EE412 Foundation of Big Data Analytics, Fall 2018
HW2

Name: 김정만

Student ID: 20150073

Discussion Group (People with whom you discussed ideas used in your answers):

On-line or hardcopy documents used as part of your answers:

Answer to Problem 1

Exercise 7.2.6

Wikipedia의 edit distance 정의를 따를때, 문자의 삽입, 삭제, 변경 횟수이다.

A = "abcdfg", B = "abcefg", C = "abc", D = "adfgbc" 일때 각 edit distance는

A↔B = 1, A↔C = 3, A↔D = 4, B↔D = 5, C↔D = 3, B↔C = 3 이다.

각 점에서 다른 점들까지의 거리합은 A : 8, B : 9, C : 9, D : 12 로 A가 clusteroid 이다.

각 점에서 다른 점들까지의 최대거리는 A : 4, B : 5, C : 3, D : 5 로 C가 clusteroid 가된다

∴ {"abcdfg", "abcefg", "abc", "adfgbc"}

Exercise 7.3.2

세 클러스터의 diameter 3, $\sqrt{13}$, $\sqrt{18}$, 각 intercluster 거리 $\sqrt{17}$, $\sqrt{17}$, $\sqrt{25}$ 이다

모든 diameter < 모든 intercluster 거리를 만족하면 자명하지만 그렇지 않으므로

각각 증명하겠다

(2,2) → (12,6) → (4,10)

(3,4) → (12,6) → (9,10)

(5,2) → (9,10) → (12,3)

(4,8) → (12,3) → (2,2)

(4,10) → (12,3) → (2,2)

(6,8) → (12,3) → (2,2)

(9,10) → (2,2) → (12,3)

(10,5) → (2,2) → (4,10)

(9,3) → (4,10) → (2,2)

(12,6) → (2,2) → (4,10)

(11,4) → (4,10) → (2,2)
↳ (2,2) → (4,10) tie

(12,3) → (4,10) → (2,2)

다음과 같은 12쌍의 각 시작점들은 모두 3개의 클러스터 각각에 속해있는 점들이다.

Exercise 7.4.1

정들이 줄어들면, 가장 가까운 두 대표점 사이의 거리는

$0.8v - 0.8c$ 가 된다. 이 거리가 d 이하가 되면 merge 되기때문에

링과 원이 하나의 클러스터로 합체될 조건은 $\frac{4}{5}(v-c) \le d$ 이다.

(여기서의 가정은 centroid와 두 클러스터의 대표점이 일직선이되는

적어도 하나의 대표점 쌍이 존재한다는 것이다)

Answer to Problem 2

Exercise 11.1.5

$$\det\begin{bmatrix} 1-\lambda & 1 & 1 \\ 1 & 2-\lambda & 3 \\ 1 & 3 & 6-\lambda \end{bmatrix} = (1-\lambda)\left[(2-\lambda)(6-\lambda)-9\right] - \left[6-\lambda-3\right] + \left[3-(2-\lambda)\right]$$

$= \lambda^3 - 9\lambda^2 + 9\lambda - 1 = (\lambda-1)(\lambda^2-8\lambda+1) = 0$ 이므로 $\lambda = 1,\ 4\pm\sqrt{15}$ 이다.

i) $\lambda=1$ 일때, $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$, $y=-z$, $x+z=y$, $x=2y$ 이므로 $\begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix}$ 이다.

normalize 하면 $\lambda=1$ 일때 eigen vector는 $\begin{pmatrix} 2/\sqrt{6} \\ 1/\sqrt{6} \\ -1/\sqrt{6} \end{pmatrix}$ 이다.

ii) $\lambda = 4+\sqrt{15}$ 일때, $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = (4+\sqrt{15})\begin{pmatrix} x \\ y \\ z \end{pmatrix}$, $y+z = (3+\sqrt{15})x$, $x+3z = (2+\sqrt{15})y$,

$x+3y = (\sqrt{15}-2)z$ 이므로 $y = \dfrac{1+\sqrt{15}}{5+\sqrt{15}}z$, $x = \dfrac{6+2\sqrt{15}}{30+8\sqrt{15}}z$ 이며 normalize하면

$\lambda = 4+\sqrt{15}$ 일때 eigenvector는 $\begin{pmatrix} 0.1938 \\ 0.4722 \\ 0.8599 \end{pmatrix}$ 이다.

iii) $\lambda = 4-\sqrt{15}$ 일때, $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = (4-\sqrt{15})\begin{pmatrix} x \\ y \\ z \end{pmatrix}$, $y+z = (3-\sqrt{15})x$, $x+3z = (2-\sqrt{15})y$,

$x+3y = (-2-\sqrt{15})z$ 이므로 $y = \dfrac{1-\sqrt{15}}{5-\sqrt{15}}z$, $x = \dfrac{6-2\sqrt{15}}{30-8\sqrt{15}}z$ 이며 normalize 하면

$\lambda = 4-\sqrt{15}$ 일때 eigenvector는 $\begin{pmatrix} 0.5438 \\ -0.7812 \\ 0.3065 \end{pmatrix}$ 이다.

Exercise 11.1.7

(a) approximate value of the principal eigenvector $= \begin{pmatrix} 0.1938 \\ 0.4722 \\ 0.8599 \end{pmatrix}$ (뒤에 python 코드참고)

(b) $\lambda = \begin{pmatrix} 0.1938 & 0.4722 & 0.8599 \end{pmatrix}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{pmatrix}\begin{pmatrix} 0.1938 \\ 0.4722 \\ 0.8599 \end{pmatrix} = 7.8727$ 이다.

(c) 새로운 행렬 $M^* = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{pmatrix} - \lambda \cdot x \cdot x^T = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{pmatrix} - 7.8727 \cdot \begin{pmatrix} 0.1938 \\ 0.4722 \\ 0.8599 \end{pmatrix}\begin{pmatrix} 0.1938 & 0.4722 & 0.8599 \end{pmatrix}$

$$= \begin{pmatrix} 0.7043 & 0.2196 & -0.3120 \\ 0.2196 & 0.2446 & -0.1961 \\ -0.3120 & -0.1961 & 0.1787 \end{pmatrix}$$ 이다.

(d) 우선 python을 이용하여 계산한 eigen vector는 $\begin{pmatrix} 0.8165 \\ 0.4084 \\ -0.4080 \end{pmatrix}$ 이다. 그러므로

$\lambda = (0.8165 \quad 0.4084 \quad -0.4080)$ (c)에서 구한 M^* $\begin{pmatrix} 0.8165 \\ 0.4084 \\ -0.4080 \end{pmatrix} = 1$ 이다. (python 코드 참고)

(e) 새로운 행렬 $M^{**} = M^* - 1 \cdot x \cdot x^T$, 여기서 $x = \begin{pmatrix} 0.8165 \\ 0.4084 \\ -0.4080 \end{pmatrix}$ 이므로 계산하면

$M^{**} = \begin{pmatrix} 0.0316 & -0.0539 & 0.0211 \\ -0.0539 & 0.0978 & -0.0301 \\ 0.0211 & -0.0301 & 0.0122 \end{pmatrix}$ 이다. 이를 통해 python으로 계산한

세 번째 eigen vector는 $\begin{pmatrix} 0.5435 \\ -0.1820 \\ 0.3050 \end{pmatrix}$ 이다. 그러므로 $\lambda = \begin{pmatrix} 0.5435 \\ -0.1820 \\ 0.3050 \end{pmatrix}^T \cdot M^{**} \cdot \begin{pmatrix} 0.5435 \\ -0.1820 \\ 0.3050 \end{pmatrix}$

(python 코드 참고)

$= 0.1270$ 이다.

Exercise 11.3.1

(a) $M^TM = \begin{pmatrix} 1 & 3 & 5 & 0 & 1 \\ 2 & 4 & 4 & 2 & 3 \\ 3 & 5 & 3 & 4 & 5 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 5 & 4 & 3 \\ 0 & 2 & 4 \\ 1 & 3 & 5 \end{pmatrix} = \begin{pmatrix} 36 & 37 & 38 \\ 37 & 49 & 61 \\ 38 & 61 & 84 \end{pmatrix}$

$MM^T = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 5 & 4 & 3 \\ 0 & 2 & 4 \\ 1 & 3 & 5 \end{pmatrix} \begin{pmatrix} 1 & 3 & 5 & 0 & 1 \\ 2 & 4 & 4 & 2 & 3 \\ 3 & 5 & 3 & 4 & 5 \end{pmatrix} = \begin{pmatrix} 14 & 26 & 22 & 16 & 22 \\ 26 & 50 & 46 & 28 & 40 \\ 22 & 46 & 50 & 20 & 32 \\ 16 & 28 & 20 & 20 & 26 \\ 22 & 40 & 32 & 26 & 35 \end{pmatrix}$ 이다.

(b) python을 이용하여 M^TM 과 MM^T의 eigen value를 구하였다. (python 코드 참고)

M^TM 의 eigen value $\Rightarrow \lambda_1 = 153.5610 \quad \lambda_2 = 15.4330 \quad , \lambda_3 = 0$

MM^T의 eigen value $\Rightarrow \lambda_1 = 153.5610, \quad \lambda_2 = 15.4330, \quad \lambda_3 = \lambda_4 = \lambda_5 = 0$

(c) M^TM 의 eigen vector $\Rightarrow V_1 = \begin{pmatrix} 0.4093 \\ 0.5635 \\ 0.7176 \end{pmatrix} \quad V_2 = \begin{pmatrix} 0.8160 \\ 0.1259 \\ -0.5642 \end{pmatrix}, \quad V_3 = \begin{pmatrix} 0.4082 \\ -0.8165 \\ 0.4082 \end{pmatrix}$

MM^T의 eigen vector $\Rightarrow V_1 = \begin{pmatrix} 0.2977 \\ 0.5705 \\ 0.5207 \\ 0.3226 \\ 0.4590 \end{pmatrix} V_2 = \begin{pmatrix} -0.1591 \\ 0.0332 \\ 0.7359 \\ -0.5104 \\ -0.4143 \end{pmatrix} V_3 = \begin{pmatrix} 0.1251 \\ -0.4532 \\ 0.3255 \\ 0.7200 \\ -0.3932 \end{pmatrix} V_5 = \begin{pmatrix} 0.9413 \\ -0.1148 \\ -0.0403 \\ -0.1883 \\ -0.2152 \end{pmatrix}$

$V_4 = \begin{pmatrix} 0.0752 \\ -0.0729 \\ -0.1057 \\ -0.7257 \\ 0.6717 \end{pmatrix}$

(d) U는 MM^T 의 eigen vector 행렬, Σ는 M^TM 의 eigen value 의 제곱근이며

V는 M^TM 의 eigen vector 행렬이므로

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 5 & 4 & 3 \\ 0 & 2 & 4 \\ 1 & 3 & 5 \end{pmatrix} = \begin{pmatrix} 0.2977 & -0.1591 \\ 0.5705 & 0.0332 \\ 0.5207 & 0.7359 \\ 0.3226 & -0.5104 \\ 0.4590 & -0.4143 \end{pmatrix} \begin{pmatrix} 12.3922 & 0 \\ 0 & 3.9285 \end{pmatrix} \begin{pmatrix} 0.4093 & 0.5635 & 0.7176 \\ 0.8160 & 0.1259 & -0.5642 \end{pmatrix}$$

(e) 더 작은 특이값을 날리면

$$\begin{pmatrix} 0.2977 \\ 0.5705 \\ 0.5207 \\ 0.3226 \\ 0.4590 \end{pmatrix} (12.3922) \begin{pmatrix} 0.4093 & 0.5635 & 0.7176 \end{pmatrix} = \begin{pmatrix} 1.5100 & 2.0788 & 2.6473 \\ 2.8936 & 3.9838 & 5.0733 \\ 2.6411 & 3.6361 & 4.6304 \\ 1.6363 & 2.2527 & 2.8688 \\ 2.3281 & 3.2052 & 4.0817 \end{pmatrix}$$ 이다

(f) 전체 에너지는 $(12.3922)^2 + (3.9285)^2 = 168.9997$ 이며 보존된 에너지는

$(12.3922)^2 = 153.5666$ 으로 90.868 % 에너지가 보존된다.

Exercise 11.4.2

(a) 각 열에 대한 확률은 0.210 이므로 $\sqrt{2 \times 0.210} = 0.648$ 로 나누면 $C = \begin{pmatrix} 1.54 & 1.54 \\ 4.63 & 4.63 \\ 6.17 & 6.17 \\ 7.72 & 7.72 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$ 이 된다.

($\frac{1^2+3^2+4^2+5^2}{243}$, $\frac{3^2+3^2+3^2}{243}$, $\frac{4^2+4^2+4^2}{243}$)

각 행에 대한 확률은 0.111, 0.198 이므로 0.471, 0.629 로 나누면

$$R = \begin{pmatrix} 6.37 & 6.37 & 6.37 & 0 & 0 \\ 6.36 & 6.36 & 6.36 & 0 & 0 \end{pmatrix}$$ 이 된다 따라서 2x2 행렬 $W = \begin{pmatrix} 3 & 3 \\ 4 & 4 \end{pmatrix}$ 이다.

Python을 이용해 W를 SVD 하면 $W = \begin{pmatrix} 3 & 3 \\ 4 & 4 \end{pmatrix} = \begin{pmatrix} -0.6 & -0.8 \\ -0.8 & 0.6 \end{pmatrix} \begin{pmatrix} 7.0711 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{pmatrix}$

(코드 참고)

이므로 $U = \begin{pmatrix} -0.7071 & -0.7071 \\ -0.7071 & 0.7071 \end{pmatrix} \begin{pmatrix} 0.02 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} -0.6 & -0.8 \\ -0.8 & 0.6 \end{pmatrix} = \begin{pmatrix} 0.0085 & 0.0113 \\ 0.0085 & 0.0113 \end{pmatrix}$ 이며, 대입하면

$$M \approx \begin{pmatrix} 1.54 & 1.54 \\ 4.63 & 4.63 \\ 6.17 & 6.17 \\ 7.72 & 7.72 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0.0085 & 0.0113 \\ 0.0085 & 0.0113 \end{pmatrix} \cdot \begin{pmatrix} 6.37 & 6.37 & 6.37 & 0 & 0 \\ 6.36 & 6.36 & 6.36 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0.3881 & 0.3881 & 0.3881 & 0 & 0 \\ 1.1669 & 1.1669 & 1.1669 & 0 & 0 \\ 1.555 & 1.555 & 1.555 & 0 & 0 \\ 1.9456 & 1.9456 & 1.9456 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(b) 위와 같은 방식으로 하면, $C = \begin{pmatrix} 1.54 & 1.54 \\ 4.63 & 4.63 \\ 6.17 & 6.17 \\ 7.72 & 7.72 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$ $R = \begin{pmatrix} 6.36 & 6.36 & 6.36 & 0 & 0 \\ 0 & 0 & 0 & 7.78 & 7.78 \end{pmatrix}$ 이다.

$W = \begin{pmatrix} 5 & 5 \\ 0 & 0 \end{pmatrix}$ 이며 python을 통해 $W = \begin{pmatrix} 5 & 5 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 7.071 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0.7071 & 0.7071 \\ -0.7071 & 0.7071 \end{pmatrix}$

이므로 $U = \begin{pmatrix} 0.7071 & -0.7071 \\ 0.7071 & 0.7071 \end{pmatrix} \begin{pmatrix} 0.02 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.0141 & 0 \\ 0.0141 & 0 \end{pmatrix}$ 이며, 대입하면

$M = \begin{pmatrix} 1.54 & 1.54 \\ 4.63 & 4.63 \\ 6.17 & 6.17 \\ 7.72 & 7.72 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0.0141 & 0 \\ 0.0141 & 0 \end{pmatrix} \cdot \begin{pmatrix} 6.36 & 6.36 & 6.36 & 0 & 0 \\ 0 & 0 & 0 & 7.78 & 7.78 \end{pmatrix} = \begin{pmatrix} 0.2162 & 0.2162 & 0.2162 & 0 & 0 \\ 0.8304 & 0.8304 & 0.8304 & 0 & 0 \\ 1.1066 & 1.1066 & 1.1066 & 0 & 0 \\ 1.3846 & 1.3846 & 1.3846 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

(c) 위와 같은 방식으로 하면, $C = \begin{pmatrix} 1.54 & 0 \\ 4.63 & 0 \\ 6.17 & 0 \\ 7.72 & 0 \\ 0 & 6.58 \\ 0 & 8.22 \\ 0 & 3.29 \end{pmatrix}$ $R = \begin{pmatrix} 6.45 & 6.45 & 6.45 & 0 & 0 \\ 0 & 0 & 0 & 7.78 & 7.78 \end{pmatrix}$ 이다.

$W = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ 이며 python을 통해 $W = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ 이므로

$U = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0.25 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0.25 \end{pmatrix}$ 이며, 대입하면

$M = \begin{pmatrix} 1.54 & 0 \\ 4.63 & 0 \\ 6.17 & 0 \\ 7.72 & 0 \\ 0 & 6.58 \\ 0 & 2.22 \\ 0 & 3.29 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0.25 \end{pmatrix} \begin{pmatrix} 6.45 & 6.45 & 6.45 & 0 & 0 \\ 0 & 0 & 0 & 7.78 & 7.78 \end{pmatrix} = \begin{pmatrix} 9.933 & 9.933 & 9.933 & 0 & 0 \\ 29.8635 & 29.8635 & 29.8635 & 0 & 0 \\ 39.7965 & 39.7965 & 39.7965 & 0 & 0 \\ 49.794 & 49.794 & 49.794 & 0 & 0 \\ 0 & 0 & 0 & 12.7981 & 12.7981 \\ 0 & 0 & 0 & 15.9879 & 15.9879 \\ 0 & 0 & 0 & 6.3991 & 6.3991 \end{pmatrix}$

Exercise 9.3.1 Answer to Problem 3

(a) (A,B) : $1-\frac{4}{8}=\frac{1}{2}$ $\quad (B,C)$: $1-\frac{4}{8}=\frac{1}{2}$

(A,C) : $1-\frac{4}{8}=\frac{1}{2}$ $\quad \therefore$ 모든 자카드 거리는 $\frac{1}{2}$ 이다

(b) $\text{cosine}(A,B)=\frac{4}{\sqrt{6}\cdot\sqrt{6}}=\frac{2}{3}$

$\text{cosine}(A,C)=\frac{4}{\sqrt{6}\cdot\sqrt{6}}=\frac{2}{3}$

$\text{cosine}(B,C)=\frac{4}{\sqrt{6}\cdot\sqrt{6}}=\frac{2}{3}$

코사인 거리를 두 벡터 사이 각으로 정의했을 때 $\frac{A\cdot B}{|A|\cdot|B|}=\cos\theta$ 이므로

(A,B) 코사인거리 $=\arccos\left(\frac{2}{3}\right)=48.1948^\circ$ 이다

나머지 쌍도 모두 같다.

(A,B) 코사인거리 $=48.1948^\circ$, (A,C) 코사인거리 $=48.1948^\circ$, (B,C) 코사인거리 $=48.1948^\circ$

(c) (A,B) : $1-\frac{2}{5}=\frac{3}{5}$ $\quad (B,C)$: $1-\frac{1}{6}=\frac{5}{6}$ $\quad (A,C)$: $1-\frac{2}{6}=\frac{2}{3}$ 이다.

(d) $\cos(A,B)=\frac{2}{\sqrt{4}\cdot\sqrt{3}}=\frac{1}{\sqrt{3}}=0.5774$ $\quad (A,B)$ 코사인거리 $=54.7321^\circ$

$\cos(B,C)=\frac{1}{\sqrt{3}\cdot\sqrt{4}}=\frac{1}{\sqrt{12}}=0.2887$ $\quad (B,C)$ 코사인거리 $=73.2199^\circ$

$\cos(A,C)=\frac{2}{\sqrt{4}\cdot\sqrt{4}}=\frac{1}{2}$ $\quad (C,A)$ 코사인거리 $=60^\circ$

(e) 각 사용자의 평균은 $A=\frac{10}{3}$, $B=\frac{7}{3}$, $C=3$ 이므로

	a	b	c	d	e	f	g	h
A	$\frac{2}{3}$	$\frac{5}{3}$		$\frac{5}{3}$	$-\frac{7}{3}$		$-\frac{1}{3}$	$-\frac{4}{3}$
B		$\frac{2}{3}$	$\frac{5}{3}$	$\frac{2}{3}$	$-\frac{4}{3}$	$-\frac{1}{3}$	$-\frac{4}{3}$	
C	-1		-2	0		1	2	0

이다.

(f) $\cos(A,B)=\frac{\frac{1}{9}(10+10+28+4)}{\frac{1}{3}\sqrt{2^2+5^2+5^2+7^2+1^2+4^2}\cdot\frac{1}{3}\sqrt{2^2+5^2+2^2+4^2+1^2+4^2}}=\frac{52}{\sqrt{120}\cdot\sqrt{66}}=0.5843$

$\cos(B,C)=\frac{\frac{1}{3}\cdot(-10-(-8)}{\frac{1}{3}\cdot\sqrt{66}\cdot\sqrt{1^2+2^2+1^2+2^2}}=-\frac{19}{\sqrt{66}\cdot\sqrt{10}}=-0.7396$

$\cos(A,C)=\frac{\frac{1}{3}(-2-2)}{\frac{1}{3}\cdot\sqrt{120}\cdot\sqrt{10}}=-\frac{4}{\sqrt{120}\cdot\sqrt{10}}=-0.1155$ 그러므로

$(A,B)=54.2464^\circ$

$(B,C)=137.6974^\circ$

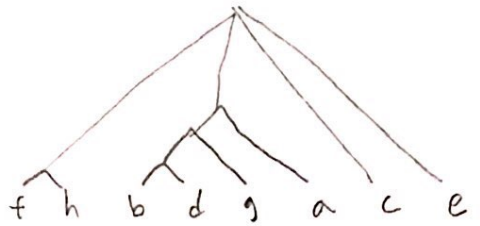
$(C,A)=96.6325^\circ$ 이다.

Exercise 9.3.2

( ) 는 자카드 유사도

(a)

	a	b	c	d	e	f	g	h
	1	1	0	1	0	0	1	0
	0	1	1	1	0	0	0	0
	0	0	0	1	0	1	1	1



1) 가장 유사한 쌍 (f,h) 이다. (1)

2) 다음으로는 (b,d) 이다. (d,g)도 tie (2/3)

3) 다음으로는 (d,g) 이다 (2/3)

4) 다음으로는 (a,b)이다. (b,c),(f,g)도 tie (1/2)

군집으로 { (f,h), (b,d,g,a), (c), (e) }
{ (f,h,b,d,g), (a), (c), (e) }
{ (f,h), (b,d,g,c), (a), (e) }

총 3가지 경우의 4개 클러스터가 있다.

(b) 4개의 클러스터가 { (f,h), (b,d,g,a), (c), (e) } 로 되어있다고 한다.

	(f,h)	(b,d,g,a)	(c)	(e)
A	2	17/4		1
B	2	7/3	4	1
C	7/2	10/3	1	

(c) $\cos(A,B) = \dfrac{4 + \frac{119}{12} + 1}{\sqrt{2^2+(4.25)^2+1^2}\cdot\sqrt{2^2+(\frac{7}{3})^2+4^2+1^2}} = \dfrac{\frac{179}{12}}{\sqrt{23.0625}\cdot\frac{1}{3}\cdot\sqrt{238}} = 0.604$

$\cos(B,C) = \dfrac{7 + \frac{70}{9} + 4}{\frac{1}{3}\cdot\sqrt{238}\cdot\sqrt{(\frac{7}{2})^2+(\frac{10}{3})^2+1^2}} = 0.7398$

$\cos(C,A) = \dfrac{7 + \frac{85}{6}}{\sqrt{(\frac{7}{2})^2+(\frac{10}{3})^2+1^2}\cdot\sqrt{23.0625}} = 0.8930$

각각 (A,B) = 52.8431°
(B,C) = 42.2856°
(C,A) = 26.7413° 이다.

Exercise 9.4.3

(a) $\begin{bmatrix} x & 1 \\ 1 & 1 \\ 1.178 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1.617 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1.617x+1 & x+1 & x+1 & x+1 & x+1 \\ 2.617 & 2 & 2 & 2 & 2 \\ 2.905 & 2.178 & 2.178 & 2.178 & 2.178 \\ 2.617 & 2 & 2 & 2 & 2 \\ 2.617 & 2 & 2 & 2 & 2 \end{bmatrix}$ 이고

$(4-1.617x)^2 + (1-x)^2 + (3-x)^2 + (3-x)^2 + (2-x)^2$ 을 미분하면

$-2\cdot(1.617\times(4-1.617x) + (1-x) + (3-x) + (3-x) + (2-x)) = 0$

즉, $15.468 - 6.614689x = 0$, 즉 $x = 2.338$ 이다. $U_{11} = 2.338$ 이다.

(b) $\begin{pmatrix} 2.338 & 1 \\ 1 & 1 \\ 1.178 & 1 \\ 1 & 1 \\ 1 & x \end{pmatrix} \begin{pmatrix} 1.617 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 4.781 & 3.338 & 3.338 & 3.338 & 3.338 \\ 2.617 & 2 & 2 & 2 & 2 \\ 2.905 & 2.178 & 2.178 & 2.178 & 2.178 \\ 2.617 & 2 & 2 & 2 & 2 \\ 1.617+x & 1+x & 1+x & 1+x & 1+x \end{pmatrix}$

이므로 $(2.383-x)^2+(3-x)^2+(4-x)^2+(3-x)^2$ 을 미분하면

$-2\cdot\left((2.383-x)+(3-x)+(4-x)+(3-x)\right)=0$ 이므로

$12.383-4x=0$ 이므로 $x=3.096$ 이다. $V_{52}=3.096$ 이다.

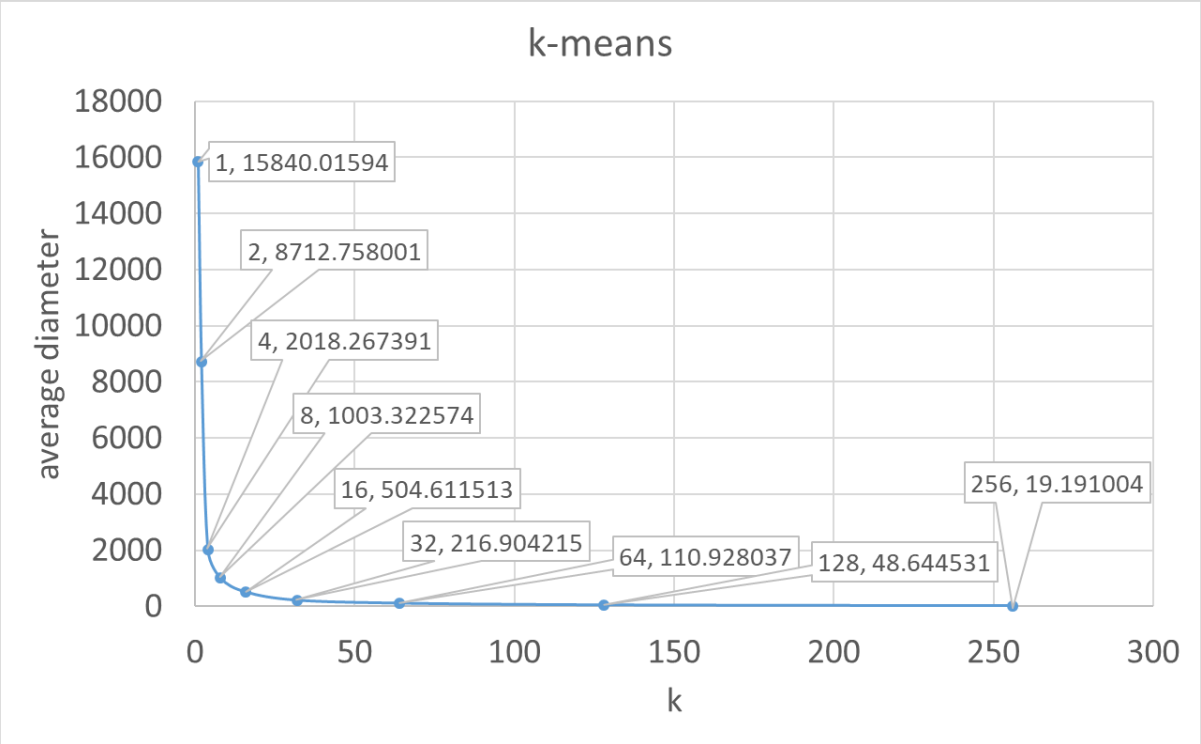
(c) $\begin{pmatrix} 2.338 & 1 \\ 1 & 1 \\ 1.178 & 1 \\ 1 & 1 \\ 1 & 3.096 \end{pmatrix} \begin{pmatrix} 1.617 & 1 & 1 & 1 & 1 \\ 1 & y & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 4.781 & 2.338+y & 3.338 & 3.338 & 3.338 \\ 2.617 & 1+y & 2 & 2 & 2 \\ 2.905 & 1.178+y & 2.178 & 2.178 & 2.178 \\ 2.617 & 1+y & 2 & 2 & 2 \\ 4.713 & 1+3.096y & 4.096 & 4.096 & 4.096 \end{pmatrix}$

이므로 $(-0.338-y)^2+(-y)^2+(4-y)^2+(3-3.096y)^2$ 을 미분하면

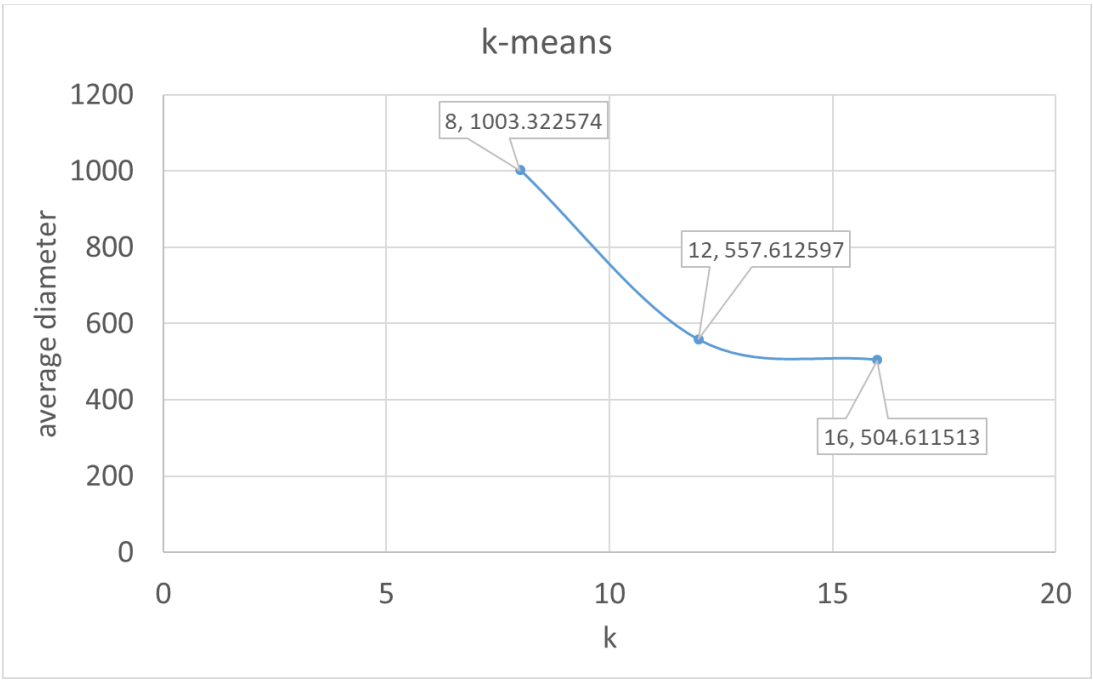
$-2\cdot\left((-0.338-y)+(-y)+(4-y)+3.096\cdot(3-3.096y)\right)=0$ 이므로

$12.95-12.585216y=0$ 이므로 $y=1.029$ 이다. $V_{22}=1.029$ 이다.

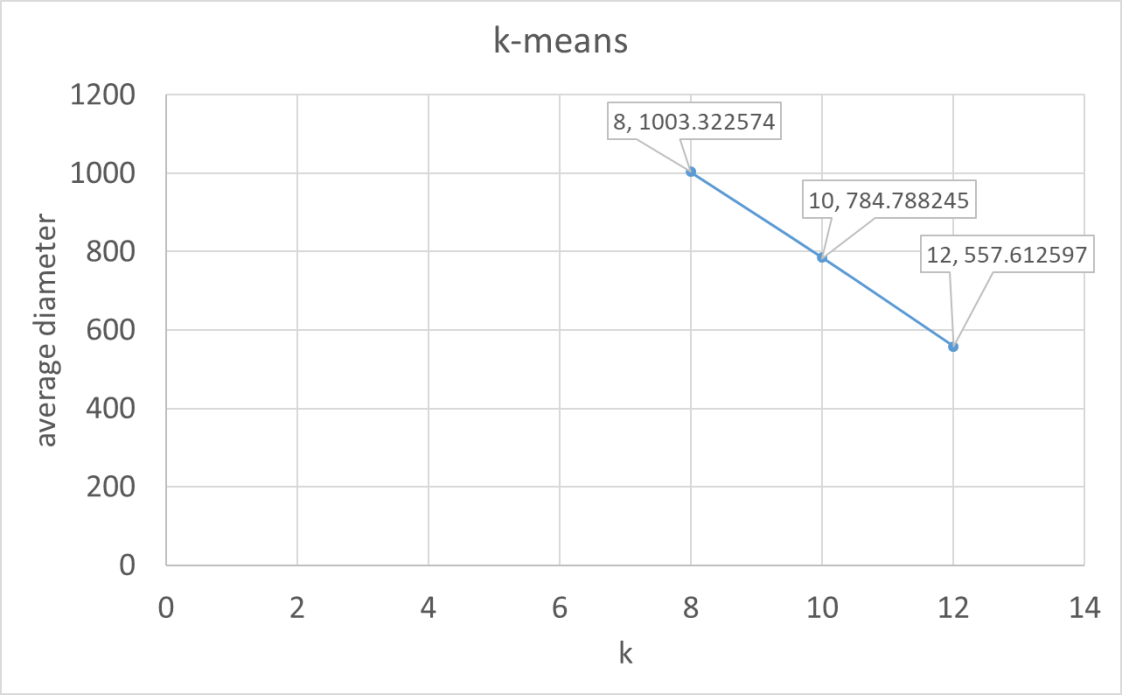
1-(b).



그래프를 보면 average diameter가 k값이 증가함에 따라 급격히 줄어들다가 k=16 ~ k=32 사이에서 거의 줄어들지 않게 되는 것을 볼 수 있다. 따라서 정확한 k 값은 k=8 ~ k=16 사이에 위치한다는 결론을 내릴 수 있다. 따라서 이진 탐색을 위해 중앙 값인 k=12일 때를 측정해보았다.



급격한 변화는 k=8에서 k=12 사이에 있기 때문에 다시 이진 탐색을 위해 k=10일때를 보았다.



k=8 ~ k=12 사이에서는 diameter가 선형으로 감소하는 모습을 보였다. 급격히 감소하는 구간이 없기 때문에 이 데이터와 맞는 클러스터 개수는 선형 구간의 중점인 **10개**라고 할 수 있다.

2.

Exercise 11.1.7

(a).

```
import numpy as np

M = np.array([[1,1,1], [1,2,3], [1,3,6]])
x = np.array([[1], [1], [1]])
diff = 1
while diff > 0.0001:
    x_old = x
    x = np.dot(M, x) / np.linalg.norm(np.dot(M, x))
    diff = np.linalg.norm(x - x_old)
print(x)
```

while문을 이용하여 power iteration을 구현하였다.

(d).

```
import numpy as np

M = np.array([[0.7043,0.2796,-0.312],
              [0.2796,0.2446,-0.1967],
              [-0.312,-0.1967,0.1787]])
x = np.array([[1], [1], [1]])
diff = 1
while diff > 0.0001:
    x_old = x
    x = np.dot(M, x) / np.linalg.norm(np.dot(M, x))
    diff = np.linalg.norm(x - x_old)
print(x)
```

(e).

```
import numpy as np

M = np.array([[0.0376,-0.0539,0.0211],
              [-0.0539,0.0778,-0.0301],
              [0.0211,-0.0301,0.0122]])
x = np.array([[1], [1], [1]])
diff = 1
while diff > 0.0001:
    x_old = x
    x = np.dot(M, x) / np.linalg.norm(np.dot(M, x))
    diff = np.linalg.norm(x - x_old)
print(x)
```

Exercise 11.3.1

(b) ~ (c)

```
import numpy as np
from numpy import linalg as LA

M = np.array([[1,2,3], [3,4,5], [5,4,3], [0,2,4], [1,3,5]])
MTM_value, MTM_vector = LA.eig(np.dot(M.T, M))
MMT_value, MMT_vector = LA.eig(np.dot(M, M.T))
print(MTM_value)
print(MTM_vector)
print(MMT_value)
print(MMT_vector)
```

Exercise 11.4.2

(a).

```
import numpy as np
from numpy import linalg as LA

W = np.array([[3, 3], [4, 4]])
print(np.linalg.svd(W))
```

(b).

```
import numpy as np
from numpy import linalg as LA

W = np.array([[5, 5], [0, 0]])
print(np.linalg.svd(W))
```

(c).

```
import numpy as np
from numpy import linalg as LA

W = np.array([[1, 0], [0, 2]])
print(np.linalg.svd(W))
```

3-(c).

실행방법

Python hw2_3c.py path/to/ratings.txt path/to/movies.txt path/to/ratings_test.txt

알고리즘

이 문제에서는 1. Utility Matrix 정규화, 2. Matrix Factorization, 3. Gradient Descent, 4. Regularization, 5. Genre가 사용되었다. 먼저 3-(b)와 같이 Utility Matrix에서 각 User의 average rating을 뺐다. 여기서 뺐을 때 0이 되는 element는 0.00001로 두었는데, 이는 Gradient Descent 과정에 이 element가 포함되도록 하기 위함이었다.

그 후 k = 100으로 설정하여 U = [#_user X 100], V = [100 X #_movie] 행렬을 평균이 0이 되도록 초기화하였으며 이를 Gradient Descent 알고리즘을 통해 UV와 Utility Matrix M의 MSE가 최소가 되도록 했다. 여기서 MSE는 0이 아닌 element들에 대해서만 비교했으며, 따라서 0.00001로 설정

한 것이다.

Gradient Descent 알고리즘은 Cost를 최소화하는 방향으로 각 U, V element들을 조정하였다. 여기서 Over-fitting을 피하기 위해 Regularization Term을 두었는데, 따라서 Cost를 MSE + (U, V의 각 element 제곱의 합) 로 설정하였다. 따라서 Learning Rule은 element = element – learning_rate * (d_Cost / d_element) 이며 여기서 Cost에는 MSE 뿐만 아니라 Regularization Term도 포함되어 있는 것이다.

이를 통해 학습이 모두 끝나면, M* = UV가 되어 M*를 참고하여 ratings_test.txt에 있는 (user, movie) 쌍들에 대해 predict rating을 출력할 수 있다. 하지만 문제는 ratings.txt에 포함되어있지 않은 영화들이었는데, 이때 movies.txt에 있는 영화 장르를 이용했다.

User A에 대해 처음 보는 movie B에 대한 평점을 예측할 때, user A가 평가했던 영화들 중 movie B와 단 하나라도 공통된 장르가 있는 영화들을 골라 평점을 평균내었다. 이 평균이 movie B에 대한 예측 평점이 된다.