통계연구(2016), 특별호, 17-34

토픽모형을 이용한 빅데이터 기반 마이크로 세그멘테이션 방법론 연구

김용대[1] · 정구환[2]

요약

마케팅의 주요 방법 중 하나는 구매자료를 이용하여 고객들을 여러 그룹으로 세분화하고, 그룹마다 맞춤형 마케팅을 실시하는 것이다. 최근의 빅데이터 출현은 고객을 점점 더 크기가 작은 그룹에 속하게 하여 궁극적으로 개별 고객 맞춤형 마케팅을 하려는 노력들이 여러 분야에서 이루어지고 있다. 하지만 고객세분화를 위한 일반적인 방법인 군집분석은 군집의 개수가 커지면서 성능이 현저히 저하된다. 본 논문에서는 고객의 다양성은 최대한 보장하면서 동시에 예측력이 우수한 고객 세그멘테이션을 위하여 토픽모형을 이용한 마이크로 세분화 방법을 소개한다. 토픽모형은 문서의 분류를 위하여 개발된 방법인데, 본 논문에서는 토픽모형이 구매자료를 이용한 고객 마이크로 세그멘테이션에 어떻게 이용될 수 있는지를 설명하고, 실제 자료의 분석을 통하여 토픽모형의 우수성을 실증적으로 보여준다.

주요용어 : 토픽모형, 고객세분화, 마이크로 세그멘테이션

1. 서론

다양한 종류의 정형/비정형 빅데이터를 모으고 분석하고 이를 기반으로 효율적인 의사결정을 하는 빅데이터 생태계가 많은 관심을 모으고 있다. 특히 우리 모두를 놀라게 했던 알파고 충격은 새삼 빅데이터의 중요성을 각인시키고 있다.

빅데이터의 응용분야는 무궁무진하다. 스마트 공장(smart factory)으로 대변되는 제조업이나 정부 3.0을 추진 중인 정부 등 다양한 분야에서 빅데이터가 적용되고 있다. 특히, 마케팅에서의 빅데이터 활용이 기업의 생존문제로 여겨지면서, 많은 기업들이 경쟁적으로 빅데이터 마케팅을 수행하고 있다. 빅데이터 마케팅의 가장 큰 성공사례로는 아마존을 꼽을 수 있다. 빅데이터를 기반으로 한 도서추천으로 세간의 관심을 모은 작은 인터넷 쇼핑 회사가 2015년에는 미국 전체 유통기업에서 월마트를 제치고 시가총액 1위의 회사로 자리매김하였다. 또한, 최근 알파고를 이용한 마케팅으로 시가총액을 25조나 상승시킨 구글의 빅데이터 마케팅도 높게 평가 할 수 있다. 마케팅에서 빅데이터 응용에 대한 자세한 내용은 이서구 (2015)를 참조하기 바란다.

빅데이터 마케팅에서 가장 중요하고 핵심적인 것 중에 하나는 고객세분화이다. 빅데이터를 기반으로 기업이 관리하는 고객에게서 여러 가지 특성을 추출하고, 이를 바

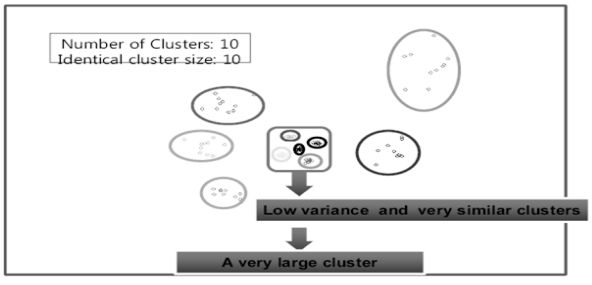
1) 서울특별시 관악구 관악로 1, 서울대학교 자연과학대학 통계학과, 교수. E-mail: ydkim0903@gmail.com
2) 서울특별시 관악구 관악로 1, 서울대학교 통계학과, 박사과정. E-mail: kuhemian@gmail.com

탕으로 고객을 여러 개의 세그먼트(그룹)로 나눈 후, 각 세그먼트별로 다양한 마케팅 활동을 수행하는 것이 마케팅의 핵심이 되고 있다. 카드사의 다양한 카드 출시나 유통사의 기획 상품 등이 빅데이터를 이용한 고객세분화를 기반으로 수행된 마케팅의 예이다.

고객의 인구통계학적 정보(예: 성별, 나이, 주소, 수입 등) 외에 위치정보와 쇼핑정보 등의 고객의 행동을 기반으로 한 자료가 고객세분화에 널리 사용되고 있다. 이러한 다양한 종류의 빅데이터를 기반으로 고객세분화를 하는 가장 대표적인 분석 방법으로는 군집분석이 있다. 군집분석이란 주어진 자료(즉, 고객)를 유사한 것들을 묶고 유사하지 않은 것은 나누어서 전체 자료를 몇 개의 세그먼트로 나누는 방법을 지칭한다 (박창이 외, 2013). 군집분석 방법론으로는 크게 계층적 군집분석과 비계층적 군집분석(K-평균 군집분석)을 들 수 있다.

고객세분화를 통하여 각 기업은 고객 맞춤형 관리를 수행 할 수 있는데, 이를 위하여 보통은 수십 개의 세그먼트를 이용한다. 하지만 최근에 목격되는 고객 니즈의 다양화와 빠른 변화로 인해 수십 개의 세그먼트를 이용한 고객관리의 허점이 노출되고 있다. 기존 방법과는 다른 보다 세분화된 고객 분류를 통하여 효과적인 마케팅의 필요성이 절실히 요구되고 있다. 기존의 방법과는 양적으로 다른 수천 개 이상의 고객 세그먼트를 구축하는 작업을 마이크로 세그멘테이션이라고 부른다 (Funk, 2002). 모바일 시대에서 고객 니즈 별 타켓광고가 마이크로 세그멘테이션을 활용할 수 있는 예이다. 뉴욕의 상징인 Macy's 백화점은 캘리포니아주로 진출하기 위하여 빅데이터를 기반으로 수만 개의 세그먼트를 추출하여 성공적인 마케팅을 수행하였다 (IBM, 2010).

양적변화가 질적변화를 가져오듯이 수십 개의 세그먼트를 위하여 사용되어졌던 다양한 군집분석방법들은 수천 개에서 수 만개의 세그먼트를 추출해야하는 마이크로 세그멘테이션에서는 성능이 매우 떨어진다. 〈그림 1.1〉은 군집에 따라 분산이 다른 자료에 대해서 K-평균 군집분석을 실시한 결과이다. 분산이 작은 여러 개의 군집이 K-평균 군집분석에서는 같은 군집으로 묶였다. 이는 자료간의 거리를 기반으로 하는 모든 군집분석 방법에서 공통적으로 발생하는 문제로 왜 기존의 군집분석 방법들이 좋지 않은 결과를 제공하는지를 잘 설명하고 있다.



〈그림 1.1〉 군집에 따라 분산이 다른 경우, K-평균 군집분석 방법은 분산이 작은 세그먼트를 잘 구분하지 못하는 경향이 있다.

본 논문에서는 고객의 구매이력을 바탕으로 마이크로 세그멘테이션을 할 수 있는 새로운 빅데이터 분석방법을 소개하고자 한다. 토픽모형이라고 불리는 LDA(Latent Dirichlet Allocation)방법을 소개하고, 이를 이용하여 고객의 구매이력을 바탕으로 마이크로 세그멘테이션을 구축하는 방법을 설명한다.

토픽모형은 문서를 출현하는 단어의 빈도를 기반으로 분류하기 위하여 개발된 분석방법이다. 단어-빈도 자료를 기반으로 요인분석을 통하여 문서를 분류하는 방법이 Deerwester et al. (1990)에 의하여 연구되었으며, 이를 바탕으로 한 확률모형을 Hofmann (1999)이 제안하였고, 베이지안 방법을 이용한 모수의 추정이 Blei et al. (2003)에 의하여 개발되었다. 토픽모형에서의 토픽은 군집분석에서의 세그먼트와 같은 개념인데, 토픽모형의 가장 큰 특징은 하나의 문서가 여러 개의 토픽을 가질 수 있다는 것이다. 토픽모형을 고객 구매이력 빅데이터에 적용할 때에는 고객은 문서에, 고객이 구매한 상품은 문서가 포함한 단어에 대응된다. 즉, 토픽모형을 이용한 고객 세그멘테이션에서는 한 고객이 여러 개의 세그먼트에 할당될 수 있는 다중 멤버십(multi-membership)이라는 특징이 있다. 보통 쇼핑에 대한 고객의 니즈는 다양하기 때문에 이러한 특징은 매우 자연스럽다. 예를 들면 하나의 신용카드를 부부가 같이 사용하는 경우에는 적어도 두 개의 쇼핑 니즈가 공존할 것이고, 이 경우 신용카드 보유 고객을 두 개의 세그먼트에 할당하는 것은 매우 자연스럽다.

마이크로 세그멘테이션에서 다중 멤버십은 매우 효율적으로 수천 개 이상의 세그먼트를 추출할 수 있는 길을 제공한다. 예를 들어, 10개의 토픽을 기반으로 고객 세그멘테이션을 구축할 때, 다중 멤버십을 허용하면 1024(2의 10제곱)개의 서로 다른 고객 세그먼트를 만들 수 있다. 따라서 수천 개 혹은 수만 개의 고객 세그멘테이션을 위해서는 수십 개의 토픽이면 충분하다는 매우 중요한 장점이 있다.

토픽모형의 또 다른 장점은 확률모형을 기반으로 하기 때문에 보통의 군집분석 방법에서 요구하는 거리를 정의하지 않아도 된다는 것이다. 특히, 구매이력자료는 보통 고객/상품 별 구매 횟수로 구성되는데, 횟수에 대하여 거리를 정의하는 것이 매우 어렵다. 특히, 자료가 매우 성긴(sparse) 경우에는 일반적으로 사용되는 유클리드 거리 등은 자료의 특징을 탐지하는데 한계가 있다. 반면에 토픽모형은 구매횟수 자료에 다항분포를 가정하기 때문에 매우 자연스럽고 효율적으로 구매이력 자료를 분석 할 수 있다.

본 본문에서는 빅데이터 기반 마이크로 세그멘테이션을 위한 토픽모형을 소개하고, 실제 온라인 쇼핑회사의 고객 구매이력자료를 기반으로 R에서 토픽모형을 통하여 세그먼트를 추출하는 방법에 대해서 설명하고자 한다. 2장에서는 자료의 구조 및 확률적 토픽모형을 소개하고, 3장에서는 모수 추정을 위한 베이지안 계산방법을 설명한다. 4장에서는 토픽모형의 빅데이터 적용을 위한 분산처리 알고리즘에 대하여 다루고, 5장에서는 온라인 쇼핑 구매이력 자료를 소개하고, R에서 토픽모형을 이용한 마이크로 세그멘테이션 분석을 하는 방법을 설명한다.

2. 자료 및 모형소개

2.1 자료구조

문서자료는 n개의 문서로 구성되어 있고, j번째 문서($j=1,...,n$)에는 n_j개의 단어가 포함되어 있다. 전체 단어의 종류는 W개로 각각의 단어를 단어사전 $\{1,...,W\}$의 원소에 하나씩 대응하여 앞으로는 단어를 1부터 W까지의 숫자로 간주한다. j번째 문서에서 i번째 단어($i=1,...,n_j$)를 $x_{ji} \in \{1,...,W\}$로 나타낸다면 j번째 문서는 $\boldsymbol{x}_j = (x_{ji})_{i=1}^{n_j}$로, 문서자료 전체는 $\boldsymbol{x} = (\boldsymbol{x}_j)_{j=1}^{n}$로 나타낼 수 있다.

쇼핑자료는 n명의 고객으로 구성되어 있고, j번째 고객($j=1,...,n$)은 n_j개의 상품을 구매했다. 전체 상품의 종류는 W개로 각각의 상품을 품목 $\{1,...,W\}$의 원소에 하나씩 대응하여 앞으로는 상품을 1부터 W까지의 숫자로 간주한다. j번째 고객이 i번째로 구매한 상품($i=1,...,n_j$)을 $x_{ji} \in \{1,...,W\}$로 나타낸다면 j번째 고객은 $\boldsymbol{x}_j = (x_{ji})_{i=1}^{n_j}$로, 쇼핑자료 전체는 $\boldsymbol{x} = (\boldsymbol{x}_j)_{j=1}^{n}$로 나타낼 수 있다.

2.2 LDA 모형

확률적 토픽 모형에서는 문서를 단어 가방(bag of words)으로 가정한다. 즉, 문서 안에서 단어들의 순서는 모형에 영향을 끼치지 않고, 단어들의 빈도만 모형에 영향을 끼친다. 쇼핑자료의 경우에는 고객이 상품을 구매한 순서는 모형에 영향을 끼치지 않고, 상품을 구매한 빈도만 모형에 영향을 끼치는 것이다.

k번째 토픽($k=1,...K$)을 단어사전에 대한 확률분포 $\boldsymbol{\phi}_k = (\phi_{k1},...,\phi_{kW})$로 나타내고, $\boldsymbol{\theta}_j = (\theta_{j1},...,\theta_{jK})$는 토픽집합 $\{1,...,K\}$에 대한 확률분포로 j번째 문서가 각각의 토픽을 어떠한 확률로 갖는지를 나타낸다. 쇼핑자료의 경우에는 토픽을 고객의 관심사로 해석할 수 있다. k번째 토픽이 "육아"라면 $\boldsymbol{\phi}_k = (\phi_{k1},...,\phi_{kW})$는 "기저귀", "분유"와 같은 육아와 관련된 상품에서 큰 확률 값을 가질 것이다. θ_{jk}는 j번째 고객이 k번째 토픽인 "육아"에 얼마나 관심을 갖고 있는지를 나타내는 확률 값이다.

확률적 토픽 모형에서는 관측된 자료의 우도(likelihood) 함수를 다음과 같이 가정한다.

$$p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{\phi}) = \prod_{j=1}^{n}\prod_{i=1}^{n_j} p(x_{ji}|\boldsymbol{\theta}_j,\boldsymbol{\phi}) = \prod_{j=1}^{n}\prod_{i=1}^{n_j}\left(\sum_{k=1}^{K}\theta_{jk}\phi_{kx_{ji}}\right)$$

이 때, $\boldsymbol{\theta} = (\boldsymbol{\theta}_j)_{j=1}^{n}, \boldsymbol{\phi} = (\boldsymbol{\phi}_k)_{k=1}^{K}$이다. 즉, $(\boldsymbol{\theta},\boldsymbol{\phi})$가 주어졌을 때 문서에 속한 각각의 단어는 토픽들의 혼합모형에서 추출된다고 가정하는 것이다. 쇼핑자료의 경우에는 고객이 어떤 상품을 구매하는지가 관심사들의 혼합모형에서 추출된다는 뜻이다.

LDA 모형은 $\boldsymbol{\theta}_j$와 $\boldsymbol{\phi}_k$에 대한 사전분포로 각각 $D(\alpha,...,\alpha)$와 $D(\beta,...,\beta)$를 사용한다. 여기서 $D(\alpha_1,...,\alpha_K)$는 모수가 $(\alpha_1,...,\alpha_K)$인 디리클레 분포를 뜻한다. 디리클레 분포의 확률밀도함수는 식 (2.1)과 같이 정의된다.

$$\frac{\Gamma\left(\sum_{k=1}^{K}\alpha_k\right)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}x_k^{\alpha_k-1} \tag{2.1}$$

이 때, $x_k \in (0,1)$, $\sum_{k=1}^{K}x_k = 1$이고 $\Gamma(t) = \int_0^{\infty} x^{t-1}e^{-x}dx$이다.

x_{ji}가 어떤 토픽에서 추출되었는지(어떤 관심사로부터 상품의 구매가 이루어졌는지)를 나타내는 잠재변수 z_{ji}를 도입하면 LDA 모형을 다음과 같이 계층적으로 나타낼 수 있다.

$$\begin{aligned}
\boldsymbol{\theta}_j|\alpha &\sim D(\alpha,...,\alpha),\\
\boldsymbol{\phi}_k|\beta &\sim D(\beta,...,\beta),\\
p(z_{ji}=k|\boldsymbol{\theta}_j) &= \theta_{jk},\\
p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\phi}) &= \prod_{j=1}^{n}\prod_{i=1}^{n_j}p(x_{ji}|z_{ji},\boldsymbol{\phi}_{z_{ji}})
\end{aligned}$$

이 때, $p(x_{ji}=w|z_{ji}=k,\phi_k)=\phi_{kw}$, $\boldsymbol{z}=(\boldsymbol{z}_j)_{j=1}^{n}$, $\boldsymbol{z}_j=(z_{ji})_{i=1}^{n_j}$이다.

집합 $\{i : x_{ji}=w, z_{ji}=k\}$의 원소의 개수를 N_{jkw}로 정의하고, 아래첨자에 점($\cdot$)은 해당 인덱스에 대해서 더하라는 뜻이다. 예를 들어, $N_{jk\cdot} = \sum_w N_{jkw}$, $N_{j\cdot\cdot} = \sum_k\sum_w N_{jkw}$이다.

$(\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{z})$의 사후분포는 다음과 같이 구할 수 있다.

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

여기서 $I(\cdot)$는 지시함수이다.

3. 다양한 계산 알고리즘 소개

LDA 모형에서 사후분포를 직접 다루는 것은 힘들기 때문에 사후분포의 근사가 필요하다. 이 장에서는 깁스 샘플러(Gibbs sampler) 방법을 기반으로 한 다양한 알고리즘을 소개한다.

3.1 전체 깁스 샘플러

$\boldsymbol{\theta}$와 $\boldsymbol{\phi}$에 대한 조건부 사후분포는 다음과 같이 분해된다.

$$P(\boldsymbol{\theta}|\boldsymbol{x},\boldsymbol{z},\boldsymbol{\phi}) = P(\boldsymbol{\theta}|\boldsymbol{z}) = \prod_{j=1}^{n} P(\boldsymbol{\theta}_j|\boldsymbol{z}_j),$$

$$P(\boldsymbol{\phi}|\boldsymbol{x},\boldsymbol{z},\boldsymbol{\theta}) = P(\boldsymbol{\phi}|\boldsymbol{x},\boldsymbol{z}) = \prod_{k=1}^{K} P(\boldsymbol{\phi}_k|\boldsymbol{x},\boldsymbol{z})$$

이 때,

$$\boldsymbol{\theta}_j|\boldsymbol{z}_j \sim D(\alpha + N_{j1\,\cdot},\ldots,\alpha + N_{jK\,\cdot}), \tag{3.1}$$

$$\boldsymbol{\phi}_k|\boldsymbol{x},\boldsymbol{z} \sim D(\beta + N_{\cdot\,k1},\ldots,\beta + N_{\cdot\,kW}) \tag{3.2}$$

이다. 다음으로 $\boldsymbol{z}$에 대한 조건부 사후분포는 다음과 같이 분해된다.

$$P(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\phi}) = \prod_{j=1}^{n} P(\boldsymbol{z}_j|\boldsymbol{x}_j,\boldsymbol{\theta}_j,\boldsymbol{\phi}) = \prod_{j=1}^{n}\prod_{i=1}^{n_j} P(z_{ji}|x_{ji},\boldsymbol{\theta}_j,\boldsymbol{\phi}).$$

이 때,

$$P(z_{ji}|x_{ji},\boldsymbol{\theta}_j,\boldsymbol{\phi}) \propto P(z_{ji}|\boldsymbol{\theta}_j)P(x_{ji}|z_{ji},\boldsymbol{\phi})$$

이다. 따라서

$$P(z_{ji} = k|x_{ji} = w,\boldsymbol{\theta}_j,\boldsymbol{\phi}) \propto \theta_{jk}\phi_{kw}$$

이므로 쉽게 계산할 수 있다.

〈알고리즘 3.1〉 전체 깁스 샘플러

---

1. $\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{z}$ 초기화
2. 수렴할 때 까지 반복 :
   (a) $\boldsymbol{\theta}$를 $P(\boldsymbol{\theta}|\boldsymbol{z})$에서 추출
   (b) $\boldsymbol{\phi}$를 $P(\boldsymbol{\phi}|\boldsymbol{x},\boldsymbol{z})$에서 추출
   (c) $\boldsymbol{z}$를 $P(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta},\boldsymbol{\phi})$에서 추출

3.2 붕괴 깁스 샘플러 (Collapsed Gibbs Sampler)

전체 깁스 샘플러 방법은 변수간의 의존성이 커서 수렴이 늦은 단점이 있다. $\boldsymbol{\theta}$와

$\boldsymbol{\phi}$를 적분하여 제거하는 것을 통해 변수간의 의존성을 줄일 수 있는데 이러한 방법을 붕괴 깁스 샘플러 방법이라 부른다 (Griffiths and Steyvers, 2004). z_{ji}의 조건부 사후분포는 다음과 같다.

$$P(z_{ji}=k|\boldsymbol{z}^{-ji},\boldsymbol{x}^{-ji},x_{ji}=w)\propto\frac{\beta+N_{\cdot kw}^{-ji}}{W\beta+N_{\cdot k\cdot}^{-ji}}\left(\alpha+N_{jk\cdot}^{-ji}\right).$$

윗첨자 $-ji$는 원소의 개수를 셀 때 j번째 문서의 i번째 단어(j번째 고객이 i번째로 구매한 상품)를 제외한다는 것을 뜻한다.

〈알고리즘 3.2〉 붕괴 깁스 샘플러

1. $\boldsymbol{z}$ 초기화
2. 수렴할 때 까지 반복 :
   (a) $z_{ji}(j\in\{1,\ldots,n\},i\in\{1,\ldots,n_j\})$를 $P(z_{ji}|\boldsymbol{z}^{-ji},\boldsymbol{x})$에서 추출

$\boldsymbol{z}$가 생성되면 확률적 토픽 모형에서 궁극적으로 알고 싶은 값인 $\boldsymbol{\theta}$와 $\boldsymbol{\phi}$는 식 (3.1)과 식 (3.2)를 통해 얻을 수 있다.

3.3 부분 붕괴 깁스 샘플러 (Partially Collapsed Gibbs Sampler)

붕괴 깁스 샘플러 방법의 단점은 $\boldsymbol{\phi}$를 적분하여 없애기 때문에 문서(고객)의 조건부 독립성이 사라지는 것이다. 문서(고객)의 조건부 독립성은 4장에서 소개할 분산알고리즘에 반드시 필요한 성질이다. 전체 깁스 샘플러 방법에서 $\boldsymbol{\theta}$만을 적분하여 없애면 문서(고객)의 조건부 독립성을 유지하면서 붕괴 깁스 샘플러 방법처럼 변수간의 의존성을 줄일 수 있다.

$\boldsymbol{\phi}_k$와 z_{ji}의 조건부 사후분포는 다음과 같다.

$$\boldsymbol{\phi}_k|\boldsymbol{x},\boldsymbol{z}\sim D(\beta+N_{\cdot k1},\ldots,\beta+N_{\cdot kW}), \tag{3.3}$$

$$P(z_{ji}=k|\boldsymbol{z}^{-ji},\boldsymbol{x},\boldsymbol{\phi})\propto\left(\alpha+N_{jk\cdot}^{-ji}\right)\phi_{kx_{ji}}. \tag{3.4}$$

〈알고리즘 3.3〉 부분 붕괴 깁스 샘플러

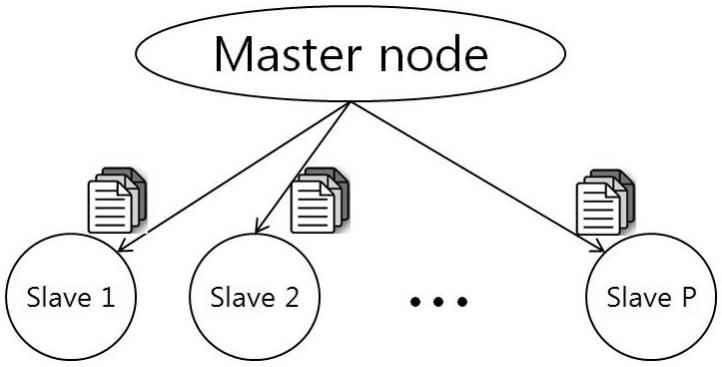
1. $\boldsymbol{\phi},\boldsymbol{z}$ 초기화
2. 수렴할 때 까지 반복 :
   (a) $\boldsymbol{\phi}$를 $P(\boldsymbol{\phi}|\boldsymbol{x},\boldsymbol{z})$에서 추출
   (b) $z_{ji}(j\in\{1,\ldots,n\},i\in\{1,\ldots,n_j\})$를 $P(z_{ji}|\boldsymbol{z}^{-ji},\boldsymbol{x},\boldsymbol{\phi})$에서 추출

붕괴 깁스 샘플러 방법과 마찬가지로 $\boldsymbol{\theta}$는 식 (3.1)을 통해 얻을 수 있다.

4. 분산알고리즘

일반적으로 깁스 샘프러 방법을 기반으로 한 알고리즘들은 많은 계산을 필요로 한다. 따라서 빅데이터 분석을 위해 3장의 알고리즘들을 직접 적용한다면 시간의 제약과 메모리의 한계에 직면하게 된다. 이 장에서는 이 문제들에 대한 해결책으로 분산알고리즘을 소개한다.

전체 문서(고객) $D=\{1,...,n\}$를 P개의 그룹$\left(D_{(1)},...,D_{(P)}\right)$으로 나누어 P개의 프로세스가 각각 한 그룹씩 맡아서 처리하고자 한다. 이를 위해서 〈그림 4.1〉처럼 네트워크를 구성해야 한다. 마스터 노드는 전체적인 정보를 슬레이브 노드들에게 제공하고, 슬레이브 노드들은 마스터 노드에서 받은 정보를 이용하여 해당 그룹의 정보를 갱신하고, 갱신된 정보를 마스터 노드에게 전달한다. 마스터 노드는 슬레이브 노드들에서 얻어진 정보를 통해 전체적인 정보를 갱신하여 다시 슬레이브 노드들에게 제공하는 방식으로 분산처리가 이루어진다.



〈그림 4.1〉 분산처리를 위한 네트워크

LDA 모형의 분산알고리즘은 부분 붕괴 깁스 샘플러 방법을 이용한다. 〈알고리즘 3.3〉의 (a)를 마스터 노드에서 처리하는 것이고, (b)를 그룹마다 나누어 슬레이브 노드들에서 처리하는 것이다. 이 때, 토픽의 개수에 비해 문서에 속한 단어(혹은 고객이 구매한 상품)의 수가 훨씬 많기 때문에 (b)에서 많은 시간이 소요된다. 따라서 (b)의 분산처리를 통해 큰 속도의 향상을 얻을 수 있다.

보다 자세히 설명하면 마스터 노드에서 토픽 ϕ를 슬레이브 노드들에게 제공한다. 슬레이브 노드들은 z_{ji}를 식 (3.4)에서 추출하기 위해서 $N_{jk\cdot}$와 ϕ가 필요하다. ϕ는 마스터 노드로부터 제공되었고, $N_{jk\cdot}$은 다른 문서에 의존하지 않는 해당 문서만으로 계산할 수 있는 값이기 때문에 식 (3.4)를 그룹마다 독립적으로 수행하는 것이 가능하다. 슬레이브 노드에서는 $\boldsymbol{z}_{(p)}\left(=\left(\boldsymbol{z}_j\right)_{j\in D_{(p)}}\right)$의 추출이 끝난 후 $N_{(p)kw}\left(=\sum_{j\in D_{(p)}} N_{jkw}\right)$를 마스터 노드에 전달한다. 마스터 노드는 각각의 슬레이브 노드들에서 전달된 정보로

$N_{\cdot kw}\left(=\sum_{p=1}^{P} N_{(p)kw}\right)$를 얻을 수 있고, 따라서 식 (3.3)을 통해 $\boldsymbol{\phi}$를 추출한다. 분산 알고리즘은 이 과정을 수렴할 때까지 반복하는 것이다.

〈알고리즘 4.1〉 분산 알고리즘

1. 마스터 노드 : $\boldsymbol{\phi}$ 초기화
   슬레이브 노드 : $\boldsymbol{z}_{(p)}$ 초기화
2. 수렴할 때 까지 반복 :
   (a) 마스터 노드 :
   $\boldsymbol{\phi}$를 $P(\boldsymbol{\phi}|\boldsymbol{x},\boldsymbol{z})$에서 추출하여 슬레이브 노드들에게 전달
   (b) 슬레이브 노드 :
   $z_{ji}(j \in \{1,\ldots,n\}, i \in \{1,\ldots,n_j\})$를 $P(z_{ji}|\boldsymbol{z}^{-ji},\boldsymbol{x},\boldsymbol{\phi})$에서 추출
   $N_{(p)kw}$를 마스터 노드에 전달

5. 자료분석

5.1 R 패키지를 이용한 LDA 분석 방법

인터넷 쇼핑몰의 쇼핑자료를 R 패키지를 이용하여 LDA 모형으로 분석하였다. 쇼핑몰에서 판매되는 상품들을 70가지 상품으로 중분류(예: 의류, 화장품, 전자제품, 건강제품 등)하였고, 인터넷 쇼핑몰 회원 중 임의 추출된 10,000명을 대상으로 2015년 한 해 동안 70가지 상품에 대한 구매 횟수를 저장하였다.

(a)

A0
A1
A2
A3
A4
A5
A6
A7
A8

(b)

고객	상품	횟수
1	6	12
1	10	2
1	11	6
1	13	4
1	29	2
1	37	5
2	2	1
2	6	6

〈그림 5.1〉 (a) “품목.txt”의 자료 형태
(b) “쇼핑.csv”의 자료 형태

자료는 “품목.txt”와 “쇼핑.csv”로 저장되어 있다. 〈그림 5.1〉은 “품목.txt”와 “쇼핑.csv”의 일부분을 나타낸 것이다. “품목.txt”에는 70가지 상품들의 이름인 $A0,\ldots,A9$, $B0,\ldots,B9,\ldots,G0,\ldots,G9$이 줄바꿈(newline)으로 구분되어 한 줄에 하나씩 저장되어 있다. “쇼핑.csv”에는 고객이 어떤 상품은 몇 번 구매했는지가 저장되어 있다. 이 때 “품목.txt”에 있는 상품들을 순서대로 $\{1,\ldots,70\}$에 대응시켜 상품을 상품명 대신 숫자로 나타내었다. 첫 번째 고객은 $A5$를 12회, $A9$을 2회, $B0$를 6회, $B2$를 4회, $C8$을 2회, $D6$을 5회 구매했다는 의미이다.

이 장에서는 붕괴 깁스 샘플러 알고리즘을 구현한 R의 “lda” 패키지를 이용해 분석하였다. “lda” 패키지를 사용하기 위해서는 먼저 “쇼핑.csv”를 “lda” 패키지에서 요구하는 〈그림 5.2〉의 형태인 “쇼핑.dat”로 변형해야한다. 앞에서는 “품목.txt”에 있는 상품들을 $\{1,\ldots,70\}$에 대응시켰지만, “lda” 패키지를 사용하기 위해서는 $\{0,\ldots,69\}$에 대응시켜야 한다. “쇼핑.dat”는 각각의 줄이 한 고객에 해당하며 첫 번째 숫자는 해당 고객이 몇 종류의 상품을 구매하였는지를 나타낸다. 이어지는 $a:b$는 상품 a를 b회 구매했다는 뜻이다. 즉, 첫 번째 고객은 6종류의 상품을 구매하였고, $A5$를 12회, $A9$을 2회, $B0$를 6회, $B2$를 4회, $C8$을 2회, $D6$을 5회 구매했다는 뜻이다. 〈그림 5.3〉은 “쇼핑.csv”를 “쇼핑.dat”로 변환하는 R 코드이다.

```
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
```

〈그림 5.2〉 “쇼핑.dat”의 자료 형태

```
shop=read.csv("D:\\쇼핑.csv")
mj=table(shop[,1])
outfile=file("D:\\쇼핑.dat")
line=c();k=1
for(j in 1:length(mj)){
  line=paste0(line,mj[j])
  for(i in k:(k+mj[j]-1)){
    line=paste0(line," ",shop[i,2]-1,":",shop[i,3])
  }
  line=paste0(line,"\n")
  k=k+mj[j]
}
writeLines(line,outfile)
close(outfile)
```

〈그림 5.3〉 “쇼핑.csv”를 “쇼핑.dat”로 변환하는 R 코드

“쇼핑.dat”는 read.documents() 함수로 불러올 수 있고, “품목.txt”는 read.vocab() 함수로 불러 올 수 있다. 붕괴 깁스 샘플러 방법은 lda.collapsed.gibbs.sampler() 함수로 구현되어 있다. 인수 documents는 쇼핑자료, K는 토픽의 개수, vocab은 품목자료, alpha는 LDA 모형의 모수 α, eta는 모수 β, num.iterations는 반복 횟수로 $\boldsymbol{z}$를 몇 번 추출하느냐를 나타낸다. compute.loglikelihood는 이진값(T,F)을 갖는 인수로 기본 값

은 F로 지정되어 있다. compute.loglikelihood가 T일 때 두 종류의 로그우도를 계산하는데, 이를 통해 깁스 샘플러 알고리즘의 수렴 여부를 파악할 수 있다. 실험에서는 토픽의 개수를 11개, 모수 α와 β는 모두 1.0으로 하였고 총 500번 반복하였다.

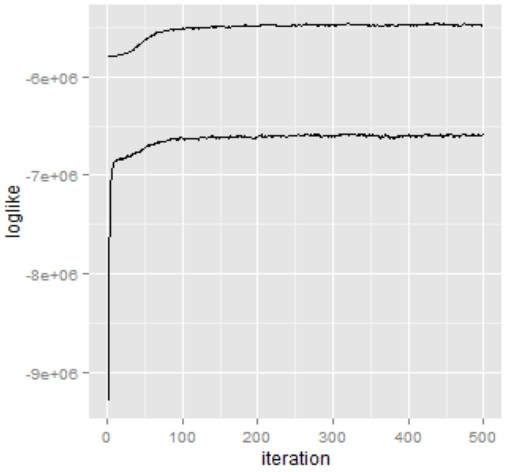
```
library(lda)
shop2=read.documents("D:\\쇼핑.dat")
item_name=read.vocab("D:\\품목.txt")
K=11; alpha=1.0; beta=1.0; iter=500
lda=lda.collapsed.gibbs.sampler(shop2,K,item_name,
    iter,alpha,beta,compute.log.likelihood=T)
```

〈그림 5.4〉 "lda" 패키지를 이용한 LDA 분석 코드

lda.collapsed.gibbs.sampler()의 결과 값으로는 document_sums, topics, log.likelihoods가 있다. documents_sums에는 마지막으로 추출된 $\boldsymbol{z}$로 얻어지는 N_{jk}^{T}.가 저장되고, topics에는 $N_{.kw}$가 저장되고, log.likelihoods에는 $\boldsymbol{z}$가 추출될 때마다 계산된 두 종류의 로그우도가 저장된다. N_{jk}.과 $N_{.kw}$를 바탕으로 $\boldsymbol{\theta}$와 $\boldsymbol{\phi}$는 각각 식 (3.1)과 식 (3.2)를 통해 얻을 수 있다.

5.2 자료분석 결과

$\boldsymbol{\theta}$와 $\boldsymbol{\phi}$를 추정하기에 앞서 깁스 샘플러 알고리즘의 수렴 여부를 파악해야 한다. 〈그림 5.5〉는 lda.collapsed.gibbs.sampler() 함수에서 제공하는 두 가지 로그우도를 그래프로 나타낸 것이다. 100번의 반복이 지난 후에 두 가지 로그우도 모두 충분히 수렴했음을 관찰할 수 있다. $\boldsymbol{\theta}$와 $\boldsymbol{\phi}$는 〈그림 5.7〉의 코드를 통해 추정할 수 있다.



〈그림 5.5〉 LDA 모형의 두 가지 로그우도

```
library(ggplot2)
plot=data.frame(iteration=c(1:iter,1:iter),
  loglike=c(lda$log.likelihoods[1,],lda$log.likelihoods[2,]),
  type=c(rep(1,iter),rep(2,iter)))
ggplot(plot,aes(x=iteration,y=loglike,group=type))+
  geom_line()
```

〈그림 5.6〉 로그우도 그래프를 그리는 R 코드

```
n=10000; W=70
theta=matrix(0,nrow=n,ncol=K)
for(i in 1:n)
  theta[i,]=t(lda$document_sums[,i])/
    sum(lda$document_sums[,i])
phi=matrix(0,nrow=K,ncol=W)
for(i in 1:K)
  phi[i,]=lda$topics[i,]/
    sum(lda$topics[i,])
```

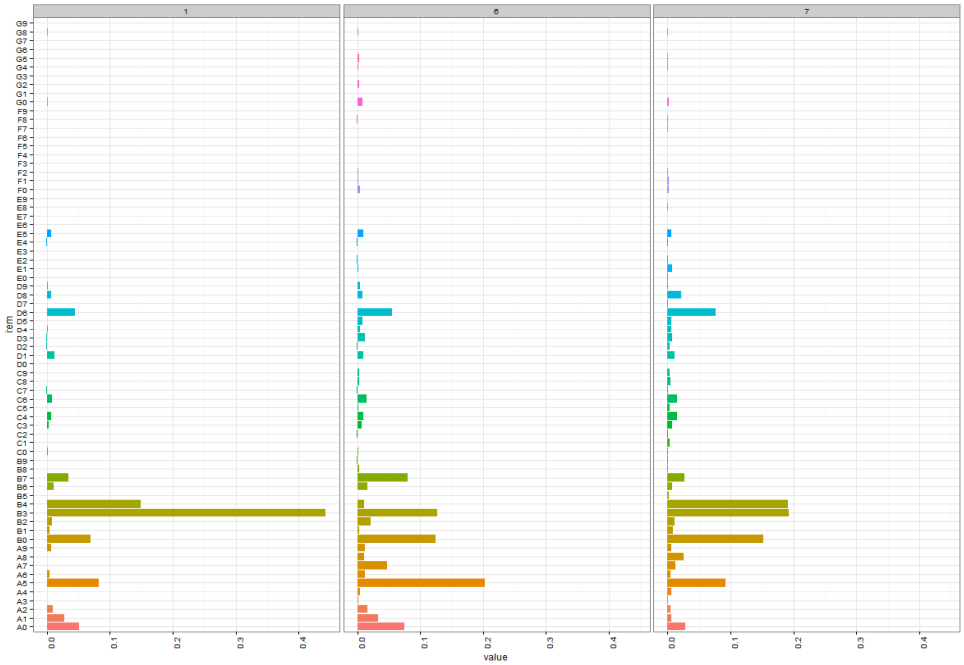
〈그림 5.7〉 θ와 ϕ를 추정하는 R 코드

토픽은 품목에 대한 확률 분포를 나타내기 때문에 토픽을 대표하는 주요 상품으로 토픽의 이름을 붙이면 토픽을 효과적으로 나타낼 수 있다. 쉽게 생각할 수 있는 방법은 단순히 확률이 큰 상품들을 주요 상품으로 보는 것이다. 이것은 상품간의 구매 빈도 차이가 상당히 클 때 의미가 없어질 수 있다. 예를 들어, A 상품의 구매 빈도가 다른 상품들의 구매 빈도보다 훨씬 크다면, 대부분의 토픽들에서 A의 확률이 클 것이고, 결국 여러 토픽들의 주요 상품이 같아질 것이다. 따라서 단순히 확률이 큰 것을 주요 상품으로 보는 것은 문제가 있다. 본 논문에서는 토픽에서 상품의 확률을 쇼핑 자료 전체에서 해당 상품이 차지하는 비율로 나눈 값을 리프트(lift)라 부르고, 이 값이 큰 상품을 주요 상품으로 보았다. 이는 해당 토픽에서만 특별히 확률이 큰 상품들을 주요 상품으로 보는 것이다. 토픽마다 리프트가 큰 두 상품을 주요 상품으로 정의하고 이것으로 토픽의 이름을 붙였다. 〈그림 5.8〉은 리프트를 계산하는 R코드이다.

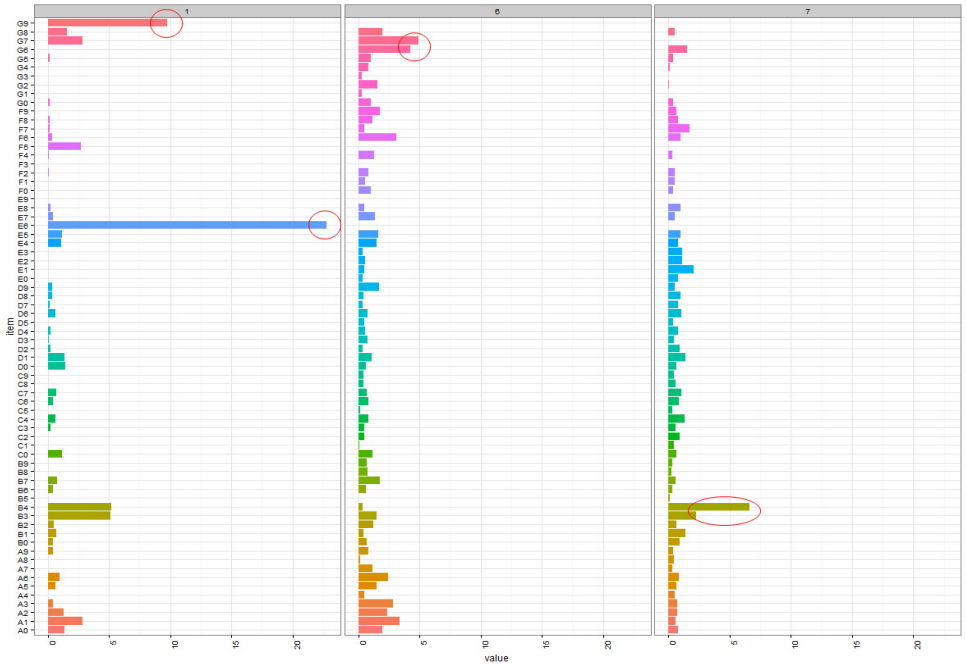
```
p=colSums(lda$topics)/sum(lda$topics)
lift=matrix(0,nrow=K,ncol=W)
colnames(lift)=item_name
topic_name=c()
for(i in 1:K){
  lift[i,]=phi[i,]/p
  sorted=sort(lift[i,],decreasing=T)[1:2]
  topic_name=c(topic_name,
    paste(names(sorted),collapse="."))
}
```

〈그림 5.8〉 리프트를 계산하는 R 코드

〈그림 5.9〉는 1, 6, 7번째 토픽을 나타낸다. 앞서 언급하였던 것과 같이 토픽마다 확률이 큰 상품들은 크게 다르지 않다. 〈그림 5.9〉는 〈그림 5.11〉의 코드로 그릴 수 있다.



〈그림 5.9〉 1, 6, 7번째 토픽

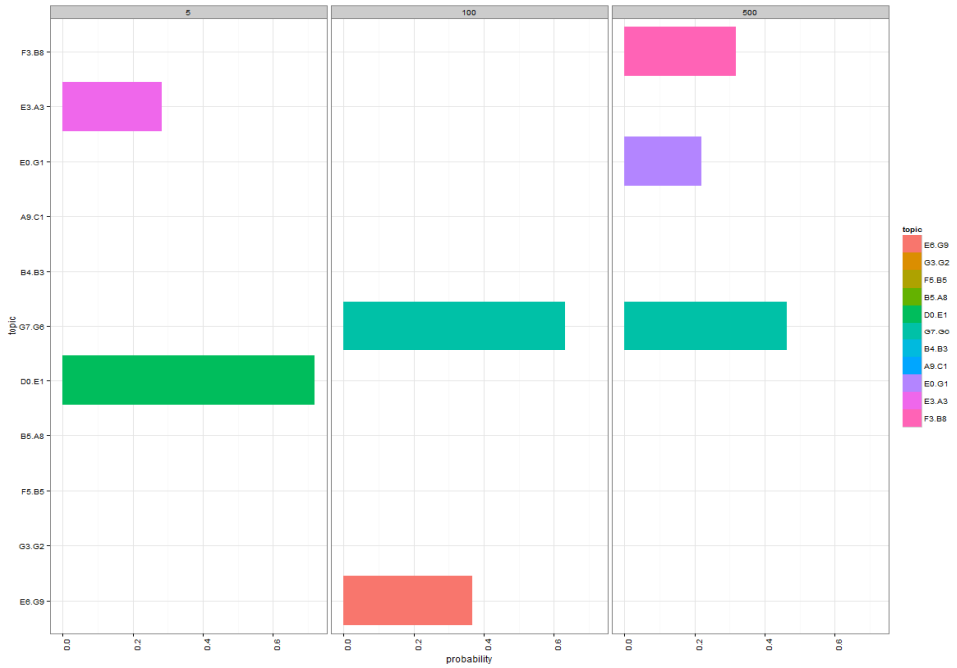


〈그림 5.10〉 1, 6, 7번째 토픽의 리프트

```
library(ggplot2)
library(reshape2)
theme_set(theme_bw())
colnames(phi)=item_name; idx=c(1,6,7)
phi.df <- melt(cbind(data.frame(phi[idx,]),
  topic=factor(idx)),variable.name="item",id.vars="topic")
qplot(item,value,fill=item,ylab="value",data=phi.df,geom="bar",stat="identity")+
  theme(axis.text.x = element_text(angle=90, hjust=1),legend.position="none") +
  coord_flip() + facet_wrap(~ topic, ncol=length(idx))
```

〈그림 5.11〉 토픽을 그리는 R 코드

〈그림 5.10〉은 1, 6, 7번째 토픽의 리프트를 나타내는데, 토픽마다 리프트가 큰 상품들이 매우 다른 것을 확인할 수 있다. 1번째 토픽은 E6와 G9이, 6번째 토픽은 G7과 G6가, 7번째 토픽은 B4와 B3가 리프트가 크다. 따라서 1번째 토픽은 "E6.G9", 6번째 토픽은 "G7.G6", 7번째 토픽은 "B4.B3"로 이름을 붙였다. 〈그림 5.10〉은 〈그림 5.11〉의 코드에서 topic을 lift로 대체하여 그릴 수 있다.



〈그림 5.12〉 5, 100, 500번째 고객의 11개 토픽에 대한 분포

〈그림 5.12〉는 5, 100, 500번째 고객의 11개 토픽에 대한 분포를 나타낸다. 5번째 고객은 D0.E1과 E3.A3를, 100번째 고객은 G7.G6와 E6.G9를, 500번째 고객은 G7.G6, F3.B8, E0.G1을 갖는다. 따라서 LDA모형에서 토픽의 개수를 11개로 할 경우 고객을 $2047 (= 2^{11} - 1$, 아무 토픽도 갖지 않는 경우는 없으므로)가지로 마이크로 세그멘테이션이 가능하다.

```
theme_set(theme_bw())
colnames(theta)=char_name; idx=c(5,100,500)
theta.df <- melt(cbind(data.frame(theta[idx,]),
  client=factor(idx)),variable.name="topic",id.vars="client")
qplot(topic,value,fill=topic,ylab="probability",
  data=theta.df,geom="bar",stat ="identity")+
  theme(axis.text.x = element_text(angle=90, hjust=1)) +
  coord_flip() + facet_wrap(~ client, ncol=length(idx))
```

〈그림 5.13〉 고객의 토픽에 대한 분포($\boldsymbol{\theta}$)를 그리는 R 코드

6. 결론

본 논문에서는 LDA 모형을 이용한 고객의 마이크로 세그멘테이션을 제안하였다. 마이크로 세그멘테이션에서 세그먼트의 개수는 토픽의 개수에 대해 기하급수적으로 증가하기 때문에 비교적 적은 토픽으로 훌륭한 마이크로 세그멘테이션이 가능하다. 하지만 기업의 입장에서는 너무 많은 수의 세그먼트는 마케팅의 어려움을 야기할 수 있다. 다중 멤버십을 이용한 마이크로 세그멘테이션의 경우에는 이러한 문제를 비교적 쉽게 해결할 수 있다. K개의 토픽이 있을 때 마케팅 전략을 토픽마다 하나씩 총 K개를 세워서 여러 토픽을 갖는 고객에게는 해당 토픽들에 대한 마케팅 전략을 섞어서 사용하는 방식이 가능하다.

첫 번째 고객이 첫 번째 토픽을 $0.9(\theta_{11}=0.9)$, 두 번째 토픽을 $0.1(\theta_{12}=0.1)$ 가졌고 두 번째 고객이 첫 번째 토픽을 $0.1(\theta_{21}=0.1)$, 두 번째 토픽을 $0.9(\theta_{22}=0.9)$ 가졌다면, 두 고객은 토픽 1과 2를 가지는 세그먼트에 속하도록 하였다. 하지만 고객이 토픽들을 어느 정도 가지는지 정확히 알 수 있기 때문에 보다 정교한 마이크로 세그멘테이션이 가능하다. 예를 들면 토픽 1과 2를 가지는 세그먼트 $\{j:\theta_{j1}>0,\theta_{j2}>0, \theta_{jk}=0 \text{ for } k>2\}$는 토픽 1을 많이, 2를 적게 가지는 세그먼트 $\{j:\theta_{j1}>0.5,\theta_{j2}>0, \theta_{jk}=0 \text{ for } k>2\}$와 토픽 1을 적게, 2를 많이 가지는 세그먼트 $\{j:\theta_{j1}>0,\theta_{j2}>0.5, \theta_{jk}=0 \text{ for } k>2\}$로 나눌 수 있다. 이러한 방법을 통해 무궁무진한 마이크로 세그멘테이션이 가능하다.

본 논문에서는 R 패키지를 이용하여 쉽게 LDA 모형을 사용할 수 있는 방법을 설명하였다. 하지만 R 패키지로는 빅데이터를 다루기가 쉽지 않다. 1차적인 해결 방법은 R보다 속도가 빠른 프로그래밍 언어를 사용하는 것이다. 하지만 R 패키지의 함수들도 내부적으로는 C언어로 구현되어 있기 때문에 이를 통해 얻을 수 있는 속도 향상은 거의 없다. 근본적인 해결 방법은 분산처리를 통해 속도를 향상시키는 것이다. 4장에서 소개한 LDA 모형의 분산알고리즘을 구현한다면 빅데이터 분석이 가능하다. 토픽의 개수가 적기 때문에 마스터 노드에서 걸리는 시간은 슬레이브 노드에서 걸리는 시간에 비해 미미하다. 따라서 P개의 슬레이브 노드를 사용하여 분산알고리즘으로 분석할 경우 기존 알고리즘에 비해 시간이 약 $\frac{1}{P}$배 걸리게 된다. 따라서 단순히 코어

의 숫자를 늘려줌으로써 빅데이터 분석이 가능하다. 분산알고리즘은 MPI(message passing interface)와 같은 병렬처리 언어를 통해 쉽게 구현이 가능하다. LDA 모형의 분산알고리즘은 아직 R로 구현되어 있지 않아서 R에서는 사용이 가능하지 않다.

LDA 모형을 사용하기 위해서는 사전에 토픽의 개수 K를 정해주어야 한다. 토픽의 개수는 고객들의 관심사의 종류이기 때문에 미리 정하는 것은 쉽지 않다. AIC (Akaike information criterion)와 BIC(Bayesian information criterion) 같은 모형 선택 기준을 사용하여 K를 정할 수 있지만 이는 많은 양의 계산을 필요로 한다. 최근에는 토픽 모형 분야에서 비모수적인 방법을 사용하여 K를 추론하는 방법들이 많이 연구되고 있다. 마이크로 세그멘테이션에서도 비모수적인 방법을 도입한다면 모형 선택 필요 없이 자료에 알맞은 K를 추론할 수 있다. 이는 추후 연구에서 진행할 것이다.

참고문헌

박창이, 김용대, 김진석, 송종우, 최호식 (2013). 〈R을 이용한 데이터마이닝〉, 교우사.

이서구 (2015). 빅데이터 분석에 관한 마케팅적 접근, 〈대한경영학회지〉, 28(1), 21-35.

Blei, D. M., Ng, A. Y. and Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, 3, 993-1022.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6), 391.

Funk, Daniel C. (2002). Consumer-based Marketing: The use of micro-segmentation strategies for understanding sport consumption, *International Journal of Sports Marketing and Sponsorship*, 4(3), 39-64.

Griffiths, T. L., and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1), 5228-5235.

Hofmann, T. (1999). Probabilistic latent semantic indexing. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 50-57.

IBM (2010). macys.com: Focusing on each customer as the brand goes national (white paper).

Micro-Segmentation Strategy for Big Data Analytics Using a Topic Model

Yongdai Kim[1] · Kuhwan Jeong[2]

Abstract

One of the key techniques for data base marketing is to divide customers into several segments and to apply different marketing actions to different segments. Due to recent advent of big data, personalized marketing by making large number of segments has been developed and applied. However, it is known that standard clustering algorithms do not work well when the number of segments required is very large. In this paper, we introduce an efficient method for micro-segmentation, which makes several thousands segments, based on the topic model. Originally, the topic model has been developed to classify a large set of documents based on term-frequency data. In this paper, we explain how the topic model can be used for the micro-segmentation of customers based on behavioral histories and illustrate its superiority by analyzing a real data set using R-package.

Key words : topic model, customer segmentation, micro-segmentation

1) Professor, Dept. of Statistics, Seoul National University, 1, Gwanak-ro, Gwanak-gu, Seoul, 08826, Korea. E-mail: ydkim0903@gmail.com

2) Graduate Student, Dept. of Statistics, Seoul National University, 1, Gwanak-ro, Gwanak-gu, Seoul, 08826, Korea. kuhemian@gmail.com