차원축소 기법 훑어보기 PCA, SVD, NMF

최범균, 2014-04-25

	i1	i2	i3	...	i11998	i11999	i12000
u1	0	0	0	...	1	0	0
u2	1	0	0	…	0	0	0
u3	1	0	1	...	0	0	1
...	...	...	...	...	...	...	...
u499998	0	0	0	...	0	0	0
u499999	0	0	0	...	0	1	0
u500000	0	0	0	...	0	0	0

u간 또는 i간 유사도를 구하려면?
많은 계산 필요, 0이 너무 많음

차원축소가 필요한 이유

- 계산 비용 축소
- 노이즈 제거
- 도출된 결과 이해

차원축소 알고리즘 몇 가지

- 주요 구성요소 분석(principal component analysis; PCA)
- 특이 값 분해(Singular Value Decomposition; SVD)
- 비음수 행렬 인수분해(Non-negative Matrix Factorization; NMF)

* “머신러닝 인 액션”과 “집단 지성 프로그래밍” 책 참고

1. PCA

주요 구성요소 분석(PCA)

- PCA란?
  - 데이터에서 두드러지게 데이터를 나누는
- PCA를 이용하면,
  - 중요한 속성 도출
  - 데이터 압축
  - 얼굴 인식(eigenface)
  - 기타 등등

PCA 구하기 과정 요약

- 데이터 집합에 대해 공분산 행렬 구함
  - 데이터의 집합의 각 열을 데이터의 속성이라고 가정
    - 즉, 열이 n개면, 속성 $x_1, x_2, \ldots, x_n$개 존재
- 공분산 행렬에서 고유값과 고유벡터 구함
- 고유값 목록에서 값이 높은 k개 인덱스 구함
  - 인덱스 값이 i1, i2, .., ik라고 할 경우
  - $x_{i1}, x_{i2}, .., x_{ik}$가 주요 구성요소가 됨

PCA 구하기 1: 공분산 행렬 구하기

데이터 행렬: 세 개의 속성

x1	x2	x3
2	1	3
8	4	7
1	4	4
2	8	2

공분산 행렬

Cov(x1, x1) 10.25	Cov(x1, x2) -0.41666667	Cov(x1, x3) 6
Cov(x2, x1) -0.41666667	Cov(x2, x2) 8.25	Cov(x2, x3) -1.66666667
Cov(x3, x1) 6	Cov(x3, x2) -1.66666667	Cov(x3, x3) 4.66666667

* 공분산: Cov(X, Y) = (∑ (x-x평균)(y-y평균)) / (n-1)
* Cov(X, X) = Var(X), 즉 분산

PCA 구하기2: 고유값 구하기

- 공분산 행렬에서 고유값, 고유벡터 구함

Cov(x1, x1) 10.25	Cov(x1, x2) -0.41666667	Cov(x1, x3) 6
Cov(x2, x1) -0.41666667	Cov(x2, x2) 8.25	Cov(x2, x3) -1.66666667
Cov(x3, x1) 6	Cov(x3, x2) -1.66666667	Cov(x3, x3) 4.66666667

고유값 목록:
[0.6541291, 14.33608183, 8.17645574]

고유벡터 목록:

0.51879659	0.81567701	0.25597093
-0.15597286	-0.20408152	0.96644876
-0.84054897	0.54131484	-0.02134668

* n차 정사각 행렬 A가 있을 때, 고유벡터V와 고유값 λ의 관계: AV = Vλ

PCA 구하기3: 고유값 목록에서 주요 구성요소 뽑기

x1	x2	x3
2	1	3
8	4	7
1	4	4
2	8	2

고유값 목록:
[0.6541291, 14.33608183, 8.17645574]

고유값 크기 순선에 따른 주요 구성요소의 순서:
x2 > x3 > x1

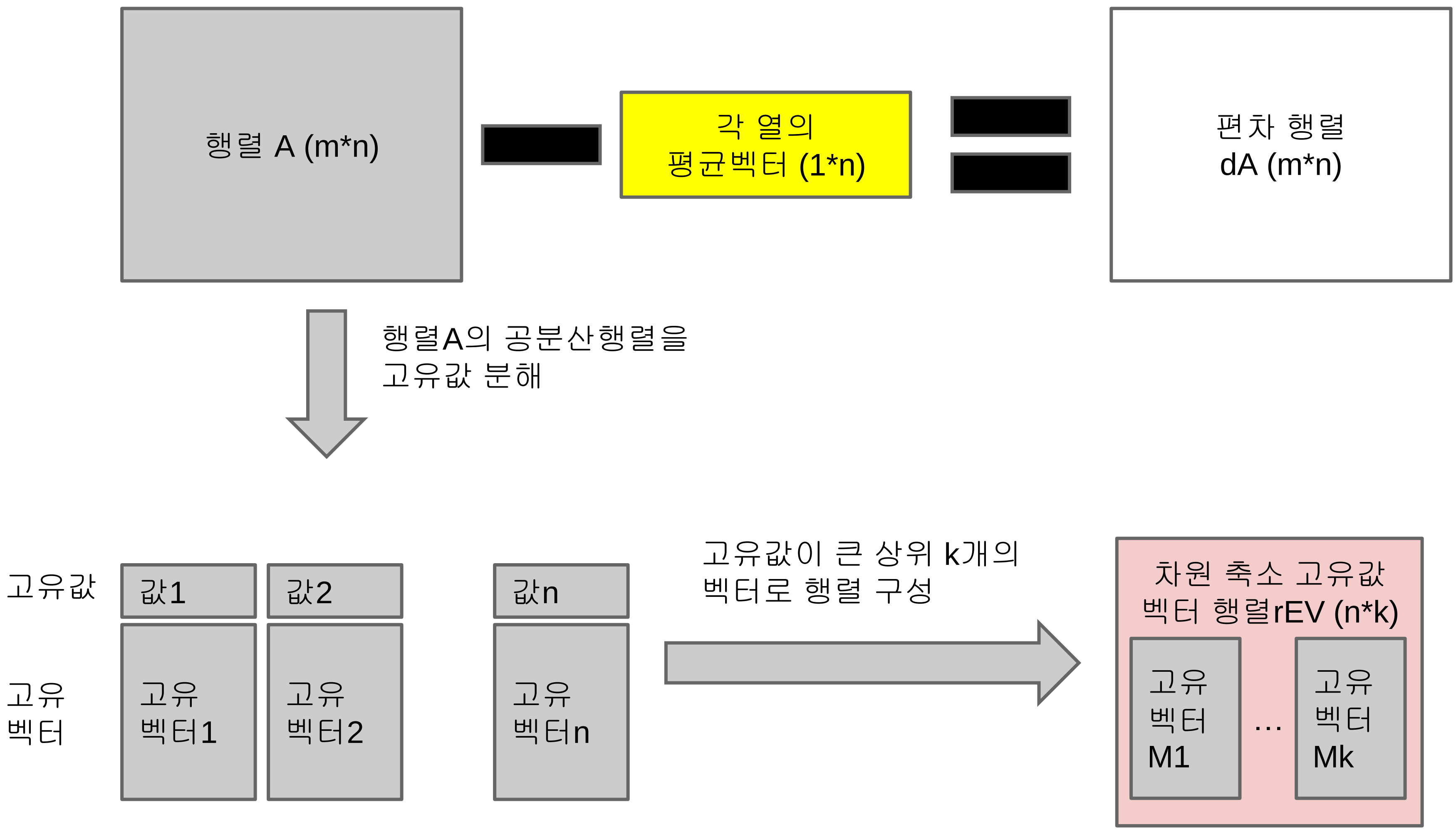
주요 구성 요소 2개를 추린다면,
97% 변화량을 반영하면서 3차원을 2차원으로 축소

x2	x3
1	3
4	7
4	4
8	2

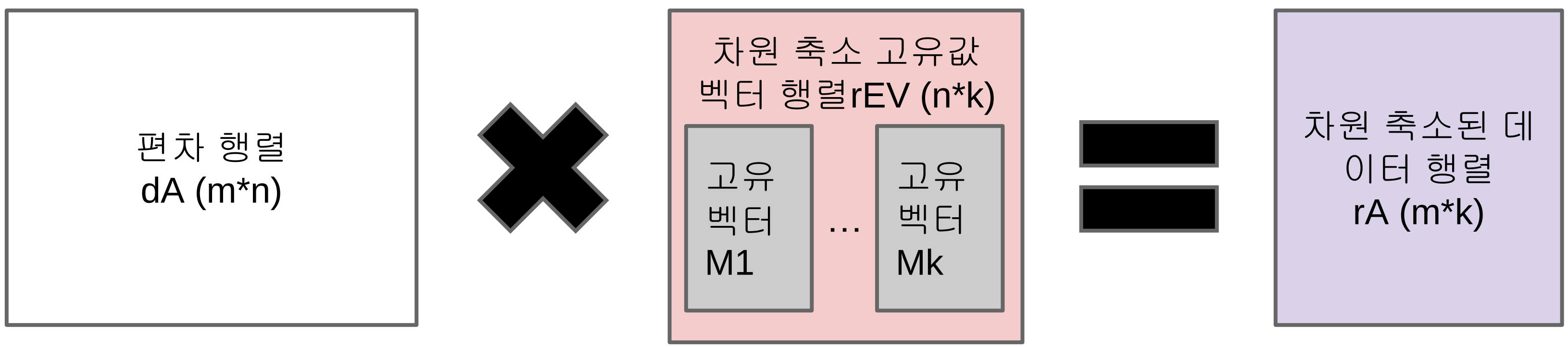
두 속성이 전체 변화량의 97% 차지

(14.336+8.176) / (14.336+8.176+0.654)

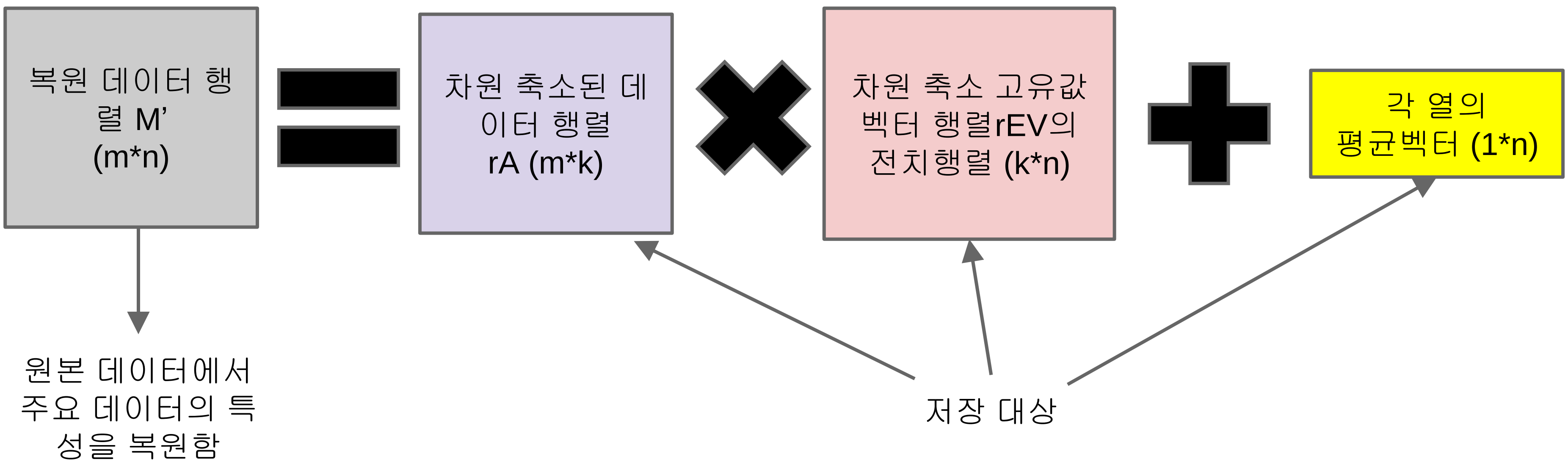
PCA를 이용한 데이터 압축 1/3



PCA를 이용한 데이터 압축 2/3



PCA를 이용한 데이터 압축 3/3



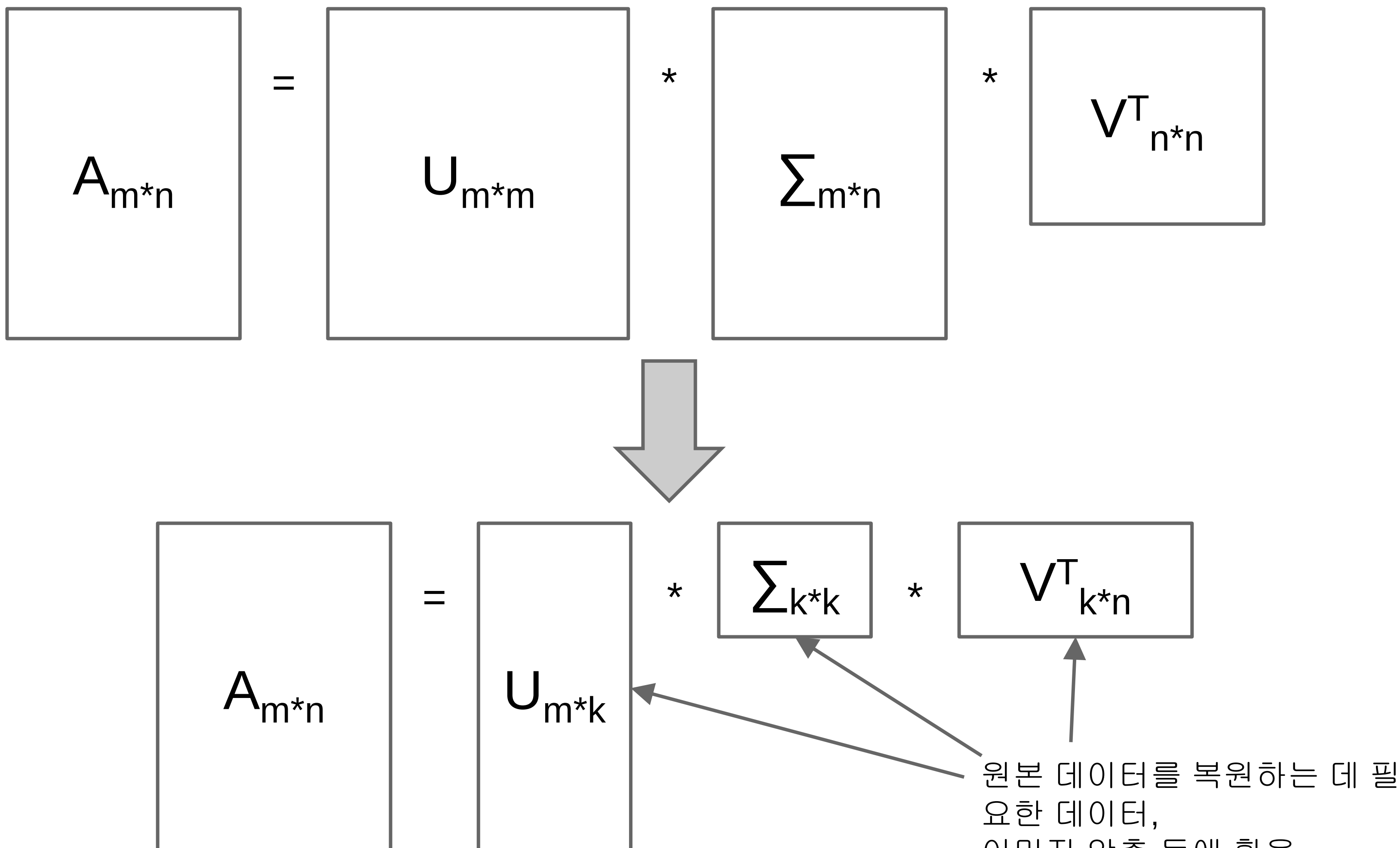
2. SVD

특이 값 분해**singular value decomposition(SVD)**

- 공식: $A = U\Sigma V^T$
  - A: m * n 행렬
  - U: m * m 직교 행렬 ($AA^T = U(\Sigma\Sigma^T)U^T$)
    - U의 열벡터는 AA^T의 고유벡터
  - V: n * n 직교 행렬 ($A^TA = V(\Sigma^T\Sigma)V^T$)
    - V의 열벡터는 A^TA의 고유벡터
  - ∑: m * n 특이 값 행렬!, 대각행렬
    - AA^T와 A^TA의 0이 아닌 고유값의 루트 값 목록
    - 값의 순서 (1,1) > (2,2) > … > (n, n) (m>n 경우)

* 직교 행렬: $AA^{-1}=AA^T=I$ 일 때, A를 직교 행렬이라 함
* I는 단위 행렬 (단위 행렬: AI=IA=A인 행렬 I, 대각선 1, 나머지 0)
* 대각 행렬: 대각선만 값을 갖고 나머지는 0인 행렬

SVD를 이용한 차원 축소: 데이터 축소



SVD를 이용한 차원 축소: 열 차원 축소

$$A_{m*n} = U_{m*m} * \Sigma_{m*n} * V^{T}_{n*n}$$

$$A'_{m*k} = U_{m*k} * \Sigma_{k*k}$$

열의 개수를 줄여서, 행 간 비교 연산 비용 낮춤.
행 간 유사도 연산 비용을 낮추는데 활용

SVD를 이용한 차원 축소: 행 차원 축소

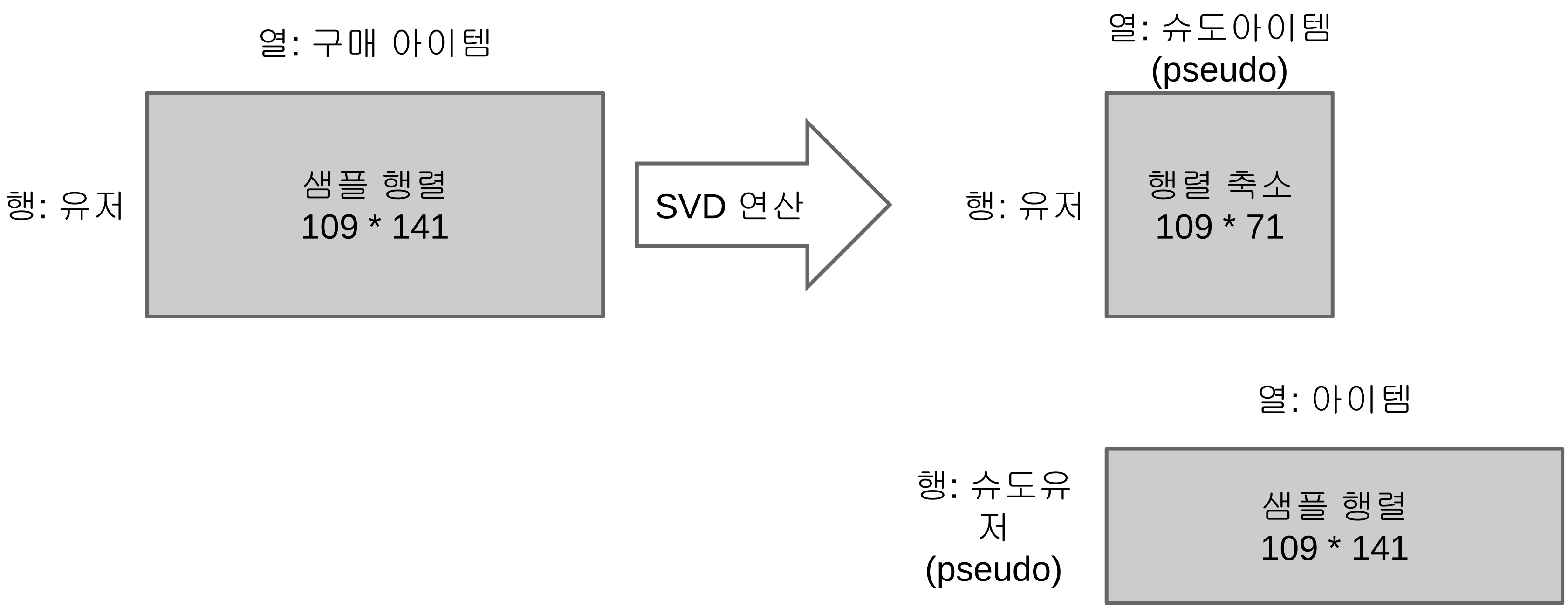
$$A_{m*n} = U_{m*m} * \Sigma_{m*n} * V^T_{n*n}$$

$$A'_{k*n} = \Sigma_{k*k} * V^T_{k*n}$$

행의 개수를 줄여서, 열 간 비교 연산 비용 낮춤.
열 간 유사도 연산 비용을 낮추는데 활용

SVD 적용 예, 유사도 분석을 위한 축소

90%를 넘기는 특이값은 71개



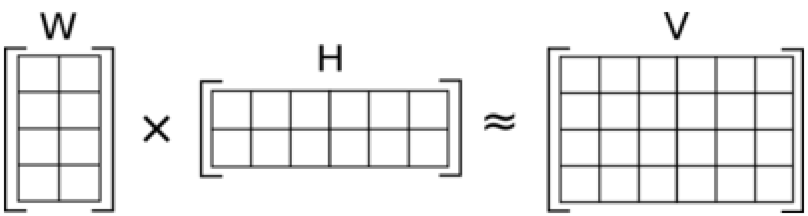
3. NMF

독립 특성

- 여러 데이터 중에서 주요 독립 특성을 이루는 데이터는?
  - 예,
    - 기사들 중에서 주제(독립 특성)들을 뽑아내고, 각 주제에 해당하는 기사 찾기
- 비음수 행렬 인수분해
  - NMF(Non-negative Matrix Factorization)
  - 독립 특성 추출 기법 중 하나

NMF

- 행렬 V가 있을 때, V를 W*H=V인 두 행렬 W와 H로 분해



출처: 위키피디

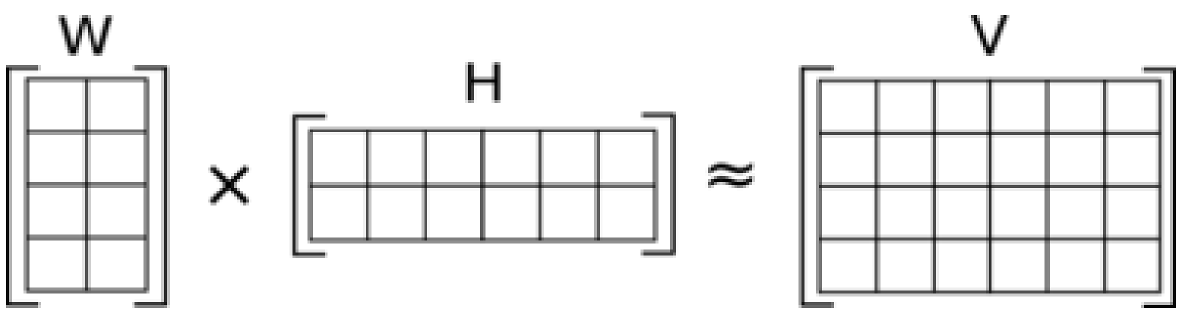
인수분해 알고리즘 (증배 갱신 규칙multiplicative update rule 사용)

1. W, H에 임의 값을 채움
2. W*H와 V가 유사해 질 때까지 까지 아래 과정 반복
   a. H 행렬 갱신
      i. $HN = W^T * V$, $HD = W^T * W * H + 0.000000001$
      ii. $H(r,c) = H(r,c) * HN(r, c) / HD(r,c)$
   b. W 행렬 갱신
      i. $WN = V * H^T$, $WD = W * H * H^T + 0.000000001$
      ii. $W(r,c) = W(r,c) * WN(r,c) / WD(r,c)$

NMF의 해석

W의 각 열이 특성이 됨:
- 열 길이가 2일 때 특성 f1, f2가 존재

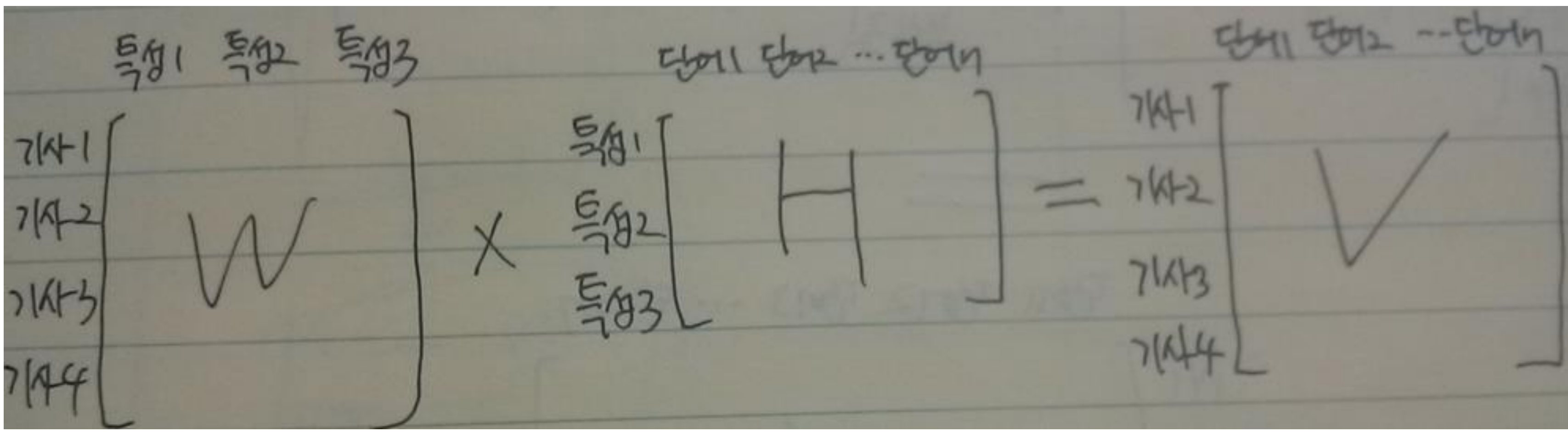
W의 각 행은 V의 같은 행이 '특성에 얼마나 적합'
한지(즉, 특성에 대한 가중치)를 나타낸다.



H의 각 행이 특성이 됨:
H의 각 열은 V의 같은 열이 '특성에 얼
마나 중요'한지를 나타낸다.

NMF를 독립 특성 발견에 활용

- 예, 온라인 기사의 주제 찾기



가중치 행렬:
- 각 특성이 기사에 얼마나 적합한가?
- 각 열에서 값이 높은 기사n개가 해당 특성을 갖는 기사
- 각 행에서 값이 높은 특성n개가 기사를 잘 반영하는 특성

특성 행렬:
- 각 단어가 특성에 중요한 정도
- 각 행에서 값이 높은 단어 상위n개가 해당 특성을 대표하는 단어가 됨 (즉, 주제가 됨)

* 특성의 개수 선택이 분류에 영향

NMF 활용 예, 구매자 금액대별 성향