

빅데이터 시대에서 확률모형의 역할

김용대¹

서울대학교 통계학과¹

목차

- 1절: 빅데이터 분석의 3가지 모드
- 2절: 군집분석에서 확률모형
- 3절: 문서분류에서 확률모형
- 4절: 개인화추천에서의 확률모형

1절: 빅데이터 분석의 3가지 모드

알고리즘

K-means clustering 개념/ 원리

(when $k = 2$, variable $x = (x_1, x_2)$)

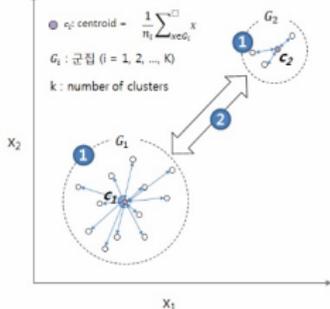
x : 변수(variable), m 개

○ 관찰값(observation), n 개

$$c_i: \text{centroid} = \frac{1}{n_i} \sum_{x \in G_i} x$$

G_i : 군집 ($i = 1, 2, \dots, K$)

k : number of clusters



1 군집 내 응집도 최대화 (maximizing cohesion within cluster)

- 군집1 데이터와 centroid c_1 과의 거리 합 최소화
- 군집2 데이터와 centroid c_2 과의 거리 합 최소화

$$\text{Min} \sum_{i=1}^k \sum_{x \in G_i} d(c_i, x)$$

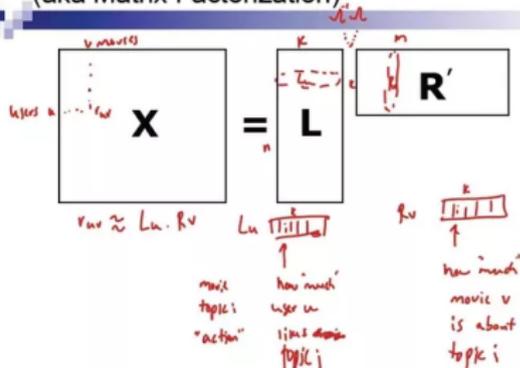
2 군집 간 분리도 최대화 (maximizing separation between clusters)

- 군집1의 centroid c_1 과 군집2의 centroid c_2 의 거리 최대화

$$\text{Max} \sum_{i=1}^k d(c_i, c_j), i \neq j$$

[R 분석과 프로그래밍] <http://rfriend.tistory.com>

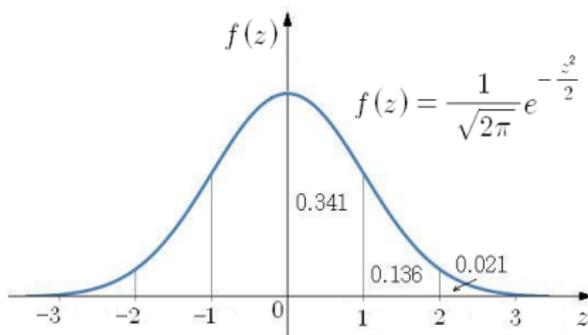
Interpreting Low-Rank Matrix Completion (aka Matrix Factorization)



©Carnegie Mellon 2013

7

확률모형



Patshala
पाठशाला



Starting point for Bayesian inference

- ▶ Write joint posterior density:

$$p(\beta, \mathbf{u}, \sigma_u^2, \sigma_\epsilon^2 | \mathbf{y}, \beta_o, \mathbf{V}_\beta) \propto (\sigma_\epsilon^2)^{-n+2} \exp\left(-\frac{1}{2\sigma_\epsilon^2}(\mathbf{y} - \mathbf{X}\beta - \mathbf{Z}\mathbf{u})'(\mathbf{y} - \mathbf{X}\beta - \mathbf{Z}\mathbf{u})\right) \\ \exp\left(-\frac{1}{2}(\beta - \beta_o)' \mathbf{V}_\beta^{-1}(\beta - \beta_o)\right) (\sigma_u^2)^{-n+2} \exp\left(-\frac{1}{2\sigma_u^2} \mathbf{u}' \mathbf{A}^{-1} \mathbf{u}\right) p(\sigma_\epsilon^2) p(\sigma_u^2)$$

- ▶ Want to make fully Bayesian probability statements on β and \mathbf{u} ?

- Integrate out uncertainty on all other unknowns.

$$p(\beta, \mathbf{u}, | \mathbf{y}, \beta_o, \mathbf{V}_\beta) = \int_0^\infty \int_0^\infty p(\beta, \mathbf{u}, \sigma_\epsilon^2, \sigma_u^2 | \mathbf{y}, \beta_o, \mathbf{V}_\beta) d\sigma_\epsilon^2 d\sigma_u^2$$

- 데이터를 숫자가 아닌 확률변수로 이해

본 강연의 목적

- 알고리즘적 분석방법과 확률모형 기반의 분석방법을 비교
- 알고리즘에 비교한 확률모형의 장점을 설명
- 3가지 분야를 고려
 - 군집분석: K-means vs 혼합모형
 - 문서분류: Tf-Idf vs 토픽모형
 - 개인화추천: 행렬분해 vs 개인화회귀모형

2절: 군집분석에서의 확률모형

군집분석

- 군집분석
 - 모집단 또는 범주에 대한 사전정보가 없는 경우에 사용하는 비지도 학습법
- 목적
 - 각 군집별 특성파악
 - 각 군집내의 개체들을 다른 군집에 속하는 개체들보다 서로 더 유사하도록 여러 군집으로 나눈 후 각 군집별 특성 파악
 - 특이값을 갖는 개체 발견
 - 결측값의 보정

군집분석

- 군집분석의 예
 - 고객 세분화
 - 고객을 인구통계, 구매패턴, 생활패턴 등과 관련된 변수들을 이용하여 여러 집단으로 나눔
 - 사기 방지 또는 이상 탐지
 - 정상적인 제품으로 군집 분석 후, 새로 생산된 제품이 군집과의 거리가 크면 자동적으로 불량품으로 인식

군집분석

- 유사성 척도

- 확률모형에 기초하지 않는 군집방법에서는 관측값들이 서로 얼마나 유사한지 또는 유사하지 않은지를 측정하는 척도 필요
 - 보통 비유사성을 이용하여 군집화
- 변수들이 모두 연속형인 경우
 - 유클리디언 거리

$$d(x, y) = \left(\sum_{i=1}^p (x_i - y_i)^2 \right)^{1/2}$$

- Cosine 거리

$$d(x, y) = \cos\theta = \frac{x \cdot y}{\|x\| \|y\|} = \frac{\sum_{i=1}^p x_i y_i}{\sqrt{\sum_{i=1}^p x_i^2} \sqrt{\sum_{i=1}^p y_i^2}}$$

군집분석

- 유사성 척도

- 변수들이 모두 범주형인 경우

- 차원이 p 인 두 이항 데이터 점 $\{x\}$, $\{y\}$ 의 좌표 값들에 대한 분할표

| | | y | | 계 |
|---|---|-------|-------|-------|
| | | 1 | 0 | |
| x | 1 | a | b | a + b |
| | 0 | c | d | c + d |
| 계 | | a + c | b + d | p |

- 이항데이터에서 0과 1이 동일한 중요도와 가중치를 가지는 경우 (대칭인 경우)

$$d(x, y) = \frac{b + c}{p}$$

- 비대칭인 경우(1이 더 중요한 경우)

$$\text{자카드 거리} = \frac{b + c}{a + b + c}$$

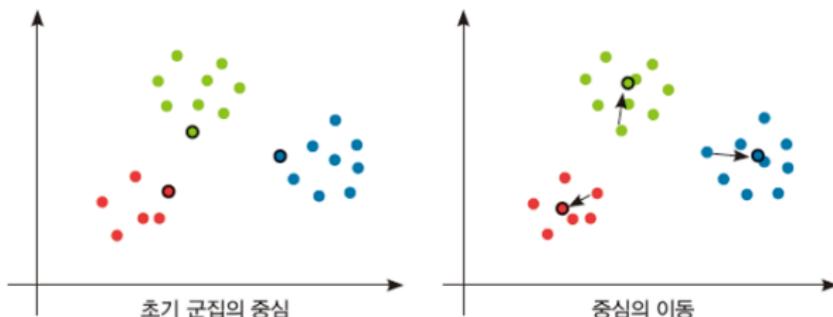
K-평균 군집분석

- K-평균 군집분석
 - 사전에 결정된 군집수 K 가 주어지면 유클리드 거리를 이용하여 전체 데이터를 상대적으로 유사한 K 개의 군집으로 나누는 방법

K-평균 군집분석

- K-평균 군집분석의 알고리즘

- ① 군집수 K 가 주어지면 랜덤하게 초기 K 개 군집의 중심 선택
- ② 각 관측값을 그 중심과 가장 가까운 거리에 있는 군집에 할당
- ③ 군집 중심을 새로 계산
- ④ 기존의 중심과 새로 계산된 중심 간에 차이 없을 때까지 2-3번 반복



K-평균 군집분석

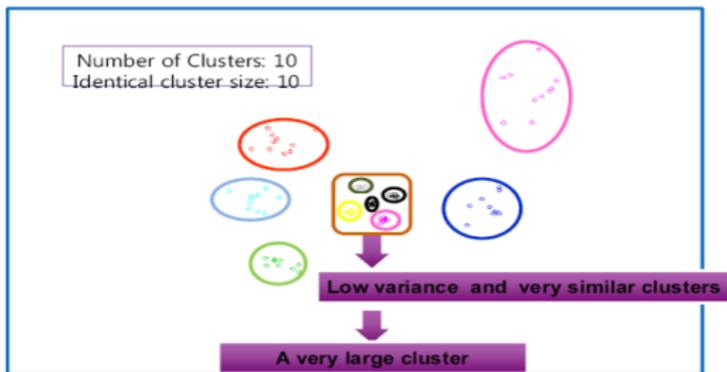
- K의 선택
 - 실제 데이터분석에서 적절한 군집 수를 미리 아는 경우는 드뭄
 - 군집수의 변경해 가면서 결과를 보고 최적의 군집 수를 결정
 - 일반적으로 K 값이 증가함에 따라 거리 제곱 합이 감소
 - K 값을 증가시켜 가면서 거리 제곱 합을 계산하고 $K - 1$ 때의 값보다 감소량이 줄어드는 K 값을 선택

K-평균 군집분석

- K-평균 군집분석 병렬처리 알고리즘
 - ① 데이터의 수를 n 개 slave node의 수를 p 개라 하면, 각 slave에 n/p 개의 데이터를 할당
 - ② Master는 각 slave에 현재의 K 개의 군집의 평균을 보냄
 - ③ Slave는 주어진 K 개의 평균을 이용하여 데이터를 군집으로 나눈 후, 새로운 군집의 평균을 master로 보냄
 - ④ Master에서 각 slave에서 구한 군집들의 평균을 구함
 - ⑤ 2-4단계를 master에서 구한 군집들의 평균이 수렴할 때까지 반복

모형기반 군집분석

- 기존의 군집분석의 문제점
 - 군집간의 분산이 다른 경우 안 좋음
 - 자료들간의 상관관계를 반영하지 못함
 - 이종의 변수가 섞여 있는 경우 거리를 정의하기 어려움



모형기반 군집분석

- 혼합모형 (Mixture model)

$$X_1, \dots, X_n \stackrel{iid}{\sim} \sum_{k=1}^K \pi_k f(x|\theta_k)$$

- 모형해석: K 개의 군집이 있고, 각 군집의 자료수는 π_k 에 비례하고 각 군집의 관측치의 분포는 $f(x|\theta_k)$ 이다.
- 모형추정: EM algorithm
- 군집할당

$$\operatorname{argmax}_k \pi_k f(x|\theta_k)$$

모형기반 군집분석

- 가우스 혼합모형

$$X_1, \dots, X_n \stackrel{iid}{\sim} \sum_{k=1}^K \pi_k \mathcal{N}(x | \mu_k, \Sigma_k)$$

- $\Sigma_k = I$ 이면 K-평균 방법과 동일
- 군집별 다른 분산을 줄 수 있으며 상관관계도 반영 가능

모형기반 군집분석

- EM 알고리즘

1 초기값 설정 $\hat{\pi}_k, \hat{\mu}_k, \hat{\Sigma}_k$ ($k = 1, \dots, K$)

2 Expectation 단계

$$\hat{\gamma}_{i,k} = \frac{\hat{\pi}_k \phi(\mathbf{x}_i; \hat{\mu}_k, \hat{\Sigma}_k)}{\sum_{j=1}^K \hat{\pi}_j \phi(\mathbf{x}_i; \hat{\mu}_j, \hat{\Sigma}_j)}, \quad i = 1, \dots, n; \quad k = 1, \dots, K.$$

3 Maximization 단계

$$\hat{\pi}_k = \sum_{i=1}^n \hat{\gamma}_{i,k} / n,$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^n \hat{\gamma}_{i,k} \mathbf{x}_i}{\sum_{i=1}^n \hat{\gamma}_{i,k}},$$

$$\hat{\Sigma}_k = \frac{1}{\sum_{i=1}^n \hat{\gamma}_{i,k}} \sum_{i=1}^n \hat{\gamma}_{i,k} (\mathbf{x}_i - \hat{\mu}_k) (\mathbf{x}_i - \hat{\mu}_k)^T.$$

모형기반 군집분석

- 다항분포 혼합모형

$$X_1, \dots, X_n \stackrel{iid}{\sim} \sum_{k=1}^K \pi_k \text{Cat}(x | p_{k1}, \dots, p_{kD})$$

- EM 알고리즘

- 1 초기값 설정 $\hat{\pi}_k, \hat{p}_{kd}$ ($k = 1, \dots, K; d = 1, \dots, D$)
- 2 Expectation 단계

$$\hat{\gamma}_{i,k} = \frac{\hat{\pi}_k \hat{p}_{kx_i}}{\sum_{j=1}^K \hat{\pi}_j \hat{p}_{jx_i}}, \quad i = 1, \dots, n; k = 1, \dots, K.$$

- 3 Maximization 단계

$$\hat{\pi}_k = \sum_{i=1}^n \hat{\gamma}_{i,k} / n, \quad \hat{p}_{kd} = \frac{\sum_{i=1}^n \hat{\gamma}_{i,k} I(x_i = d)}{\sum_{i=1}^n \hat{\gamma}_{i,k}}.$$

3절: 문서분류에서 확률모형

서론

Information Retrieval

- Query 와 문서 리스트

The screenshot shows a Naver search results page for the query '정보검색'. The page layout includes a search bar at the top, a left sidebar with navigation links, and a main content area with search results. The search results are organized into sections: '관련 검색어' (Related Search Terms), '내어비율' (Internal Ratio), '지서지' (Indexing), and '관련 검색어' (Related Search Terms). The '관련 검색어' section contains a list of related terms with their respective search volumes and trends.

| 관련 검색어 | 검색량 | 트렌드 |
|--------|---------------------|-----|
| 정보검색 | 426 | ↑ |
| 정보검색 | 157 | ↓ |
| 정보검색 | 402 | ↓ |
| 정보검색 | 603 | ↓ |
| 정보검색 | 424 | ↓ |
| 정보검색 | 129 | ↓ |
| 정보검색 | 3% | ↓ |
| 정보검색 | 42% | ↓ |
| 정보검색 | 2014.08.20 13:07 현재 | |

서론

TF-IDF

- 주어진 문서를 벡터로 만드는 방법
- 사전에 W 개의 단어가 있을 경우, 각 문서를 W 차원의 벡터로 만든다.
- W 차원의 벡터의 각 원소는 관련 단어가 문서에서 나온 빈도이다.
- 하지만 단순 빈도를 사용할 경우 자주 나오는 단어 (예: 'the', 'a')의 빈도가 너무 높아진다.
- 이러한 문제를 해결하기 위해서 TF-IDF를 사용한다.
- k 번째 문서에서 i 번째 단어의 TF-IDF값은

$$\text{TF-IDF}(k, i) = \text{Freq}(k, i) \times \log(N/n_i)$$

이고 여기서 N 은 전체 문서 수, n_i 는 i 번째 단어를 포함하는 문서의 수이다.

- Query와 문서들의 유사성을 TF-IDF의 유사성 (cos유사성, 상관계수)를 이용하여 측정한다.

서론

키워드 빈도를 이용한 IR 예제

- Query: "Applied multivariate analysis"
- Term-frequency Data Base

| | regression | histogram | Factor analysis | Multivariate | Asymptotic | clustering | Dimension reduction | Analysis | REL | MATCH |
|------|------------|-----------|-----------------|--------------|------------|------------|---------------------|----------|-----|-------|
| Doc1 | x | x | x | | | X | x | | R | |
| Doc2 | | | | X* | x | | | x* | | M |
| Doc3 | | | x | X* | | | | X* | R | M |

서론

숨겨진 의미: Latent Semantic

- 비슷한 단어 문제를 해결하기 위한 방법
- IR 은 단어들의 동시사용빈도(co-occurrent frequency)를 이용한 latent semantic을 추출
- "Applied" 와 "Regression" 또는 "Histogram"의 동시사용빈도는 "Asymptotic"보다 훨씬 크다.

서론

Latent semantics 예제

- psychology, history, english, philosophy => Human and social science
- mathematics, statistics, chemistry, physics, biology => Natural science

확률적 토픽 모형

Singular value decomposition for IR (Deerwester *et al.*, 1990)

$$\begin{array}{ccccccc} \boxed{C} & \approx & \boxed{T} & \times & \boxed{S} & \times & \boxed{D} \\ n \times W & & n \times K & & K \times K & & K \times W \end{array}$$

- K : semantic의 수
- T 를 바탕으로 문서들의 유사성을 측정

확률적 토픽 모형

SVD의 확률 모형: Latent Dirichlet Allocation (Hofmann, 1999)

- 문서: $j = 1, \dots, n$
- j 번째 문서의 i 번째 위치에 있는 단어: $x_{ji}, i = 1, \dots, n_j$
- 사전 : $w = 1, \dots, W$
- 토픽 (semantic): $k = 1, \dots, K$.

확률적 토픽 모형

SVD의 확률 모형: Latent Dirichlet Allocation (Hofmann, 1999)

- 확률적 토픽 모형

- j 번째 문서에 속한 단어들은 θ_{jk} 의 확률로 k 번째 토픽에 배정됨

$$p(z_{ji} = k) = \theta_{jk}$$

- 토픽이 주어졌을 때, 단어들은 다항분포를 따름

$$p(x_{ji} = w | z_{ji} = k) = \phi_{kw}$$

- 단어들의 주변분포는 다음과 같음

$$p(x_{ji} = w) = \sum_{k=1}^K \theta_{jk} \phi_{kw}$$

LDA의 베이지안 추론

사전분포

- 토픽의 수 K 가 알려졌다고 가정
- 다항분포의 켈레사전분포인 디리클레 분포 사용:

$$\theta_j \stackrel{\text{indep}}{\sim} \mathcal{D}(\alpha, \dots, \alpha)$$

$$\phi_k \stackrel{\text{indep}}{\sim} \mathcal{D}(\beta, \dots, \beta)$$

- 이 사전분포를 사용하는 토픽모형을 “Latent Dirichlet Allocation” (Blei *et al.*, 2003)라 부름
- α 와 β 는 토픽에 대한 사전 정보를 나타내는 초모수
 - α 가 작으면 문서당 토픽 수가 적음
 - β 가 작으면 토픽당 단어 수가 적음

LDA의 베이지안 추론

사후분포

$$\begin{aligned} P(\theta, \phi, \mathbf{z}|\mathbf{x}) &\propto P(\mathbf{x}|\phi, \mathbf{z})P(\mathbf{z}|\theta)P(\phi)P(\theta) \\ &\propto \left[\prod_{j=1}^n \prod_{i=1}^{n_j} \prod_{w=1}^W \phi_{z_{ji}, w}^{I(x_{ji}=w)} \right] \left[\prod_{j=1}^n \prod_{i=1}^{n_j} \prod_{k=1}^K \theta_{jk}^{I(z_{ji}=k)} \right] \\ &\quad \times \left[\prod_{k=1}^K \frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^W \phi_{kw}^{\beta-1} \right] \left[\prod_{j=1}^n \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^K \theta_{jk}^{\alpha-1} \right] \\ &\propto \left[\prod_{k=1}^K \frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^W \phi_{kw}^{\beta+N_{jkw}-1} \right] \left[\prod_{j=1}^n \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^K \theta_{jk}^{\alpha+N_{j\cdot k}-1} \right], \end{aligned}$$

where $N_{jkw} = \#\{i : x_{ji} = w, z_{ji} = k\}$.

MCMC 알고리즘

Gibbs sampler

- 가장 간단한 알고리즘은 모수들(θ, ϕ)과 잠재변수(\mathbf{z})를 모두 추출함 (full Gibbs sampler):

$$\theta_j | \mathbf{z}_j \sim \mathcal{D}(\alpha + N_{j1}, \dots, \alpha + N_{jK})$$

$$\phi_k | \mathbf{z} \sim \mathcal{D}(\beta + N_{.k1}, \dots, \beta + N_{.kW})$$

$$p(z_{ji} = k | \theta_j, \phi) \propto \theta_{jk} \phi_{kx_{ji}}$$

- 변수들간의 의존성 때문에 이 방법은 수렴이 느림

MCMC 알고리즘

Collapsed Gibbs sampler (Griffiths and Steyvers, 2004)

- θ 와 ϕ 를 적분하여 없애고 \mathbf{z} 만을 추출함:

$$p(z_{ji} = k | \mathbf{z}^{-ji}) \propto (N_{jk\cdot}^{-ji} + \alpha) \frac{N_{\cdot kx_{ji}}^{-ji} + \beta}{N_{\cdot k\cdot}^{-ji} + W\beta}$$

- 추출된 \mathbf{z} 를 통해 θ 와 ϕ 를 추출 가능
- full Gibbs sampler 방법에 비해 수렴속도가 빠름

HDP 토픽모형

LDA의 문제점

- K 를 사전에 지정해야 함
 - K 가 너무 작으면 중요한 토픽들을 놓칠 수 있음
 - K 가 너무 크면 불필요한 계산량이 많아짐
- 토픽별 빈도의 차이를 모형에 반영하지 못함 (현재 자주 쓰이는 토픽들이 미래 자료에서도 자주 쓰일 것이라 기대할 수 있음)

HDP 토픽모형

HDP 사전분포: Bayesian nonparametrics (Teh *et al.*, 2006)

$$\begin{aligned}G_0 &\sim \mathcal{DP}(\alpha_0, H) \\G_j | G_0 &\stackrel{\text{indep}}{\sim} \mathcal{DP}(\alpha_1, G_0) \\ \theta_{ji} | G_j &\stackrel{\text{indep}}{\sim} G_j\end{aligned}$$

HDP 토픽모형

Collapsed Gibbs sampler (Teh *et al.*, 2006)

- \mathbf{z} , π 를 반복적으로 추출:

$$p(z_{ji} = k \mid \mathbf{z}^{-ji}, \pi) \propto \begin{cases} (N_{jk\cdot}^{-ji} + \alpha_1 \pi_k) \frac{N_{\cdot kx_{ji}}^{-ji} + \beta}{N_{\cdot k\cdot}^{-ji} + W\beta} & \text{if } k \text{ previously used} \\ \alpha_1 \pi_u / W & \text{if } k = k^{\text{new}} \end{cases}$$

$$p(m_{jk} = m \mid \mathbf{z}, \pi) \propto s(N_{jk\cdot}, m) (\alpha_1 \pi_k)^m$$
$$(\pi_1, \dots, \pi_K, \pi_u) \mid (\mathbf{m}, \mathbf{z}) \sim \mathcal{D}(m_{\cdot 1}, \dots, m_{\cdot K}, \alpha_0),$$

where $s(n, m)$ is the Stirling's number and $\pi_u = \sum_{k=K+1}^{\infty} \pi_k$.

HDP 토픽모형

Infinite LDA representation (Teh *et al.*, 2006)

- HDP를 LDA의 극한으로 표현 가능
- 즉, K 가 커짐에 따라 LDA의 \mathbf{x} 의 주변분포는 HDP 사전분포를 사용하였을 때의 주변분포로 분포수렴함

HDP 토픽모형

Large K LDA vs HDP

- K 가 크면 LDA의 계산 속도가 매우 느려짐 (K 에 비례)
- HDP는 $K = \infty$ 임에도 불구하고 계산속도는 토픽수에 거의 영향을 받지 않음.
- 또한 K 에 대한 사후분포 계산도 가능하여 K 에 대한 추론도 가능

HDP 토픽모형

예: KBS 자료

- 2015년 사회 분야의 KBS 기사 자료를 전처리하여 분석에 사용
- HDP 토픽 모형으로 기사들을 분석하여 20개의 토픽을 찾아냄

| | | | | | | | |
|-----------|-------|------------|-------|-----------|-------|---------|-------|
| 대화형 기사 | | 해양사고 | | 법률, 헌법 | | 건설 | |
| 1450개 | 2.59% | 2365개 | 4.22% | 2091개 | 3.73% | 3621개 | 6.47% |
| 기업의 위법 행위 | | 개인 비리, 성범죄 | | 자연 | | 식품, 의약품 | |
| 2130개 | 3.80% | 3384개 | 6.04% | 2914개 | 5.20% | 2294개 | 4.10% |
| 질병, 전염병 | | 어린이집 아동 폭행 | | 날씨 | | 살인, 상해 | |
| 2366개 | 4.22% | 1740개 | 3.11% | 2913개 | 5.20% | 3774개 | 6.74% |
| 사기관련 범죄 | | 교육 | | 기업 비리(뇌물) | | 집회 | |
| 3850개 | 6.87% | 2637개 | 4.71% | 2762개 | 4.93% | 2949개 | 5.27% |
| 교통사고 | | 화재 | | 근로, 연금 | | 판결 | |
| 3544개 | 6.33% | 4410개 | 7.87% | 2369개 | 4.23% | 2402개 | 4.29% |

| | |
|--------|-------|
| 단발성 기사 | |
| 43개 | 0.08% |

HDP 토픽모형

예: KBS 자료

- 대화형 기사
 - 주요단어: 사람, 생각, 문제, 우리, 정도, 사실, 저희, 해서, 얘기, 상황
- 기업의 위법 행위
 - 주요단어: 업체, 회사, 직원, 대표, 투자, 계약, 공사, 대출, 사업, 운영
- 질병, 전염병
 - 주요단어: 병원, 환자, 치료, 건강, 발생, 결과, 의사, 수술, 금연, 담배
- 사기관련 범죄
 - 주요단어: 경찰, 혐의, 전화, 구속, 인터넷, 입건, 휴대, 서울, 불구, 피해자

4절: 개인화 추천에서 확률모형

추천 시스템의 예

생활 속의 추천 시스템

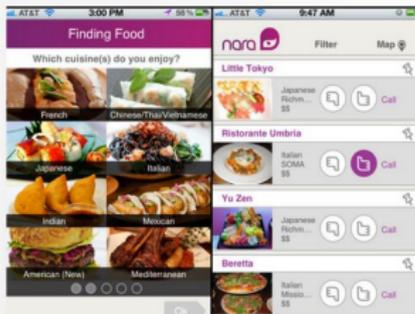
- 정보 홍수의 시대
 - 즐길 수 있는 다양한 작품들과 구매할 수 있는 온갖 상품들
 - 기호에 맞는 취사 선택 및 리를 위한 추천 시스템이 중요.
- 온라인에서의 추천 시스템들은 **통계학적 모델링과 분석에 기초함**



추천 시스템의 예

생활 속의 추천 시스템

- 본질적으로 객체 사이의 잠재적인 관계를 파악
 - 추천할 수 있는 것들
 - News Articles, Tags, Online Mates, Restaurants
 - Courses in e-Learning, Movies, Books, Various Goods
 - Research Papers, Citations, ...



내용 기반 추천 시스템

내용기반 추천 방법(Content based model)

- "Recommend items that are similar to those the user liked in the past"
- 고객이 선호하는 상품과 비슷한 내용의 상품을 추천.
 - 예 : 사용자가 영화 '스파이더맨'을 보았다면, RS는 '스파이더맨'에 대한 설명(heroes, adventure,...)을 참고하여 비슷한 영화를 찾아 추천.



내용 기반 추천 시스템

내용기반 추천 방법(Content based model)

- 장점 :
 - 다른 사용자의 정보나 평가 내역이 필요하지 않음.
 - 새로 추가된(혹은 아직 평가되지 않은) 항목 또한 추천이 가능.
 - 단점 :
 - 명시적으로 표현된 특징만을 다룰 수 있고, 질적인(Qualitative) 부분을 포착해내지 못함.
 - 추천하는 항목이 비슷한 장르에 머무름.
 - 예 : 로맨스 영화를 좋아하는 고객에게 로맨스 영화만을 추천.
 - 로맨스 영화를 좋아하는 고객에게 특정 자동차를 추천하는 것이 목표.
- ✓ 협력적 정화(Collaborative Filtering) 방법 탄생!

협력적 정화 방법

협력적 정화방법(Collaborative Filtering)

- 협력적 정화방법이란?
 - 개인화된 추천을 위한 통계적 방법
 - 개인의 선호도와 과거 상품 구매이력 등을 분석하여 개인에게 최적인 상품을 추천.
 - 기본 아이디어
 - 주어진 고객과 상품들에 대한 선호도가 비슷한 고객을 조사
 - 선호도가 비슷한 고객들이 좋아하는 상품 중에 주어진 고객이 모르고 있는 상품을 추천.
 - 종류
 - 고객 중심의 협력적 정화방법(User-based)
 - 상품 중심의 협력적 정화방법(Item-based)

상품 중심 협력적 정화 방법

품목 중심 협력적 정화방법



상품 중심 협력적 정화 방법

표기법

- $u = 1, \dots, n$: 고객
- $i = 1, \dots, l$: 상품
- r_{ui} : u 번째 고객의 상품 i 에 대한 선호도
- $\mathcal{R} = \{(u, i) : r_{ui} \text{ is observed}\}$.
- $s(i, j)$: 상품 i 와 j 의 유사도 (similarity)

$$s(i, j) = \frac{\sum_{u \in \mathcal{R}_{ij}} r_{ui} r_{uj}}{\sqrt{\sum_{u \in \mathcal{R}_{ij}} r_{ui}^2} \sqrt{\sum_{u \in \mathcal{R}_{ij}} r_{uj}^2}},$$

여기서 $\mathcal{R}_{ij} = \{u : r_{ui} \text{ and } r_{uj} \text{ are observed}\}$ 이다.

상품 중심 협력적 정화 방법

선호도 추정

•

$$\hat{r}_{ui} = \frac{\sum_{j \in \mathcal{R}_{ui}^{(k)}} s(i, j) r_{uj}}{\sum_{j \in \mathcal{R}_{ui}^{(k)}} s(i, j)},$$

여기서

$\mathcal{R}_{ui}^{(k)}$: $r_{uj} \in \mathcal{R}$ 중 상품 i 와 유사성이 큰 k 개 상품들 집합이다.

상품 중심 협력적 정화 방법

전통적인 협력적 정화방법의 문제점

- 자료의 sparsity로 유사성의 측정이 어렵다.
- 고객의 demographic정보나 상품의 내용정보를 분석에 사용하기가 어렵다.
- 새로운 고객이나 새로운 상품에 대한 추천이 어렵다.
 - ✓ Cold start problem
- ✓ 대안: 행렬분해를 이용한 협력적 정화방법

행렬분해 (Matrix Factorization)

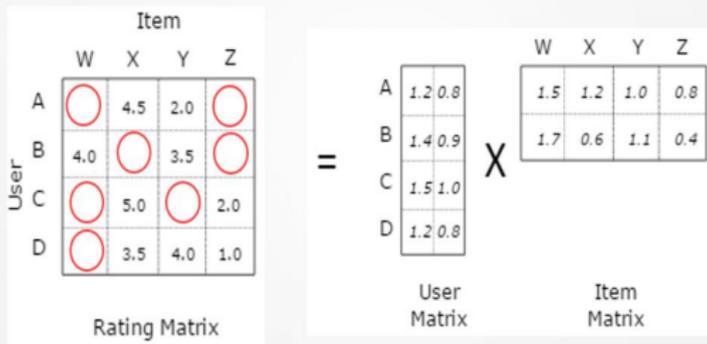
- \mathbf{R} 을 $n \times l$ 의 선호도 행렬이라 하자.
- $\mathbf{R} \approx \mathbf{AB}$ 을 가정한다. 이때 \mathbf{A} 와 \mathbf{B} 은 각각 $n \times k$ 와 $k \times l$ 행렬이다.
- 우리는 평가 되지 않은 \mathbf{R} 의 원소를 다음을 최소화 하도록 하는 \mathbf{A} 와 \mathbf{B} 를 추정함으로써 추정한다.

$$\sum_{(u,i) \in \mathcal{R}} (r_{ui} - a_u b_i)^2$$

이 때 \mathbf{A} 와 \mathbf{B} 의 추정에서 ridge 패널티와 같은 벌점 함수를 부여하고, a_u 는 \mathbf{A} 의 u 번째 행 벡터, b_i 는 \mathbf{B} 의 열 벡터이다.

행렬분해 (Matrix Factorization)

- 행렬 채워 넣기



- 잠재 요인 모델(Latent factor model)을 적용한 행렬 분해 방법 이용.

- 문제점

- 계산 속도가 느리다.
- 정확도가 최적이지 아닐 수 있다.(not optimal)

개인화 추천을 위한 확률모형: 다변량 정규 분포(Multivariate Gaussian distribution)

- R_{ui} 를 고객 u 가 상품 i 에 대해 평가한 선호도를 나타내는 확률변수라 하자.
- r_{ui} 를 R_{ui} 의 관측값이라 하고,
- $R_u = (R_{u1}, \dots, R_{ul})'$ 는 고객 u 에 대한 선호도 벡터라 하겠다.
- R_u 는 서로 독립인 확률 벡터이며 평균이 $\mu \in R^l$, 분산이 Σ 이다.

추정 : Method of Moment approach

Motivation

- 만약 μ 와 Σ 를 알고 있다고 가정하면,
- 선호도 자료가 주어졌을 때 R_{ui} 의 조건부기대값 $\mathbb{E}(R_{ui} | R_{uj} = r_{uj}, (u, j) \in \mathcal{R})$ 은 다음과 같다.

$$\mu + \mathbf{c}_{ui}' \Sigma_{ui}^{-1} (r_{u(-i)} - \mu_{(-i)}). \quad (1)$$

단, $\mathbf{c}_{ui} = (\sigma_{ij}, (u, j) \in \mathcal{R}, j \neq i)$, $\Sigma_{ui} = (\sigma_{jk}, j \in \mathcal{R}_u^U, k \in \mathcal{R}_u^U, j \neq i, k \neq i)$, $r_{u(-i)} = (r_{uj}, j \in \mathcal{R}_u^U, j \neq i)$, 그리고 $\mu_{(-i)} = (\mu_j, j \in \mathcal{R}_u^U, j \neq i)$ 이다.

- 따라서 μ 와 Σ 를 추정함으로써 관측되지 않은 선호도를 모두 추정할 수 있다.

추정 : Method of Moment approach

모수 추정

-

$$\hat{\mu}_j = \frac{\sum_{u:(u,j) \in \mathcal{R}} r_{uj}}{\sum_{u:(u,j) \in \mathcal{R}} 1}$$

- Σ 의 (j,k)번째 원소를 추정하기 위해 다음과 같은 \hat{cov}_{jk} 을 제안한다.

$$\hat{cov}_{jk} = \frac{\sum_{u \in \mathcal{R}_j^! \cap \mathcal{R}_k^!} (r_{uj} - \mu_{uj})(r_{uk} - \mu_{uk})}{\sum_u I(j, k \in \mathcal{R}_u^U)}.$$

- cov 의 추정에 희박조건(sparsity condition)을 줄 수도 있다.

$$\hat{cov}_{jk}^{soft} = (\hat{cov}_{jk} - \lambda / \sqrt{n_{jk}})_+, \quad n_{jk} = \sum_u I((j, k) \in \mathcal{R}_u^U).$$

추정 : Method of Moment approach

별점화(Regularized) 예측

- R_{ui} 의 별점화된 예측은 다음과 같다.

$$\mu_u + \mathbf{c}'_{ui}(\Sigma_{ui} + \lambda I_{n_{ui}})^{-1}(r_{u(-i)} - \mu_{(-i)}) \quad (2)$$

for some $\lambda > 0$, 여기서 $n_{ui} = \sum_{j \neq i} I(j \in \mathcal{R}_u^U)$ 이고, I_k 는 $k \times k$ 단위행렬이다.

- (2) 에 대한 예측은 Ridge 회귀분석을 통해 얻은 것이다.
- 이러한 별점화를 통해 예측 성능을 더욱 높일 수 있다.

실험 결과 비교

실험 설정

- 두가지 모형의 비교
 - 행렬 분해(Matrix factorization)
 - MME 방법
- 자료
 - R 의 "jester5k" 자료
 - $U = 5000, I = 100$
- 예측 정확도 평가
 - 70%의 학습 자료와 30%의 시험자료로 랜덤하게 분할
 - 분할을 10번 반복하여 시행

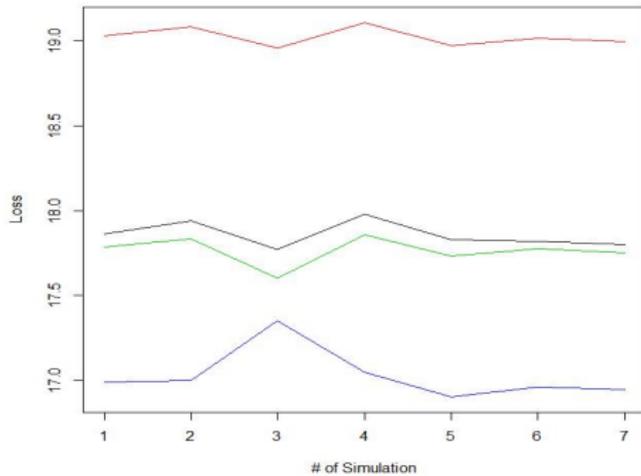
실험 결과 비교

조율 모수 λ 의 선택

- MME : Sign match 알고리즘
 - 관측값들과 예측값들의 부호를 가장 잘 매치시키는 λ 를 선택한다.
- MF : 예측 오차를 가장 작게 하는 λ 를 선택한다.

실험 결과 비교

정확도



실험 결과 비교

계산 시간

- MME가 iterative 알고리즘이 아니기 때문에 MF와의 계산 시간의 수치적 비교는 무의미 하다.
- MME 방법은 Hadoop이나 Spark로 쉽게 구현될 수 있다.

Thank you!!