고리즘을 설명하시오.** |
| 도메인 | 경영전략 |
| 정의 | K-Means알고리즘 :군집(Clustering) 별 중심값에서 중심과의 거리를 기반으로 데이터를 분류하는 군집분석 기법 |
| 키워드 | 군집(Cluster), 중심값(Centroids), 거리기반 분류, 군집갯수 사전 설정, 빠른 결과산출 |

* **군집분석(Cluster Analysis)**

- 대상들을 분류하기 위한 명확한 기준이 존재하지 않거나 기준이 밝혀지지 않은 상태에서 다양한 특성을 지는 대상자들을 집단으로 분류하는데 사용하는 기법

- 개별 데이터들간의 유사성을 측정하여 유사한 자료를 같은 그룹으로 모으는 기법

- 대상을 군집(Cluster)이라 불리는 상대적으로 동질적인 집단으로 분류하는데 이용되는 기법

* **군집분석 알고리즘의 분류**

|  |  |  |
| --- | --- | --- |
| 분류 | 알고리즘 | 사례 |
| Partitioning methods | - 사전에 결정된 군집들의 수를 사용 하는 것  - 이방법은 레코드들을 각각의 군집에 할당하는 방법  -일반적으로 계산량이 많지 않기 때문에 대량의 데이터베이스에서 유용 | K-Means,  K-medoids, PAM, CRARA, CRARANS |
| 계층적 방법  (Hierarchical methods) | - 이방법은 병합 또는 분할방법을 사용한다. 병합방법은 n개의 군집들을 가지고 시작해서 최종적으로 하나의 군집이 남을 때까지 순차적으로 유사한 군집들을 병합함  - 분할방법은 이와 반대방향으로 작용하는데, 모든 레코드들을 포함하고 있는 하나의 군집에서 출발함  - 이러한 계층적 방법은 목적이 군집들을 자연적인 계층으로 정리하고자 할 때 특히 유용함 | CURE, CHAMELON, BIRCH |
| Model-based methods | - 어떤 정보가 hidden 되어 있는 경우 가장 그럴듯한 모델을 추정할 때 사용하는 효과적인 반복 알고리즘  - 보통 Maximum Likelihood estimate 방법을 사용해서 관측된 데이터에 알맞은 모델의 변수(parameter)를 추정 | EM(Expectation Maximization) |

* 군집기법을 프로토타입기반 군집기법, 계층적기법, 밀도기반 군집기법, 그리드기반 기법으로 구분하기도 함. 프로토타입이라 함은 클러스터의 특징(중심점)을 의미
* **K-means 알고리즘의 정의**

**-** 주어진 데이터를 사전 정의된k개의 클러스터로 묶는 알고리즘, 각 클러스터와 거리차이의 분산을 최소화하는 방식으로 동작.

**-** 군집(Cluster)별 중심값에서 중심과의 거리를 기반으로 데이터를 분류하는 군집 기법의 종류

- 입력 값으로 K를 취하고 군집내 유사성은 높게, 군집간 유사성은 낮게 되도록 N개의 객체집합을 K개의 군집으로 군집하는 기법

* **K-means 알고리즘의 특징**

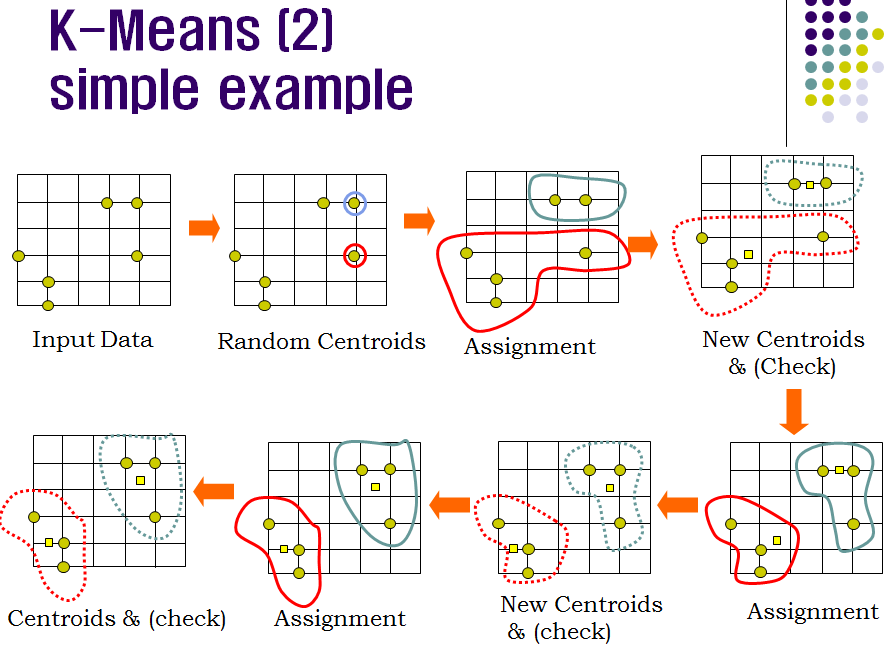
|  |  |
| --- | --- |
| 특징 | 설명 |
| Data중심점(Centeroid) | 군집화된 데이터는 중복되지 않으며 상호배타적 포함 |
| 거리 기반의 분류기법 | 각 군집의 중심점과의 유클리디안 거리 최소화 |
| 데이터군양자화 | 데이터군을 양자화하여 분할시 발생하는 오류 최소화 |
| 속도 및 구현 | 거리기반의 군집기법으로 빠른 결과 산출, 구현 용이성 |

- 군집갯수가 사전에 정의되고, 거리계산이 가능한 Data만 적용

- 간단한 구조, 많은 환경에서 빠르게 수렴(처음 주어진 데이터 개수보다 적은 반복 필요)

- 전역 최적값을 보장하지 않음(맨 처음 K값에 따라서 실제 최적값보다 나쁜 값이 나올수 있음)

* **K-means 알고리즘의 원리**



|  |  |
| --- | --- |
| 단계 | 절차 |
| 1 | 군집의 수 K를 정의 |
| 2 | 초기 K개 군집의 중심(Centroids) 선택 |
| 3 | 각 관측 값들을 가장 가까운 중심의 군집에 할당 |
| 4 | 새로운 군집의 중심 계산 |
| 5 | 재정의 된 중심값 기준으로 다시 거리기반의 군집 재분류 |
| 6 | 군집 경계가 변경되지 않을 때까지 반복 |

* **군집분석을 위한 유사성의 측정(Similarity)**

- 군집으로 분류될 대상들 사이의 일치성 또는 비슷함의 척도

- 주로 유클리디안 거리를 주로 사용

|  |  |
| --- | --- |
| 유사성 측정 척도 | 설명 |
| 유클리디안거리  (=유클리드 거리) | 임의의 두 지점간의 최단거리, 양지점간의 직선거리  각 변수값에서의 차이를 제곱한 것을 합하고 이를 제곱근 |

* **군집분석의 장점과 단점**

|  |  |
| --- | --- |
| 장점 | 단점 |
| 탐색적인 기법  다양한 형태의 데이터에 적용가능  분석방법의 적용 용이성 | 가중치와 거리 정의  초기 군집수의 결정에 민감  결과 해석의 어려움 |

* 구하려는 클러스터의 개수가 미리 정의 필요, 너무 많은 클러스터 개수 설정은 큰 클러스터가 여러 개로 나뉘는 결과 발생 가능
* K-means 군집기법을 사용하기 위해서는 초기에 군집의 수를 미리 제공해야 하는데 군집분석은 데이터를 분석하면서 결정하는 경험적 분석방법이므로 적절한 군집의 수를 안다는 것은 어려운 일.
* 따라서 군집의 수를 변화시켜 가면서 수행된 결과들을 비교해가는 분석방법이 요구(F-검정법등)
* **군집분석(K-means)의 활용사례**
* Data Mining에서 데이터 분류 및 군집 알고리즘으로 활용
* 시장과 고객 분석, 패턴인식, 공간데이터 분석, Text Mining 등
* 최근에는 패턴인식, 음성인식의 기본 알고리즘으로 활용
* 데이터가 불규칙하고 내부 특징이 알려지지 않은 분류 초기 단계에 적합. “끝”

|  |  |  |
| --- | --- | --- |
| **8** | | **K-means clustering algorithm** |
| **출제도메인** | 데이터베이스 | |
| **주요 키워드** | -Training Set, Code-Vector, Euclidean 거리, 군집(Clustering), Infinite/Finite Training, 반복 | |
| **난이도** | ★★★☆☆ (별5개기준) | |
| **참고문헌** | -95회 전자계산기조직응용 김준한 PE서브노트, 위키피디아 | |
| **문제소견** | -어렵지 않은 토픽이나 응용 수험자 입장에서는 접근하기 힘든 문제였을 것으로 보임  - 명확히 답이 있는 문제이므로 절차를 정확히 기술하고, 예시까지 들었을 경우 고득점 가능 | |
| **기출풀이**  **작성기술사** | 박찬혁 기술사(제98회 컴퓨터시스템응용기술사, ffdnc@naver.com) | |

**1. Clustering을 통한 데이터 분류 기법, K-means clustering algorithm의 개요**

**가.K-means clustering algorithm의 정의**

- 데이터를(Training Set)를 기준점(Code-Vector)을 중심으로 Euclidean 거리가 최소가 되도록 K개의 묶음으로 군집(clustering)하여 분류하는 데이터마이닝 기법

**나. K-means clustering algorithm의 특징**

- 반복적 :초기의 잘못된 병합을 알고리즘을 반복 수행하며 회복함

- 알고리즘이 간단하고 대규모 적용이 가능함 :계산시간이 짧음

- 연관성이 높은 데이터는 근거리에 위치하는 특성을 이용(데이터가 패턴 없이 분산 시 비효율적)

**2. K-means clustering algorithm의 개념도와 분석 기법 유형,수행 절차**

**가.K-means clustering algorithm의 개념도와 분석 기법 유형**

|  |  |
| --- | --- |
|  | |
| 유형 | 특징 및 활용 |
| Infinite Training | 중심점에 대한 정보를 지속변화, 추가정보 지속 반영 |
| Finite Training | 일정 데이터로 중심점 결정, 추가데이터는 소속결정에 사용 |

**나. K-means clustering algorithm의 수행 절차**

|  |  |
| --- | --- |
| 수행 절차 | 절차 별 수행 Logic |
| ① 초기설정 | K개의 군집, 초기 Code Vector 결정 |
| ②군집과 분류 | Training Set와 Code-Vector간 거리계산, 최소가 되도록클러스터링 |
| ③ Code-Vector 갱신 | 군집된 데이터를 기준으로 평균을 구하여 Code-Vector 갱신 |
| ④ 검증 및 반복 | Code-Vector의 오차범위 확인, 종료 또는 분류 반복 |

- ②~④를 반복하며, 중심 값의 변화가 없을 경우 알고리즘을 종료함

**3. K-means clustering algorithm의문제점 및 적절한 활용 분야**

**가.K-means clustering algorithm의문제점**

- 초기 조건(중심점위치, 집단수)에 따라 결과가 상이할 수 있음

- 분별력 문제:데이터, 가중치가 모두 동일

- 클러스터링이 원의 형태이면 경우에 따라 애매한 데이터 발생

**나. K-means clustering algorithm의 적절한 활용 분야**

- 트랜드 또는 성향이 불분명한 시장을 분석하는 경우

- 개체가 불규칙적이고, 개체 간 관련성을정확히 알 수 없는 분류초기단계

**[참고자료]**

|  |
| --- |
| **<문제 예시>**  입력 값 : A=[1,3] B = [2,2] C = [2,3] D = [3,4] E = [5,6] F = [6,4] G = [7,3]  중심 값 : C1 = [2,2] C2 = [5,6] |
| **<문제 풀이>**  ①초기설정 :K개의 군집, 초기 Code Vector 결정   * 문제에서 입력 값 7개, 중심 값 기준으로 한 군집 2개가 초기 값으로 주어짐   ②군집과 분류 :Training Set와 Code-Vector간 거리계산, 최소가 되도록클러스터링  1) 거리 계산   * C1[2,2] 에서 부터의 거리 A[1,1] B[0,0] C[0,1] D[1,2] E[3,4] F[4,2] G[5,1] * C2[5,6] 에서 부터의 거리 A[4,3] B[3,4] C[3,3] D[2,2] E[0,0] F[1,2] G[2,3]   2) 클러스터링   * C1에서 가까운 Code-Vector : A, B, C, D * C2에서 가까운 Code-Vector : E, F, G   ③ Code-Vector 갱신 :군집된 데이터를 기준으로 평균을 구하여 Code-Vector 갱신  1) 클러스터링의 중심 좌표 계산   * C1 군집 A, B, C, D의 중심 좌표 [(1+2+2+3)/4, (3+2+3+4)/4] = [2,3] * C1 군집 E, F, G의 중심 좌표 [(5+6+7)/3, (6+4+3)/3] = [6,4.3]   2) 기존의 중심 값 C1[2,2], C2[5,6]와 다르므로 중심 값을 갱신  ④ 검증 및 반복 :Code-Vector의 오차범위 확인, 종료 또는 분류 반복  1) ②부터 반복하여 새로운 중심 값으로 다시 계산하여 클러스터링 수행  2) 클러스터링의 중심 좌표를 다시 계산(C1[2,3] C2[6,4.3]), 변화가 없으므로 알고리즘 종료  \* 중심에 변화가 있으면 알고리즘 반복(②부터 반복) |
| **<최종 결과 도출>**   * **중심 값 : C1[2,3] C2[6,4.3]** * **클러스터링 : {A, B, C, D}, {E, F, G}** |

|  |  |  |
| --- | --- | --- |
| **5** | **K-Means 알고리즘은 군집화(Clustering) 문제를 해결하는 가장 간단한 자율학습 (Unsupervised Learning) 알고리즘 중 하나이다. 사전에 정해진 어떤 수의 클러스터를 통해서 주어진 데이터 집합을 분류하는 간단하고 쉬운 방법이다. 좌표 [x,y]로 주어진 아래의 입력값과 중심값에 대해 K-Means 알고리즘을 수행 후 변화된 중심값과 입력값들의 군집을 구하시오.**  **○ 입력값 : A = (1,3) B = (2,2) C = (2,3) D = (3,4) E = (5,6)**  **F = (6,4)**  **○ 중심값 : G1= (2,2) G2= (5,6)** | |
| **출제도메인** | | 알고리즘 | |
| **주요 키워드** | | 유클리디안 거리, 클러스터링, Centroid, Traning유형 | |
| **출제배경** | | 알고리즘 기출 | |
| **난이도** | | ★ ★ ★ ☆ ☆ (별5개 기준) | |
| **점수가이드** | | ① 알고리즘의 개념 서술(1~9점)  ② K-Means 알고리즘 정의, 특징, 원리 서술 (7~12점)  ③ K-Means 알고리즘 정의, 특징, 원리 서술하고 문제를 풀었으나 답이 틀린 경우(13~15점)  ④ K-Means 알고리즘 정의, 특징, 원리 서술하고 문제를 풀고 답이 맞은 경우 (+α) | |
| **참고문헌** | | 실무 제안서 | |

**1. 군집화(Clustering) 문제를 해결하는 가장 간단한 자율학습 (Unsupervised Learning) 알고리즘**

**K-Means 알고리즘**

**1) K-Means 알고리즘의 정의**

- 주어진 데이터를 특정 성질에 기초해서 k개 묶음으로 나누는 대표적 방법

- 공간상의 점들을 서로 가까운 점들끼리 묶어서 몇 개의 군으로 나누는

군집화기법

**2) K-Means의 특징**

- 각 관찰치를 상호 배반적인 K개의 군집으로 형성

- 초기에 부적절한 병합(분리)이 일어나더라도 이에 대한 회복 가능

- 군집의 수 K를 사전에 정의

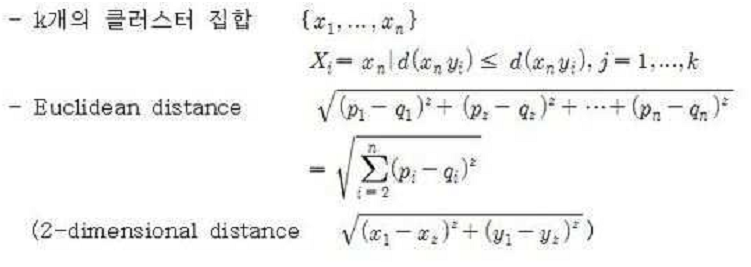
- 알고리즘이 비교적 간단하고 계산시간이 오래 걸리지 않아 대용량 자료에 유용

**3) K-Means 알고리즘의 원리**

|  |  |
| --- | --- |
| **구분** | **주요내용** |
| 그룹 분류 | 각 그룹의 점들은 중심까지의 유클리디안 거리가 최소가 되게 분류 |
| Data Centroid | 클러스터링된 Data는 Overlapping 되지 않으며, 하나의 데이터 군에 포함 |
| 데이터군 양자화 | 데이터군을 양자화 하여 분할시 발생하는 오류를 최소화, 왜곡을 감소 |
| 반복적 최적의 해 | K-Means는 유일한 최적해가 존재하는 것이 아님. 반복법을 통하여 최적의 해를 선택 |

**2. K-Means 알고리즘 수행 절차**

|  |  |  |
| --- | --- | --- |
| http://people.revoledu.com/kardi/tutorial/kMean/Image/Algorithm_clip_image002_0002.gif | | |
| **단계** | **절차** | **수행내용** |
| STEP 1 | 초기화 | 입력값과 클래스값(중심값) 선정.  random하게 cluster centroid를 선택  - 클러스터의 수 k개  - n개의 초기데이터 집합 x1... xn  - k개의 초기 중심 집합 y1...yk |
| STEP 2 | 거리 계산 | . 입력값과 중심값 사이의 거리를 계산**(유클리디안 거리)**  . 각 vector들을 가장 가까운 centroid을 가진 cluster로 분류한다. |



|  |  |  |
| --- | --- | --- |
| **STEP 3** | **군집화** | **. 가까운 중심값을 기준으로 입력값들을 분류**  **. 유클리디안 거리가 최소가 되도록 군집화** |
| STEP 4 | 군집의  중심 좌표 | . 분류된 군집(클래스)의 중심좌표를 계산  . 각 vector들과 cluster centroid 사이의 값을 전부 더한 후 평균을 낸다.  . (X축의 합, Y축의 합)/각 클래스에 속한 입력값의 수 |
| STEP 5 | 중심값  비교 | **. 초기 중심값과 새로 구해진 중심좌표를 비교**  . 4단계에서 나온 결과로 cluster centroid 갱신  - 두 중심값이 같은경우 : 알고리즘 완료   * 두 중심값이 다른경우 : 중심좌표를 신규값으로 변경,   **STEP2 ~ STEP5 를 반복** |

**3.데이터 처리 방식에 따른 두가지 training 방식**

|  |  |
| --- | --- |
| **유형** | **설명** |
| Infinite training | 집단 중심점에 대한 정보를 지속적으로 변화시키면서 추가적으로 들어오는 모든 데이터에 대하여 반복적으로 재해석하여 **집단 중심점을 재파**악하고 모든 데이터의 소속을 수정하면서 진행하는 방식 |
| Finite training | 일정량의 데이터를 미리주고, 주어진 데이터를 토대로 집단 중심점과 데이터들의 소속을 결정하고 난 후에는 추가로 들어오는 데이터에 대해서는 **고정된 중심점 정보를 이용해 소속을 결정짓는 방식** |

**4.풀이과정**

입력값 : A = (1,3) B = (2,2) C = (2,3) D = (3,4) E = (5,6) F = (6,4)

중심값 : G1= (2,2) G2= (5,6)

**가. STEP 1 – 초기화 단계**

- 주어진 문제에서 입력값 6개, 클래스(군집) 2개 임. (중심값이 2개 이므로)

**나. STEP 2 – 거리의 계산**

- 중심값을 기준으로 각 입력값의 X, Y 좌표의 차에 대한 절대값으로 거리 계산

- G1 ~ A = ( 1, 1 ) / G2 ~ A = ( 4, 3 )

- G1 ~ B = ( 0, 0 ) / G2 ~ B = ( 3, 4 )

- G1 ~ C = ( 0, 1 ) / G2 ~ C = ( 3, 3 )

- G1 ~ D = ( 1, 2 ) / G2 ~ D = ( 2, 2 )

- G1 ~ E = ( 3, 4 ) / G2 ~ E = ( 0, 0 )

- G1 ~ F = ( 4, 2 ) / G2 ~ F = ( 1, 2 )

**다. STEP 3 – 군집화**

- 가까운 중심값을 기준으로 각 입력값들을 군집화 (X값과 Y값의 합으로 거리계산)

- G1 에 가까운 입력값 : A, B, C, D

- G2 에 가까운 입력값 : E, F

**라. STEP 4 – 군집의 중심 좌표 계산**

- G1 군집 : A , B , C , D

중심좌표 = [ (1 + 2 + 2 + 3) / 4, (3 + 2 + 3 + 4) / 4 ] = [ 2, 3 ]

- G2 군집 : E, F

중심좌표 = [ (5 + 6) / 2, (6 + 4) / 2 ] = [ 5.5, 5 ]

**마. STEP 5 – 중심값 비교**

- 기존의 중심값과 다르므로 중심값 대체 STEP 2 부터 반복

- 중심값 : G1 = [ 2, 3 ] G2 = [ 5.5, 5 ]

- 거리 계산 후 군집화 하면,

G1 에 가까운 입력값 : A, B, C, D

G2 에 가까운 입력값 : E, F

- 다시 군집의 중심값을 계산하면,

G1 = [ 2, 3 ], G2 = [ 5.5, 5 ]

- 중심값의 변화가 없으므로 알고리즘을 종료함

**바. 도출된 중심값과 군집**

- 중심값 : G1 = [ 2, 3 ], G2 = [ 5.5, 5 ]

- 군집 : { A, B, C, D }, { E, F }

“끝”

“이하 여백”

**Apriori 알고리즘**

**Ⅰ. Apriori알고리즘의 개요**

가. **Apriori알고리즘의 정의**

- 데이터들에 대한 발생 빈도를 기반으로 각 데이터 간의 연관관계를 밝히기 위한 방법

- 구현이 간단하고 성능 또한 만족할 만한 수준을 보여주는 알고리즘으로 패턴 분석을 위해 자주 이용되는 알고리즘

- Apriori 알고리즘는 K번째 항목 집합이 K+1 번째 항목 집합을 발견하기 위해 사용되는 레벨 단위로 진행되는 반복 접근법을 사용

- 어 프라이오리(a priori) : 시스템 기술등을 종합적으로 검토하고 선험성(경험에 선행하는 선천적인 인식)을 높이는 인식

나. **선험적 규칙(Apriori Principle)**

- 모든 항목집합에 대한 지지도를 계산하지 않고, 원하는 빈발 항목 집합을 찾아내는데 이용되는 선험적 규칙은 다음과 같다.

1) 한 항목 집합이 빈발한다면, 이 항목 집합의 모든 부분집합은 역시 빈발 항목 집합이다.

2) 한 항목 집합이 비빈발한다면, 이 항목 집합을 포함하는 모든 집합은 비빈발 항목 집합이다.

ref.)

**<지지도 계산>**

TV 구매가 DVD 구매에 영향을 주는지 확인

|  |  |
| --- | --- |
| 판매품목 | 거래수 |
| TV | 4,000 |
| DVD | 2,000 |
| TV와 DVD 동시 구매 | 1,000 |
| 전체 합계 | 10,000 |

TV 에 대한 DVD 의 지표 분석 (TV를 사면 DVD 도 같이 산다.)

- 지지도란, 전체 거래중 TV 와 DVD를 구매한 사람 비율

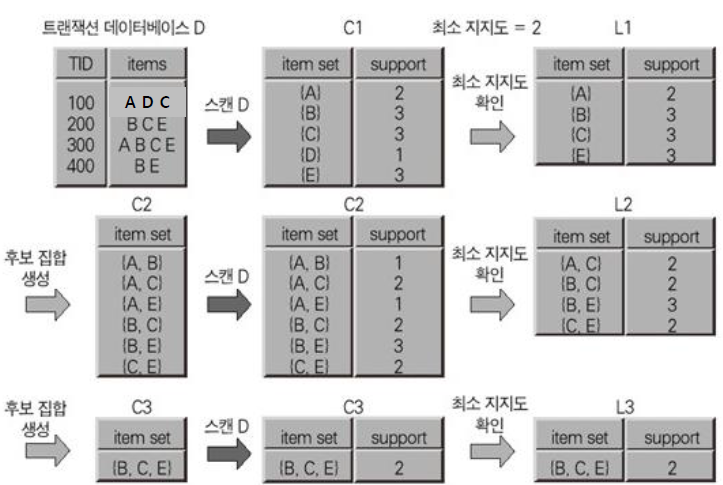
= TV & DVD / 전체 거래수 = 1,000 / 10,000 = 10%

**(X ∪ Y) 를 포함하는 트랜잭션 총수**

**전체 포함하는 트랜잭션 총수**

**지지도 =**

**Ⅱ. Apriori 알고리즘의 수행과정**



- 모든 항목들의 집합을 Ⅰ = {a, b, c, d} 라고 하자. 만일, {b, c, d} 가 빈발 항목 집합이라면 이 항목의 부분집합 {b, c}, {b, d}, {b}, {c}, {d} 는 역시 빈발 항목 집합이 되는데, 이를 선험적 규칙이라 한다. 만일, {a, b} 가 최소 지지도 기준을 넘지 못한 비빈발 항목 집합이라면, 이 집합을 포함하는 {a, b, c}, {a, b, d}, {a, b, c, d} 는 빈발 항목 집합이 될 수 없다. 이 사실을 이용하면 최소 지지도 기준을 넘지 못하는 항목 집합들을 쉽게 가지치기할 수 있는데, 이를 선험적 규칙을 이용한 빈발항목 집합 추출 알고리즘이라고 한다

- 선헙적 알고리즘을 이용한 빈발 항목 집합의 생성은 먼저, 1-항목으로 이루어진 집합(1-항목집합)에 대하여 지지도를 계산한다. 만일, 1-항목집합의 지지도가 최소지지도를 넘지 못하면 모두 버린다. 최소 지지도를 넘는 1-항목집합을 이용하여 불필요한 2-항목집합(2개 항목으로 이루어진 집합)에 대한 가지치기를 하고 2-항목 빈발항목집합 후보를 생성한다. 이 빈발항목 집합 후보들 중에서 최소지지도 기준을 넘는 2-항목 빈발항목집합을 확정하고, 이들을 이용하여 불필요한 3-항목집합에 대한 가지치기를 하고 3-항목 빈발항목 집합 후보를 생성한다. 같은 방법을 k-항목 빈발항목집합을 확정할 때까지 반복한다.

- 선헙적 알고리즘의 특성은 1-항목집합, 2-항목집합, k-항목집합 등 각 수준별(level-wise)로 접근하는 방식이다. 각 수준에서 빈발항목집합 후보를 생성하고, 이 후보들이 지지도를 만족하는지 시험한다(generate-and-test). 알고리즘에서 Ck는 k-빈발항목집합 후보를 의미하고, Lk는 k-빈발항목집합을 의미한다.

“끝”

**Apriori 알고리즘 소개**

연관규칙(Association Rule)의 대표적인 형태의 하나로 현재도 많이 사용되는 알고리즘이 다. 기본 개념은 데이터들에 대한 발생 빈도를 기반으로 각 데이터 간의 연관관계를 밝히기위한 방법을 얘기한다. 에를 들면, 대형 마켓에서 소비자의 물건 구매 패턴을 들 수 있다. 시리얼을 구매하는 고객의 경우, 우유를 같이 구매하는 경우가 많다. 이와 같은 상품들의 연관관계를 분석하면 상품을 전시할 때 좀더 효율적으로 전시할 수 있는 효과를 가져 올 수 있다.

**1. Apriori 알고리즘**

구현이 간단하고 성능 또한 만족할 만한 수준을 보여주는 알고리즘으로 패턴 분석을 위해 자주 이용되는 알고리즘이다.

**1.1 Apriori 연관 규칙**

트랜잭션과 각 트랜잭션에 대한 아이템 리스트가 주어졌을 때, 아이템 집합과 다른 아이템 집합들 사이에 연관관계에 대한 규칙을 찾는 것을 목표로 한다. 우리는 여기서 [표-1]과 같은 시리얼과, 우유와 같은 아이템들에 대한 연관관계에 대한 규칙을 발견하고자 한다.

|  |
| --- |
| 90%의 사람이 시리얼을 구매할 때, 우유를 같이 구매한다. |

연관 관계를 계산은 아이템들의 출현 빈도를 이용하여 계산하게 된다. 여기서는 간단히 아이템들이 동시에 출현하게 될 경우의 확률에 대해서 여기서는 신뢰도 s라고 부르고, 아이템 X가 출현할 때 또 다른 아이템 Y 역시 포함되어 있을 조건부 확률을 신뢰도 c라고 부르겠다.

[표-2]에서 주어진 데이터를 보자. 최소 지지도(Minimum Support)를 50%, 최소 신뢰도(Minimum Confidence)를 50%라고 두고 한번 따라가 보도록 하겠다.

|  |  |
| --- | --- |
| Transaction ID | Items Bought |
| 1 | 우유, 버터, 시리얼 |
| 2 | 우유, 시리얼 |
| 3 | 우유, 빵 |
| 4 | 버터, 맥주, 오징어 |

우선 우유와 시리얼의 연관관계를 살펴보자.

우유를 샀을 때, 시리얼을 살 조건부 확률은 대략 66.6%정도가 나오고, 전체 아이템에서 우유와 시리얼이 동시에 출현할 확률은 50%가 된다. 따라서, 지지도는 50%, 신뢰도는 66.6%가 됨을 알 수 있다. 이와 반대의 경우로 시리얼을 샀을 때, 우유를 동시에 구매할 확률에서는 지지도 50%와 신뢰도 100%가 됨을 알 수 있다. 이를 간단히 표현하면 다음와 같다.

- 우유 => 시리얼 (50%, 66.6%)

- 시리얼 => 우유 (50%, 100%)

연관 규칙 생성은 다음과 같이 2가지 단계를 거쳐 이루어 진다.

**1) 빈발 항목집합들(Large Itemsets) 찾기**

빈발 항목집합들(Large Itemsets)을 찾기 위해서 미리 결정된 최소 지지도(Minimum Support) 이상의 트랜잭션 지지도를 가지는 항목 집합들의 모든 집합들을 빈발 항목 집합들이라고 하며 그 외 모든 항목 집합들은 작은 항목 집합들(Small Itemsets)이라 한다.

**2)연관 규칙 생성**

데이터베이스로부터 연관 규칙을 생성하기 위하여 빈발 항목집합을 이용한다.

지지도(Support)와 신뢰도(Confidence)는 다음의 식을 통해 계산된다.

|  |
| --- |
| DRW000008388c0f  DRW000008388c11 |

이제 [표-2]의 데이터에 대해서 위와 같이 계산해 보면 최소 지지도를 50%로 주었기 때문에 50% 이하의 지지도를 가지는 아이템을 모두 버리면 [표-4]과 같은 아이템 집합이 나온다.

|  |  |
| --- | --- |
| 아이템 집합 | Support |
| {우유} | 75% |
| {버터} | 50% |
| {시리얼} | 50% |
| {우유, 시리얼} | 50% |

이를 이용해 후보집합을 만들면 {우유, 버터}, {우유, 시리얼}, {버터, 시리얼}의 3가지가 됨을 알 수 있다. 이를 다시 이용해 빈발 항목집합을 [표-2]의 데이터를 기반으로 다시 계산하면 [표-5]와 같은 결과가 나오게 된다.

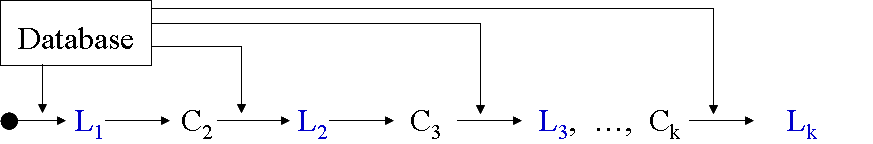
|  |  |
| --- | --- |
| 아이템 집합 | Support |
| {우유, 시리얼} | 50% |

하나의 빈발 항목 집합이 생성되었고, 더 이상 후보집합을 구할 수 없으므로 {우유, 시리얼}이 최종 빈발 항목집합이 된다. 이를 이용해 연관 규칙을 추출하면 [표-6]와 같다. 여기서 신뢰도가 50% 이상인 규칙을 가져오면 원하는 답이 나오게 된다.

|  |  |
| --- | --- |
| 아이템 집합 | Support |
| 우유 =>시리얼 | 2/3 = 66.6% |
| 시리얼 => 우유 | 2/2 = 100% |

**1.2 빈발 항목집합(Large Itemset) 탐색**

빈발 항목집합을 찾는 것은 기본적으로 [그림-2]와 같은 과정을 얘기한다. Database로부터 후보 항목집합(Candidate Itemset)을 생성하고, 이를 데이터베이스 트랜잭션과 비교하여 빈발 항목집합(Large Itemset)을 찾아내는 과정을 반복하게 된다. 최종적으로 빈발 항목집합에 DRW000008388c13가 될 때까지 반복하는 과정을 거친 후 최종 빈발항목집합을 생성해내게 된다.



알고리즘은 [표-7], [표-8]과 같다. 간단한 알고리즘이기에 따로 설명은 하지 않겠다.

|  |
| --- |
| **Input :**  DRW000008388c15// DRW000008388c17-1 크기의 Large Itemset  **Output :**  DRW000008388c19// DRW000008388c1b크기의 후보 집합  **Apriori-gen algorithm :**  DRW000008388c1d= 0;  **for each DRW000008388c1fdo**  **for each DRW000008388c21do**  **if DRW000008388c23of the elements in DRW000008388c25and DRW000008388c27equal then**  DRW000008388c29= DRW000008388c2b{DRW000008388c2d }; |

|  |
| --- |
| **Input :**  DRW000008388c2f// 아이템 집합  DRW000008388c31// 트랜잭션들의 데이터  s // 지지도  **Output :**  DRW000008388c33// Large itemsets  **Apriori algorithm :**  DRW000008388c35= 0; // 스캔 번호  DRW000008388c37= 0;  DRW000008388c39= DRW000008388c3b; // 초기 후보 집합  **repeat**  DRW000008388c3d= DRW000008388c3f+1;  DRW000008388c41= 0;  **for each DRW000008388c43do**  DRW000008388c45= 0;  **for each DRW000008388c47do**  **for each DRW000008388c49do**  **if DRW000008388c4bthen**  DRW000008388c4d= DRW000008388c4f+ 1;  **for each DRW000008388c51do**  **if DRW000008388c53do**  DRW000008388c55= DRW000008388c57;  DRW000008388c59;  DRW000008388c5b= Apriori-Gen(DRW000008388c5d )  **until** DRW000008388c5f= 0; |

**1.3 규칙 생성 알고리즘**

연관 규칙 생성은 앞서 구해진 빈발 항목집합들(Large Itemsets)들을 이용하게 되는데 모든 빈발 항목집합 *L*에 대해서 *L*의 모든 공집합이 아닌 부분집합들을 찾는다. 각각의 그러한 부분집합 A에 대하여, 만약 Supp(*A*)에 대한 Supp(*L*)의 비율이 적어도 최소 신뢰도(Minimum Confidence) 이상이면 *A ->(L-A)*의 형태의 규칙을 출력한다. 연관 규칙에 대한 계산은 [표-3]에 나타난 신뢰도 계산을 통해 최소 신뢰도 이상인 규칙들을 추출하면 된다.

**2. Apriori 대한 고찰**

Apriori 알고리즘은 구현하기 쉽고, 이해하기 쉬우며 어느정도 만족할 만한 결과를 주기 때문에 자주 쓰이고 있다. 그러나, 이 알고리즘은 후보 집합 생성시에 아이템의 개수가 많아지면 계산 복잡도가 엄청나게 증가하게 된다. 대규모의 데이터에 대해서는 FP-Tree를 고려해 볼수 있다. FP-Tree는 후보집합 생성에서 발생되는 cost를 줄여 줄 수 있다.

**KNN 기출문제**

|  |  |
| --- | --- |
| 관리 107-2 | 5. 새로운 Fingerprint를 기존의 클러스터에 매칭하기 위해서 사용되는 KNN(K-Nearest Neighbor) 알고리즘의 동작원리를 설명하고 장단점을 논하시오. |

|  |  |
| --- | --- |
| **토픽** | **KNN(K-Nearest Neighbor)** |
| **상위** | 기계학습알고리즘 |
| **접두어** | 측위기반의 |
| **정의** | 새로운 fingerprint를 기존 클러스터 내의 모든 데이터와 Instance 기반거리를 측정하여 가장 많은 속성을 가진 클러스터에 할당하는 군집알고리즘 |
| **비교,연관** | NN(Nearest Neighbors) |
| **키워드** | 최고인접, 다수결, 가중합계, 유사도(거리)기반, NN개선(noise제거), 유클리디안거리, 마할로노비스의 거리, 다수결(majority voting), 가중합계방식 |

**참고사이트**

<http://blog.naver.com/santalsm/220526205088>

http://yeunhwa.tistory.com/11

<http://javacan.tistory.com/350>

**1. 측위기반의 KNN(K-Nearest Neighbor) 알고리즘의 개요**

**가. KNN 알고리즘의 정의**

- 새로운 fingerprint를 기존 클러스터 내의 모든 데이터와 Instance 기반거리를 측정하여 가장 많은 속성을 가진 클러스터에 할당하는 군집알고리즘

- Sample 에 주어진 x(샘플)에서 가장 가까운 k개의 원소가 많이 속하는 class로 x를 분류하는 비모수적 확률밀도 추정방법

**나. KNN 알고리즘의 특징**

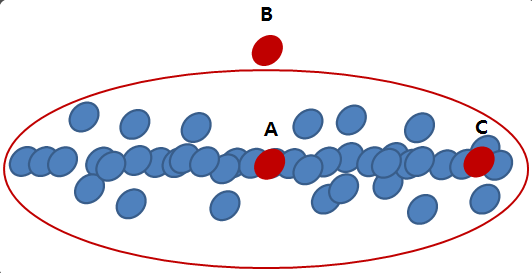
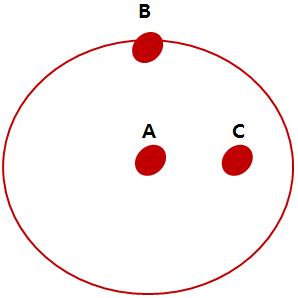
|  |  |
| --- | --- |
| 특징 | 설명 |
| **최고인접 다수결** | 기존 데이터중 가장 유사한 k개의 데이터를 측정하여 분류 |
| **유사도(거리) 기반** | 유클리디안 거리(Euclidian's distance), 마할라노비스의 거리(Mahalanobis' distance), 코사인 유사도(cosine similarity)등을 활용 |
| **Lazy learning 기법** | 새로운 입력 값이 들어온 후 분류시작 |
| **단순유연성** | 모형이 단순하며 파라미터의 가정이 거의 없음 |
| **NN(Nearest Neighbors)개선** | - **KNN**은 **가장 근접한 k개의 데이터에 대한 다수결 내지 가중합계 방식**으로 분류  - **NN의 경우** 새로운 항목을 분류할 때 가장 유사한 instance를 찾아서 같은 class에 일방적으로 분류 시 **잡음 섞인 데이터에는 성능이 좋지 못함** |

**다. KNN 알고리즘에 대한 거리의 개념**

- **유클리디안 거리(Euclidian's distance):** 점과 전간의 최단 거리

- **마할라노비스의 거리(Mahalanobis' distance):** 두 모집단들을 판별하는 문제에서 두 집단 사이의 거리(확률분포를 고려한 거리- 이때 화이트닝 변환(whitening transform)을 사용)

- **코사인 유사도(cosine similarity):** 내적 공간의 두 벡터간 각도의 코사인값을 이용하여 측정된 벡터간의 유사한 정도

- 유클리디안 거리는 A와 B가 가장 가까움

- 마할라노비스의 거리는 A와 C가 가장 가까움(확률분포를 고려)

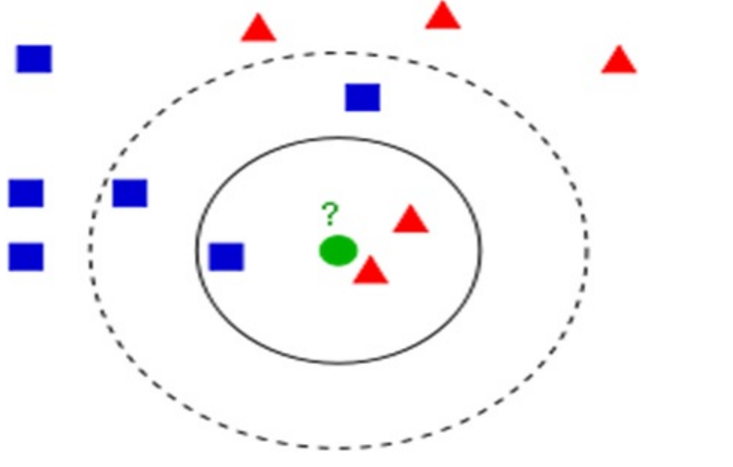
확률 분포를 고려하기 위해 화이트닝 변환을 하며 오른쪽에 A,B,C만 있는 그림이 변환후 모습임

- **이외 KNN에 사용되는 유사도 측정방법: 상관계수, 마니모토점수 등**

**2. KNN 알고리즘의 동작원리**

**가. K값 결정과 분류의 원리**

새로운 fingerprint(원)을 네모 또는 삼각형의 클러스터에 매칭원리



1) 새로운 fingerprint입력(물음표, 동그라미) 확인

2) 거리기반 k개 데이터를 training set에서 추출

3) 추출데이터들의 클러스터, lable 확인

4) 다수결(Majority voting) 의한 클러스터 매칭

**- 결과 새로운 fingerprint는 K가 3인 경우 삼각형, 5인 경우 사각형 클러스터에 매칭됨**

**나. KNN 알고리즘의 상세 동작원리**

|  |  |
| --- | --- |
| 동작원리 | 설명 |
| **fingerprint확인** | - 새로운 입력값 확인  - 가까운 데이터는 같은 **Label(클러스터 가능성 큼)**  - 기존의 모든 데이터와 새로운 fingerprint와 비교준비 |
| **명목변수기반 그룹분류** | - 기존의 저장되어있는 데이터 셋의 label화  - 서로 다른 범주 데이터를 정규화 수행  - 분류기 검사 수행  예: 데이터의 90%를 훈련데이터 10%를 테스트로 활용 |
| **거리측정** | - 유클리디언 거리  - 메모리기반 fingerprint와 모든 데이터간의 거리계산  - 계산된 거리의 정렬수행 |
| **K 선정** | - 양의 정수값, 정렬된 거리중 가장 가까운 k개 데이터 선정  - 여러 k값을 모델링 후 가장 성능이 좋은 k값 선정  - 노이즈 클수록 큰 k값 선정이 좋음  - 작은 k는 극단값 및 노이즈를 허용하여 클러스터링 오류가능 |
| **클러스터 매칭** | - 명목 데이터 경우, 다수결(majority voting)기반의 클러스터 매칭 수행, k개 데이터가 많이 속해있는 클러스터로 새로운 값을 분류  - 수치형 데이터의 경우 k개 데이터의 평균(or 가중평균)을 이용하여 클러스터 매칭 |

**3. KNN 알고리즘의 장단점**

**가. 장점**

|  |  |
| --- | --- |
| 장점 | 설명 |
| **효율성** | 훈련데이터에 잡음이 있는 경우에도 적용가능 |
| **결과일관성** | - 훈련데이터의 크기가 클 경우 효율적임  (데이터의 수가 무한대로 증가시 오차율이 베이즈 오차율의 두배보다 항상 나쁘지 않음을 보장)  - 임의의 k값에 대해 베이즈 오차율에 항상 근접함을 보장 |
| **학습간단** | - 모형이 단순하고 쉬운 구현 가능 |
| **유연한 경계** | - 거리의 변형, 가중치 적용용이  - 유클리디언,코사인유사도, 가중치 적용, 정규화적용 용이 |
| **모델의 유연성** | - 데이터에 대한 가정 반영 및 변형이 간편  - 변형데이터의 training data set 기반 분류기 검증용이 |
| **높은 정확도** | - 사례기반으로 높은 정확성  - 훈련데이터 클수록 클러스터 매칭 정확성 좋아짐 |

**나. 단점**

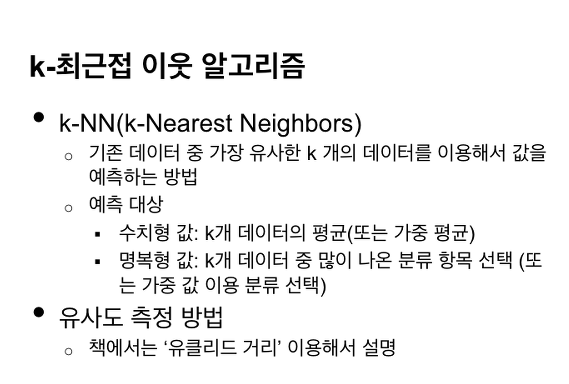
|  |  |
| --- | --- |
| 단점 | 설명 |
| **성능가변성** | -k값 선정에 따라 알고리즘의 성능이 좌우됨  -k값 최적화, under/overfitting의 고려필요 |
| **높은 자원요구량** | -데이터 셋 전체를 읽어서 메모리에 기억  -새로운 개체 n을 읽어서 메모리 내의 데이터 셋과 비교 |
| **고비용** | -모든 훈련샘플과의 거리를 계산하여야 하므로 연산비용(cost)이 높음 |
| **공간예측 부정확** | - 공간정보 예측모델에서는 영향변수 많이 적용이 어려움 |
| **거리계산 복잡성** | - 모든 데이터와의 유사도, 거리측정 수행필요 |
| **노이즈에 약함** | - 노이즈로 인해 큰 k 설정을 필요로 함  - 민감하고 작은 데이터 무시되는 under fitting 문제야기 |

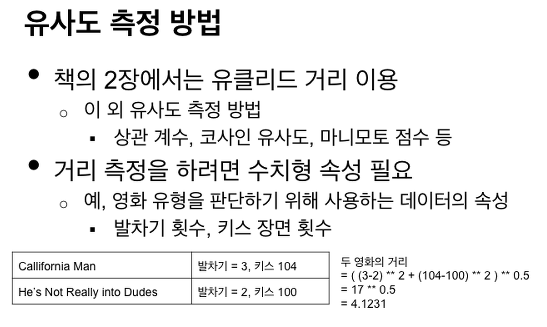
**4. KNN 알고리즘의 활용방안**

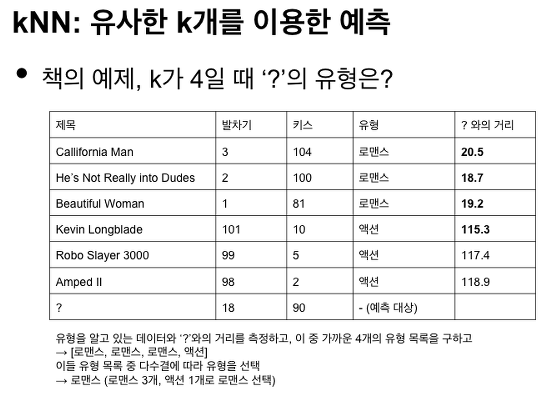
|  |  |  |
| --- | --- | --- |
| 활용방안 | 설명 | 사례 |
| **위치측위** | 이동객체의 위치에서 AP 신호강도를 측정하고 이를 KNN 알고리즘을 활용하여 이동객체의 위치를 최종적으로 추정 | **MS사의 RADAR - Ekahau Inc의 Wi-Fi RLS, COMPAS** |
| **선호도 분류** | 사용자의 추천정보 기반 성향/구매패턴분류 | **내용기반 추천 시스템** |
| **데이터필터링** | 포털등의 중복, 유사 게시글 필터링 | **문서분류(빈발항목집합, 빈발단어집합 등)** |
| **고속도로 통행시간 예측** | TCS 교통량 및 DSRC 구간 통행시간의 실시간 자료를 KNN 기반으로 분석 | **차량 근거리 무선통신(DSRC) 활용 통행시간 예측** |

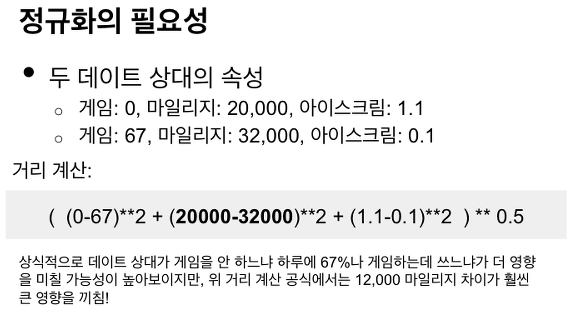
**“끝”**

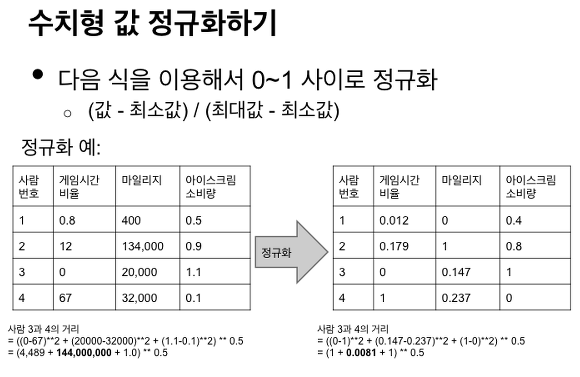
**참고: KNN의 개념**

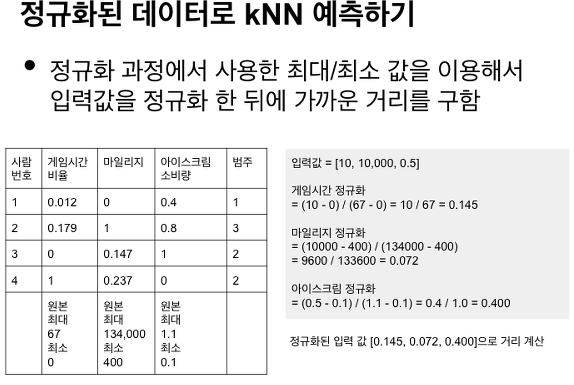
****







****

****

**107회 기출풀이**

시스템 생성 대체 텍스트:
107
 
 
 
 
 
 
 
 
 
 
 
k
p
c
 
(
h
t
t
p
:
/
/
c
a
f
e
.
n
a
v
e
r
.
c
o
m
/
8
1
t
h
 
KPC 
 
 
5 
KNN 
 
 
 Fingerprint
 
 
 
 
 
  KNN(K-
Nearest Neighbor) 
 
 
 
 
. 
 
 
, 
 
 
Sample
 
 x
 
 
 k
 
 
 
 class
 x
 
 
 
 
 
 
 
Lazy Learning, Overfitting/Underfitting 
 
 
 
 
, 
(Fingerprint)
 
 
 
 
 
 
 KNN
 
 
 
 
 
-KNN
 
 
 
 
 
 
 
 
 
-
 k
 
 
 KNN
 
/
 
 
 
 
 
 
 
 
 
 
 
 
 
 
(
 
, 2013), 
 
(
 
, 2015.07) 
AP
 
 WLAN
 
 
(
, 2011) 
k-NN
(
) 
 
1. 
 
 
, KNN(K-Nears Neighbor) 
2. KNN 
 
 
3. KNN 
 
 
4. KNN 
 
 
 
 PE (
105
 
 / crazypterpan@naver.com) 
1.
 
 
 
, KNN(K-Nearest Neighbor) 
 
. KNN
 
 
 - Sample
 
 x
 
 
 k
 
 
 
 class
 x
 
 
  
 
 
 
 
 
  - k
 
 
 
(Nearest Neighbor)
 
 
 
 
 
 
  
. KNN
 
 
 
 
NN(Nearest 
Neighbors)
 
-k-NN
 
 
 k
 
 
 
 
 
 
 
 
*NN
 
 
 
 
 
 
 
 Instance
 
 
 
 class
 
 
 
 
 
 
 
 
 
 
시스템 생성 대체 텍스트:
107
 
 
 
 
 
 
15 
 
KPC 
 
 
-
 
 
 
 
 
 
 
(Lazy Learining) 
-
 
 
 
 
 
 
 
 
 
    - 
 
 
 
 k
 
, 
 
 
 
 
         
2.
 
KNN(K-Nearest Neighbor) 
 
 
 
 
Fingerprint DB 
 
-Fingerprint DB
 : 
 
(RP,Reference Point)
 
,  
 
 
AP
 
 
 
 
 
Fingerprint DB
 
 
 
 
-Fingerprint DB
 
 
  
-
 
 
 
 
 
 
(
) 
 
-
 
(Fingerprint) x
 
 
 {x
1
, x
2
, ... , x
n
}
 
 
. (
 
 
 
) 
-
 
 
 
 
(
 
) 
 
* 
 
 
 
  dist(p, q) = 
 
  - p
 q
 
 
, p
n
 p
 n
 
 
 
 
(k
 
) 
-
 
 
 
 
 
 
 k
 
 
 ex) k = N^1/2 
-
 k
 Overfitting 
 
, 
 
 
 
 
 
. 
 
 
-
 k
 Unverfitting 
 
. 
 
 
 
 
 
 
(
 
) 
-
 
 
 
 : 
 
 
 
  
 k
 
 
 
 
 
 k
 
 
 
 
 
 
* 
 : N(x) = {x
1
, x
2
, ... , x
k
} 
Label
 
 
 
 
 
 
 
 Label
 
 
* Label
 : y(x
1
), y(x
2
), ... , y(x
k
) 
 
 
 Label
 
 
 
 
 
 
 
 x
  
 
 
 
    -KNN 
 
 1)k
 
, 2)Underfiting 
 Overfitting
 
 
 
 
 
3.
 
KNN 
 
/
 
 
 
 
 
-
 
 
 
 
 
 
 
 
-
 
 
 
 
 
 
 
 
 
 
 
 
 
 
(
 
 
 
 
) 
-
 k
 
 
 
 
 
 
 
 
 
 
-k
 
 
 
 
 
 


시스템 생성 대체 텍스트:
107
 
 
 
 
 
 
 
 
 
 
 
k
p
c
 
(
h
t
t
p
:
/
/
c
a
f
e
.
n
a
v
e
r
.
c
o
m
/
8
1
t
h
 
KPC 
 
-k
 
, Under/Overfitting
 
 
 
 
 
-
 
 
 
 
 
.  
-
 
 n
 
 
 
 
 
 
 
 
-
 
 
 
 
 
(cost)
 
 
     -KNN
 
 
 
 
 
 
, 
 
 
 
 
 
      KD
(
 
)
 
 
 
 
 
4.
 
KNN 
 
 
 
 
 
 
 
 AP
 
 
KNN 
 
 
 
 
 
 
-MS
 RADAR 
-Ekahau Inc.
 Wi-
Fi 
RTLS, COMPAS 
 
 
 
 
 
/
 
 
-
 
 
 
 
 
 
, 
 
 
 
-
(
, 
 
) 
 
 
-TCS
 
 DSRC
 
 
 KNN
 
 
-
 
 
(DSRC) 
 
 
 
 
 
 
 
 
 
 
 
 
 
 
 
 
 
 
 
 
 
 
 
 
 

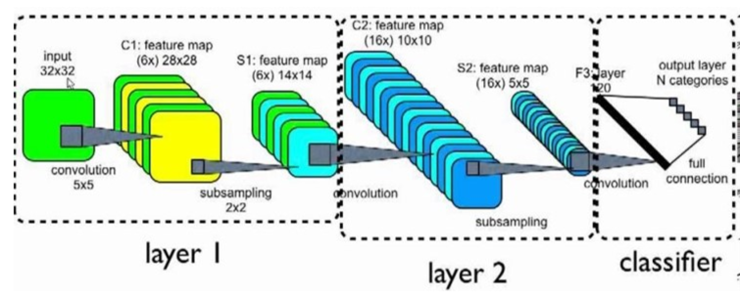

**CNN (Convolutional Neural Networks)**

**1. CNN(Convolutional Neural Networks) 의 개념**

- 영상 인식에 적용이 용이하도록 만들어진 인공 신경망의 한 종류

- 일반 다층퍼셉트론에서사용되는 구조와 다르게 컨볼루션 레이어와 풀링 레이어로 구성

**2. CNN 구조도 및 동작원리**



- Convolution: 대상을 일부분을 여러 개의 필터를 사용하여 재 구성, 이때 활성화(Activation) 함수로서ReLU 함수를 사용

- Max pooling : 구성된 컨볼루션 레이어를 샘플링을 통해 재 구성

   - Classifier를 통해 최종 영상 인식

**RNN (Recurrent Neural Networks)**

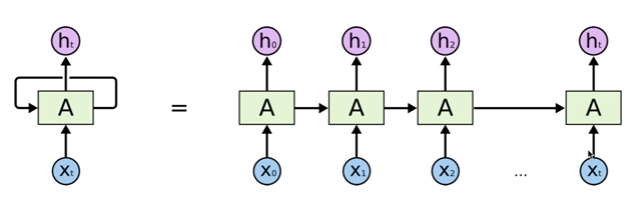
**1. RNN(Recurrent Neural Networks)의 개념**

- 음성이나언어 등 연속된 입력 데이터에 적용이 용이한 딥러닝 모델

- 연속된데이터 상에서 이전 순서의 히든 노드(hidden node)의 값을 저장 이후다음 순서의 입력 데이터로 학습할 때 이전에 저장해 놓은 값을 이용

- 과거학습의 정보를 잃지 않고 연속적인 정보의 흐름을 학습에 반영

**2. RNN의 구조도 및 동작 원리**



- 좌측의 recurrent한 구조를 펼치면 우측의 시계열구조가 됨

**3. mRNN의 한계 및 진화된 모델**

- RNN은 학습방법의 역전파 거리가 늘어나면 gradient 값이 폭증하거나 사라지는 문제점이 발생

  - 이를 극복하기 위해 LTSM(Long short term memory) 및 GRU(Gated Recurrent Unit)이 주로 사용되어짐