특성분석과 순위분석

빅데이터 최신 기법 소개

김용대[1]

[1]서울대학교 통계학과

2015년 2월 26일

R을 이용한 빅데이터 분석 2015

목차

1부: 특성모형

목차

군집분석

- "군집분석"은 주어진 관측값들의 거리 또는 유사성을 이용하여 전체를 몇개의 집단으로 나누는 분석 방법
- 군집분석 종류
  - 계층적 군집분석
  - k-평균 군집분석
  - 모형기반 군집분석 (예:가우스혼합모형)
  - 커널기반 군집분석
  - ....
- 군집분석의 두 가지 특징
  - 거리 (예: 유클리드거리) 또는 유사성 (예:cosine값)이 필요하다.
  - 각 군집이 상호배반적이다.

0이 많은 고차원 이산형 자료

- 문서 뭉치
  - 차원: 사전에 있는 단어의 수
  - 관측치: 각 문서에서 각 단어가 나온 횟수
- 인터넷 쇼핑
  - 차원: 판매하는 상품 수
  - 관측치: 각 고객이 특정 기간 동안에 각 상품별 구입 횟수 (또는 구매 개수)

특성모형

- 0이 많은 고차원 이산형 자료의 분석 방법
- 일반적 군집분석의 문제점
  - 거리 또는 유사성 측도를 정하기 어렵다 (0이 많아서)
  - 상호배반적 군집이 적절하지 않다
    - 하나의 관측값이 여러 개의 속성을 가질 수 있다.
    - “대통령이 야구 시구를 하다"는 정치 기사이면서 동시에 스포츠 기사
    - 엄마와 딸이 같은 아이디로 인터넷 쇼핑
    - 쇼핑 이유가 여러 개 (예:아기 용품, 문화생활)
- Latent Dirichlet Allocation (LDA)

* Not “Linear Discriminant Analysis”

목차

자료구조

- 대상: n개
- 품목: W개
- n_j: j번째 대상이 품목을 선택한 횟수 $(j = 1 \ldots, n)$
- $x_{ji} \in \{1, \ldots, W\}$: j번째 대상이 i번째 선택한 품목 $(i = 1, \ldots, n_j)$

자료구조 예제 1

- 문서뭉치
- 대상: n개의 문서
- 품목: W개의 단어
- n_j: j번째 문서에 사용된 단어의 수 $(j = 1 \ldots, n)$
- $x_{ji} \in \{1, \ldots, W\}$: j번째 문서의 i번째 단어

자료구조 예제 2

- 인터넷 쇼핑
- 대상: n명의 고객
- 품목: W개의 상품
- n_j: j번째 고객이 구입한 상품의 수 $(j = 1 \ldots, n)$
- $x_{ji} \in \{1, \ldots, W\}$: j번째 고객이 i번째 구입한 상품

확률분해 모형 (Hofmann,1999)

- 확률적 주제 모형 (Probabilistic Topic model)
- 예) 문서 뭉치 분석
  - 문서에 나오는 각 단어에 주제 (예: 정치, 경제, 사회 등등)를 할당한다
  - j번째 문서의 i번째 단어의 주제가 $k \in \{1, \ldots, K\}$일 확률은

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

  - 주제가 주어진 경우에 i번째 단어의 출현 확률은

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

확률분해 모형 (cont.)

- 결론적으로, 확률주제모형은 단어출현 확률을 다음과 같이 분해한다

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

- $\theta_j = (\theta_{j1}, \ldots, \theta_{jK})$: j번째 문서에 나타나는 주제의 빈도
- $\phi_k = (\phi_{k1}, \ldots, \phi_{kW})$: 주제 k에 쓰이는 단어의 빈도

(*) 확률분해모형에서는 확률이 단어의 위치 (즉 i)에 의존하지 않는다 (Bag of words assumption).

확률분해 모형 (cont.)

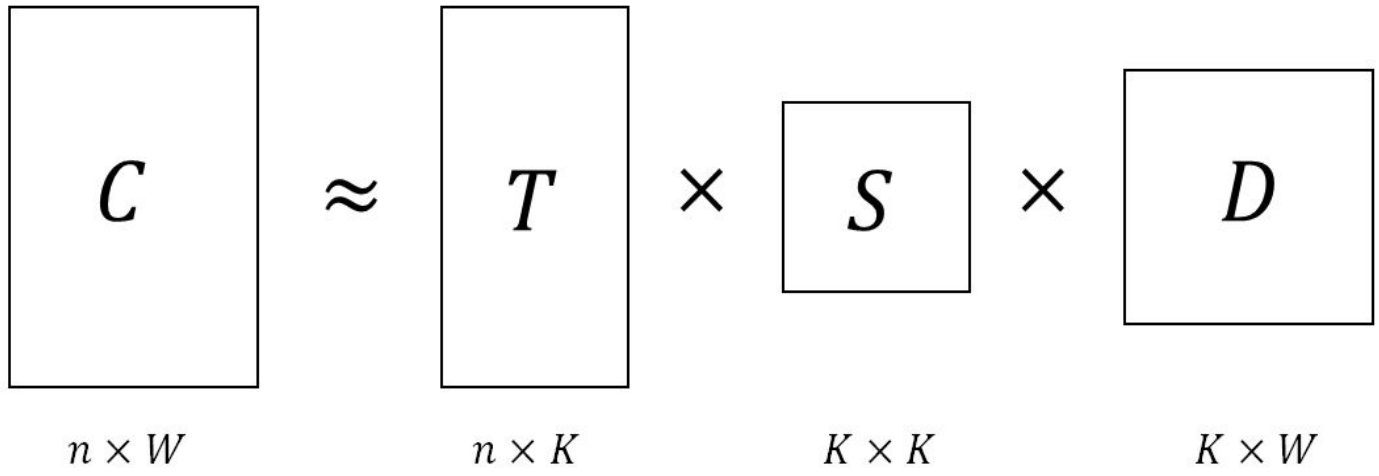
- 인터넷 쇼핑 자료인 경우에는 주제는 사용의도로 해석할 수 있음.
- 문서 뭉치의 주제 또는 인터넷 쇼핑의 사용의도 등을 특성(feature)라고 함.
- 특성은 관측되지 않는 잠재적 (latent) 변수이고, 따라서 잠재적 특성모형 (Latent Feature model)이라고도 불려짐.
- 군집분석에서 각 군집의 특징도 특성으로 해석할 수 있음. 즉, 군집분석은 한 개체가 하나의 특성을 갖는 반면에 확률분해모형은 한 개체 (문서 또는 고객)가 여러 개의 특성을 가질 수 있음.
- K-평균 군집법과 같이 특성의 수 K는 정해주어야 함.

확률분해 모형 (cont.)

- 인터넷 쇼핑의 간단한 예
  - 품목집합: {아기용품, 공연, 옷, 생필품, 전자제품}
  - $\phi_1 = (0.8, 0.05, 0.05, 0.05, 0.05)$: 주제는 '아기'
  - $\phi_2 = (0.0, 0.40, 0.50, 0.10, 0.00)$: 주제는 '젊은 여자'
  - $\theta_j = (0.2, 0.7, \ldots)$: j번째 고객은 주로 젊은 여자의 취향을 가지고 있지만 가끔은 아기관련 제품도 구매한다.
  - 젊은 엄마일 가능성이 있다로 해석할 수 있다.

확률분해와 행렬분해

- 확률분해는 행렬분해 (matrix factorization)의 특수한 경우
- 특성치 분해 (Singular Value Decomposition, SVD)



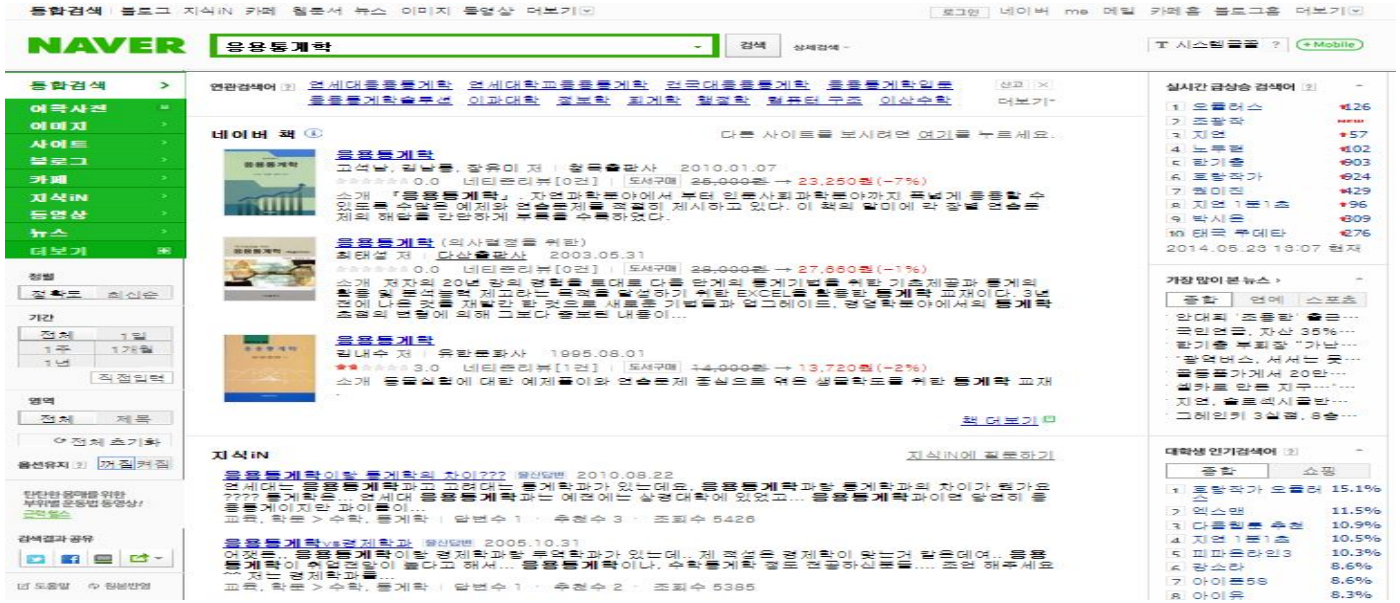
- $n \times W$차원 행렬을 $n \times K, K \times K$ 그리고 $K \times W$의 곱으로 근사
- 즉, 큰 차원의 행렬을 작은 차원의 행렬의 곱으로 분해

확률분해와 행렬분해 (cont.)

- 확률분해 모형에서는
  - C 행렬의 (j, w) 원소는 $p_{jw} = p(x_{ji} = w)$
  - T 행렬의 (j, k) 원소는 θ_{jk}
  - S 행렬은 identity 행렬
  - D 행렬의 (k, w) 원소는 ϕ_{kw}.

행렬분해와 의미분석

- 문서 검색의 예



행렬분해와 의미분석 (cont.)

- 단순한 단어-빈도 비교를 통한 검색의 문제점
  - Query: “Applied multivariate analysis”
  - 단어-빈도 비교 결과

	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

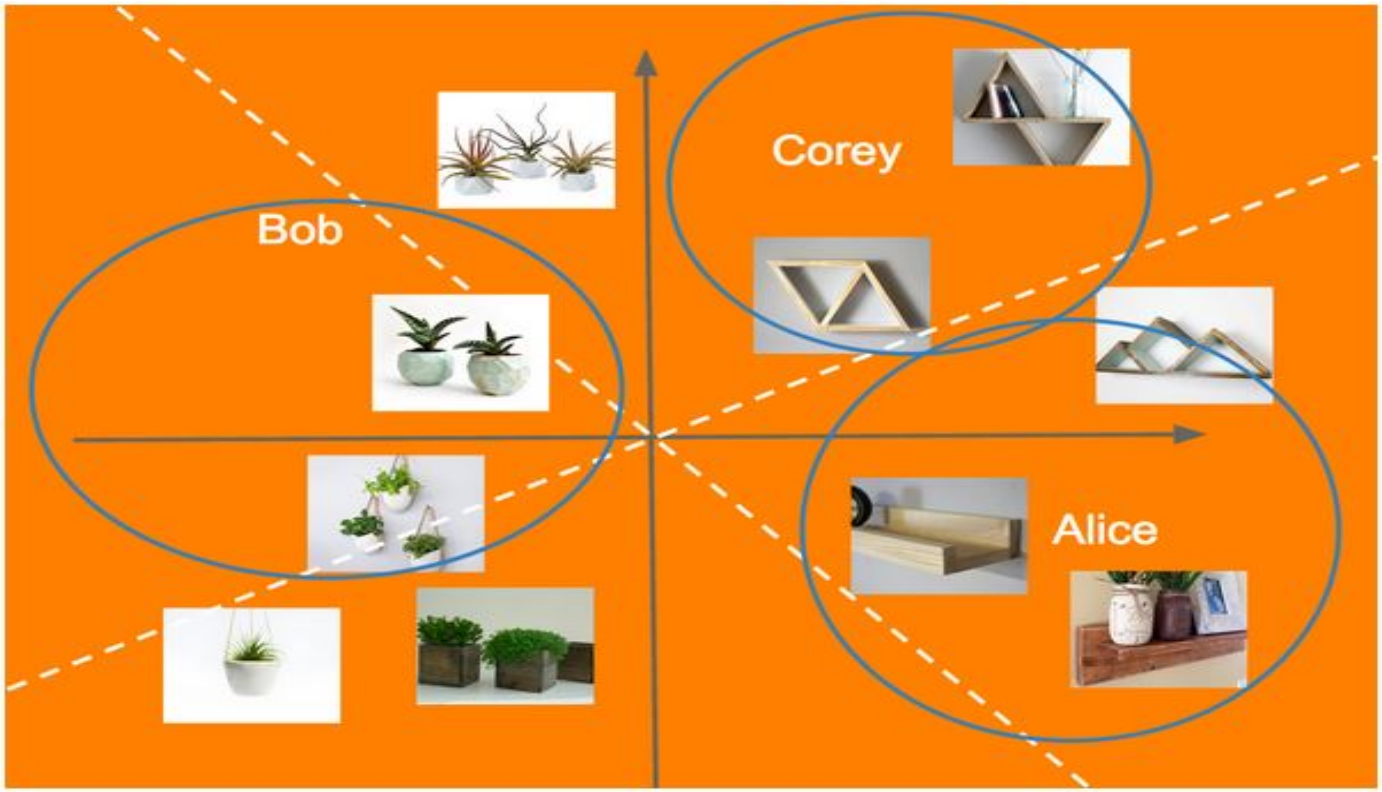
행렬분해와 의미분석(cont.)

- 문제 해결 방법: 잠재적 의미 (Latent Semantics) 이용
  - 비슷한 단어들의 의미를 파악
  - 단어들의 상관계수(동시 출현 빈도)를 통하여 비슷한 단어들을 파악
  - "Applied"와 "Regression" 또는 "Histogram" 의 상관계수가 "Applied"와 "Asymptotic"의 상관계수보다 매우 큼.

행렬분해와 의미분석(cont.)

- 잠재적 의미의 예
  - psychology, history, English, philosophy => Human and social science:
  - mathematics, statistics, chemistry, physics, biology => Natural science: .
- 행렬분해에서 K가 잠재적 의미의 개수이고 T는 각 문서에서의 잠재적 의미 (예: 문서뭉치 에서 주제, 인터넷쇼핑에서의 사용의도)의 출현 양
- T를 이용하여 문서들 사이의 유사성을 파악 (Latent Semantic Analysis)
- D는 각 단어 (또는 인터넷 쇼핑에서의 상품)가 가지고 있는 잠재적 의미의 양

행렬분해와 의미분석(cont.)



목차

추론의 목적

- 관측치로부터 잠재적 특성 추정
- 관측치: $x_{ji}, j = 1, \ldots, n, i = 1, \ldots, n_i$
- 잠재적 특성과 관련된 모수
  - $\theta_{jk}, j = 1, \ldots, n, k = 1, \ldots, K$: 대상 (문서, 고객)별 의미 분포
  - $\phi_{kw}, k = 1, \ldots, K, w = 1, \ldots, W$: 의미 (주제,의도)별 품목 (단어, 상품) 출현 분포
- 관측치에 비하여 (하나의 대상에서 대부분의 품목의 출현 빈도가 0이다) 추정하고자 하는 모수의 수가 매우 많다.
- 베이지안 방법을 사용한다.

베이지안 방법 소개

- 관측치 x, 모수 θ
- 모형

$$x|\theta \sim p(x|\theta)$$

- 사전분포

$$\theta \sim p(\theta)$$

- 베이즈정리로부터 구한 사후분포

$$\theta|x \propto p(x|\theta)p(\theta).$$

- 베이지안 방법은 (전적으로) 사후분포를 이용하여 모수에 대한 추론을 한다.

MCMC 알고리즘 소개

- 모수의 치원이 큰 경우 사후분포를 계산하는 것이 매우 어렵다.
- 사후분포를 계산하지 않고 사후분포로부터 모수를 생성시킨다 (Monte-Carlo방법)
- 모수의 차원이 큰 경우 사후분포로부터 모수를 생성하는 것도 어렵다.
- 이 문제를 해결하기 위하여, 사후분포로 **수렴하도록** 적당한 방법으로 모수를 생성한다 (Markov-Chain Monte-Carlo 알고리즘, MCMC)

Gibbs 샘플링 알고리즘 소개

- MCMC 알고리즘의 한 종류
- $\theta = (\theta_1, \ldots, \theta_p)$ 이고

$$\theta_{(-i)} = (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_n).$$

- 초기회: $m = 0, \theta^{(0)}$
- 수렴할 때 까지 반복
  - $m = m + 1$
  - For $i = 1$ to p
    - Generate $\theta_i \sim p(\theta_i | x, \theta_{(-i)}^{(m-1)})$
    - Update $\theta_i^{(m-1)}$ by θ_i
  - $\theta^{(m)} = \theta^{(m-1)}$.

사전분포

- 사후분포의 계산이 용이한 Dirichlet분포를 사용
- α와 β는 주어진 양의 상수
- θ
  - $\theta_j = (\theta_{j1}, \ldots, \theta_{jK})'$
  - $\theta_1, \ldots, \theta_n$은 독립이고
  - $\theta_j \sim \boldsymbol{Dir}(\alpha, \ldots, \alpha)$
- ϕ
  - $\phi_k = (\theta_{k1}, \ldots, \theta_{kW})'$
  - $\phi_1, \ldots, \phi_K$는 독립이고
  - $\phi_k \sim \boldsymbol{Dir}(\beta, \ldots, \beta)$

Dirichlet 분포 소개

- $\theta \in R^p, \theta_j \geq 0$ 그리고 $\sum_{j=1}^{p} \theta_j = 1$을 만족하는 확률벡터를 위한 분포
- $Dir(\alpha_1, \ldots, \alpha_p)$의 확률밀도함수

$$\frac{\Gamma(\alpha_1 + \cdots + \alpha_p)}{\prod_{i=1}^{p} \Gamma(\alpha_i)} \prod_{i=1}^{p} \theta_i^{\alpha_i}.$$

- p가 2이면 beta분포라고 불린다.
- 모형이 multinomial분포이고 사전분포가 Dirichlet분포이면 사후분포도 Dirichlet분포가 된다 (공액류 사전분포, conjugate prior).
- Dirichlet분포를 따르는 확률벡터를 쉽게 생성할 수 있다 (MCMC알고리즘이 쉬워진다).

α 와 β의 선택

- α가 작이지면, 개별 대상(문서, 고객)이 가지는 의미(주제, 의도)의 수가 작아진다. $\alpha = 0$이면 군집분석과 비슷해진다.
- β가 작아지면 의미별로 중요 품목 (단어, 상품)의 수가 줄어든다.
- 실제 문제에서는 α와 β값을 바꾸어가면서 결과를 보고 가장 해석이 용의하고 좋은 결론을 주는 값을 선택한다 (K-평균 군집분석에서 K를 정하는 방법과 유사).

사후분포

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

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

Naive Gibbs 샘플링 알고리즘

- $\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{z}$에 Gibbs 샘플링 알고리즘을 적용한다.

$$\begin{aligned} \boldsymbol{\theta}_j|\mathbf{z}_j &\sim \mathcal{D}(\alpha + N_{j1\cdot}, \ldots, \alpha + N_{jK\cdot}) \\ \boldsymbol{\phi}_k|\mathbf{z} &\sim \mathcal{D}(\beta + N_{\cdot k1}, \ldots, \beta + N_{\cdot kW}) \\ p(z_{ji} = k|\boldsymbol{\theta}_j, \boldsymbol{\phi}) &\propto \theta_{jk}\phi_{kx_{ji}} \end{aligned}$$

- 속도가 매우 느리다.

Collapsed Gibbs 샘플러 알고리즘

- $\boldsymbol{\theta}$ 와 $\boldsymbol{\phi}$를 적분한 후 $\mathbf{z}$에 Gibbs 샘플링 알고리즘을 적용한다.

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

- $\boldsymbol{\theta}$ 와 $\boldsymbol{\phi}$ 는 $\mathbf{z}$가 생성된 후 쉽게 생성할 수 있다.
- 수렴속도가 빠르다.

목차

인터넷 쇼핑몰 자료

- 인터넷 쇼핑몰에서 판매되는 상품들을 70가지 품목으로 중분류 (예 : 의류, 화장품, 전자제품, 건강제품 등)

- 인터넷 쇼핑몰 회원 중 임의 추출된 10,000명을 대상으로 2013년 한 해 동안 70가지 품목에 대한 구매 횟수를 저장

```
> shoping=read.csv("C:\\Users\\JKH\\Desktop\\lda실습\\shoping.csv")
> head(shoping,10)
   고객 품목 횟수
1     1    6   12
2     1   10    2
3     1   11    6
4     1   13    4
5     1   29    2
6     1   37    5
7     2    2    1
8     2    6    6
9     2    9    1
10    2   11    5
```

LDA 패키지

- R의 패키지 "lda"가 Collapsed Gibbs 샘플러를 구현
- "lda"를 사용하기 위해서는 자료를 아래와 같은 형식으로 변환해야 함

```
6 5:12 9:2 10:6 12:4 28:2 36:5
11 1:1 5:6 8:1 10:5 13:5 14:2 16:2 17:9 26:1 36:4 45:1
10 5:10 10:8 12:2 17:1 20:1 31:2 36:8 39:1 45:1 60:1
11 5:5 10:10 12:1 13:18 14:1 17:7 26:1 31:1 34:3 36:5 39:2
```

- 첫 번째 사람은 6가지 품목을 구매
  - 6번째 품목(A6) : 12회, 10번째 품목(A10) : 2회
    11번째 품목(B1) : 6회, 13번째 품목(B3) : 4회
    29번째 품목(C9) : 2회, 37번째 품목(D7) : 5회

자료 전처리

- preproc(자료, 저장할 위치, 저장할 이름)
  : 자료를 "lda"에서 사용할 수 있도록 변환하여 dat파일로 저장하는 함수

```
> preproc(shoping,"C:\\Users\\JKH\\Desktop\\lda실습","shoping2")
Preprocessed data are saved at 'C:\Users\JKH\Desktop\lda실습\shoping2.dat'.
```

- dat 파일을 "lda"의 read.documents로 불러옴

```
> library(lda)
> shoping2=read.documents("C:\\Users\\JKH\\Desktop\\lda실습\\shoping2.dat")
Read 10000 items
> item_name=read.vocab("C:\\Users\\JKH\\Desktop\\lda실습\\품목.txt")
> head(shoping2,2)
[[1]]
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    5    9   10   12   28   36
[2,]   12    2    6    4    2    5

[[2]]
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
[1,]    1    5    8   10   13   14   16   17   26    36    45
[2,]    1    6    1    5    5    2    2    9    1     4     1
```

Collapsed Gibbs 샘플러

- Collapsed Gibbs 샘플러 실행

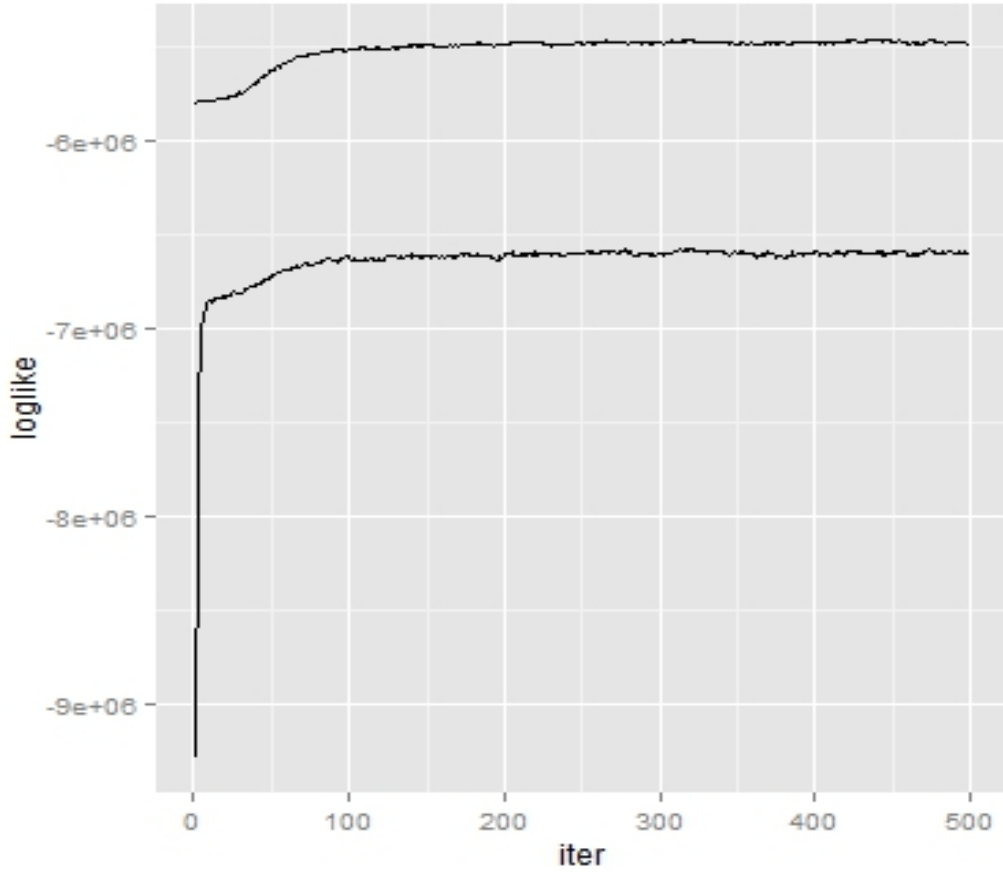
```
> n=10000; K=11; W=70; alpha=1.0; beta=1.0
> lda=lda.collapsed.gibbs.sampler(shoping2, K, item_name, 500, alpha, beta, compute.log.likelihood=T)
```

  - lda$documents_sums : $N^T_{jk\cdot}$
  - lda$topics : $N_{\cdot kw}$
  - lda$log.likelihoods : 두 종류의 log likelihood 값

수렴 여부 판단

- log likelihood를 통한 수렴 여부 판단

```
> plot_loglike(lda)
```



θ와 ϕ 추정

- θ와 ϕ를 각각 $N_{jk\cdot}$와 $N_{\cdot kw}$를 이용하여 추정

```
# theta를 추정
theta=matrix(0,nrow=n,ncol=K)
for(i in 1:n){
  theta[i,]=t(lda$document_sums[,i])/sum(t(lda$document_sums[,i]))
}

# phi를 추정
phi=matrix(0,nrow=K,ncol=W)
for(i in 1:K){
  phi[i,]=lda$topics[i,]/sum(lda$topics[i,])
}
```

lift 계산

- 특성을 대표하는 주요 품목으로 특성을 나타내고 싶음
- 단순히 확률이 큰 것을 주요 품목으로 보는 것은 문제가 있음
- 확률보다는 확률의 상대적인 값인 lift를 보는 것이 타당함

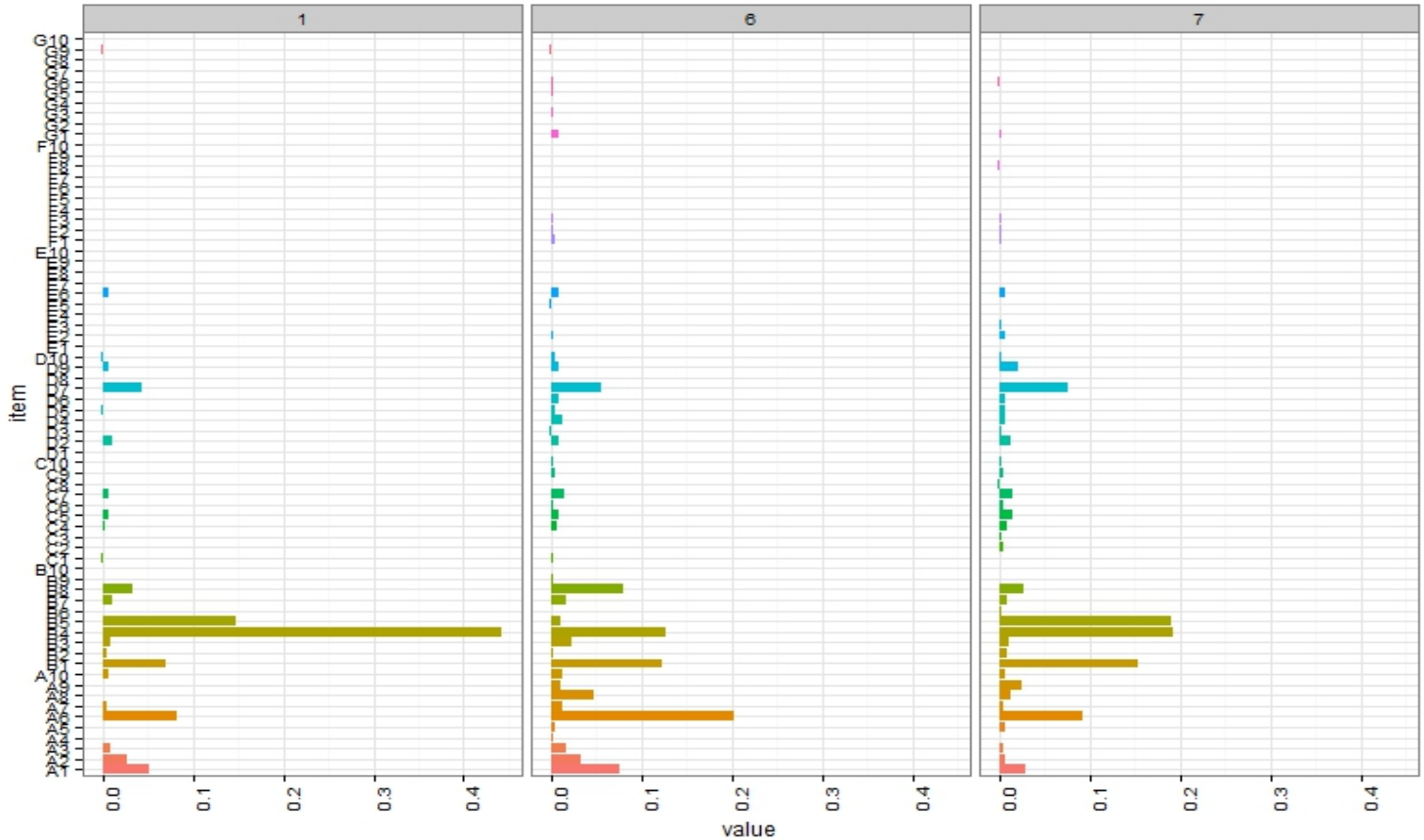
```
# 품목의 비율
p=colSums(lda$topics)/sum(lda$topics)

# lift 계산
lift=matrix(0,nrow=K,ncol=W)
colnames(lift)=item_name
for(i in 1:K){
  lift[i,]=phi[i,]/p
}
```

특성 이름 붙이기

- 특성에서 lift가 큰 두 개의 품목을 주요 품목으로 정의
- 주요 품목으로 특성의 이름을 붙임

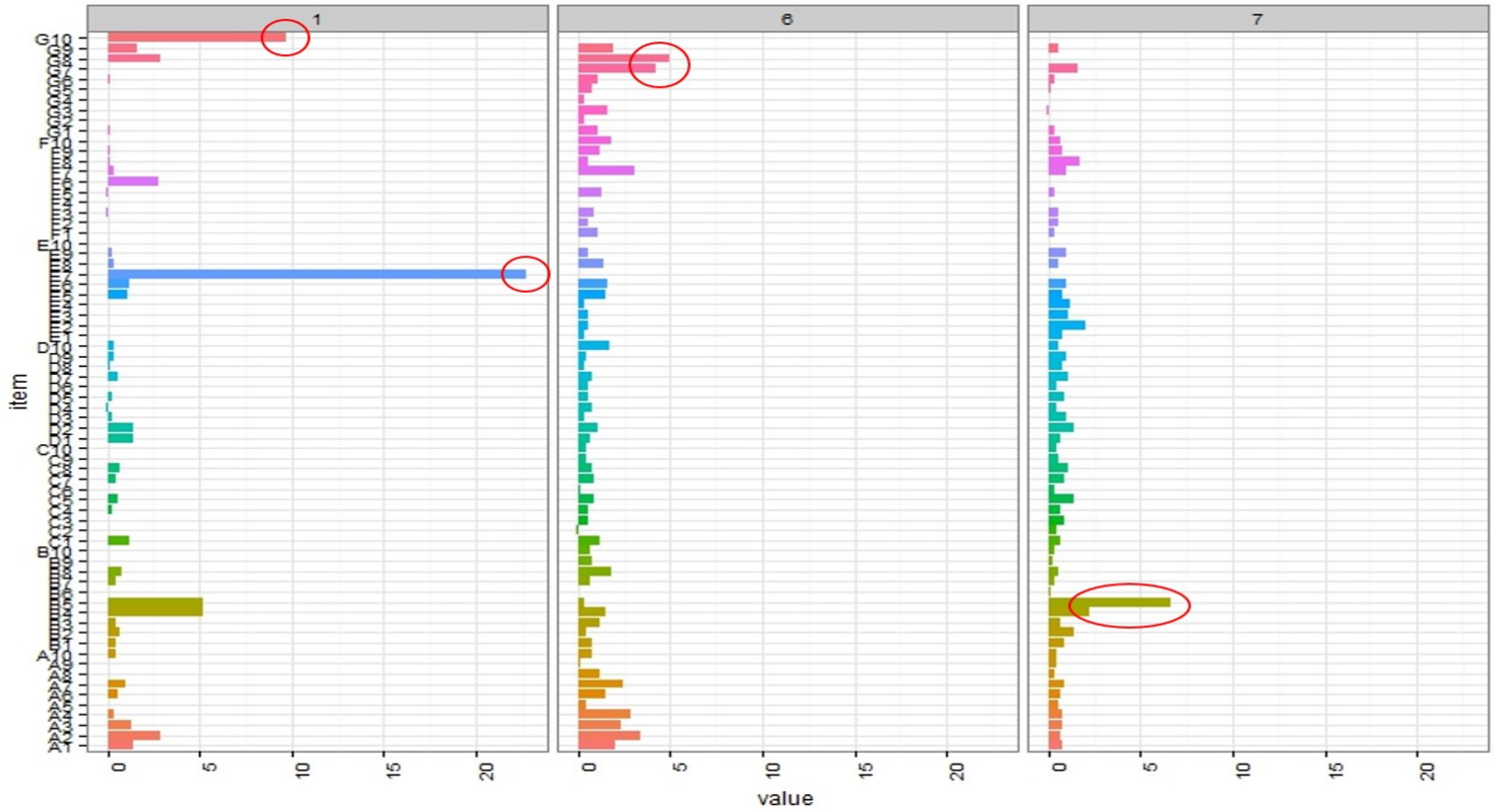
```
> # 특성마다 lift 상위 2개 품목으로 이름을 붙임
> char_name=c()
> for(i in 1:K){
+   sorted=sort(lift[i,],decreasing=T)[1:2]
+   char_name=c(char_name,paste(names(sorted),collapse=" "))
+ }
> char_name
 [1] "E7 G10" "G4 G3"  "F6 B6"  "B6 A9"  "D1 E2"  "G8 G7"
 [7] "B5 B4"  "A10 C2" "E1 G2"  "E4 A4"  "F4 B9"
```

LDA 결과 - 1. 특성



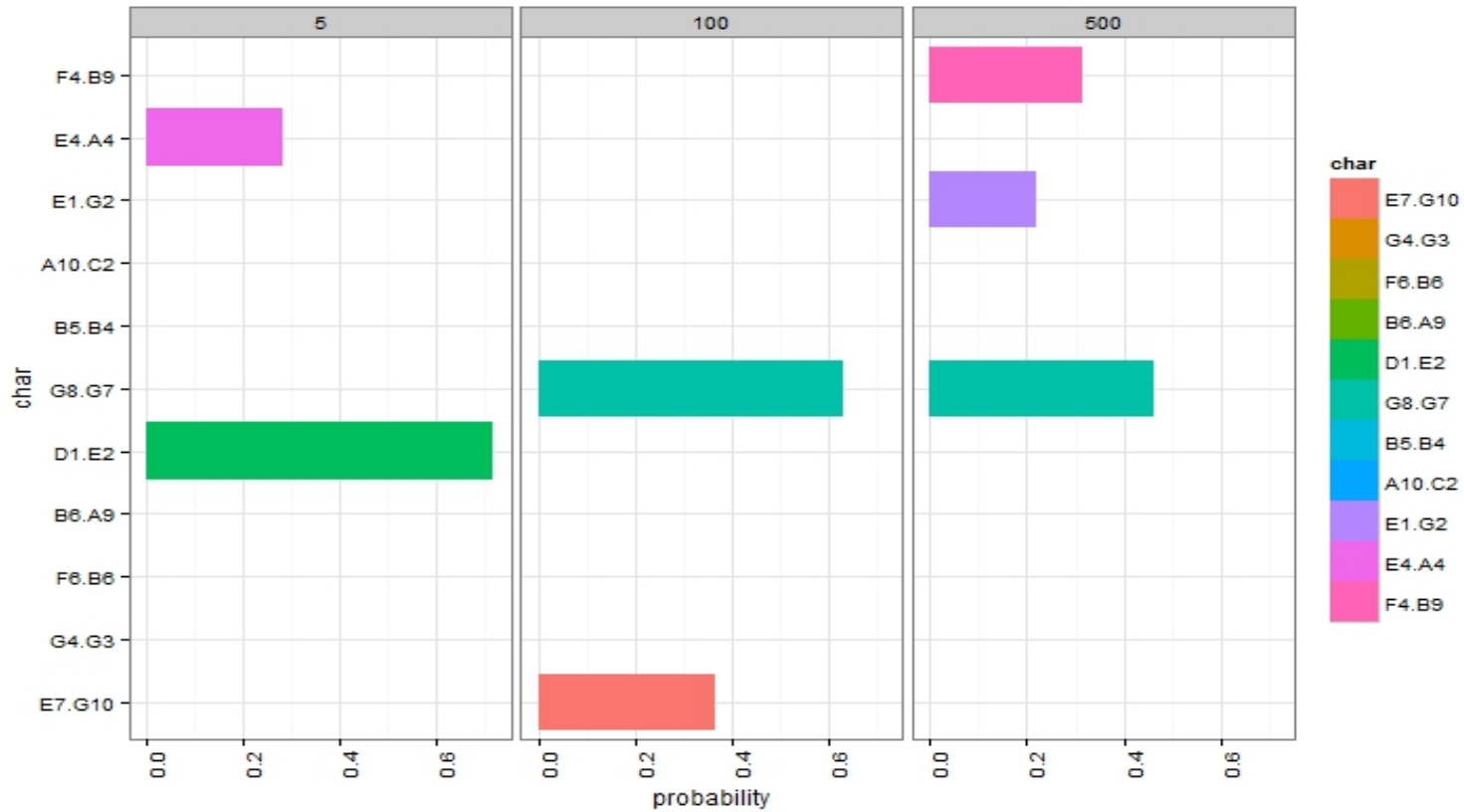
- 1, 6, 7번째 특성의 분포(ϕ_1, ϕ_6, ϕ_7)
- 특성마다 확률이 큰 품목들은 크게 다르지 않음

LDA 결과 - 1. 특성



- 1, 6, 7번째 특성의 lift
- 1번째 특성은 E7과 G10 품목이, 6번째 특성은 G8과 G7 품목이, 7번째 특성은 B5와 B4 품목이 lift가 큼
- 특성마다 lift가 큰 품목들은 매우 다름

LDA 결과 - 2. 고객



- 5번째 고객은 D1.E2와 E4.A4 특성을,
  100번째 고객은 G8.G7과 E7.G10 특성을,
  500번째 고객은 G8.G7, F4.B9, E1.G2 특성을 가짐

2부: 순위모형

목차

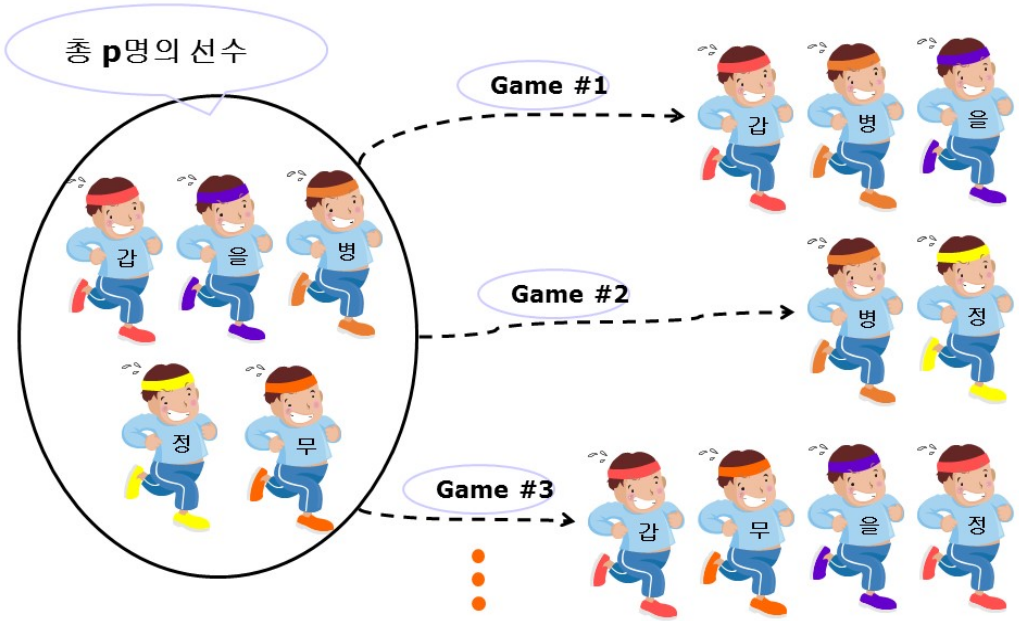
순위자료 구조

- n : 시도 (trial) 횟수
- p : 항목 (item)의 수
- $A_i \subset \{1, \ldots, p\}$: i-번째 시도에 참가한 항목의 수
- $R_i = (R_{i1}, \ldots, R_{i|A_i|})$: i 번째 시도에 참가한 항목의 등수

예제 1: 스포츠/게임

- 스포츠 경기나 다자 대결 게임의 결과
  - n : 게임의 수
  - p : 전체 선수의 수
  - $A_i \subset \{1, \ldots, p\}$: i-번째 게임에 참가한 선수의 수
  - $R_i = (R_{i1}, \ldots, R_{i|A_i|})$: i번째 경기에 참가한 선수의 등수
- 축구나 바둑 등에서는 보통 $|A_i| = 2$.
- 육상경기나 인터넷 게임 등에서는 $|A_i| \geq 2$

예제 1: 스포츠/게임 (cont.)



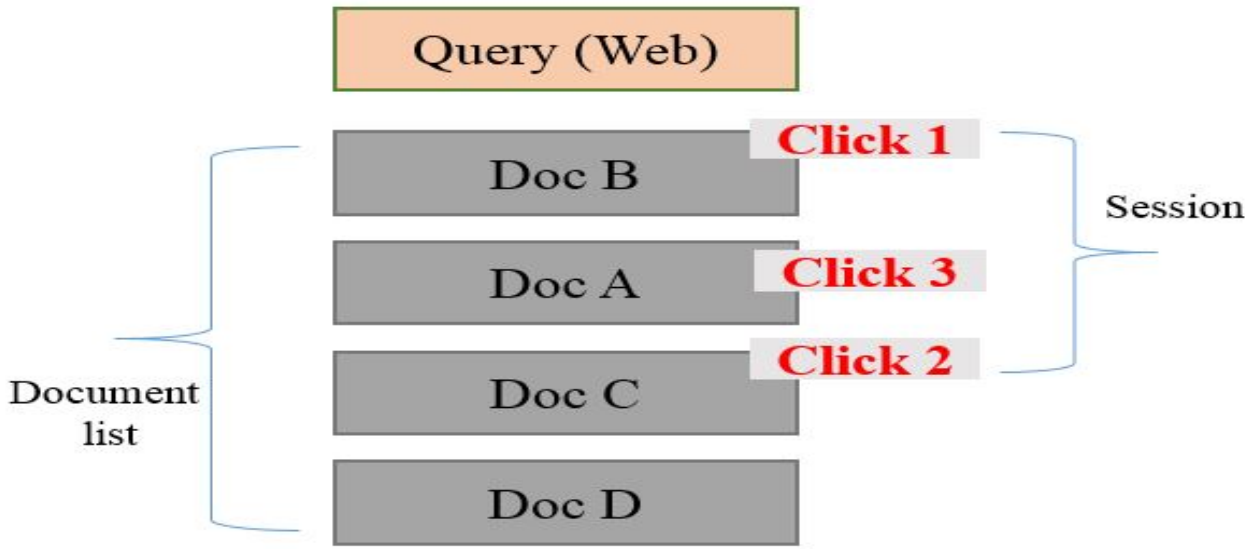
예제 2: 선호도 자료

- 선호도 서베이 자료
  - n : 서베이 대상 사람의 수
  - p : 품목의 수
  - $A_i \subset \{1, \ldots, p\}$: i-번째 서베이에서 비교된 품목의 수
  - $R_i = (R_{i1}, \ldots, R_{i|A_i|})$: i번째 비교에서의 선호도
- 예: 100명의 사람들에게 5종류의 차종의 선호도 조사
  - $n = 100, p = 5, |A_i| = 5$.

예제 3: 인터넷 검색

- 각 문서에 대한 사용자의 선호도
  - n : 검색의 수
  - p : 문서의 수
  - $A_i \subset \{1, \ldots, p\}$: i-번째 검색에서 클릭을 받은 문서
  - $R_i = (R_{i1}, \ldots, R_{i|A_i|})$: i번째 검색에서 클릭의 순서

예제 3: 인터넷 검색 (cont.)



- $A_i = \{A, B, C\}$이고 $R_i = (3, 1, 2)$.

목차

Thustone 모형

- 각 항목은 능력치/선호도 $\lambda_j, j = 1, \ldots, p$를 갖는다.
- i번째 시도에서 함목 j의 능력치/선호도는 $Z_{ij} + \lambda_j$가 되면, Z_{ij}는 독립이고 분포가 같다.
- i번째 시도에서 $\{Z_{ij} + \lambda_j, j \in A_i\}$의 내림차순의 순위를 R_i로 관측한다.
- 예
  - $A_i = \{1, 3, 5\}$이면

$$p\{R_i = (1, 3, 2)\} = p\{Z_{i1} + \lambda_1 > Z_{i5} + \lambda_5 > Z_{i3} + \lambda_3\}.$$

- 능력치/선호도 λ_i를 추정하는 것이 목표 (예: 골프선수 랭킹)

Luce 모형

- 각 항목은 능력치/선호도 $\lambda_j > 0, j = 1, \ldots, p$를 갖는다.
- 예
  - $A_i = \{1, 3, 5\}$이면

$$p\{R_i = (1, 3, 2)\} = \frac{\lambda_1}{\lambda_1 + \lambda_3 + \lambda_5} \frac{\lambda_5}{\lambda_3 + \lambda_5} \frac{\lambda_3}{\lambda_3}.$$

- 일반적 Luce 모형
  - 두개의 항목 j_1과 j_2에 대해서 $R_{ij_1} < R_{ij_2}$이면 $j_1 \to j_2$라고 쓰자. 이 경우

$$p(j_1 \to j_2 \to \cdots \to j_m) = \prod_{k=1}^{m} \frac{\lambda_{j_k}}{\sum_{l=k}^{m} \lambda_{j_l}}.$$

Bradley-Terry 모형

- 쌍비교 자료, 즉 $|A_i| = 2$
- 각 항목은 능력치/선호도 $\lambda_j > 0, j = 1, \ldots, p$를 갖는다.
-

$$p(j_1 \to j_2) = \frac{\lambda_{j_1}}{\lambda_{j_1} + \lambda_{j_2}}.$$

모형 비교

- Thurstone 모형에서 Z_{ij}의 분포가 Gumbel분포인 경우에 Thurstone모형과 Luce 모형은 같다.
- Gumbel분포의 확률밀도함수는

$$f(x) = \exp\left\{x + e^{-x}\right\}.$$

- Bradley-Terry모형은 Luce모형의 특별한 형태 (즉, $|A_i| = 2$).

Mallows-Bradley-Terry 모형

- Bradley-Terry 모형을 $|A_i| \geq 2$인 경우로 확장
- 예
  - $A_i = \{1, 3, 5\}$이면

$$p\{R_i = (1, 3, 2)\} = Cp(1 \to 3)p(1 \to 5)p(5 \to 3),$$

    이고 $p(j_1 \to j_2)$는 Bradley-Terry모형을 따른다.
  - C는 normalizing 상수이다.
- 일반적으로

$$p(j_1 \to j_2 \to \cdots \to j_m) = C \prod_{k<l} p(j_k \to j_l).$$

- Normalizing 상수의 계산 때문에 $|A_i| = p$인 경우 (예: 선호도 조사)에만 사용 가능.

목차

Thurstone 모형

- Z_{ij}분포 선택 (예: 정규분포, Gumble분포)
- 우도함수 계산

$$L(\lambda) = \prod_{i=1}^{n} p(R_i|\lambda).$$

- 우도함수를 최대로 하는 λ값으로 추정
- 일반적으로 우도함수의 계산이 어려워서 잘 쓰이지 않음.

Luce 모형

- 최대우도추정량 사용
- 우도함수는 생존분석에서 사용 되는 Cox의 비례위험모형에서의 부분우도함수
- 이론적 복잡성 및 계산량이 커서 빅데이터 분석에는 잘 사용되지 않음

Bradley-Terry 모형

- 쌍비교자료이며 확률은 다음과 같이 주어진다:

$$p(j_1 \to j_2) = \frac{\lambda_{j_1}}{\lambda_{j_1} + \lambda_{j_2}}.$$

- $u_j = \log \lambda_j$로 놓으면

$$\begin{aligned} p(j_1 \to j_2) &= \frac{\exp(u_{j_1})}{\exp(u_{j_1}) + \exp(u_{j_2})} \\ &= \frac{\exp(u_{j_1} - u_{j_2})}{1 + \exp(u_{j_1} - u_{j_2})}. \end{aligned}$$

- 즉, 로지스틱 회귀모형을 이용하여 능력치/선호도를 추정할 수 있다.

Bradley-Terry 모형 (cont.)

- i번째 시도에서 $j_1 \to j_2$ 이면 $y_i = 1$이라 놓고 $j_2 \to j_1$이면 $y_i = 0$이라 놓자.
- 우도함수는 다음과 같이 주어진다

$$L(\lambda) = \prod_{i=1}^{n} \frac{y_i \exp(u_{j_1} - u_{j_2})}{1 + \exp(u_{j_1} - u_{j_2})}.$$

Bradley-Terry 모형 (cont.)

- $\mathbf{x}_i \in R^p$이며 $x_{i,j_1} = 1, x_{i,j_2} = -1$ 그리고 $j \notin \{j_1, j_2\}$에 대해서 $x_{ij} = 0$이라 놓자.
- $u = (u_1, \ldots, u_p)'$이라 놓자.
- 그러면, 우도함수는

$$L(\lambda) = \prod_{i=1}^{n} \frac{y_i \exp(\mathbf{x}_i' u)}{1 + \exp(\mathbf{x}_i' u)}.$$

- 즉, $(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)$을 자료로 갖는 로지스틱 회귀모형의 우도함수가 되며, 따라서 u의 최대우도추정량은 쉽게 구할 수 있다.

슈도 최대우도 추정량

- Bradley-Terry 모형의 우도함수는 $|A_i| = 2$인 경우에만 사용이 가능하다.
- $|A_i| \geq 2$경우, Bradley-Terry우도함수를 다음과 같이 확장한다. (일반화 Bradley-Terry 모형)
- i번째 시도에서 $j_1^{(i)} \to j_2^{(i)} \to \cdots \to j_{m_i}^{(i)}$ 관측하였다고 가정하자.
- 여기서 $A_i = \{j_1^{(i)}, \ldots, j_{m_i}^{(i)}\}$ 이고 $m_i = |A_i|$ 이다.

슈도 최대우도 추정량 (cont.)

- 이러한 관측치를 $\binom{m_i}{2}$개의 쌍비교 자료를 독립적으로(?) 관측하였다고 가정한다.
- 즉, 새로운 관측치는 모든 $k < l$에 대해서 $j_k \to j_l$이 된다.
- 새로운 관측치로 부터 구한 우도함수는

$$L(\lambda) = \prod_{i=1}^{n} \prod_{1 \le k < l \le m_i} \frac{\exp\left(u_{j_k^{(i)}} - u_{j_l^{(i)}}\right)}{1 + \exp\left(u_{j_k^{(i)}} - u_{j_l^{(i)}}\right)}. \tag{1}$$

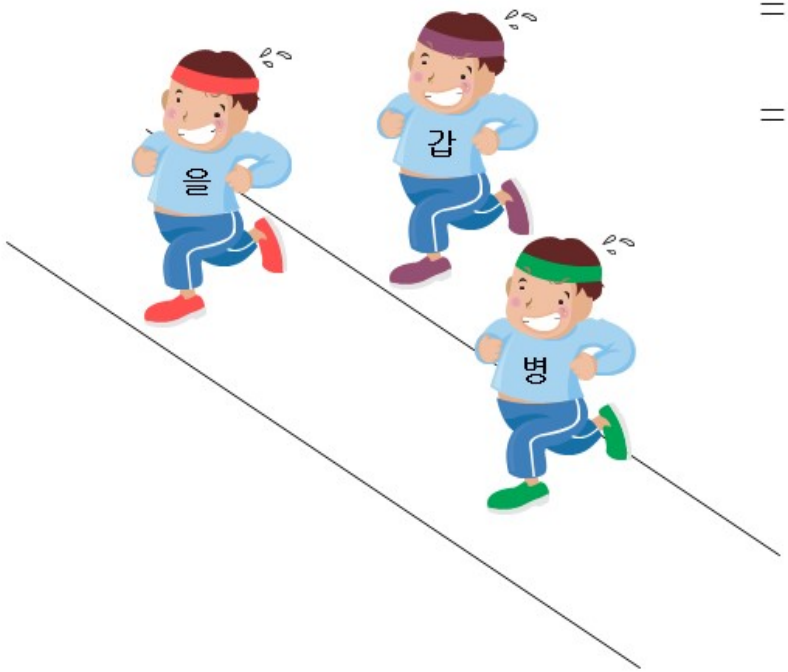
슈도 최대우도 추정량 (cont)

- $j_k^{(i)} \to j_l^{(i)}$ 이면 $y_{ikl} = 1$로 놓고 $j_l^{(i)} \to j_k^{(i)}$ 이면 $y_{ikl} = 0$로 놓자.
- $\mathbf{x}^{ikl} \in R^p$ 이고 $x_k^{ikl} = 1, x_l^{ikl} = -1$ 그리고 $h \notin \{k, l\}$에 대해서 $x_h^{ikl} = 0$ 로 놓자.
- 그러면 식 (1)는 다음과 같다:

$$L(\lambda) = \prod_{i=1}^{n} \prod_{(k,l)\in A_i, k\neq l} \frac{y_{ikl} \exp(\mathbf{x}^{ikl'} u)}{1 + \exp(\mathbf{x}^{ikl'} u)}. \tag{2}$$

- 식 (2)를 슈도(Pseudo) 우도함수라고 부른다.
- 슈도우도함수는 로지스틱 회귀모형의 우도함수가 되어서 쉽게 u의 최대우도추정량을 구할 수 있다.

슈도 최대우도 추정량 (cont)



$$
\begin{aligned}
&P(\text{을} > \text{병} > \text{갑}) \\
&= P(\text{을} > \text{병}) P(\text{을} > \text{갑}) P(\text{병} > \text{갑}) \\
&= \frac{\pi_{\text{을}}}{\pi_{\text{을}} + \pi_{\text{병}}} \frac{\pi_{\text{을}}}{\pi_{\text{을}} + \pi_{\text{갑}}} \frac{\pi_{\text{병}}}{\pi_{\text{병}} + \pi_{\text{갑}}}
\end{aligned}
$$

선수가 **N** 명이면
${}_{N}\mathbf{C}_{2}$ 번의 대결 존재

목차

공변량을 포함한 순위자료 구조

- n : 시도 (trial) 횟수
- p : 항목 (item)의 수
- $A_i \subset \{1, \ldots, p\}$: i-번째 시도에 참가한 항목의 수
- $R_i = (R_{i1}, \ldots, R_{i|A_i|})$: i 번째 시도에 참가한 항목의 등수
- $\mathbf{x}_j^{(i)}, j \in A_i$: i번째 시도에서 j번째 항목과 관련된 공변량 벡터
- 공변량의 예
  - 축구경기: 홈경기 여부, 과거 전적 등등
  - 자동차 선호도 조사: 엔진크기, 가격 등등
  - 인터넷 게임: 옵션선택사항 등등
  - 인터넷 검색: 문서의 노출 위치, 문서의 주제 등등

순위자료 회귀모형

- i번째 시도에서 j번째 항목의 능력치/선호도를 다음과 같이 모형화:

$$\log(\lambda_j^{(i)}) = u_j + \mathbf{x}_j^{(i)\prime}\beta.$$

- 예를 들어, Bradley-Terry 모형에서는

$$\begin{aligned} p(j_1^{(i)} \to j_2^{(i)}) &= \frac{\exp(u_{j_1}^{(i)} - u_{j_1}^{(i)})}{1 + \exp(u_{j_1}^{(i)} - u_{j_1}^{(i)})} \\ &= \frac{\exp\left(u_{j_1} - u_{j_2} + (\mathbf{x}_{j_1}^{(i)} - \mathbf{x}_{j_2}^{(i)})'\beta\right)}{1 + \exp\left(u_{j_1} - u_{j_2} + (\mathbf{x}_{j_1}^{(i)} - \mathbf{x}_{j_2}^{(i)})'\beta\right)}. \end{aligned}$$

우도함수

- i번째 시도에서 $j_1 \to j_2$ 이면 $y_i = 1$이라 놓고 $j_2 \to j_1$이면 $y_i = 0$이라 놓자.
- $\mathbf{z}_i \in R^p$이며 $z_{i,j_1} = 1, z_{i,j_2} = -1$ 그리고 $j \notin \{j_1, j_2\}$에 대해서 $z_{ij} = 0$이라 놓자.
- $\mathbf{w}_i = \mathbf{x}_{j_1}^{(i)} - \mathbf{x}_{j_2}^{(i)}$.
- 우도함수는 다음과 같이 주어진다:

$$L(\lambda) = \prod_{i=1}^{n} \frac{y_i \exp(\mathbf{z}_i' u + \mathbf{w}_i' \beta)}{1 + \exp(\mathbf{z}_i' u + \mathbf{w}_i' \beta)}.$$

우도함수 (cont.)

- 즉, $(y_1, \mathbf{z}_1, \mathbf{w}_1), \ldots, (y_n, \mathbf{z}_n, \mathbf{w}_n)$을 자료로 갖는 로지스틱 회귀모형의 우도함수가 되며, 따라서 u와 β의 최대우도추정량은 쉽게 구할 수 있다.
- $|A_i| \geq 2$의 경우에도 슈도우도함수를 비슷하게 구할 수 있다.

목차

온라인 레이싱 게임 자료

Figure : 온라인 레이싱 게임의 스크린샷

- 하나의 게임에 여러 사용자가 참여를 하여 순위 경쟁을 함
- 전체 사용자 수는 많으나 한 게임 당 참여할 수 있는 사용자 수는 상대적으로 적음

온라인 레이싱 게임 자료

Figure : 게임 내에서의 커스터마이징 과정

- 각각의 사용자는 여러 자동차 중 한 대를 선택하여 게임에 참여함
- 4210명의 전체 사용자 중 각 게임마다 최대 16명의 사용자가 43가지의 자동차 중 하나를 선택하여 게임에 참여

온라인 레이싱 게임 자료

```
> setwd("D:\\ranking")
> data = read.csv("racing_data.csv",header=FALSE)
> head(data)
  V1   V2   V3   V4   V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21 V22 V23 V24 V25
1  4 1387 2298  848 2143  0  0  0  0   0   0   0   0   0   0   0   0  39  39  25   3   0   0   0   0
2  4 3577  148 1047  733  0  0  0  0   0   0   0   0   0   0   0   0  32  25  25  39   0   0   0   0
3  2 1779 1077    0    0  0  0  0  0   0   0   0   0   0   0   0   0  25  39   0   0   0   0   0   0
4  2 2952  764    0    0  0  0  0  0   0   0   0   0   0   0   0   0  25  16   0   0   0   0   0   0
5  3 1695 1745 1246    0  0  0  0  0   0   0   0   0   0   0   0   0  15  24  24   0   0   0   0   0
6  2 4147  265    0    0  0  0  0  0   0   0   0   0   0   0   0   0  15  18   0   0   0   0   0   0
  V26 V27 V28 V29 V30 V31 V32 V33
1   0   0   0   0   0   0   0   0
2   0   0   0   0   0   0   0   0
3   0   0   0   0   0   0   0   0
4   0   0   0   0   0   0   0   0
5   0   0   0   0   0   0   0   0
6   0   0   0   0   0   0   0   0
```

- 각 행은 하나의 게임에 해당
- V1 : 한 게임에 참가한 사용자의 수
- V2 - V17 : 게임에 참가한 사용자의 이름(먼저 들어온 순서로 나열됨.)
- V18 - V33 : 참가한 사용자가 게임에서 사용한 자동차의 이름(순서는 위와 마찬가지)

일반화 Bradley-Terry 모형 구축 방법

- 본 자료에서는 두 가지 모형 구축 방법에 대해 분석을 실행

1. 사용자가 선택하는 자동차들 사이의 능력치 차이만을 고려하여 모형을 구축
   (모형1로 표기)
2. 사용자의 실력 차이와 사용자가 선택하는 자동차들 사이의 능력치 차이를 모두 고려하여 모형을 구축(모형2로 표기)

자료 전처리

- 먼저, 일반화 Bradley-Terry 모형은 기본적으로 두 개체 간의 승패에 대한 정보를 필요로 하므로 이를 추출한다.

```
> data.user = data.car = c()
> for (n in 1:16){
+ data.tmp = data[data[,1]==n,]
+ if (nrow(data.tmp)!=0){
+ for (j in 1:(n-1)){
+ for (k in (j+1):n){
+ tmp.user = data.tmp[,c(j+1,k+1)]
+ tmp.car = data.tmp[,c(j+17,k+17)]
+ colnames(tmp.user) = c("user1","user2")
+ colnames(tmp.car) = c("car1","car2")
+ data.user = rbind(data.user,tmp.user)
+ data.car = rbind(data.car,tmp.car)
+ }}}}
>
> rownames(data.user) = rownames(data.car) = NULL
```

자료 전처리

- 위 과정을 통해 생성된 행렬은 이긴 사용자(자동차)의 이름이 첫번째 열에 오고, 진 사용자(자동차)의 이름이 두번째 열이 됨

```
> head(data.user)
  user1 user2
1  1779  1077
2  2952   764
3  4147   265
4   304  3358
5  1936  2025
6  1117    35
```

```
> head(data.car)
  car1 car2
1   25   39
2   25   16
3   15   18
4   23   23
5   23    3
6   39   16
```

- 하지만, glm 함수를 이용해서 일반화 Bradley-Terry 모형을 적합하려면 자료의 재가공이 필요함
- 이후의 과정은 모형의 형태에 따라 자료 구축 방법이 달라지므로 이를 나눠서 소개함

자료 전처리 : 모형1

- 모형1을 적합할 때는 사용자의 정보가 불필요하고 같은 자동차간의 대결 또한 불필요하므로 이를 제거하여 자료를 구축.

```
> nr = sum(data.car[,1]!=data.car[,2])
> x.car = matrix(0,nr,43); j = 1
> for (i in 1:nrow(data.car)){
+ if (data.car[i,1]!=data.car[i,2]){
+ x.car[j,data.car[i,1]] = 1
+ x.car[j,data.car[i,2]] = -1
+ j = j+1
+ }
+ }
> y = rep(1,nr)
> data.train1 = data.frame(x.car,y)
> data.train1 = data.train1[,-1]
```

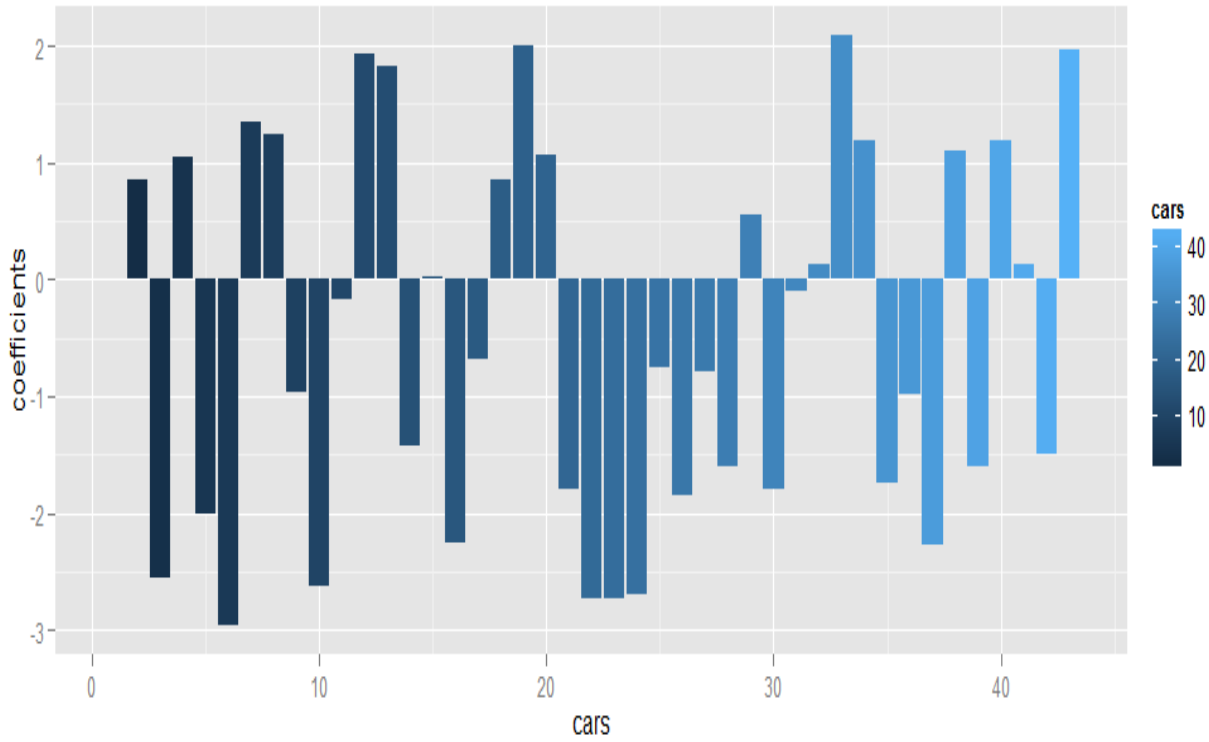
일반화 Bradley-Terry 모형 적합 : 모형1

- 가공된 자료를 바탕으로 모형1을 적합한 결과

```
> fit1 = glm(y~.-1,data=data.train1,family="binomial")
> fit1

Call:  glm(formula = y ~ . - 1, family = "binomial", data = data.train1)

Coefficients:
      X2        X3        X4        X5        X6        X7        X8        X9
 0.85416  -2.54922   1.04792  -2.01317  -2.95416   1.34327   1.24640  -0.97269
     X10       X11       X12       X13       X14       X15       X16       X17
-2.61757  -0.16572   1.92808   1.82614  -1.41906   0.02553  -2.25011  -0.68986
     X18       X19       X20       X21       X22       X23       X24       X25
 0.85163   1.99340   1.07344  -1.78801  -2.72877  -2.72530  -2.69787  -0.75090
     X26       X27       X28       X29       X30       X31       X32       X33
-1.84150  -0.78949  -1.60332   0.55119  -1.79331  -0.09143   0.12905   2.09063
     X34       X35       X36       X37       X38       X39       X40       X41
 1.19139  -1.73443  -0.98222  -2.28102   1.10119  -1.60626   1.18954   0.13549
     X42       X43
-1.48982   1.96424

Degrees of Freedom: 42372 Total (i.e. Null);  42330 Residual
Null Deviance:      58740
Residual Deviance: 51760        AIC: 51840
```

일반화 Bradley-Terry 모형 적합 : 모형1

- 계수 값을 그림으로 표시



결과 해석 : 모형1

- 기준 변수는 X1(자동차1)이며, X1의 계수값을 0으로 고정했을 때 나머지 자동차들의 계수값을 출력함
- 예를 들어 자동차1가 자동차2를 이길 확률은 다음과 같이 계산할 수 있다.

$$Prob = \frac{\exp(0)}{\exp(0) + \exp(0.85416)} = 0.2985609$$

자료 전처리 : 모형2

- 모형2을 적합할 때는 사용자의 승패 정보와 사용자가 선택한 자동차 정보가 모두 필요.

```
> x.user = matrix(0,nrow(data.user),4210)
> x.car = matrix(0,nrow(data.car),43)
> for (i in 1:nrow(data.user)){
+ x.user[i,data.user[i,1]] = 1
+ x.user[i,data.user[i,2]] = -1
+ if (data.car[i,1]!=data.car[i,2]){
+ x.car[i,data.car[i,1]] = 1
+ x.car[i,data.car[i,2]] = -1
+ }
+ }
>
> name = c()
> for (i in 1:ncol(x.user)){
+ name = c(name,paste("user",i,sep=""))
+ }
> colnames(x.user) = name
> name = c()
> for(i in 1:ncol(x.car)){
+ name = c(name,paste("car",i,sep=""))
+ }
> colnames(x.car) = name
>
> x = cbind(x.user,x.car)
> x = x[,-1]
> y = rep(1,nrow(x))
> data.train = data.frame(x,y)
```

일반화 Bradley-Terry 모형 적합 : 모형2

- 전처리된 자료를 이용하여 모형1과 같은 방법으로 모형2를 적합
- 자료의 수가 약 5만개이고 모수의 수가 약 5천개인데, 이 경우에는 모수의 수가 자료의 수에 비해 많아서 일반적인 로지스틱 회귀분석으로는 정확한 추정량을 얻기 힘들다. => Ridge 혹은 LASSO 페널티 함수를 붙여서 모형 적합

glmnet 패키지 : 모형2

- glmnet 패키지의 glmnet 함수는 로지스틱 회귀분석의 로그-우도함수에 Ridge, LASSO, elastic net 등의 여러 페널티 함수를 더한 목적 함수를 최적화하는 함수.
- glmnet 함수는 다양한 조정 모수값에 대한 모형 적합 결과를 제공하고, cv.glmnet 함수는 cross-validation을 통한 최적의 조정 모수 값을 찾고 그 값에서의 단일 모형 적합 결과를 제공함.
- 패키지 설치 및 불러오기

```
> install.packages("glmnet")
> library(glmnet)
```

샘플링 과정 : 모형2

- 모형 1에서는 모수가 42개였지만 모형2에서는 모수가 4252개로 늘어남.
- 즉, 자료의 크기가 모형1에 비해 훨씬 커져서 적합이 되지 않는 문제가 발생함 => 자료를 랜덤 샘플링 하여 분석을 진행
- 약 50000개의 전체 자료 중 2000개의 승패 자료를 샘플링
- y값이 모두 1이면 glmnet 함수를 사용할 수 없으므로 일부 x의 부호를 바꾸어 y값을 0으로 바꾸어줌

```
> sample.index = sample(1:nrow(x))[1:2000]
> x = x[sample.index,]
> x[1:1000,] = -x[1:1000,]
> y = c(rep(0,1000),rep(1,1000))
```

모형 적합 결과(Ridge) : 모형2

- cv.glmnet 함수를 이용하여 ridge 페널티 함수를 붙인 경우

```
> fit.ridge = cv.glmnet(x,y,family="binomial",intercept=FALSE,alpha=0)
> coef(fit.ridge)
```

- 적합 결과(모수의 수가 많아 일부만 표시)

```
user4201    -2.817472e-01
user4202    -2.527971e-01
user4203    -2.756326e-01
user4204     2.914404e-01
user4205     .
user4206     .
user4207     .
user4208     .
user4209     .
user4210     2.450019e-01
car1         8.186521e-02
car2        -2.559890e-02
car3        -1.394299e-02
car4         6.271671e-02
```

모형 적합 결과(Ridge) : 모형2

- 자동차의 계수값을 그림으로 표시

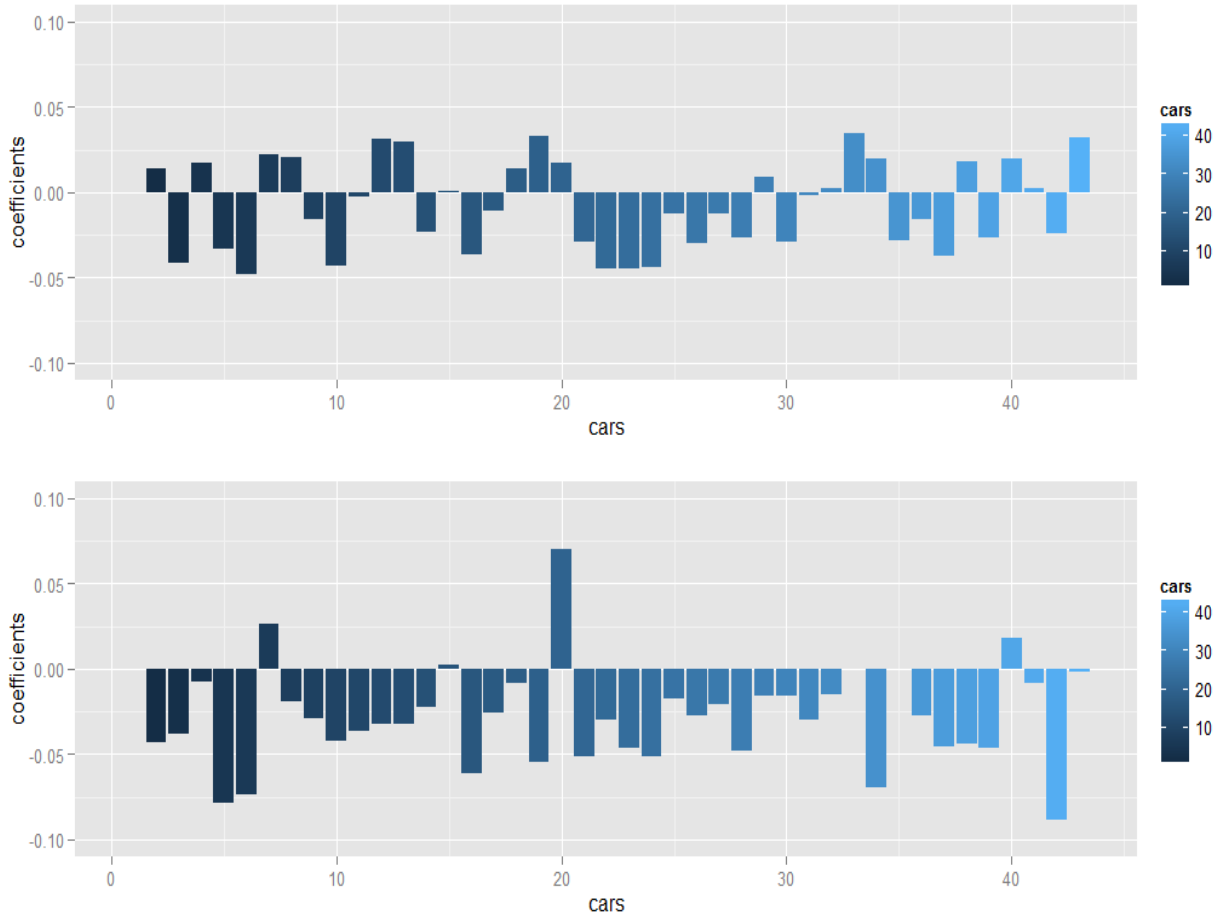
결과 해석(Ridge) : 모형2

- 기준 변수는 user1(사용자1)이며, user1의 계수값을 0으로 고정했을 때 나머지 사용자와 자동차들의 계수값을 출력함
- car1을 탄 user4201이 car2를 탄 user4208을 이길 확률은 다음과 같이 근사 계산된다.
  (계수값은 소수점 셋째 자리까지 쓰고 나머지는 버림)

$$Prob = \frac{exp(-0.281 + 0.081 - 0 - (-0.025))}{1 + exp(-0.281 - 0 + 0.081 - (-0.025))} = 0.456$$

모형1과 모형2(Ridge)의 결과 비교

- 두 모형의 계수의 크기를 맞추고 기준 변수를 car1로 맞춰서 적합 결과를 비교
  (위 : 모형1, 아래 : 모형2(Ridge))



Thank you!!