[어서와~ 머신러닝은 처음이지?]

1장. 거리개념과 MDS

- 장형석
- 국민대학교 빅데이터경영MBA과정 교수
- 숙명여자대학교 빅데이터센터 연구소장
- chjang1204@nate.com
- 010-3302-5543

1. 생활에서 만나는 문제

사례 1

- 보습학원 원장인 김민정씨는 중고등학생 아이들을 가르치는 학원을 운영하고 있다. 최근 새 학기에 접어들면서 학생들의 인원수가 늘어남에 따라 가장 효율적으로 클래스를 운영하는 문제를 고민하게 되었다. 이왕이면 비슷한 성향의 학생들끼리 서로 묶어서 가르치게 되면 서로 친해지기도 쉽고 시너지효과도 나지 않을까? 아마도 이런 아이들이 좋아 하는 선생님의 성향도 비슷해서 새로운 강사를 뽑을 때에 참고할 수있지 않을까?

사례 2

- 강남에서 순대국집을 운영하는 요리사출신인 사장님 오창규씨는 새로운 메뉴를 개발하고 싶어한다. 순대국을 좋아하는 사람이 좋아하는 또다른 메뉴는 무엇일까? 그런 목록이 있다면 기존의 요리재료와 큰 차이가 없는 요리들을 골라서 자신이 가장 자신있게 할 수 있는 메뉴를 만들어 본텐데..

1. 생활에서 만나는 문제

사례 1

	A	B	C	D	E	F
1	학생번호	국어점수평균	수학점수평균	영어점수평균	과학점수평균	학업집중도
2	1	90	75	85	60	70
3	2	65	90	60	88	80
4	3	45	53	48	50	60
5	4	77	96	62	92	70
6	5	88	89	80	82	90
7	6	90	92	90	96	100
8	7	65	70	66	76	70
9	8	60	90	70	98	80
10	9	46	56	43	55	60
11	10	88	67	90	70	70

사례 2

속성	값
좋아한다	1
무응답	0
싫어한다	1

	A	B	C	D	E	F	G	H	I	J	K
1	고객번호	추어탕	갈비탕	김치볶음밥	뼈다귀해장국	북어해장국	순대국	쭈꾸미볶음	김치찌개	쌈밥정식	삼계탕
2	1	1	0	0	1	0	1	0	0	0	1
3	2	0	0	1	0	0	0	0	0	0	-1
4	3	0	0	0	0	1	0	0	0	0	1
5	4	1	0	0	0	0	0	0	0	0	0
6	5	0	0	0	1	0	0	0	0	0	0
7	6	0	0	0	0	0	0	0	1	0	1
8	7	0	0	0	1	1	0	0	0	0	0
9	8	0	0	0	0	0	0	0	0	1	0
10	9	1	0	0	1	0	1	0	0	0	1
11	10	0	0	1	0	0	0	0	1	0	0

2. 해결책을 위한 사고실험

사례 1

- 비슷한 아이들을 같은 클래스로 묶고싶다.

사례 2

- 순대국을 좋아하는 부류의 고객집단이 좋아하는 메뉴 집단을 찾고싶다.

유사한 부류를 찾는다.

=> 유사성 / 유사도 계산

2. 해결책을 위한 사고실험

유사도 계산

- 거리개념

$$distance = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$distance = \sqrt{(수학점수1 - 수학점수2)^2 + (영어점수1 - 영어점수2)^2 + \cdots}$$

=> 다차원 척도법(Multi-Dimensional Scaling, MDS)

3. 알고리즘과 수학적 정리

수학공식

- 수치 : 민코우스키(Minkowski) 거리

$$d(x,y) = \left(\sum_{n=1}^{m} |x_n - y_n|^r\right)^{\frac{1}{r}}$$

- r = 1 : 맨하탄 거리
- r = 2 : 유클리드 거리

- Vector : 코사인 유사도

$$\mathrm{d}(\mathbf{x},\mathbf{y}) = \cos(\boldsymbol{x},\boldsymbol{y}) = \frac{\boldsymbol{x}\bullet\boldsymbol{y}}{|\boldsymbol{x}||\boldsymbol{y}|}$$

- 연속적인 수치 : 상관계수

$$d(x,y) = \frac{\sum_{j=1}^{m}(x_j - \bar{x})(y_j - \bar{y})}{\sqrt{\sum_{j=1}^{m}(x_j - \bar{x})^2}\ \sqrt{\sum_{j=1}^{m}(y_j - \bar{y})^2}}$$

4. 코딩구현과 구체적인 해결방안

사례 1 : 1단계 - 유사도 계산

- 학생 데이터

```
> academy <- read.csv("academy.csv" , stringsAsFactors = F , header = T)
> academy <- academy[-1]
> head(academy)
  국어점수평균 수학점수평균 영어점수평균 과학점수평균 학업집중도
1           90           75           85           60         70
2           65           90           60           88         80
3           45           53           48           50         60
4           77           96           62           92         70
5           88           89           80           82         90
6           90           92           90           96        100
```

- 유클리드 거리행렬 : dist( )

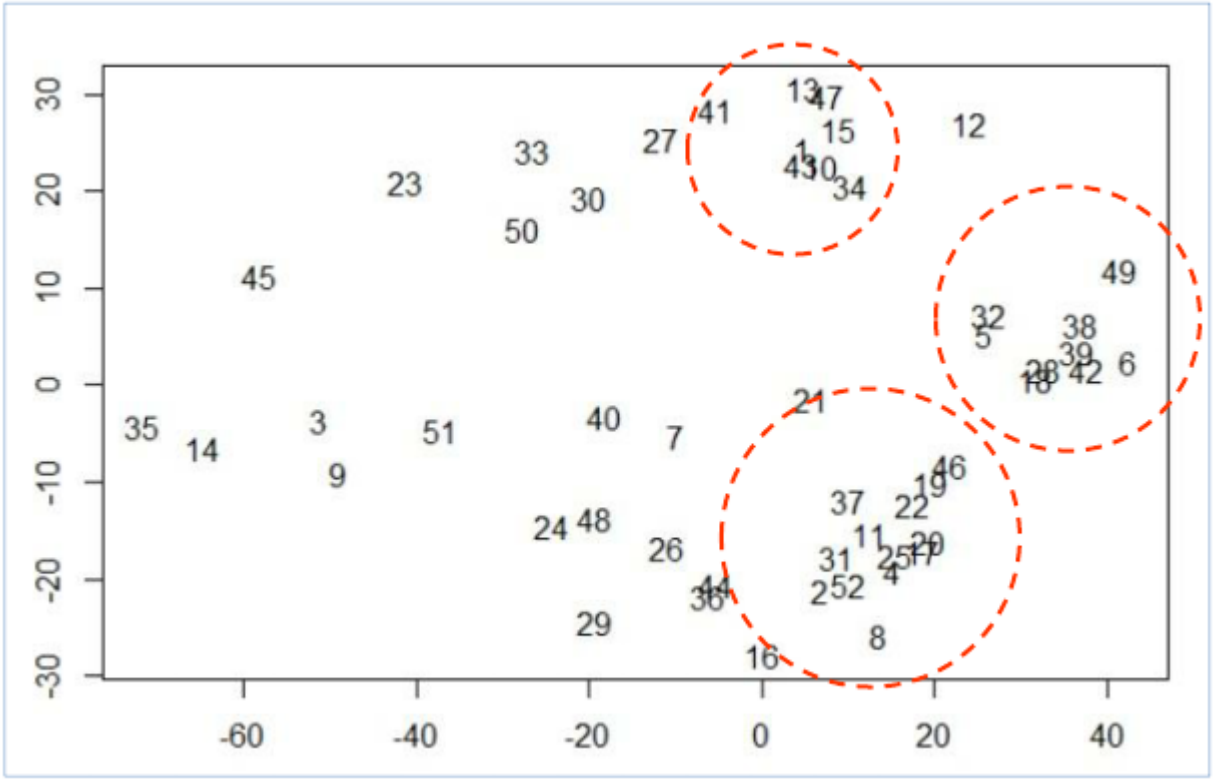
```
> dist_academy <- dist(academy , method = "euclidean")
> dist_academy
          1         2         3         4         5         6         7
2  48.569538
3  63.859220 61.294372
4  46.508064 17.320508 70.235319
5  33.301652 32.649655 78.057671 31.527766
6  50.099900 44.643029 95.005263 43.416587 20.223748
7  35.594943 26.076810 42.296572 33.045423 39.012818 54.635154
8  53.795911 15.000000 69.152006 22.912878 35.227830 41.327957 32.015621
9  64.699304 57.402091  7.745967 66.264621 76.491830 93.391648 40.336088 65.8
10 13.892444 48.805737 65.642974 47.222876 33.585711 46.957428 33.911650 50.9
11 41.581246 26.907248 67.727395 19.798990 36.027767 47.106263 30.724583 25.2
12 20.149442 52.962251 81.572054 49.122296 25.903668 36.414283 47.254629 55.0
13 15.427249 52.316345 66.513157 51.855569 33.896903 49.699095 39.812058 58.6
14 79.284299 75.372409 25.179357 82.054860 93.429118 110.308658 57.489129 83.8
15  7.280110 51.980766 68.825867 49.537864 34.292856 49.244289 40.074930 55.6
```

4. 코딩구현과 구체적인 해결방안

사례 1 : 2단계 - 2차원 그래프로 시각화

- 시각화 : cmdscale( )

```
>
> two_coord <- cmdscale(dist_academy)
> plot(two_coord , type = "n")
> text(two_coord , as.character(1:52))
```



4. 코딩구현과 구체적인 해결방안

사례 2 : 1단계 – 전치행렬로 변환

- 음식 데이터

```
> food <- read.csv("food.csv" , stringsAsFactors = F , header = T)
> food <- food[-1]
> head(food)
  추어탕 갈비탕 김치볶음밥 뼈다귀해장국 북어해장국 순대국 쭈꾸미볶음 김치찌개 쌈밥정식 삼계탕
1      1      0          0            1          0      1          0        0        0      1
2      0      0          1            0          0      0          0        0        0     -1
3      0      0          0            0          1      0          0        0        0      1
4      1      0          0            0          0      0          0        0        0      0
5      0      0          0            1          0      0          0        0        0      0
6      0      0          0            0          0      0          0        1        0      1
```

- 전치행렬 : t(food)

```
> t(food)
           [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18]
추어탕        1    0    0    1    0    0    0    0    1     0     1     0     0     0     1     0     1     1
갈비탕        0    0    0    0    0    0    0    0    0     1     0     1     0     0     0     0     0     1
김치볶음밥    0    1    0    0    0    0    0    0    0    -1     1     0     0     0     0     0     0     0
뼈다귀해장국  1    0    0    0    1    0    1    0    1     0     0     0     0     0    -1     0     1     0
북어해장국    0    0    1    0    0    0    1    0    0     0     0     0     1     0     0     1     0     0
순대국        1    0    0    0    0    0    0    0    1     0     0     0     1     1     0     0     1     0
쭈꾸미볶음    0    0    0    0    0    0    0    0    0     1     0     0     0     0     0     0     0    -1
김치찌개      0    0    0    0    0    1    0    0    0     0     0     0     0     1     1     0     0     0
쌈밥정식      0    0    0    0    0    0    0    1    0     0     0     0    -1     0     0     1     0     0
삼계탕        1   -1    1    0    0    1    0    0    1     0     1     0     1     1     0     1     0     0
```

4. 코딩구현과 구체적인 해결방안

사례 2 : 2단계 - 유사도 계산

- 유클리드 거리행렬 : dist( )

```
> dist(t(food) , method = "euclidean")
             추어탕   갈비탕 김치볶음밥 뼈다귀해장국 북어해장국   순대국 쭈꾸미볶음 김치찌개 쌈밥정식
갈비탕       4.690416
김치볶음밥   5.196152 4.795832
뼈다귀해장국 4.358899 4.582576   4.242641
북어해장국   5.000000 4.582576   4.242641     4.000000
순대국       4.358899 4.358899   4.690416     3.162278   4.242641
쭈꾸미볶음   4.795832 4.358899   3.464102     3.741657   3.464102 4.000000
김치찌개     5.099020 4.898979   3.605551     4.582576   3.872983 4.582576   3.316625
쌈밥정식     5.385165 4.582576   3.741657     4.472136   4.472136 4.690416   3.464102 3.872983
삼계탕       5.830952 4.898979   5.000000     4.795832   3.872983 4.795832   5.000000 4.242641 5.196152
```

- 다른 방식(?)

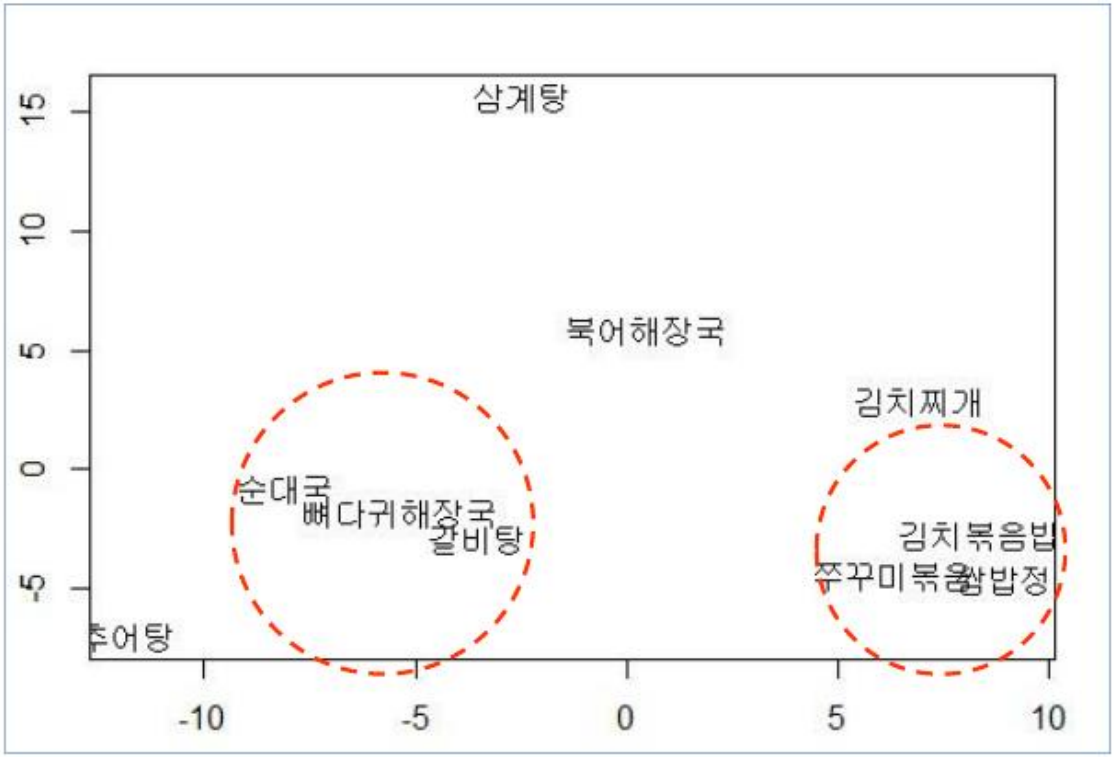
```
> food.mult <- t(as.matrix(food)) %*% as.matrix(food)
> food.mult
           추어탕 갈비탕 김치볶음밥 뼈다귀해장국 북어해장국 순대국 쭈꾸미볶음 김치찌개 쌈밥정식 삼계탕
추어탕         17      3         -1           4          1      5         -1       0      -2      1
갈비탕          3     11         -2           0          0      2         -2      -2      -1      3
김치볶음밥     -1     -2          8           0          0     -1          0       2       1      1
뼈다귀해장국    4      0          0          10          2      6          0      -1      -1      3
북어해장국      1      0          0           2         10      2          1       2      -1      7
순대국          5      2         -1           6          2     12          0       0      -1      4
쭈꾸미볶음     -1     -2          0           0          1      0          4       1       0     -1
김치찌개        0     -2          2          -1          2      0          1       9       1      5
쌈밥정식       -2     -1          1          -1         -1     -1          0       1       8      0
삼계탕          1      3          1           3          7      4         -1       5       0     19
```

4. 코딩구현과 구체적인 해결방안

사례 2 : 3단계 - 2차원 그래프로 시각화

- 시각화 : cmdscale( )

```
> two_coord2 <- cmdscale(dist.food)
> plot(two_coord2 , type = "n")
> text(two_coord2 , rownames(food.mult))
```



기타

유사도 계산 알고리즘

Table 4.1 The Pearson correlation between user 1 and other users based on the three items that user 1 has in common with the others

	Item 101	Item 102	Item 103	Correlation with user 1
User 1	5.0	3.0	2.5	1.000
User 2	2.0	2.5	5.0	-0.764
User 3	2.5	-	-	-
User 4	5.0	-	3.0	1.000
User 5	4.0	3.0	2.0	0.945

Note A user's Pearson correlation with itself is always 1.0.

피어슨 상관계수

Table 4.2 The Euclidean distance between user 1 and other users, and the resulting similarity scores

	Item 101	Item 102	Item 103	Distance	Similarity to user 1
User 1	5.0	3.0	2.5	0.000	1.000
User 2	2.0	2.5	5.0	3.937	0.203
User 3	2.5	-	-	2.500	0.286
User 4	5.0	-	3.0	0.500	0.667
User 5	4.0	3.0	2.0	1.118	0.472

유클리드 거리

기타

유사도 계산 알고리즘

스피어만 상관계수

Table 4.3 The preference values transformed into ranks, and the resulting Spearman correlation between user 1 and each of the other users

	Item 101	Item 102	Item 103	Correlation to user 1
User 1	3.0	2.0	1.0	1.0
User 2	1.0	2.0	3.0	–1.0
User 3	1.0	-	-	-
User 4	2.0	-	1.0	1.0
User 5	3.0	2.0	1.0	1.0

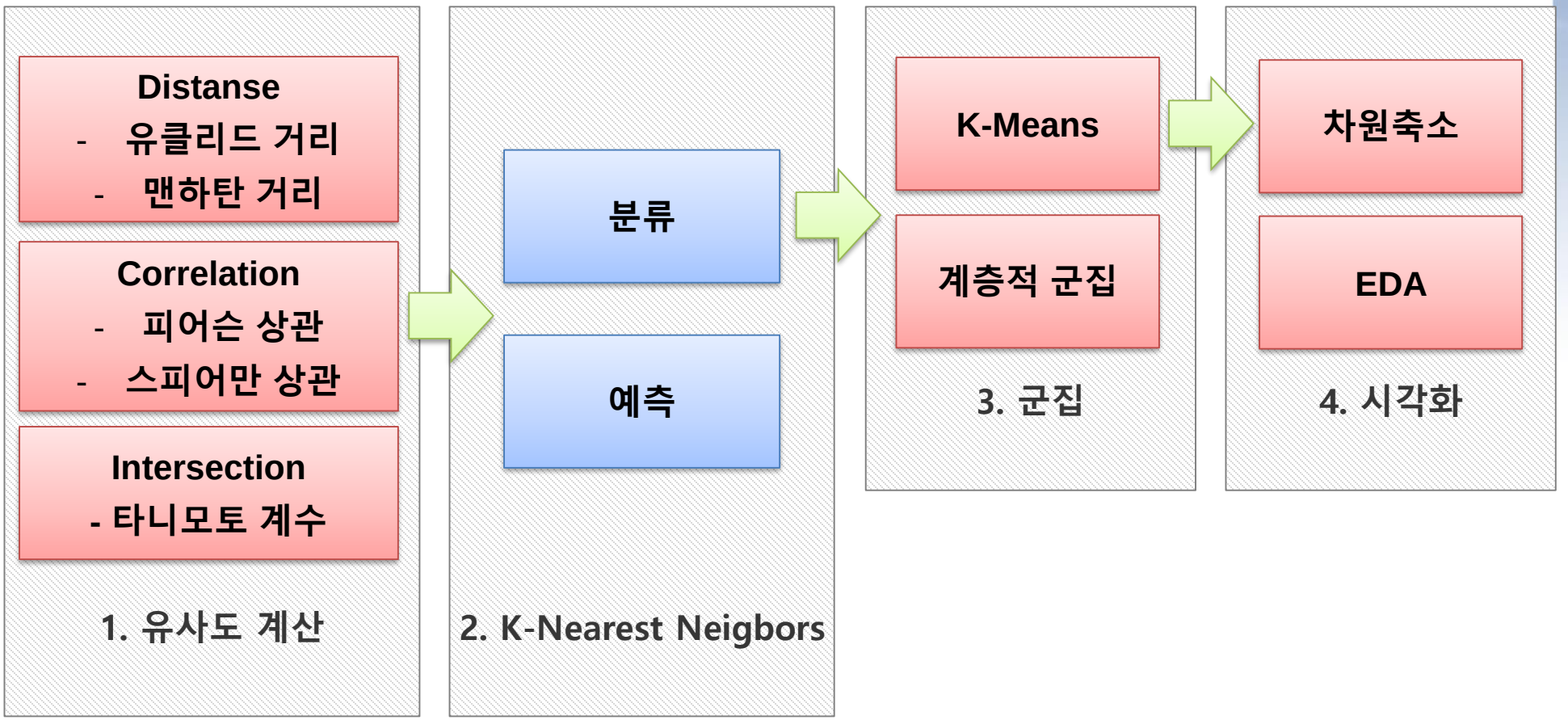
타니모토 계수

Table 4.4 The similarity values between user 1 and other users, computed using the Tanimoto coefficient. Note that preference values themselves are omitted, because they aren't used in the computation.

	Item 101	Item 102	Item 103	Item 104	Item 105	Item 106	Item 107	Similarity to user 1
User 1	X	X	X					1.0
User 2	X	X	X	X				0.75
User 3	X			X	X		X	0.17
User 4	X		X	X		X		0.4
User 5	X	X	X	X	X	X		0.5

기타

내가 생각하는 책의 목차



감사합니다.