핸즈온 머신러닝

PART5. 데이터의 유사도와 비유사도 측정

- 군집화, 이상치 분석, 최근접 이웃 분류와 같은 데이터 마이닝 응용분야에서 상호간 비교를 통해 어느 정도 같거나 다른지를 평가하는 방법이 필요함

○ 군집

- 데이터 객체의 집합

- 한 개 군집 내의 객체는 서로 유사함

- 다른 군집의 객체와는 서로 다른 점이 있음

- 이상치 분석은 군집 기반 기술을 사용하여 나머지 값과 상이한 객체를 잠정적 이상치로 구분

- 객체유사도에 대한 지식은 최근접 이웃 분류 스키마에서 사용됨

○ 유사도와 비유사도

- 유사도와 비유사도는 상호 관련이 있음

- 2개 객체 i와 j가 다를 경우 이에 대한 유사도 측정값은 0

- 유사도 값이 커질수록, 객체 사이의 유사한 정도는 큼(일반적으로 값 1은 완전일치를 말함)

- 비유사도는 반대의 성질

핸즈온 머신러닝

PART5. 데이터의 유사도와 비유사도 측정

2.4.1 데이터 행렬 대 차이 행렬

- 이전 글에서 다뤘던 1차원 객체와 달리 다차원 속성으로 표현된 객체를 다룸
- n개의 객체(사람, 상품, 과정)는 p개의 속성(연령, 신장, 체중, 성별과 같은 측정값이나 특성)에 의해 기술한다고 가정
- 객체는 $x_1=(x_{11},x_{12},\ldots,x_{1p}),x_2=(x_{21},x_{22},\ldots,x_{2p})$ 등으로 나타냄
- x_{ij}는 객체 x_i의 j번째 속성 값
- 관계형 DB에서 객체는 튜플이며, 데이터 샘플이나 속성벡터라고 함
- 주 메모리 기반 클러스터링과 최근접 알고리즘은 다음 2개의 데이터 구조 중 한 개에 근거하여 실행함

(1) 데이터 행렬(객체와 속성 구조)
- n개의 데이터 객체를 관계형 표나 n행(객체)과 p열(속성)의 행렬 형태로 저장
- 각 행은 객체 1개에 해당
- 표기법의 일환으로 p개의 속성을 인덱스하기 위해 f개를 사용할 수 있음
- 데이터 행렬을 2개 모드 행렬이라고도 함

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

핸즈온 머신러닝

PART5. 데이터의 유사도와 비유사도 측정

(2) 차이 행렬(객체와 객체 구조)

– n개 객체의 모든 쌍에 대해 근접도를 저장

– 종종 n대 n표로 표시

$$\begin{bmatrix} 0 & & & \\ d(2,1) & 0 & & \\ d(3,1) & d(3,2) & 0 & \\ \dots & \dots & \dots & \\ d(n,1) & d(n,2) & \dots\ \dots & 0 \end{bmatrix}$$

– d(i, j)는 객체 i와 j간의 비유사도를 측정한 값

– 객체 i와 j가 유사한 경우 0에 가까운 음이 아닌 값을 가짐, 큰 값을 가질수록 차이가 큼

– d(i, j) = d(j, i) : 대칭행렬

– sim(i, j) = 1 – d(i, j) : sim(i, j)는 유사도

– 차이 행렬을 1개 모드 행렬이라고도 함

핸즈온 머신러닝

PART5. 데이터의 유사도와 비유사도 측정

2.4.2 명목 속성의 유사도 측정

- 명목 속성은 2개 이상의 값을 가짐

ex) map_color는 명목 속성으로 빨강, 노랑, 녹색, 분홍, 파랑색과 같은 5개의 상태를 가짐

- 명목 속성의 상태 개수를 M개라고 할 때, 이 상태값은 1, 2, ..., M과 같이 알파벳 문자, 기호, 정수로 표시할 수 있음
- 정수로 표시할 경우 정수는 데이터 핸들링을 위해 사용하며 특별한 의미는 없음

○ 명목 속성을 갖는 객체 간 차이 계산

- 두 객체 i와 j 간 차이는 비일치 비율에 근거하여 계산할 수 있음

$$d(i, j) = \frac{p - m}{p}$$

- m은 일치하는 속성의 수(i와 j가 동일한 상태에 있는 속성의 개수)
- p는 객체를 표현하는 전체 속성의 수
- 가중치는 m개의 효과를 증가시키기 위해 사용함
- 유사도는 다음과 같음

$$sim(i, j) = 1 - d(i, j) = \frac{m}{p}$$

※ 명목 속성 객체 간 유사도는 인코딩을 하여 계산함

- M개의 상태에 대해 각각 이진값을 생성함으로써 비대칭 이진 속성을 인코딩함
- 특정 상태값을 갖는 객체에 대해 이진 속성값은 1이고 다른 이진 속성값은 0이 됨

ex) map_color를 인코딩하기 위해 이진 속성값은 5개의 색상에 대해 생성 가능

노란색 속성은 1로 설정, 나머지는 0으로 설정과 같이 인코딩

핸즈온 머신러닝

PART5. 데이터의 유사도와 비유사도 측정

2.4.3 이진 속성의 유사도 측정

- 대칭이나 비대칭 이진 속성으로 표현된 객체의 유사점과 비유사도 측정값을 구함
- 이진 속성은 2개의 상태값 중 하나의 값 (0, 1)을 가짐
- 0은 해당 속성이 없음을 의미, 1은 있음을 의미
- 이진 속성값을 숫자로 처리하면 분석오류를 범할 수도 있음 => 이진 데이터에 대해 특화된 방법을 사용하여 비유사도를 계산해야 함

○ 2개 이진 속성간 비유사도 계산

(1) 이진 데이터로 구성된 차이 행렬을 계산

- 모든 이진 속성이 동일한 가중치를 가질 경우 다음 표와 같은 분할표(contigency table)를 계산할 수 있음

	Object j			
Object i		1	0	sum
	1	q	r	q + r
	0	s	t	s + t
	sum	q + s	r + t	p

- 대칭 이진 속성은 각 상태가 똑같은 값을 가짐
- 대칭이진 속성에 근거한 비유사도는 대칭이진차이도(symmetric binary dissimilarity)라 함
- 객체 i와 j가 대칭이진 속성이면 i와 j간 차이는 다음과 같음

$$d(i, j) = \frac{r + s}{q + r + s + t}$$

핸즈온 머신러닝

PART5. 데이터의 유사도와 비유사도 측정

○ 비대칭 이진 속성

- 2개 상태의 중요도가 동일하지 않음

ex) 질병검사결과 양성(1)가 음성(0) 판정에서 양성(1) 판정이 더 중요함

- 이와 같은 이진 속성은 상태값이 하나임을 의미하는 "monary"라고 함
- 이러한 속성에 근거한 차이는 비대칭 이진 차이라하고 음성 판정의 일치 개수 t는 중요하지 않으므로 계산식에서 생략함

$$d(i, j) = \frac{r + s}{q + r + s}$$

※ 2개 이진 속성 간 차이는 차이도 대신 유사도의 개념에 의해 계산할 수 있음

ex) 객체 i와 j의 비대칭 이진 유사도

$$sim(i, j) = \frac{q}{q + r + s} = 1 - d(i, j)$$

- sim(i, j)의 계수는 자카드 계수(Jaccard coefficient)
- 대칭과 비대칭 이진 속성이 동일한 데이터 집합에서 발생하면 혼합속성접근방법을 사용함

핸즈온 머신러닝

PART5. 데이터의 유사도와 비유사도 측정

2.4.4 수치형 데이터의 유사도(민코브스키 거리)

○ 수치형 속성으로 표시한 객체간의 비유사도 계산
- 일반적으로 유클리드, 맨하탄, 민코브스키 거리가 있음
- 거리를 계산하기 전에 데이터를 정규화함 => 데이터를 좀 더 작은 범위인 [-1, 1]이나 [0.0, 1.0]으로 변환
- 데이터를 정규화하면 모든 속성은 동일한 가중치를 적용한 결과가 됨(단, 특정분야에서 효과를 못 보기도 함)

(1) 유클리드 거리(euclidean distance)
- 가장 유명한 거리측정방법
- 최단거리라고 함
- $i=(x_{i1},x_{i2},\ldots,x_{ip})$와 $j=(x_{j1},x_{j2},\ldots,x_{jp})$는 p개의 숫자 속성값
- 객체 i와 j 간 유클리드 거리는 다음과 같은 식으로 계산

$$d(i,j)=\sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\ldots+(x_{ip}-x_{jp})^2}$$

(2) 맨하탄 거리(시티블록거리)
- 도시의 두 지점 간 블록 거리

ex) 2개의 블록아래, 3개 블록 지나 총 5개 블록
- 다음 식으로 계산함

$$d(i,j)=|x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+\ldots+|x_{ip}-x_{jp}|$$

핸즈온 머신러닝

PART5. 데이터의 유사도와 비유사도 측정

(3) 민코브스키 거리

- 유클리드와 맨하탄 거리를 일반화한 방법

$$d(i,j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \ldots + |x_{ip} - x_{jp}|^h}$$

- h는 h>=1인 실수
- 거리는 L_p norm이라 하며 p는 h에 대한 표기
- p는 속성의 개수
- 맨하탄 거리는 h=1인 경우
- 유클리드 거리는 h=2인 경우

○ 최소상계거리(supremum distance)

- h가 무한대일 때 민코브스키 거리를 일반화한 결과
- 체비세프 거리라고도 함
- 계산을 위해서는 2개 객체 간 값의 최대차이를 보이는 속성 f를 찾아내야 함
- 이 차이는 최소상계 거리로 좀 더 형식적으로 다음과 같이 정의

$$d(i,j) = \lim_{h\to\infty}\left(\sum_{f=1}^{p}|x_{if} - x_{jf}|^h\right)^{\frac{1}{h}} = \max_{f\to p}|x_{if} - x_{jf}|$$

핸즈온 머신러닝

PART5. 데이터의 유사도와 비유사도 측정

2.4.5 서열 속성에 대한 유사도 측정

- 서열 속성값은 유의미한 순서나 서열을 가짐
- 하지만 연속값 간의 차이는 알 수 없음

ex) 서열 속성값 : 소, 중, 대

- 서열 속성은 실제값을 개수가 유한 개인 카테고리로 나눈 구분 결과를 얻은 결과
- 카테고리는 순러(ranks)로 구분
- 순서 속성 f로 매핑 가능한 숫자 속성의 범위는 M_f로 정의할 수 있음

ex) 간격-척도의 속성 temperature(섭씨)의 간격은 (-30, -10), (-10, 10), (10, 30)으로 표시할 수 있음(순서대로 추운 온도, 온화한 온도, 따뜻한 온도)

○ n개의 객체를 설명하는 서열 속성집합을 f라고 하면, f에 대한 차이 계산 절차는 다음과 같음

- "서열 속성은 어떻게 처리하는가?" => 객체간의 비유사도를 계산할 때 서열 속성의 처리는 숫자 속성과 유사함

1. i번째 객체에 대한 f의 값은 x_{if}이고, f는 M_f의 서열 상태(1, ..., M_f)를 가짐, 각 x_{if}를 해당 순서 $r_{if} \in \{1, \ldots, M_f\}$로 교체
2. 각 서열 속성은 다양한 상태를 갖기 때문에 각 속성의 범위를 [0.0, 1.0]으로 매핑하여 각 속성이 동일한 가중치를 갖도록 함

이와 같은 데이터 정규화는 i번째 객체의 f번째 속성을 다음과 같이 순서 r_{if}로 대체하여 실행할 수 있음

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

핸즈온 머신러닝

PART5. 데이터의 유사도와 비유사도 측정

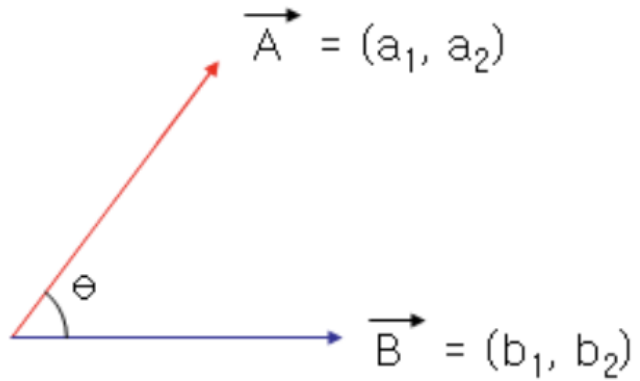
벡터공간 (Vector Space) 는 우리가 가지고 있는 데이터를 좌표계에 표현하기 위해 나온 개념이다.
음성데이터든, 이미지 데이터든, 문서 데이터든 해당 데이터가 가지고 있는 특성을 x,y축 또는 x,y,z,,,, 축 등
다차원 좌표계에 표시할수 있다면 해당 좌표들 끼리의 유사도를 계산할수가 있게 된다.

예를들어 2개의 특성으로 구분된 데이터라고 가정을 해보자.
문서데이터를 예로 들어 문서에 들어있는 단어는 오로지 A라는 단어와 B 라는 단어 2가지로 이루어져 있다면
문서들이 A라는 단어와 B라는 단어를 얼마나 많이 또는 적게 가지고 있느냐에 따라서 A축과 B축의 좌표계로 문서들을 표시 할 수 있다.
변수가 A, B 두개 밖에 없어 2차원 좌표계에 표시하였지만
데이터가 가지고 있는 특성에 따라서 3차원 10차원 200차원의 좌표계에 표시해야 할 수 있다.

좌표계에 표시된 데이터들의 벡터내적값을 이용하면 데이터들 끼리의 유사도를 구할수 있다.
유클리디안 거리는 단순히 좌표에 표시된 데이터들 간에 거리를 비교하여
거리가 멀면 유사도가 적고 가까우면 유사한 데이터라고 판별하지만
코사인 유사도 같은 경우에 문서의 단어들이 적게 있지만 A가 3개 B가 3개 있는 문서와 문서의 단어가 많이 있지만
A가 300개 B가 300개 있는 데이터는 서로 유사한 데이터라고 볼수 있다.
이런 문서간의 좌표의 실제 거리는 굉장히 멀겠지만 벡터내적을 구하면 두 데이터는 같은 데이터로 결과가 나오게된다.

이렇게 데이터가 가지고 있는 특성의 **비율**로서 데이터의 유사도를 측정하는 방법이 코사인 유사도이다.

먼저 벡터내적 공식이다.



$$\vec{A} \cdot \vec{B} = |\vec{A}||\vec{B}|\cos\theta$$

여기서 A벡터의 절대값인 $|\vec{A}|$ 은 A벡터의 크기를 말한다.

벡터는 방향과 크기를 가진 물리량인데 방향성을 소거하고 크기만을 표현할때 이렇게 표현한다.

$|\vec{A}|$ 벡터의 크기를 구하는 방법은

$$|\vec{A}| = \sqrt{a_1^2 + a_2^2}$$

A벡터가 가지고 있는 요소들의 제곱의 합에 루트를 씌워주면 A벡터의 크기가 된다.

원점으로 부터 A벡터까지의 거리 공식을 이용하면 되는것이다.

$$\vec{A} \cdot \vec{B} = a_1 b_1 + a_2 b_2$$

위 공식으로 A벡터와 B벡터의 내적값을 요소로서 구할수 있다.
코사인 내적으로 실시하면 방향이 없는 스칼라 물리량이 구해진다.

벡터의 내적을 표현하는 2가지 공식으로서 두 벡터간의 코사인 값은 아래와 같이 정리 할수 있다.

$$\cos\theta = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}||\vec{B}|} = \frac{a_1 b_1 + a_2 b_2}{\sqrt{a_1^2 + a_2^2}\sqrt{b_1^2 + b_2^2}}$$

두 데이터간의 코사인 유사도는 각 데이터들의 값을 모두 제곱하여 더한 값에 루트를 씌운 곱들을
분모로 하여 서로 간 같은 축에 있는 데이터들끼리 곱한것에 총합을 분자로 하면
두 데이터간의 코사인 값을 구할수 있다.

벡터의 성질인 크기와 방향을 이용하여 유사도를 측정하는 방법으로 가장 많이 사용되는 방법은 코사인값을 활용하는 방법
코사인은 -1 부터 1까지의 값을 갖는다.
코사인 각이 0이라면 두 벡터가 가장 가깝게 되고 유사도는 1이 된다.
코사인 각이 180이라면 두 벡터가 가장 멀리 있게 되고 유사도는 -1이 된다.

http://egloos.zum.com/metashower/v/9957584

핸즈온 머신러닝

PART5. 데이터의 유사도와 비유사도 측정

2.4.7 코사인 유사도

- 내적 공간의 두 벡터의 유사도를 측정함
- 두 벡터 간 각도의 코사인으로 측정하며 두 벡터가 동일한 방향을 향하고 있는지 여부를 판단
- 종종 텍스트 분석에서 문서의 유사도를 측정하는 데 사용함
- 문서는 특정 단어의 빈도를 기록한 수천 개의 속성이나 문서 내의 구문으로 표시할 수 있음

=> 각 문서는 term-frequency 벡터로 표시가 가능한 객체

=> 이러한 데이터는 매우 비대칭적임

○ Term-Frequency 벡터

- 일반적으로 매우 길고 데이터 값은 별로 없음
- 따라서 대부분 0의 값을 가짐

ex) 정보수집, 텍스트문서, 군집화, 생물학, 분류학, 유전자 특성 매핑 등

- 앞서 학습한 일반적인 거리 측정값은 희소한 숫자형 데이터(sparse numeric data)에 대해서는 적합하지 않음

ex) 2개의 term-frequency 벡터는 해당 문서 간에 공유하는 단어는 많지 않다는 것을 의미하는 값 0을 대부분 가짐
하지만 이러한 사실은 해당 문서들이 유사하지 않음을 의미함
따라서 2개 문서가 공통으로 보유하고 있는 단어와 해당 단어의 발생빈도에 초점을 두어 측정할 필요가 있음
=> 0을 무시한 수치형 데이터에 대한 측정값이 필요

- 코사인 유사도는 문서를 비교하는 데 사용할 수 있는 유사도 측정값
- 질의 작업을 수행한 단어 벡터에 대한 계산된 문서 랭킹값
- 비교 대상 2개 벡터를 x와 y라 하고, 유사도 함수로 코사인 측정값을 이용하면 다음 식과 같음

$$sim(x, y) = \frac{x \cdot y}{||x|| ||y||}$$

핸즈온 머신러닝

PART5. 데이터의 유사도와 비유사도 측정

- $||x||$는 벡터 $x = (x_1, x_2, \ldots, x_p)$에 대한 유클리드 노름이며 $\sqrt{x_1^2 + x_2^2 + \ldots + x_p^2}$과 같음

=> 개념적으로 이 값은 벡터의 길이

- 위 식의 값은 벡터 x와 y 간 간도의 코사인을 계산함
- 코사인 값 0은 2개의 벡터가 90도로 매칭되는 값이 없음을 의미
- 코사인 값이 1에 근접할 수록 벡터 간 각도는 작아지고 일치하는 정도는 높아짐을 의미
- 코사인 일치도는 측정값을 정의한 특성을 만족하지 않으므로 비측정값이라고 함

- 속성이 이진값일 때, 코사인 유사도 함수는 공유된 특성이나 속성으로 해석할 수 있음
- $x_i = 1$일 때 객체 x가 i번째 속성을 처리한다고 하면, $x^t \cdot y$는 x와 y의 내적이고, |x||y|는 x가 보유한 속성의 수와 y가 보유한 속성의 수에 대한 기하평균값(gemoetric mean)

=> sim(x,y)는 공통속성에 대한 상대적 소유정도(possession)가 됨

$$sim(x, y) = \frac{x \cdot y}{x \cdot x + y \cdot y - x \cdot y}$$

- x 또는 y가 소유한 속성의 수에 대해 x와 y가 고유한 속성의 개수 비율에 해당함
- 이 함수는 타니모토 계수나 타니모토 거리라하며 정보수집과 생물분류학에서 자주 사용됨