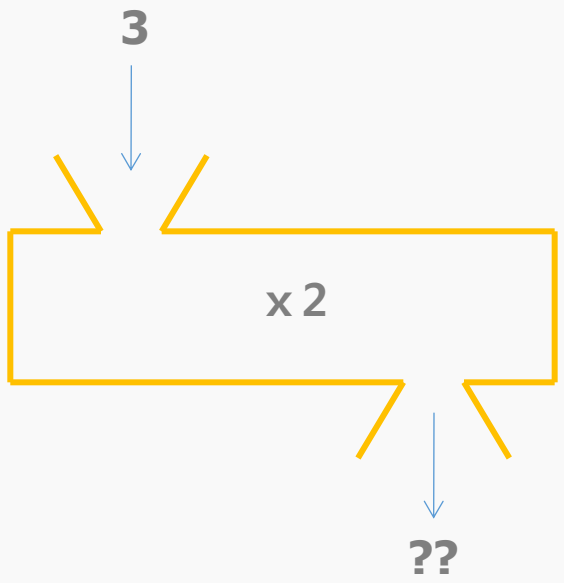
Programming Collective Intelligence

Chapter 3. Discovering Groups(군집발견)

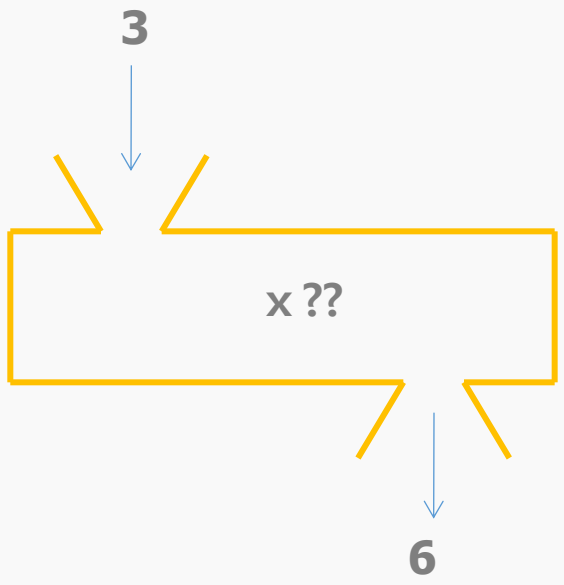
함수(Function), System, Filter, Hypothesis

Filter is a System that performs mathematical operations

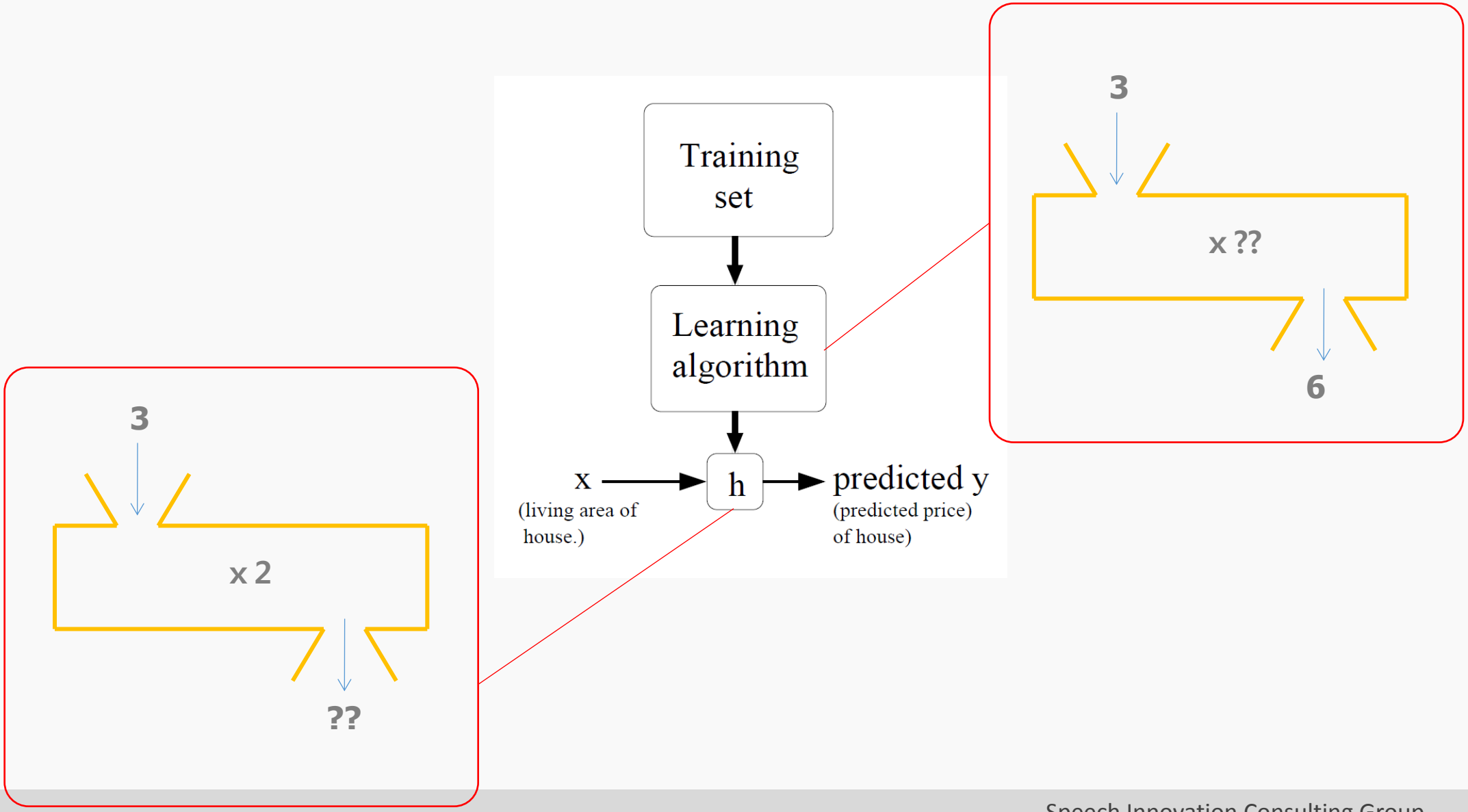


Learning, Adaptation

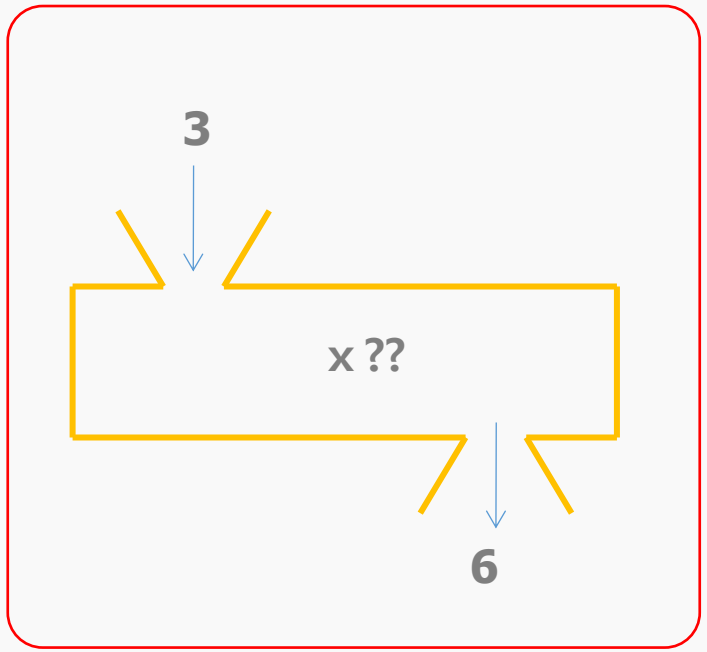
System Identification



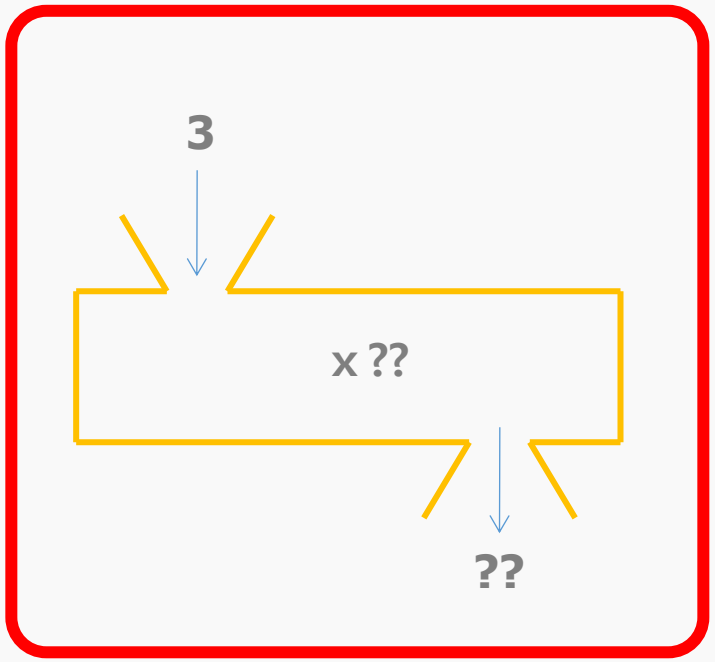
Learning Process



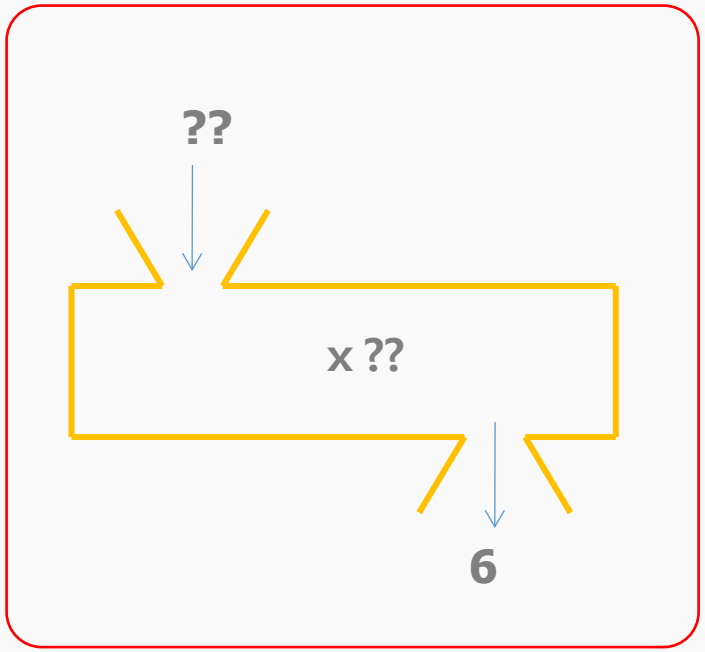
Supervised Learning vs. Unsupervised Learning



(Linear) Regression / Classification



Clustering
(뭔지 모르지만 공통된 녀석들을 묶어주자)



PCA / ICA / EM

할 수 있는 게 제한되어 있거나, 풀기 어렵거나 , 계산이 지저분하거나, heuristic 한 경우들이 많다...

Unsupervised Learning / Clustering

블로그에 나오는 단어 빈도수를 활용하여 블로거를 분류해보자.

"단어 빈도로 블로그를 군집하면,
비슷한 스타일을 가지거나
유사 주제에 대한 글이 자주 올라오는 블로그 그룹들을
찾을 수 있다."

1. Feed 내 단어 수 세기

Feedparser Link
https://pypi.python.org/pypi/feedparser#downloads

Feedlist.txt
http://whyDSP.org/263

Unicode encoding error
blog = blog.encode("utf-8")

Blogdata1.txt

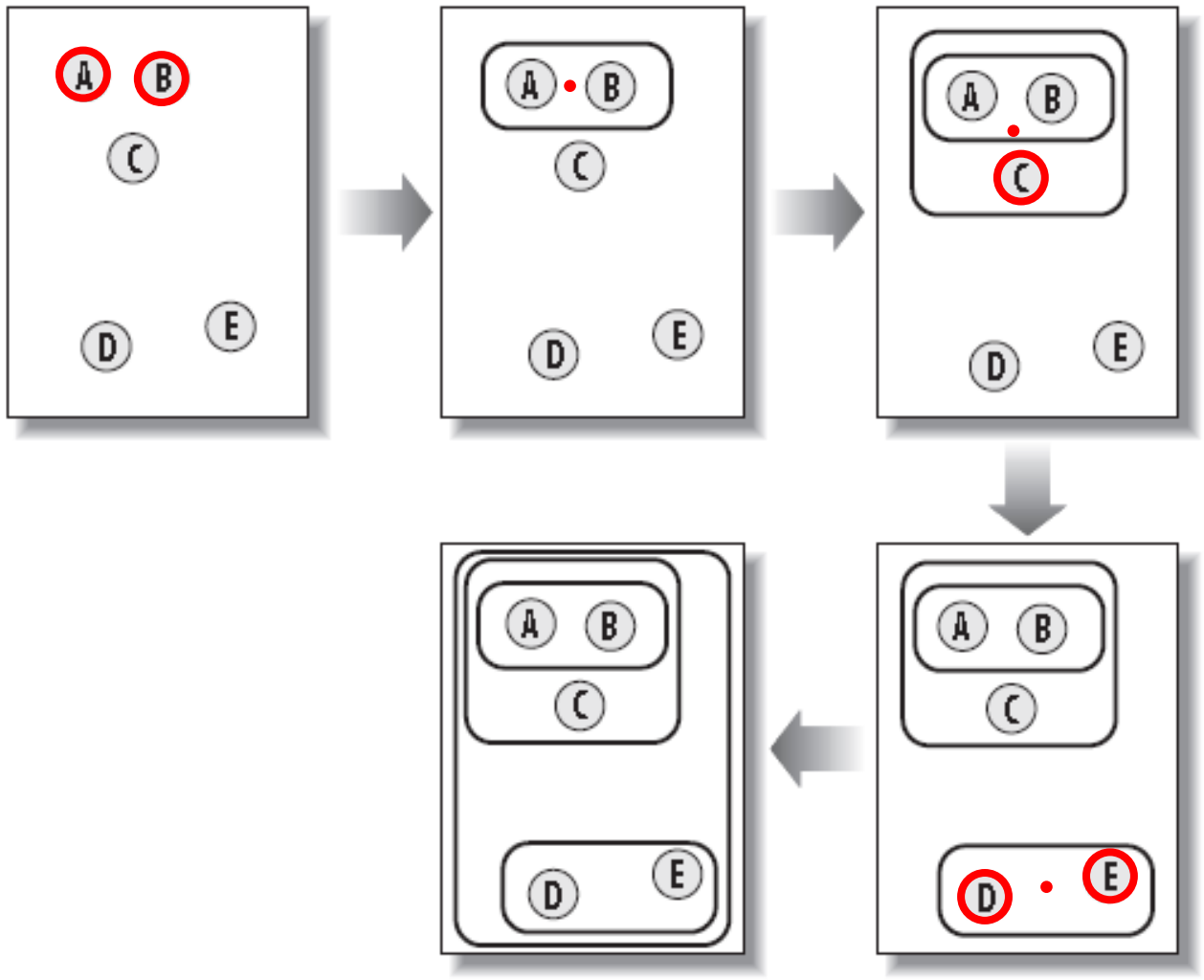
블로그별

단어별

Blog	four	looking	second	here	k	music	until	example	w
Schneier on Security	2	1	0	2	1	0	2	4	0	0	0	0
PaulStamatiou.com - Technology, Design and Photography								2	23	24	7
Search Engine Roundtable			0	0	0	4	0	0	0	0	0	0	0
Copyblogger	0	0	0	1	0	0	0	1	0	0	0	1	1	2	0
Joi Ito's Web	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
Engadget RSS Feed	0	0	4	2	0	1	0	0	0	0	0	0	0	3
TMZ.com	0	0	0	1	1	0	0	0	1	0	0	0	0	1	2	2
A Consuming Experience (full feed)					0	0	0	16	0	0	1	2	9
Latest from Crooks and Liars					0	2	0	1	0	1	0	1	1	0
Publishing 2.0	0	1	2	7	0	0	0	2	1	6	1	2	0	8
mezzoblue	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1
Eschaton	2	0	0	4	0	0	0	0	1	0	0	0	0	5	0
PerezHilton	0	0	1	4	0	0	0	0	1	0	0	0	0	3	0
Slashdot	0	1	0	0	9	1	0	0	0	1	0	0	0	0	0
Matt Cutts: Gadgets, Google, and SEO					0	0	0	2	0	0	0	1
Gothamist	1	0	1	0	2	3	1	0	0	0	0	0	0	0	0
we make money not art			1	0	0	1	0	1	0	0	0	0	1	0
BuzzMachine	0	0	0	9	0	0	0	0	1	0	0	0	0	0	0
Techdirt.	3	8	3	18	0	32	2	6	4	1	0	2	1	18	9

Word count

계층적 군집화 (Hierarchical Clustering)

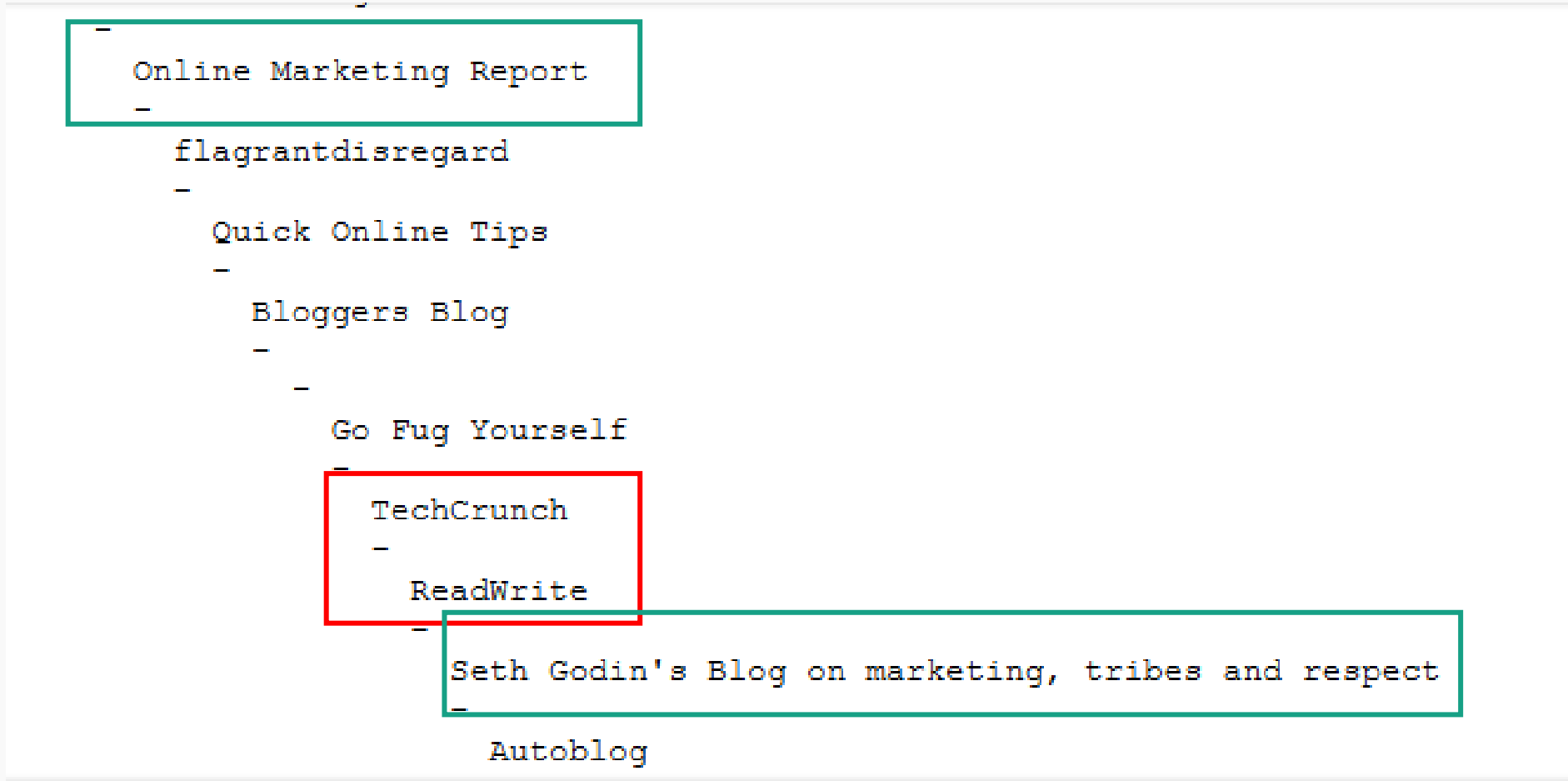


단어 빈도수 패턴이 가장 비슷한 (여기서는 Pearson Correlation Coefficient를 사용) 왜? 'Normalized'

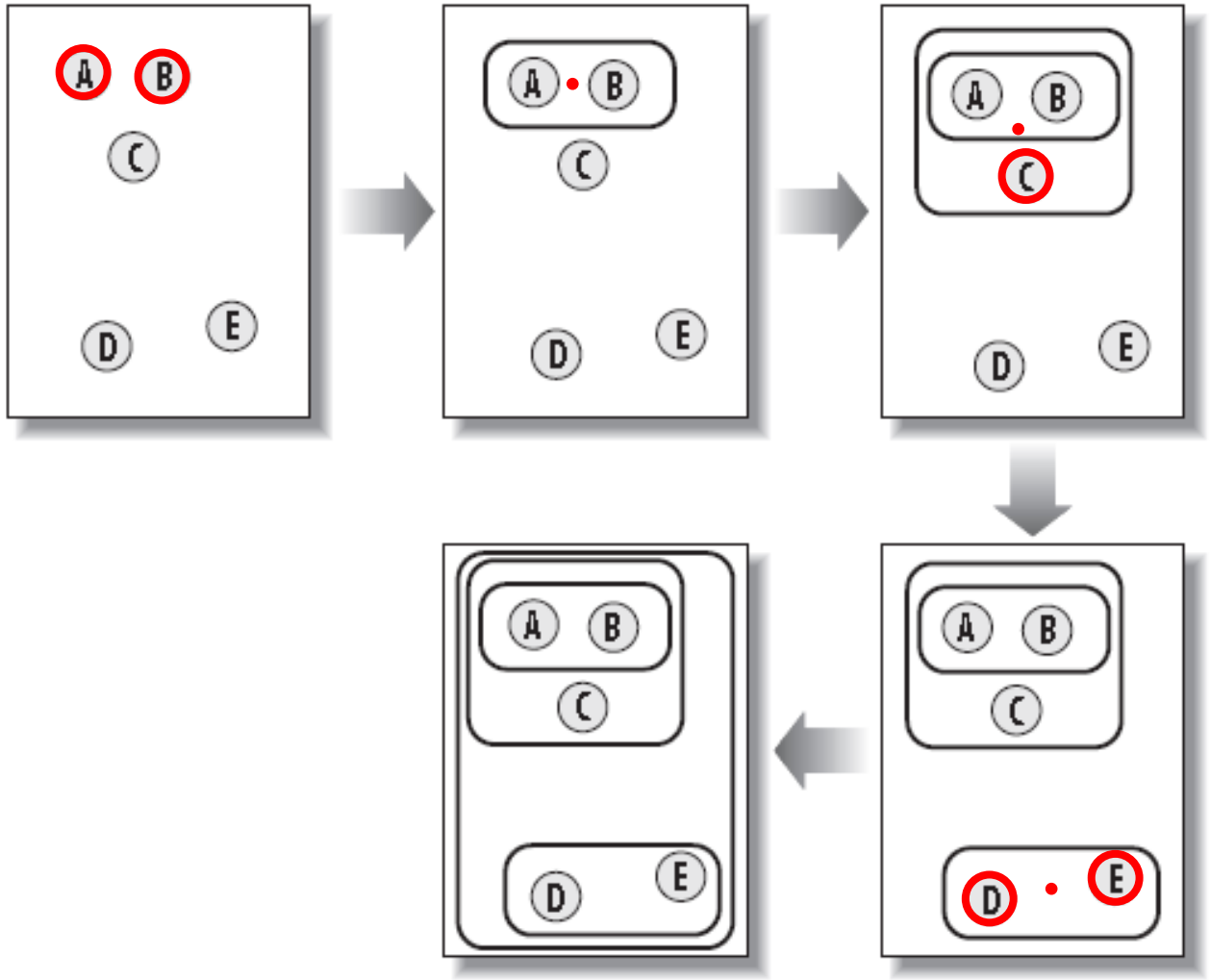
1. (유사도가) 가장 가까운 한 쌍을 찾는다.
2. 찾은 한 쌍을 하나의 그룹으로 만든다.
   (이 때, 그룹의 위치는 두 쌍의 중심이 된다.)
3. 위 과정을 1개의 그룹만 남을 때까지 반복

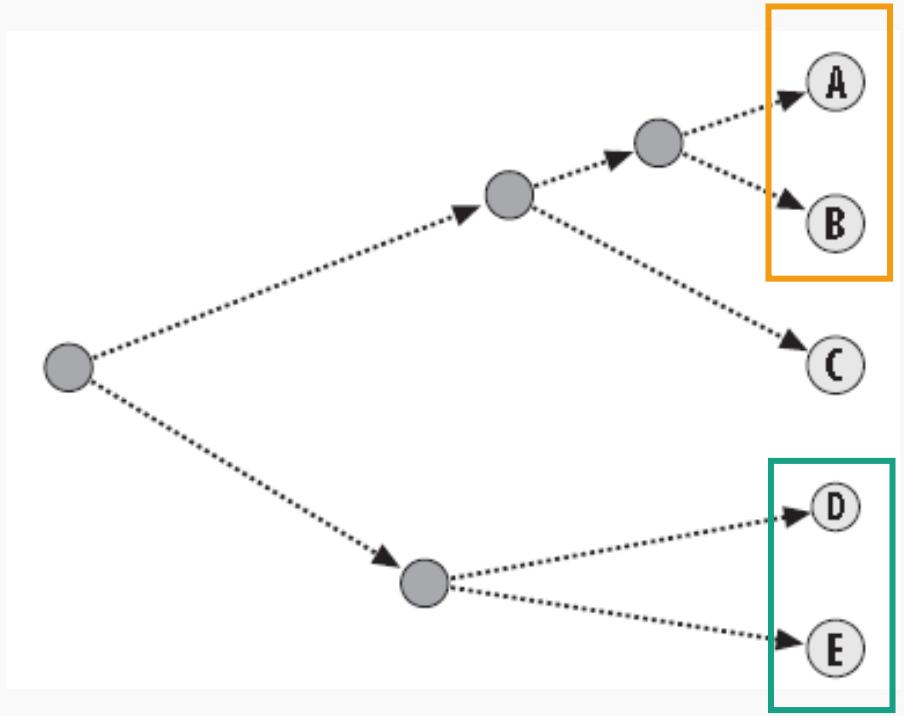
python

Print Hcluster



계통도(Dendrogram)



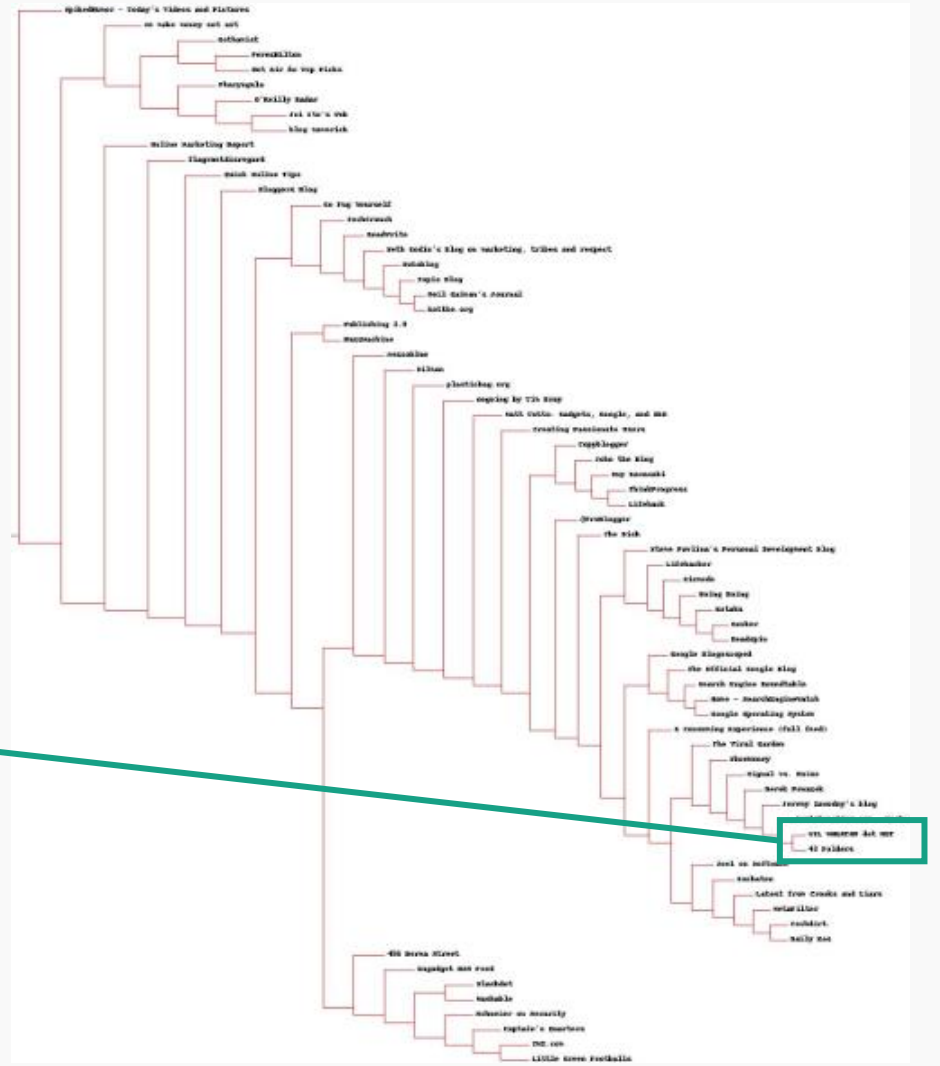


한 눈에 보기 편하다
AB가 DE보다 가깝다.
왜? 더 child node로 묶여 있다.

Draw Dendrogram

Python Image Library (PIL)

http://effbot.org/downloads/#pil



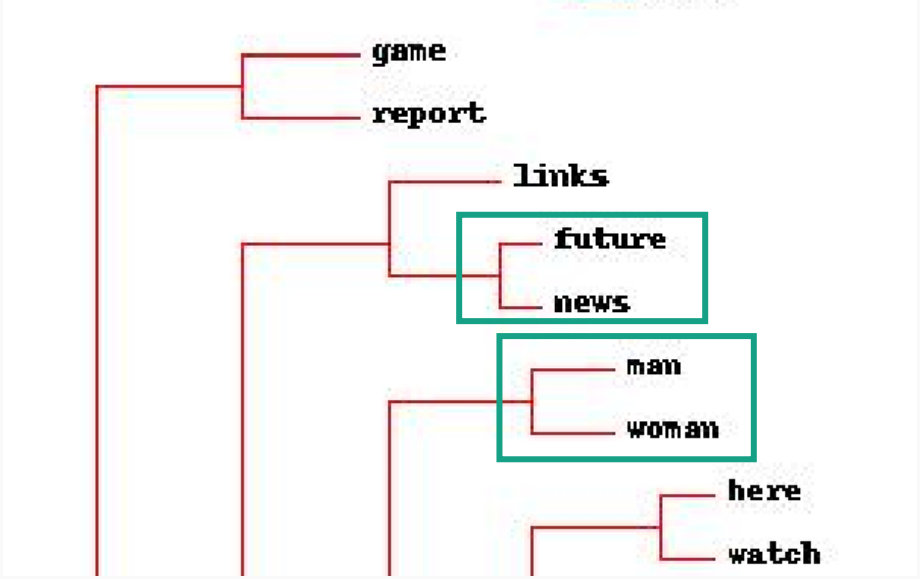
http://wilwheaton.net/about-wil-wheaton/

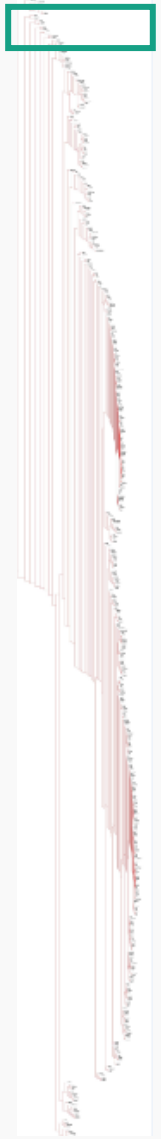
http://www.43folders.com/about

세로줄 군집화

블로그별로 clustering 했던 것을
word별로 clustering
(blogdata1.txt를 단순히 rotating 시켜주면 됨.)

Ex. 마트에서 어떤 물건을 함께 진열할까?





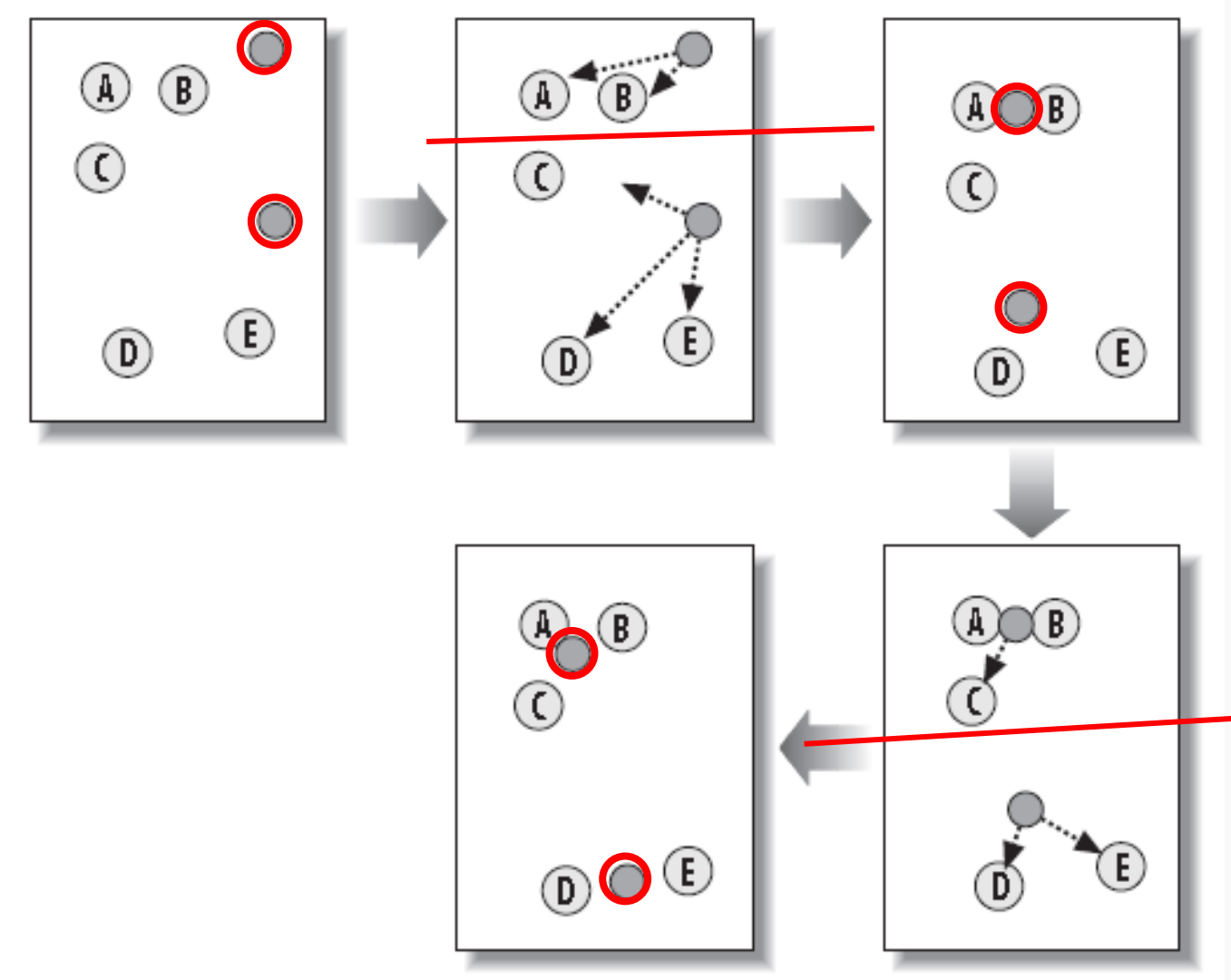
계층적 군집화(Hierarchical Clustering) 기법의 단점

계통도(dendrogram)도 그리 한눈에 들어오지 않는다. → 뚜렷한 그룹으로 쪼개지 못한다.

계산량이 많다.

→ K-means Clustering

K-means Clustering



1. 임의의 k점(예. K=2)을 initial centroid 로.
2. 각 centroid에서 가까운 node들을 하나의 그룹으로 만듬.
3. Centroid update : 각 그룹의 무게중심으로
4. 더 이상 그룹에 변화가 없을때까지 위 과정을 반복

선호도 군집

Zebo.com
(사람들이 가지고 싶은 물건 목록을 만드는 사이트)

Zebo.com 연결 안됨.
Zebo.txt 만 별도로 다운로드 하여 시뮬레이션.

가지고 싶은 아이템

익명의 사용자 index

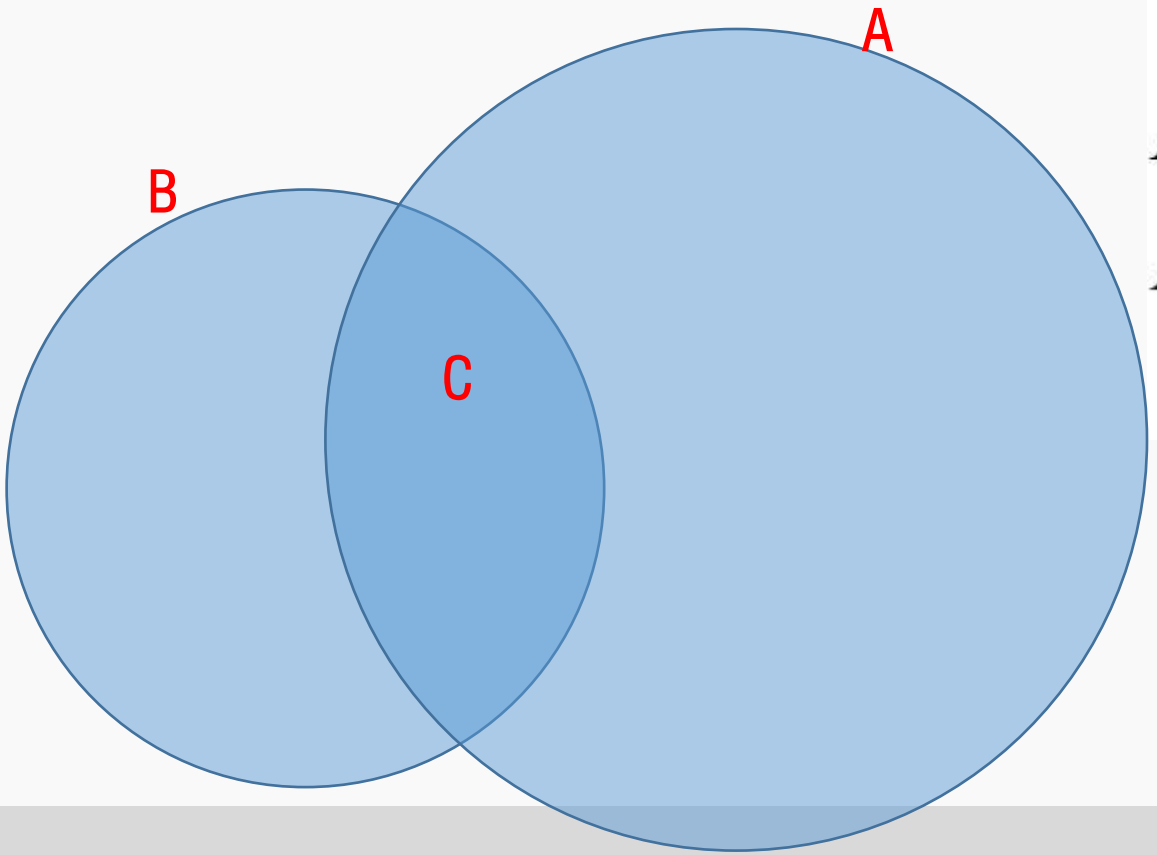
Zebo.txt

	item	U0	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U1
2	mansio		0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
3	sports	ar	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	bike	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0
5	clothes		0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
6	kids	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	phone	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	travel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	cell ph	ne	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0
10	better	ob	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	cloths	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
12	digicam		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	lots of	money	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	dog	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
15	xbox 36		0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
16	watch	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
17	life	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	laptop	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	0
19	million	dollars		0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	good jo		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	shirts	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0
22	love	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
23	nice ca		0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
24	cat	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
25	man	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	shoes	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
27	girl	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	plane	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

가지고 싶으면 1
안가지고 싶으면 0
Binary Data!!
→ Tanamoto Coefficient

Tanamoto Coefficient

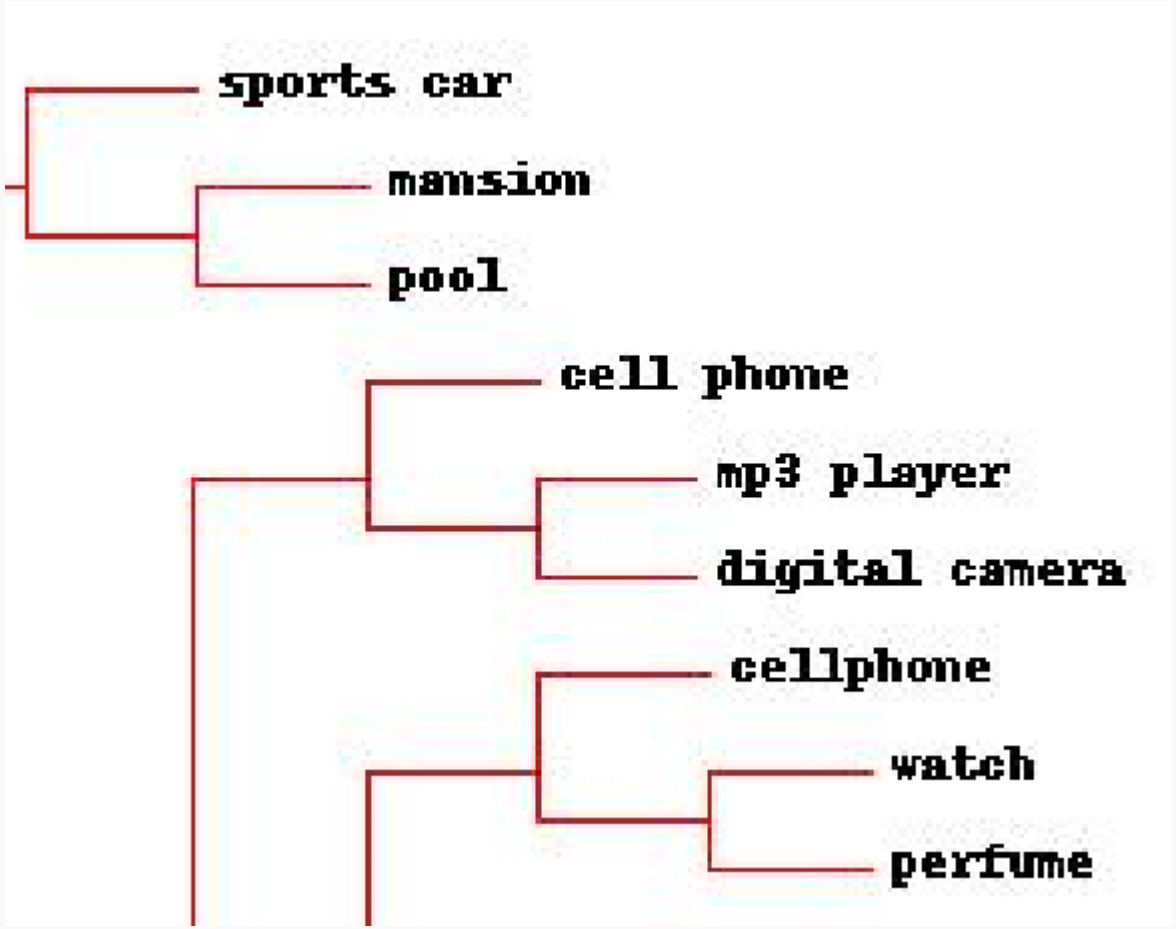
$$T(a,b) = \frac{N_c}{N_a + N_b - N_c} = \frac{\text{교집합}}{\text{합집합}}$$

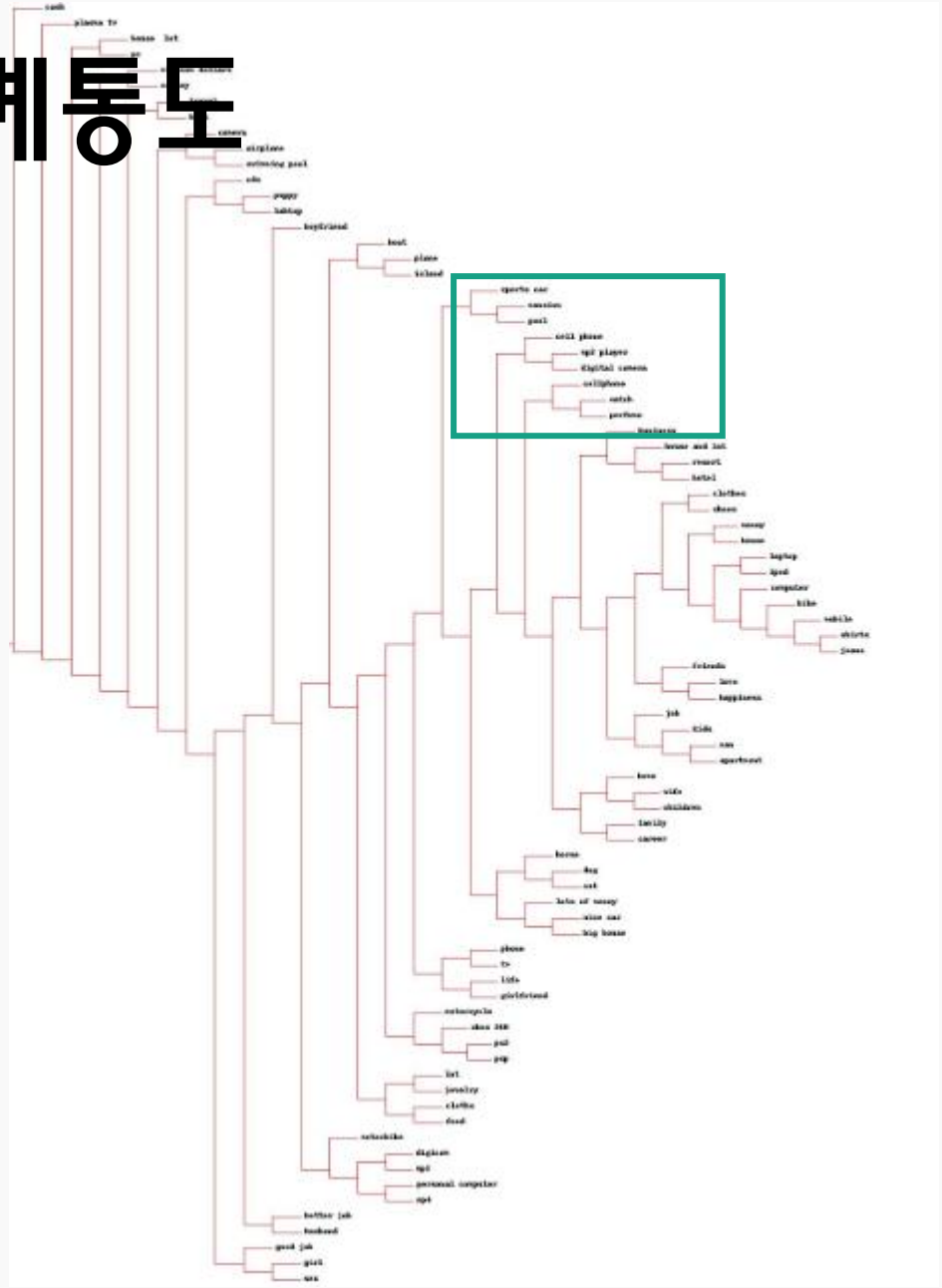


	items	users	
A		1 0 1 1 0 1	$\lvert A\rvert = 4$
B		1 1 0 1 0 0	$\lvert B\rvert = 3$
$A \wedge B$		1 0 0 1 0 0	$\lvert A \wedge B\rvert = 2$
$A \vee B$		1 1 1 1 0 1	$\lvert A \vee B\rvert = 5$

$$S_T(A,B) = \frac{2}{5}$$

선호도 군집 계통도


sports car
mansion
pool
cell phone
mp3 player
digital camera
cellphone
watch
perfume

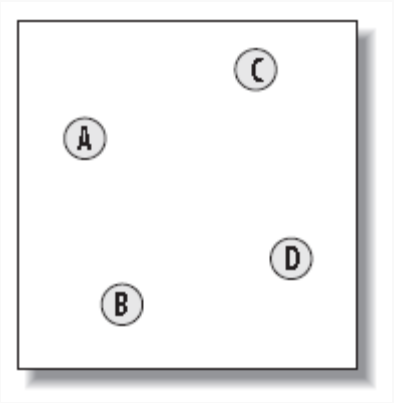


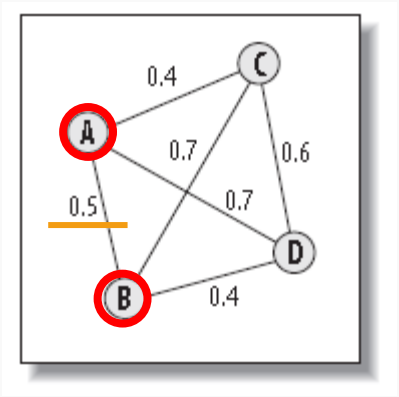
2차원으로 데이터 보기

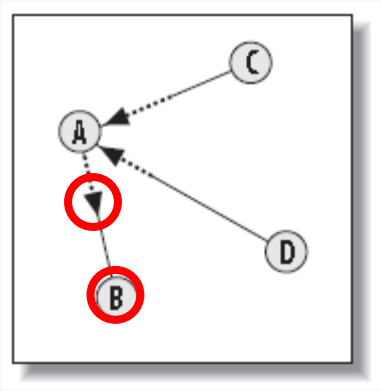
다시 블로그 군집 예제로 돌아와서…
블로그 A, B, C, D간 pearson correlation coefficient가 아래 표와 같이 분포 되어있다고 가정할 때…

	A	B	C	D
A	0.0	0.2	0.8	0.7
B	0.2	0.0	0.9	0.8
C	0.8	0.9	0.0	0.1
D	0.7	0.8	0.1	0.0

Multidimensional Scaling



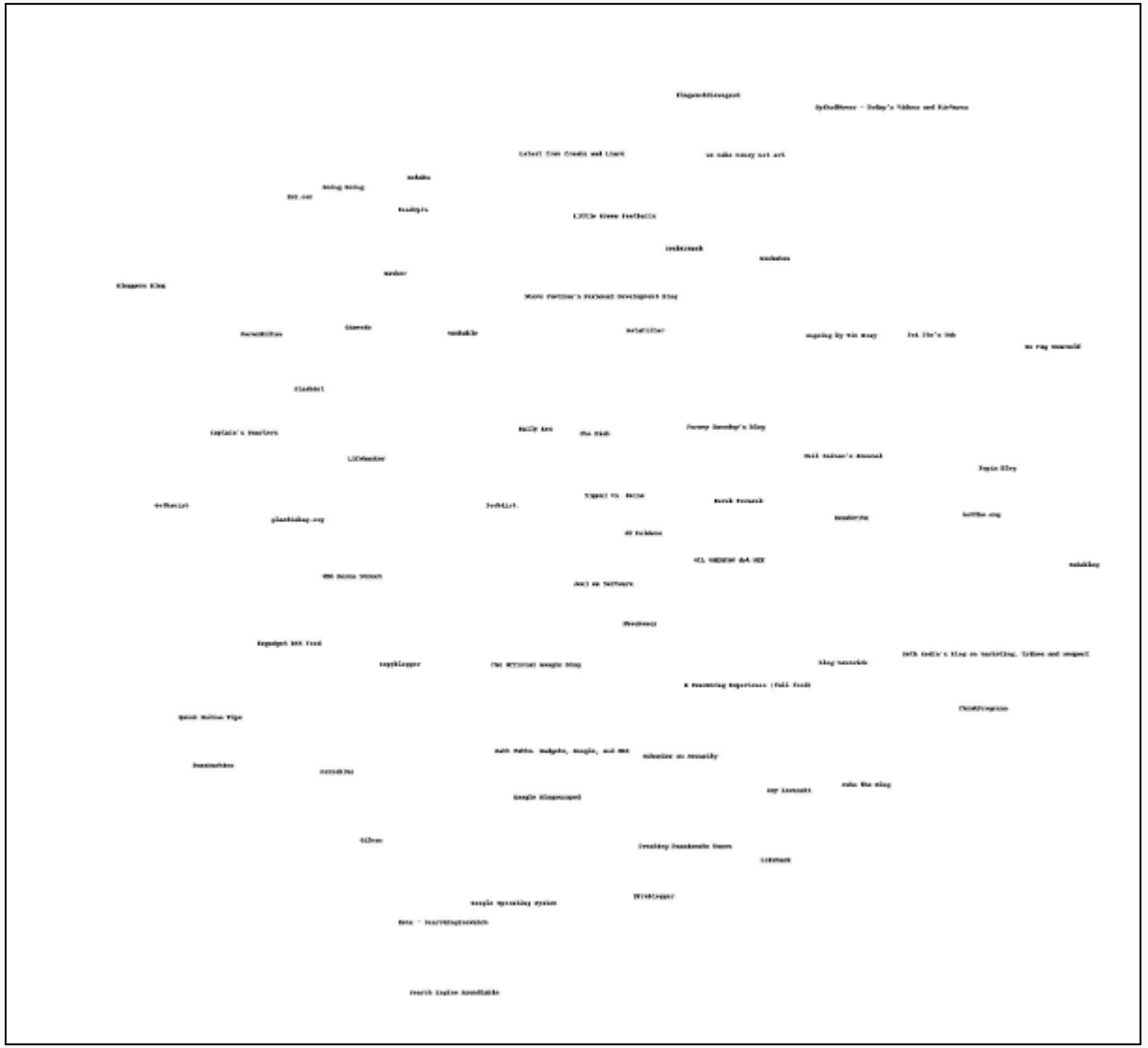




	A	B	C	D
A	0.0	0.2	0.8	0.7
B	0.2	0.0	0.9	0.8
C	0.8	0.9	0.0	0.1
D	0.7	0.8	0.1	0.0

1. Item을 2차원 공간안에 random하게 표시
2. 그 때의 실제거리(actual distance)를 표시
3. 목표거리(target distance, 표에 있는 유사도)와 실제거리를 비교.
   멀면 조금 댕겨주고, 가까우면 조금 밀어줌.
4. 위 과정을 총 error (target – actual)이 일정 이하가 될 때까지 반복

블로그 공간의 2D 출력



Seungil Kim

E-mail : goodksi@gmail.com
Blog : www.whyDSP.org
www.facebook.com/whyDSP
Twitter : @DrSonicwave