



### 저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

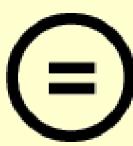
다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원 저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리와 책임은 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)



碩士學位論文

MH와 SAMC를 이용한  
LDA 토픽모형 추론



高麗大學校 大學院

應用統計學科

南 昭 姬

2018年 12月

全 秀 榮 教 授 指 導  
碩 士 學 位 論 文

MH와 SAMC를 이용한  
LDA 토픽모형 추론

이 論文을 統計學 碩士學位 論文으로  
提出함.

2018年 12月

高 麗 大 學 校 大 學 院  
應 用 統 計 學 科

南 昭 姬 (印)



南 昭 姬의 統計學 碩士學位論文  
審查를 完了함.

2018年 12月

委員長 \_\_\_\_\_ 전 수 영



委 員 \_\_\_\_\_ 홍 승 만



委 員 \_\_\_\_\_ 임 성 수



## 요 약

LDA 토픽모형은 수많은 문서집합 내의 토픽을 추출하는 통계적 모형이다. 이 모형은 하나의 문서를 여러 단어들의 집합으로 간주하며, 문서에서 높은 빈도로 동시에 발생하는 단어집합을 추출해주는 클러스터링(clustering) 기법의 일종이다. 문서집합 내 단어들을 통해서 해당 토픽이 어떤 이슈인지 파악할 수 있으며, 토픽들이 문서상에서 어떤 분포를 가지는지 계산할 수 있다. 높은 성능과 편의성으로 토픽모형 분야에서 표준적인 방법으로 인식되고 있지만 정확추론이 불가능해 근사추론을 수행하고 있다.

근사추론 연구는 Blei et al.(2003), Griffiths and Steyvers(2004), Yuan et al.(2015) 등이 있지만, Bag-of-Word 기반으로 각 단어의 가중치를 동등하게 보아 상대적 중요성을 고려하지 않았다. 또한, 마코브 체인 몬테카를로(Markov chain Monte Carlo) 방법을 이용하기 때문에 국소트랩(local trap)의 문제점이 존재한다.

LDA 토픽모형에서 기존의 근사추론의 문제점을 극복하고자 본 연구는 단어의 상대적 중요성을 반영한 PMI weighted Metropolis-Hastings within Gibbs(PWMH) 알고리즘과 Stochastic approximation Monte Carlo(Liang et al., 2007; SAMC) 알고리즘을 이용한 근사추론 방법을 제안하고자 한다. PWMH 알고리즘은 불용어 및 빈번하게 사용되는 단어를 제거하여 LDA 토픽모형의 성능을 향상시킨다. SAMC 알고리즘은 국소트랩의 문제점을 본질적으로 가지고 있지 않으며, 또한 에르고닉 성질을 만족하며 표본공간을 조절할 수 있는 능력을 갖추



고 있다. 본 연구에서 제안한 PWMH 알고리즘과 SAMC 알고리즘을 이용한 근사추론 방법은 기존 방법과 모의실험 및 실자료 분석을 통해 제안된 방법이 더욱 정확한 결과를 제공하는 우수성을 보여 주었다.

핵심어 : 텍스트 마이닝, 토픽모형, Latent Dirichlet allocation, Markov chain Monte Carlo, Pointwise mutual information, Stochastic approximation Monte Carlo, 국소트랩.



## 목 차

요 약	.....
목 차	.....
표 목 차	.....
제 1 장 서 론	..... 1
제 2 장 LDA 토픽모형 근사추론	..... 4
2.1 LDA	..... 4
2.2 LDA 모형에 대한 근사추론	..... 7
제 3 장 PWMH 알고리즘을 이용한 LDA 근사추론	... 21
3.1 PMI	..... 21
3.2 PWMH 알고리즘	..... 22
제 4 장 SAMC 알고리즘을 이용한 LDA 근사추론	..... 24
제 5 장 모의실험 및 실증분석	..... 28
5.1 토픽모형 성능평가 방법	..... 28
5.2 PWMH 알고리즘 실증분석	..... 30
5.3 SAMC 알고리즘 모의실험	..... 33
제 6 장 결 론	..... 36
참 고 문 헌	..... 37



## 표 목 차

<표 1>	30
<표 2>	31
<표 3>	31
<표 4>	32
<표 5>	33
<표 6>	34
<표 7>	34
<표 8>	34
<표 9>	34
<그림 1>	5
<그림 2>	6
<그림 3>	8
<그림 4>	18



# 제 1 장 서 론

빅데이터란 디지털 환경에서 생성되는 데이터로 그 규모가 방대하고 생성주기가 짧으며, 수치 데이터뿐만 아니라 문자와 영상 데이터까지 모두 포함한다. 2012년 세계경제포럼에서 떠오르는 10대 기술 중 하나로 빅데이터를 언급한 이후에 전 산업에 걸쳐 빅데이터에 대한 관심이 꾸준히 증가하였고, 이에 따라 국내뿐만 아니라 전 세계적으로 빅데이터 시장규모가 점차 확대되고 있는 추세다. 이로 인해서 빅데이터는 21세기의 원유라는 말까지 생겨났다. 원유를 어떻게 활용하느냐에 따라 만들어낼 수 있는 제품의 종류가 무궁무진하듯, 빅데이터 역시 데이터를 어떻게 활용하느냐에 따라 이전에는 미처 발견하지 못했던 수많은 새로운 가치들을 창조해낼 수 있기 때문이다.

이러한 빅데이터 시대에 접어들면서 기존 소셜네트워크(social network) 서비스로 대표되는 소셜미디어의 성장과 스마트폰으로 대변되는 모바일 장치의 확산이 결합되어 일상 속에서 대규모 텍스트 데이터가 급속히 생성, 유통, 저장되고 있다. 이로 인해 텍스트 마이닝 기법이 각광받고 있다.

텍스트 마이닝(text mining)은 비정형 텍스트 데이터에서 자연어처리(natural language processing)에 기반을 두어 유용한 정보를 추출, 가공하는 것을 목적으로 하는 기술이다. 텍스트 마이닝 기술을 통하여 방대한 양의 텍스트 뭉치(corpus)에서 의미있는 정보추출, 연계성 파악, 분류(classification), 군집화(clustering), 요약(summary)하기 등 단순한 정보 이상의 결과를 얻을 수 있다. 하지만, 텍스트 데이터는 인간이 직접 처리할 수 있는 양을 크게 넘어섰고 온라인 미디어의 확산에 따라 디지털화가 빠른 속도로 진행되고 있다. 이로 인해 데이터를 쉽게 사용할 수 있게 되었으나 이 대량의 데이터들 중에서 우리가 원하는 정보를 찾아내는 것은 더욱 어려워졌다. 따라서 자동적으로 비정형 텍스트의 집합을 이해하기 쉽게 만들어주고 의미있는 정보를 추출해줄 수 있는 도구가 필요한데, 이러한 역할을 해주는 도구가 바로 토픽모형(topic model)이다.

기계학습(machine learning) 및 자연어처리 과정에서 토픽모형이란 비정형 텍-



스트 집합의 추상적인 주제(topic)를 발견하기 위한 통계적 모형이다(Blei and McAuliffe, 2010). 기존 개별 단어 빈도수 분석에서 발생하는 희소성(sparsity), 동음이의어(homonym) 등의 문제점을 해결할 수 있으며, 음성, 이미지 등 다양한 종류의 데이터에도 적용 가능하다(Blei, 2012).

일반적으로 텍스트를 표현하는 방법은 단어의 순서 및 의미를 고려하지 않고 빈도를 이용하여 나타내는 것이다. 실제로 대부분의 텍스트 마이닝 관련 연구는 Bag-of-Word를 기반으로 한다. 가장 기본적인 방법론인 TF-IDF(term frequency-inverse document frequency; Salton and McGill, 1983)는 어떤 단어가 해당 문서에서 자주 사용되지만 다른 주제의 문서집합에서는 출현빈도가 낮은 값을 표현하는 것으로, 문서에서 단어의 중요도를 평가하는 방법이다(You et al., 2015). 그러나 이 알고리즘은 문서와 단어가 늘어나면 많은 시간이 소요되고 희소성의 문제가 발생하는 한계점이 있다. 이를 해결하기 위해 Deerwester et al.(1990)와 Papadimitriou et al.(2000)은 TF-IDF를 차원축소기법인 특잇값 분해(single value decomposition, SVD)를 이용한 잠재의미분석(latent semantic indexing, LSI)을 제안했다. 이 알고리즘은 최초의 토픽모형으로 여겨지는데, 문서-단어 행렬을 문서-주제 행렬과 주제-단어 행렬로 분해하는 과정을 통해 잠재 변수인 주제를 발견하고자 했다. 문맥 간의 내재적인 의미(latent/hidden meaning)를 효과적으로 보존할 수 있고 문서 간 유사도 측정 등 모형의 성능향상에 도움을 줄 수 있으며 희소성의 문제를 해결할 수 있다. 그러나 확률에 기반을 둔 모형이 아니기 때문에 최대가능도 또는 베이지안(Bayesian) 접근 방법을 사용하여 데이터에 모형을 적용할 수 없다는 한계점이 있다. Hoffmann(1999)은 문서-단어 행렬에 단어의 출현빈도 대신 출현확률로 대체한 확률적 잠재의미분석(probabilistic latent semantic indexing, PLSI)을 제안했다. 그러나 문서 내 주제 분포를 제공하지 않으며 과적합(overfitting) 현상과 문서의 크기 증가에 따라 추정해야 하는 모수의 개수가 선형적으로 늘어나는 한계점이 있다. Blei et al. (2003)는 PLSI의 한계점을 보완하는 잠재 디리클레 할당(latent Dirichlet allocation, LDA)을 제안했다. 가장 대표적으로 활용되고 있는 LDA 모형은 주제들이



디리클레 분포(Dirichlet distribution)를 따른다는 가정 하에 어떤 주제에 대해 단어들이 포함될 확률을 모델링하는 것으로 높은 성능과 편의성으로 인해 토픽 모델링 분야에서 표준적인 방법론으로 인식되고 있다. 그러나 이론적으로 정확 추론이 불가능하므로 원래 모델의 근사 형태로 접근해야 한다. LDA 모형의 대표적인 근사추론 방법은 variational expectation–maximization inference(VEM) 알고리즘(Blei et al., 2003), collapsed gibbs(C-Gibbs) 알고리즘(Griffiths and Steyvers, 2004), Metropolis–Hastings within Gibbs(MH within Gibbs) 알고리즘(Yuan et al. 2015)이 있다.

본 연구에서는 기존의 VEM과 C-Gibbs, MH within Gibbs 알고리즘 외에 새롭게 제안된 두 가지 알고리즘을 이용해 LDA 근사추론(approximate inference)을 수행하고자 한다. 제 2장에서는 LDA 모형에 대한 근사추론 관련 연구에 대해 알아본다. 선행연구인 Blei et al.(2003)에 의해 제안된 VEM 알고리즘과 Griffiths and Steyvers(2004)에 의해 제안된 C-Gibbs 알고리즘, Yuan et al.(2015)에 의해 제안된 MH within Gibbs 알고리즘을 소개한다. 제 3장에서는 PMI weighted Metropolis–Hastings within Gibbs(PWMH) 알고리즘을 소개하고, 이를 이용한 LDA 모형에서의 근사추론을 알아본다. 제 4장에서는 Stochastic approximation Monte Carlo(SAMC) 알고리즘(Liang et al., 2007)을 소개하고, 이를 이용한 LDA 모형에서의 근사추론을 알아본다. 제 5장에서는 모의실험 자료 및 실 자료를 바탕으로 LDA 모형 성능을 기존의 방법들과 비교한다.



## 제 2 장 LDA 토픽모형 근사추론

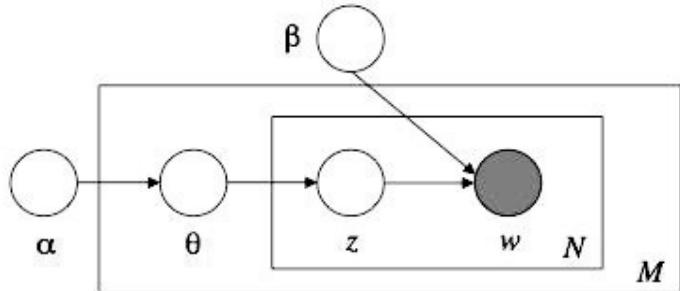
### 2.1 LDA(latent Dirichlet allocation)

LDA는 비지도 학습(unsupervised learning) 방식으로 각 문서가 어떠한 주제를 가지는지 모르는 상태에서 유일하게 관찰된 데이터인 단어들의 패턴만으로 학습이 이루어진다(Sung, 2014). 이러한 학습을 통해 문서 내에 숨겨진 주제들을 찾아내는 알고리즘으로 문서집합에 대한 확률적 생성모형(probabilistic generative model)이기도 하다. 확률적 생성 모형이란 어떤 확률분포와 파라미터가 존재할 때, 그로부터 랜덤 프로세스(random process)에 따라 데이터를 생성하는 관점의 모형을 의미한다. 즉, LDA에서 확률적 생성모형이란 실제 문서를 작성하는 과정으로 보고 문서를 작성하기 위해 각 문서에 어떤 주제를 포함시킬 것인지, 또 그에 따라 어떤 단어들을 어떤 주제에서 선택하여 배치할 것인지 각각의 파라미터로 모델링하는 것을 의미한다(Park and Song, 2013). 따라서 문서, 단어 등 관측된 변수를 통해 문서의 구조나 주제 같은 잠재변수(hidden variable)를 추론하는 것을 목적으로 하며 결과적으로 전체 문서집합의 주제들과 각 문서별 주제분포, 각 주제에 포함된 단어들의 분포를 알아낼 수 있다(Blei et al., 2003).

LDA는 문서가 여러 개의 주제를 가지고 있고, 각각의 주제는 디리클레분포(Dirichlet distribution)를 따른다고 가정한다.  $M$ 개의 문서가 주어지고, 모든 문서는 각각  $K$ 개의 주제 중 하나에 속할 때, 단어는 이산데이터(discrete data)의 기본 단위로 단어집(vocabulary)의 인덱스(index)로 나타낼 수 있다. 단어집의 크기를  $V$ 라 하면, 각각의 단어는 인덱스  $v = \{1, \dots, V\}$ 로 대응된다. 단어 벡터(vector)  $w$ 는  $V$ 벡터로 표기하며  $w^v = 1$ ,  $w^u = 0$ ,  $v \neq u$ 를 만족한다. 즉, 문서에  $v$  번째 단어가 있으면 1, 그렇지 않으면 0으로 표기한다. 문서는  $N$ 개의 연속된 단어로,  $W = (w_1, w_2, \dots, w_N)$ 로 표기한다. 문서 말뭉치(corpus)는  $M$ 개의 문서집합



으로,  $D = \{W_1, W_2, \dots, W_M\}$ 와 같이 표기 한다.

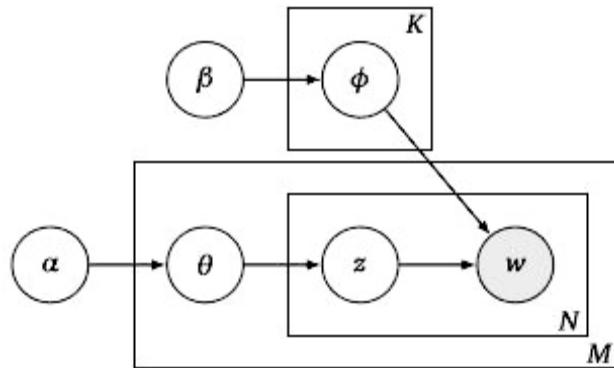


<그림 1> LDA 확률그래프 모형

LDA의 확률그래프 모형(그림 1)에서  $K$ 는 주제의 개수를 의미하고 알고 있는 고정된 값이라고 가정한다.  $M$ 은 문서의 개수를 의미하고,  $N$ 은 해당 문서의 전체 단어개수를 의미한다.  $\alpha$ 는 하이퍼 파라미터(hyper parameter)로 전체 문서집합에서 동일한 값을 가진다.  $\alpha$ 는  $K$ 차원의 디리클레 분포의 매개변수로 각각의 문서가 어떤 주제 비율로 구성되어 있는지 나타내는  $\theta$ 를 결정한다.  $\theta$ 는  $K$ 차원의 벡터로  $\theta_i$ 는 문서가  $i$ 번째 주제에 속할 확률을 의미한다. 즉, 문서의 주제분포를 의미하며  $\sum_{i=1}^K \theta_i = 1$ 을 만족한다.  $Z$ 는  $N$ 차원의 벡터로  $z_n$ 은 단어  $w_n$ 에 할당된 주제를 의미한다.  $\beta$ 는  $K \times V$  크기의 행렬로  $\beta_{ij}$ 는  $i$ 번째 주제가 단어 집의  $j$ 번째 단어를 생성할 확률을 의미한다. 여기서  $w_n$ 은 실제 문서를 통해서 주어지며, 다른 변수는 관측할 수 없는 잠재변수이다.



다음으로 <그림 2>는 smoothed LDA 확률그래프 모형이다.



<그림 2> smoothed LDA 확률 그래프 모형

LDA의 확률그래프 모형(그림 1)과 비슷하나 여기서  $\beta$ 는 하이퍼 파라미터(hyper parameter)로 전체 문서집합에서 동일한 값을 가진다.  $\beta$ 는  $K$ 차원의 디리를 레 분포의 매개변수로 각각의 단어가 어떤 주제 비율로 구성되어 있는지 나타내는  $\phi$ 를 결정한다.  $\phi$ 는  $K$ 차원의 벡터로  $\phi_i$ 는 단어가  $i$ 번째 주제에 속할 확률을 의미한다. 즉, 단어의 주제 분포를 의미하며  $\sum_{i=1}^K \phi_i = 1$ 을 만족한다.

LDA는 각각의 문서  $w \in D$  가 다음과 같은 과정을 거쳐 생성된다고 가정한다.  $\xi$ 는 포아송 분포의 매개변수로  $N$ 은 문서의 길이를 의미한다.

#### LDA의 문서 생성과정(Blei et al., 2003):

(단계 1)  $\xi (= 1, 2, \dots)$ 에 대하여  $Poisson(\xi)$  분포로부터  $N$ 을 선택한다.

(단계 2)  $\alpha (= 1, \dots, K)$ 에 대하여  $Dirichlet(\alpha)$  분포로부터  $\theta$ 를 선택한다.

(단계 3) 문서 내의 단어  $w_n \in W$ 에 대해서,



- (a)  $\theta (= 1, \dots, K)$ 에 대하여  $Multinomial(\theta)$  분포로부터  $z_n$ 을 선택한다.
- (b)  $z_n$ 이 주어졌을 때  $w_n$ 은  $p(w_n | z_n, \beta)$ 로부터 선택한다.

따라서 LDA 모형은 다음과 같이 해석될 수 있다. 각 문서에 대해  $K$ 개의 주제에 대한 가중치  $\theta$ 가 존재한다. 문서 내의 각 단어  $w_n$ 은  $K$ 개의 주제 중 하나를 가지는데,  $z_n$ 은  $\theta$ 에 의한 다항분포(multinomial distribution)로 선택된다. 마지막으로 실제 문서를 통해서 주어지는 단어  $w_n$ 은  $z_n$ 에 기반을 두어 선택된다.

## 2.2 LDA 모형에 대한 근사추론

LDA를 사용하기 위해서는 문서가 주어졌을 때 잠재변수에 대한 사후분포(po sterior distribution)를 구할 수 있어야 한다. <그림 1>의 LDA 확률그래프 모형에서 확인할 수 있듯이 각 문서에 대한 사후분포는 다음과 같다.

$$p(\theta, Z | W, \alpha, \beta) = \frac{p(\theta, Z, W | \alpha, \beta)}{p(W | \alpha, \beta)} \quad (1)$$

사후분포를 구하기 위해서는 결합분포(joint distribution)  $p(\theta, Z | W, \alpha, \beta)$ 를 잠재변수  $\theta, Z$ 에 대해서 주변화(marginalize)한  $p(W | \alpha, \beta)$ 를 계산해야 한다. 그러나  $p(W | \alpha, \beta)$ 는  $\theta, \beta$ 가 서로 결합(coupling)되어 있어  $K^V$ 개의 상태공간(state space) 값에 대한 계산이 현실적으로 불가능하다(Dickey, 1983).

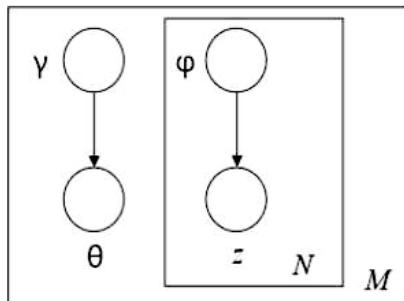
$$p(W | \alpha, \beta) = \frac{\Gamma\left(\sum_{i=1}^k \alpha_i\right)}{\prod_{i=1}^k \Gamma(\alpha_i)} \int \left( \prod_{i=1}^k \theta_i^{\alpha_i - 1} \right) \left( \prod_{n=1}^N \sum_{i=1}^k \prod_{j=1}^V (\theta_i \beta_{ij})^{w_n^j} \right) d\theta \quad (2)$$



위 사후분포에 대한 정확추론이 어렵지만 다양한 근사추론 알고리즘이 LDA 추론을 위해 고려되었다. 대표적인 근사추론 방법으로는 variational expectation–maximization(VEM) 알고리즘, 마코브 체인 몬테카를로(Markov chain Monte Carlo, MCMC)에 기반을 둔 collapsed Gibbs(C–Gibbs) 알고리즘과 Metropolis–Hastings within Gibbs(MH within Gibbs) 알고리즘이 있다.

### 2.2.1 Variational expectation–maximization(VEM) 알고리즘

VEM 알고리즘의 아이디어는 엔센 부등식(Jensen's inequality)을 사용해 로그 가능도(log likelihood)의 적절한 하한 값(lower bound)을 계산하는 것이다. 로그 가능도의 하한 값은 변분 매개변수(variational parameter)로 나타내지며 이 매개변수는 최적화 과정을 통해 가장 좋은 하한 값을 가지도록 정해진다.



<그림 3> LDA 사후분포를 근사하기 위해 사용된 variational distribution의 확률그래프 모형

이 알고리즘을 LDA에 적용하기 위해선 LDA의 확률그래프 모형(그림 1)을 단순화시켜야 한다(그림 3).  $p(W|\alpha, \beta)$ 에서  $\theta$ 와  $\beta$ 의 결합을 제거해야 하므로 기존의 그래프 도식(그림 1)에서  $\theta, Z, W$  사이의 선(edge)과 노드(node)  $W$ 를 제거한 뒤 변분 매개변수인  $\gamma$ 와  $\varphi$ 를 추가한다.  $\gamma$ 는 디리클레 분포의 매개변수이며  $\varphi = (\varphi_1, \dots, \varphi_N)$ 는 다항분포의 매개변수로 자유 변분 매개변수(free variatio



nal parameter)이다. 변분 매개변수에 대한  $\theta$ 와  $Z$ 의 조건부 확률은 다음과 같다 (Blei et al., 2003).

$$q(\theta, Z | \gamma, \varphi) = q(\theta | \gamma) \prod_{n=1}^N q(z_n | \varphi_n) \quad (3)$$

이를 바탕으로  $q(\theta, Z | \gamma, \varphi)$ 에 대해 문서의 로그 가능성도 하한 값을 구할 수 있다(Blei et al., 2003).

$$\begin{aligned} \log p(W | \alpha, \beta) &= \log \int \sum_Z p(\theta, Z, W | \alpha, \beta) d\theta \\ &= \log \int \sum_Z \frac{p(\theta, Z, W | \alpha, \beta) q(\theta, Z)}{q(\theta, Z)} d\theta \\ &\geq \int \sum_Z q(\theta, Z) \log p(\theta, Z, W | \alpha, \beta) d\theta - \int \sum_Z q(\theta, Z) \log q(\theta, Z) d\theta \\ &= E_q[\log p(\theta, Z, W | \alpha, \beta)] - E_p[\log q(\theta, Z)] \end{aligned} \quad (4)$$

식 (4)에서 우변을  $L(\gamma, \varphi; \alpha, \beta)$ 라 표기하고 변분(variational) 분포인  $q$ 와 원래 사후분포  $p$ 의 차이를 계산하기 위해 쿨백-레이블러 발산(Kullback-Leibler Divergence, KLD)을 이용하여 최적화 과정을 유도한 뒤  $\gamma$ 와  $\varphi$ 의 수렴 값을 구할 수 있다.

$$\log p(W | \alpha, \beta) = L(\gamma, \varphi; \alpha, \beta) + D(q(\theta, Z | \gamma, \varphi) \| p(\theta, Z | W, \alpha, \beta))$$

$$(\gamma^*, \varphi^*) = \operatorname{argmin} D(q(\theta, Z | \gamma, \varphi) \| p(\theta, Z | W, \alpha, \beta)) \quad (5)$$

기댓값 최대화 알고리즘(expectation maximization, EM; Andrieu et al., 2003)을 이용해서 본래의 매개변수  $\alpha, \beta$ 와 변분 매개변수  $\gamma, \varphi$ 를 추정할 수 있다. 기댓값 단계(expectation step)에서  $\alpha, \beta$ 를 안다는 가정 하에  $L(\gamma, \varphi; \alpha, \beta)$ 를 최대



화하는  $\gamma, \varphi$ 를 구한 뒤, 최대화 단계(maximization step)에서 기댓값 단계에서 구한  $\gamma, \varphi$ 를 이용하여 다시  $L(\gamma, \varphi; \alpha, \beta)$ 를 최대화하는  $\alpha, \beta$ 를 추정한다.

$$\varphi_{ni} \propto \beta_{iw_n} \exp\{E_q[\log(\theta_i)|\gamma]\}$$

$$\gamma_i = \alpha_i + \sum_{n=1}^N \varphi_{ni}$$

$$E_q[\log(\theta_i)|\gamma] = \Psi(\gamma_i) - \Psi\left(\sum_{j=1}^k \gamma_j\right) \quad (6)$$

여기서  $\Psi$ 는 테일러 근사(taylor approximation)를 이용해 계산된 로그감마 함수의 1차 미분을 의미한다(Abramowitz and Stegun, 1970). 위 과정을 바탕으로 LDA에 대한 VEM 알고리즘을 정리하면 다음과 같다.

#### VEM 알고리즘(Blei et al., 2003):

(단계 1) 모든 토픽  $i$ 와 단어  $n$ 에 대해서  $\varphi_{ni} = 1/k$ 로 초기화한다.

(단계 2) 모든 토픽  $i$ 에 대해서  $\gamma_i = \alpha_i + N/k$ 로 초기화한다.

(단계 3)  $\gamma, \varphi$ 가 수렴할 때까지 다음의 과정을 반복한다.

(1)  $\varphi_{ni}^{t+1} = \beta_{iw_n} \exp[\Psi(\gamma_i^t)]$ 를 계산한다.

(2) (1)번 식에서 구한  $\varphi_{ni}^{t+1}$ 를 정규화한다.

(3)  $\varphi_{ni}^{t+1}$ 를 이용해  $\gamma^{t+1} = \alpha + \sum_{n=1}^N \varphi_n^{t+1}$ 를 계산한다.

VEM 알고리즘은 어느 정도 주제가 한정된 비교적 단순한 문서의 경우 높은 정확도를 보장하지만 큰 데이터를 다룰 때 메모리의 문제가 발생한다. 또한  $Z, \theta$ 와  $\varphi$  사이의 의존성(dependence)을 제거하는 완전히 분해된 변분(variationa



1) 분포의 강한 독립 가정은 음의 로그 가능도(negative log likelihood)의 불확실한 상한 값(loose upper bound)으로 인해 사후분포의 부정확한 추정을 야기할 수 있다(Teh et al., 2007).

### 2.2.2 Collapsed Gibbs(C-Gibbs) 알고리즘

깁스 샘플링(Geman and Geman, 1984)은 MCMC 알고리즘 중 하나로 표본을 직접 얻기 어려운 확률분포로부터 표본을 생성할 때 사용하는 대표적인 표본추출 알고리즘이다. 추정하고자 하는 변수 외의 나머지 변수들에 대한 완전 조건부분포(full conditional distribution)에 의존하여 번갈아가면서 표본을 생성한다.

#### Gibbs 알고리즘:

$n$ 개의 확률변수  $(X_1, \dots, X_n)$ 의 결합확률분포  $p(x_1, \dots, x_n)$ 로부터  $T$ 개의 표본  $X$ 는 다음과 같은 (1)-(2)의 반복 과정을 통해 생성된다.

먼저 임의로  $X^{(0)} = (x_1^{(0)}, \dots, x_n^{(0)})$ 를 선택한다.  $t = 1, \dots, T$ 에 대하여,

- (1) 각 변수  $x_1^{(t-1)}, \dots, x_n^{(t-1)}$ 에 대하여 현재의 값을 기반으로 한 조건부 확률 분포  $p(x_i^{(t-1)} | x_1^{(t)}, \dots, x_{i-1}^{(t)}, x_{i+1}^{(t-1)}, \dots, x_n^{(t-1)})$ 에서 새로운 표본  $x_i^{(t)}$ 를 생성한다.
- (2) (1)을 통해  $X^{(t)} = (x_1^{(t)}, \dots, x_n^{(t)})$ 를 생성한다.

실제 사용 시 처음 생성되는 표본은 사용하지 않는다. 이는 깁스 알고리즘에서 생성되는 표본은 서로 독립적이지 않고 마코프 연쇄에 속하기 때문이다. 표본의 앞부분은 초기상태  $X^{(0)}$ 에 크게 의존하지만 충분한 시간이 지난 후에는 초기상태에 관계없이 확률분포에 기반을 둔 표본을 생성할 수 있다.



LDA 모형에서 나머지 변수는 고정시킨 채 한 변수만을 변화시키면서 불필요한 변수  $\theta, \phi$ 를 제외한 C-Gibbs 알고리즘을 사용하여 추론한다. 모든 잠재변수를 고려하여 사후분포를 결정하기 위해서는 많은 계산이 필요해 가장 관심 있는 주제  $Z$ 에 대한 사후분포만 고려한다.  $Z$ 에 대한 사후분포는 다음과 같다.

$$\begin{aligned}
p(Z|W, \alpha, \beta) &= \frac{p(W, Z|\alpha, \beta)}{\sum_Z p(W, Z|\alpha, \beta)} \\
&= \frac{p(W|Z, \beta)p(Z|\alpha)}{\sum_Z p(W|Z, \beta)p(Z|\alpha)} \\
&\propto p(W|Z, \beta)p(Z|\alpha)
\end{aligned} \tag{7}$$

Smoothed LDA 모형을 위해 식 (7)에서 사용자가 지정한 하이퍼 파라미터  $\alpha, \beta$ , 관측치인 단어  $w_n$ 을 제외한 모든 변수는 잠재변수가 된다. 따라서 Smoothed LDA 모형에서 모든  $\theta$ 는  $\phi$ 에 대해서 독립이므로  $Z$ 에 대한 사후분포는 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
p(Z|W, \alpha, \beta) &= \int_{\theta} \int_{\phi} p(W, Z, \theta, \phi|\alpha, \beta) d\phi d\theta \\
&= \int_{\theta} \int_{\phi} \prod_{i=1}^K p(\phi_i|\beta) \prod_{d=1}^M p(\theta_d|\alpha) \prod_{n=1}^{N_d} p(z_{dn}|\theta_d) p(w_{dn}|\phi_{Z_{dn}}) d\phi d\theta \\
&= \int_{\theta} \prod_{d=1}^M p(\theta_d|\alpha) \prod_{n=1}^{N_d} p(z_{dn}|\theta_d) d\theta \times \int_{\phi} \prod_{i=1}^K p(\phi_i|\beta) \prod_{d=1}^M \prod_{n=1}^{N_d} p(w_{dn}|\phi_{Z_{dn}}) d\phi
\end{aligned} \tag{8}$$

각 문서에 대해  $N$ 개의 단어주제를 결정하는 과정을 표본추출 과정이라고 하면 미리 지정된 반복 횟수만큼 반복적으로 표본추출이 진행되면서  $Z$ 에 대한 사후분포 계산이 가능하다. C-Gibbs 알고리즘에서 표본을 추출하기 위해서는 조건부분포가 정의되어야 하며, smoothed LDA의 확률모형에서 디리클레분포와 다항분포 사이의 관계(conjugated)로부터 다음과 같이  $Z$ 에 대한 사후분포를



정의할 수 있다(Griffiths and Steyvers, 2004; Sung, 2014).

$$w_n|z_n, \phi_{z_n} \sim Multinomial(\phi_{z_n})$$

$$\phi \sim Dirichlet(\beta)$$

$$z_n|\theta_{di} \sim Multinomial(\theta_{di})$$

$$\theta \sim Dirichlet(\alpha)$$

$$p(z_n = k | z_{-n}, W) \approx \frac{n_{k,-n}^{(v)} + \beta_v}{\sum_{v=1}^V n_{k,-n}^{(v)} + \beta_v} \times \frac{n_{d,-n}^{(k)} + \alpha_k}{\sum_{k=1}^K n_{d,-n}^{(k)} + \alpha_k} \quad (9)$$

여기서  $\alpha, \beta$ 는 디리클레 분포의 매개변수이다.  $n_{k,-n}^{(v)}$ 는 각 문서에서  $n$ 번째에 등장하는 단어  $w_n$ 를 제외한 나머지 단어들 중  $v$ 번째 단어가  $k$ 번째 주제로 관측되는 빈도수를 의미하며,  $n_{d,-n}^{(k)}$ 는 문서  $d$ 의  $n$ 번째 단어를 제외하고 해당 문서에서 주제가  $k$ 인 단어의 빈도수를 의미한다. 특히  $n_k^{(v)}$ 는 단어-주제 테이블(word-topic table)이라고 알려져 있고  $\phi_{z_n}$ 의 충분통계량(sufficient statistics) 역할을 하며,  $n_d^{(k)}$ 는 문서-주제 테이블(doc-topic table)로  $\theta_{di}$ 의 충분통계량 역할을 한다.

추가적으로 C-Gibbs 알고리즘에서 얻은 표본을 이용하게 되면 샘플링 과정에서 다루지 않았던 확률변수인  $\theta$ 와  $\phi$ 를 추정할 수 있다.

$$\phi_{k,v} \approx \frac{n_k^{(v)} + \beta_v}{\sum_{v=1}^V n_k^{(v)} + \beta_v}, \quad \theta_{d,k} \approx \frac{n_d^{(k)} + \alpha_k}{\sum_{k=1}^K n_d^{(k)} + \alpha_k} \quad (10)$$



위 과정을 바탕으로 LDA에 대한 C-Gibbs 알고리즘을 정리하면 다음과 같다.

#### LDA C-Gibbs 알고리즘(Griffiths and Steyvers, 2004):

문서집합의 각 문서에 대해서,

(단계 1) 문서집합에 있는 모든 단어에 임의로 토픽을 할당한다.

(단계 2)  $n$ 번째 단어를 제외하고 그 단어가 속한 문서별 주제분포와 주제별 단어분포를 계산한다.

(단계 3) [샘플링]

$$\pi_k = p(z_n = k | rest) = \frac{n_{k,-n}^{(v)} + \beta_v}{\sum_{v=1}^V n_{k,-n}^{(v)} + \beta_v} \times \frac{n_{d,-n}^{(k)} + \alpha_k}{\sum_{k=1}^K n_{d,-n}^{(k)} + \alpha_k} \text{로부터}$$

$\pi (= \pi_1, \dots, \pi_K)$ 를 구하고, Multinomial( $\pi$ ) 분포로부터  $z_{t+1}$ 을 선택한다.

(단계 4) 수렴할 때까지 위 단계 2-3을 반복한다.

일반적으로 C-Gibbs 알고리즘은 최소한의 메모리를 사용하여 구현하기 때문에 VEM 알고리즈다 보다 빠른 처리속도를 보여 속도 면에서 우수함을 보인다. 또한, 구현이 쉬워 다양한 분야에서 많이 사용되고 있으며(Geigle, 2016), 다양한 많은 주제를 가지고 있는 일반적인 문서에서의 추론은 C-Gibbs 알고리즘을 이용하는 것이 더 큰 정확도를 보인다(Sung, 2014). 그러나 C-Gibbs 알고리즘은 모든 문서에 대해서 문서-주제 테이블과 단어-주제 테이블을 계산해야 한다. 순차적으로 모든 문서 내 단어들의 주제를 표본추출한다고 가정하면, 문서에 존재하는 단어들의 단어-주제 테이블의 모든 행을 살펴보아야 한다. 따라서 C-Gibbs 알고리즘은 대량의 문서를 처리할 때 메모리 문제가 발생하여 많은 시간이 걸리는 한계점이 있다. 이를 보완하기 위해서 Metropolis-Hastings within Gibbs 알고리즘을 이용한 LDA 모형의 근사추론 방법이 제안되었다(Li et al., 2013).



al., 2014; Yuan et al., 2015).

### 2.2.3 Metropolis–Hastings within Gibbs(MH within Gibbs) 알고리즘

Metropolis–Hastings(Metropolis et al., 1953; Hastings, 1970; MH) 알고리즘은 표본을 직접 얻기 어려운 확률분포로부터 표본을 생성하는데 사용하는 대표적인 표본추출 알고리즘이다. 확률밀도함수의 적분 값을 1로 만드는 정규화 상수를 구하기 힘든 경우 활용할 수 있는 장점이 있다. MH 알고리즘은 다음의 과정으로 이루어진다.  $y$ 는 주어진 관측 값이고,  $f(y)$ 는  $y$ 의 확률분포이다.  $y^*$ 는 새롭게 생성되는  $y$ 의 값이다.

#### MH 알고리즘:

현재 표본을  $y_t$  ( $t = 1, 2, \dots, n$ )라고 할 때, 다음 표본  $y_{t+1}$ 은 다음과 같은 과정을 통해 생성된다.

- (1) 적절한 제안(proposal)분포  $q(y_t, y^*)$ 로부터 새로운 표본  $y^*$  값을 생성한다.
- (2) 채택확률을 다음과 같이 구한다.

$$\alpha_{y_t, y^*} = \min\left\{1, \frac{f(y^*)q(y_t, y^*)}{f(y_t)q(y_t, y^*)}\right\}.$$

- (3) 채택확률  $\alpha_{y_t, y^*}$ 의 확률로  $Y_{t+1} = y^*$ 을 채택하고,  $1 - \alpha_{y_t, y^*}$ 의 확률로  $Y_{t+1} = y_t$ 을 채택한다.

깁스 알고리즘을 구현하기 위해서는 모든 관심 있는 매개변수의 완전 조건부 분포를 사용할 수 있어야 한다. 그러나 실제로 완전 조건부분포를 구하는 것이



어려워 MH 알고리즘을 통합하여 완전 조건부 확률을 구하기 어려운 매개 변수에서 표본을 추출한다. MCMC 알고리즘의 이러한 변형 방법을 Metropolis-Hastings within Gibbs(MH within Gibbs) 알고리즘이라고 한다(Muller, 1993). MH within Gibbs 알고리즘은 다음의 과정으로 이루어진다.

#### MH within Gibbs 알고리즘:

현재 표본을  $y_t (t=1,2,\dots,n)$ 라고 할 때, 다음 표본  $y_{t+1}$ 은 다음과 같은 과정을 통해 생성된다.

(1) Gibbs 알고리즘을 이용한 완전 조건부 제안(proposal)분포  $q(y_t, y^*)$ 로부터 새로운 표본  $y^*$  을 생성한다.  $y^* \sim q(y_i^{(t)} | y_1^{(t+1)}, \dots, y_{i-1}^{(t+1)}, y_{i+1}^{(t)}, \dots, y_n^{(t)})$

(2) 채택 확률을 다음과 같이 구한다.

$$\alpha_{y_t, y^*} = \min \left\{ 1, \frac{f(y^*) q(y_t, y^*)}{f(y_t) q(y_t, y^*)} \right\}.$$

(3) 채택 확률  $\alpha_{y_t, y^*}$ 의 확률로  $Y_{t+1} = y^*$  을 채택하고,  $1 - \alpha_{y_t, y^*}$ 의 확률로  $Y_{t+1} = y_t$  을 채택한다.

LDA 모형에서 MH within Gibbs 알고리즘을 이용할 경우 높은 혼합비율(mixing rate)을 가지고 빠른 속도로 표본을 추출하기 위해서 인수분해 정리를 이용한다. 이를 수행하기 위해 MH 알고리즘의 제안분포(proposal distribution)를 고려해보자. 제안분포를 구성하기 위해서는 식 (9)로부터 시작한다.



Yuan et al.(2015)는 두 개의 항으로 분리되는 식 (9)에서  $Z$ 에 대한 조건부 확률분포를 이용하여 새로운 표본을 생성하는 두 가지 제안분포를 제시하였다.

$$q(z_n = k | rest) \propto \frac{n_k^{(v)} + \beta_v}{\sum_{v=1}^V n_k^{(v)} + \beta_v} \times (n_d^{(k)} + \alpha_k) \quad (11)$$

$$n_d^{(k)} + \alpha_k : doc-proposal$$

$$\frac{n_k^{(v)} + \beta_v}{\sum_{v=1}^V n_k^{(v)} + \beta_v} : word-proposal$$

여기서  $\alpha, \beta$ 는 디리클레 분포의 하이퍼 파라미터로 전체 문서집합에서 동일한 값을 가진다.  $n_k^{(v)}$ 는 단어-주제 테이블로 전체 문서집합에서  $k$ 번째 토픽에 할당된 단어 빈도를 의미하며,  $\phi_{z_n}$ 의 충분통계량이다.  $n_d^{(k)}$ 는 문서-주제 테이블로  $k$ 번째 토픽에 할당된  $d$ 번째 문서의 단어 빈도를 의미하며,  $\theta_{di}$ 의 충분 통계량이다.

단어 제안분포(word-proposal distribution)은 단어에는 의존하지만 문서에는 독립적이고, 문서 제안분포(doc-proposal distribution)은 문서에는 의존하지만 단어에는 독립적이다. 또한 단어의 가장 가능성 있는 주제는 단어에 의존하는 항과 문서에 의존하는 항에서 모두 높은 확률을 가진다는 것을 직관적으로 알 수 있다. 그러므로 두 개의 항 중 어느 하나 항만으로도 좋은 제안분포가 될 수 있다. 단어에 대한 주제를 정할 때  $p(z_n = k | rest)$ 가 높은 확률을 가지는 주제  $k$ 를 선택하기 때문에 어느 쪽의 항도 또한  $k$ 에 대해서 높은 확률을 가지기 때문이다.



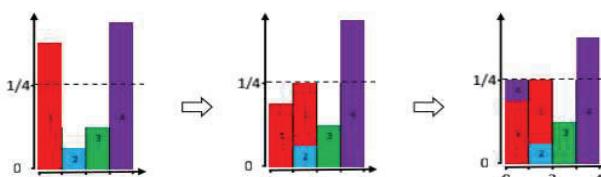
두 가지의 제안분포에 대해서 개별적으로 살펴보도록 하자. 첫 번째로 단어 제안분포이다.

$$p_w(k) \propto \frac{n_k^{(v)} + \beta_v}{\sum_{v=1}^V n_k^{(v)} + \beta_v} \quad (12)$$

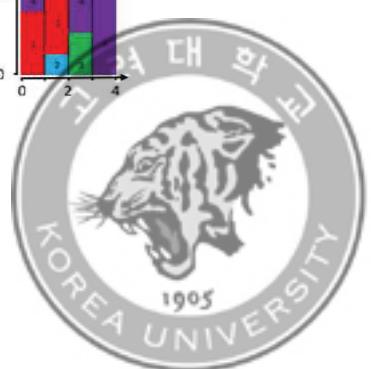
MH within Gibbs 알고리즘에서 단어 제안분포로부터 현재 표본이  $s$ 일 때, 새로운 표본  $t$ 가 채택될 확률은 다음과 같다.

$$\min \left\{ 1, \frac{p(t)p_w(s)}{p(s)p_w(t)} \right\} \quad (13)$$

단어 제안분포는 새로운 표본  $t$ 를 생성할 때 별칭 테이블(alias table)을 이용한다. 별칭 테이블은 빠르게 이산형 변수를 추출해주는 것으로 <그림 4>와 같이 비 균일 분포(non-uniform distribution)를 균일분포(uniform distribution)로 변환해준다(Yuan et al., 2015; Marsaglia et al., 2004). 이로 인해 단어들의 단어-주제 테이블의 모든 행을 살펴보지 않고 균일분포로부터 빠르게 새로운 표본을 추출하게 만들어준다. Alias table을 이용해서 추출된 주제  $k$ 는 문서 내에서 자주 나오는 주제이거나  $v$ 번째 단어가 주제  $k$ 에 가장 많이 할당되는 단어일 것이다.



<그림 4> Alias table



두 번째로 문서 제안분포이다.

$$p_d(k) \propto n_d^{(k)} + \alpha_k \quad (14)$$

MH within Gibbs 알고리즘에서 문서 제안분포로부터 현재 표본이  $s$ 일 때, 새로운 표본  $t$ 가 채택될 확률은 다음과 같다.

$$\min\left\{1, \frac{p(t)p_d(s)}{p(s)p_d(t)}\right\} \quad (15)$$

문서 제안분포는 새로운 표본  $t$ 를 생성할 때 단어 제안분포와 달리 별칭 테이블(alias table)을 이용할 필요가 없다.  $p_d(k)$ 의 첫 번째 항이  $d$ 번째 문서에서  $k$ 번째 주제에 할당된 횟수를 의미하므로 별칭 테이블 역할을 하기 때문이다.

$$n_d^{(k)} = \sum_{i=1}^{n_d} [z_{di} = k] \quad (16)$$

따라서 문서 제안분포에서는 균일분포를 이용하여 표본을 생성한다.

단어 제안분포나 문서 제안분포만으로 LDA를 위한 효율적인 MH 알고리즘이 사용될 수 있지만 실제로 적절한 혼합(mixing)을 위해서는 많은 반복이 필요하다. 빠른 혼합을 위해선 신속하게 모든 상태공간(state space)을 탐색해야 하고 높은 확률 값을 가져야하는데 한가지의 제안분포만 사용하게 되면 효과가 좋지 않기 때문이다. Yuan et al.(2015)은 높은 샘플링 효과를 유지하면서 적절한 혼합을 위하여 단어 제안분포와 문서 제안분포를 교대로 사용하는 순환 제안분포(cycle proposal distribution)을 제시하였다.



$$p_c(k) \propto p_d(k) \times p_w(k) \quad (17)$$

Tierney(1994)는 순환 MH 제안분포가 이론적으로 수렴한다는 것을 보여 주었다. 두 가지 제안분포를 결합함으로써  $p(k)$ 의 최빈값(mode)이 적어도 하나의 제안분포에 의해 충분히 높은 확률로 제안될 것이다. 또한 서로 다른 제안 분포를 혼합하는 방법의 또 다른 이점은 샘플링된 상태 사이에 자기 상관(auto-correlation)을 줄여 상태 공간을 더 빠르게 탐색하는데 도움이 된다는 것이다.

LDA에 대한 MH within Gibbs 알고리즘을 정리하면 다음과 같다.

#### LDA MH within Gibbs 알고리즘(Yuan et al., 2015):

문서집합의 각 문서에 대해서,

(단계 1) 문서집합에 있는 모든 단어에 임의로 토픽을 할당한다.

(단계 2) [샘플링] 모든 단어에 대해서,

(1) 두 가지 제안분포 중 하나를 랜덤하게 선택한다.

- 단어 제안분포
- 문서 제안분포

(2) 선택된 제안분포로부터 새로운 표본  $z^*$ 을 추출한다.

(3) 채택확률을 다음과 같이 구한다.

$$\alpha_{z_t z^*} = \min\left\{1, \frac{p(z^*)q(z^*, z_t)}{p(z_t)q(z_t, z^*)}\right\}$$

(4) 채택확률  $\alpha_{z_t z^*}$ 의 확률로  $z_{t+1} = z^*$ 을 채택하고,

$1 - \alpha_{z_t z^*}$ 의 확률로  $z_{t+1} = z_t$ 을 채택한다.

(단계 3) 수렴할 때까지 위 단계를 반복한다.



## 제 3 장 PWMH 알고리즘을 이용한 LDA 근사추론

본 장에서는 불용어나 빈번하게 사용되는 단어 제거를 위해 가중치를 적용한 개선된 MH within Gibbs 알고리즘을 제안한다. 기존 LDA 근사추론 방법인 V EM, C-Gibbs, MH within Gibbs 알고리즘은 모두 Bag-of-word 기반으로 각 단어의 가중치를 동등하게 보았다. 토픽모형의 성능을 높이기 위해서는 불용어를 잘 선정하는 것도 중요하다. 전처리 과정에서 정확하게 불용어를 제거하지 않은 경우도 있기에 한 번 더 단어의 상대적 중요성을 반영해 이러한 문제를 해결하고자 했다. 본 논문에서는 가중치로 점별 상호 정보량(pointwise mutual information, PMI)을 사용하였다.

### 3.1 PMI

문서별로 가중치가 동일해야 할 필요는 없다. 같은 단어라도 문서에 따라 다른 가중치를 부여하는 것이 상황에 따라 좋을 수 있다. 예를 들어 대부분의 문서에서 ‘the’라는 단어가 흔히 쓰이기 때문에 불용어처럼 보일 수 있을지라도 종종 ‘the’라는 단어에 관한 이야기를 하는 문서가 있을 수도 있다. 이 문서에선 ‘the’가 불용어가 아니라 주제어가 되는 것이다. 이런 경우를 고려하기 위해 본 논문에서는 점별 상호정보량(PMI)을 사용했다. PMI는 문서별 단어의 가중치로, 문서의 출현확률과 단어의 출현확률의 PMI를 계산해 이 둘의 관계성을 고려한 방법이다.

$$PMI = m(w) = \log \frac{p(w_n|d)}{p(w_n)} \quad (18)$$

여기서  $p(w_n|d)$ 는  $d$ 번째 문서에서  $n$ 번째 단어가 나올 확률을 의미하며,  $p(w_n)$ 는 전체 문서집합에서  $n$ 번째 단어가 나올 확률을 의미한다. 특정 문서에서 특



정 단어가 집중적으로 분포할 경우 그 문서와 단어의 PMI는 높아진다. 반면, 문서와 단어 사이의 상관관계가 없으면 PMI는 0이 되고, 특정 문서가 특정 단어를 피한다면 음의 PMI 값이 나오게 된다. 본 논문에서는 음의 PMI는 모두 0으로 대체하여 사용하였다.

### 3.2 PMI weighted MH within Gibbs(PWMH) 알고리즘

PWMH 알고리즘은 불용어 및 빈번하게 사용되는 단어 제거를 목적으로 기존의 MH within Gibbs 알고리즘에 PMI 가중치를 추가한 개선된 방법이다. PMI 가중치를 적용한 LDA 토픽모형 사후분포는 다음과 같다.

$$p(k) = p(z_n = k | rest) \propto \frac{(n_k^{(v)} \times m(v)) + \beta_v}{\sum_{v=1}^V n_k^{(v)} + \beta_v} \times \frac{n_d^{(k)} + \alpha_k}{\sum_{k=1}^K n_d^{(k)} + \alpha_k} \quad (19)$$

여기서  $\alpha, \beta$ 는 디리클레 분포의 하이퍼 파라미터로 전체 문서집합에서 동일한 값을 가진다.  $k$ 는 단어에 할당된 주제,  $m(v)$ 는 PMI 가중치를 의미한다.  $n_{k,-n}^{(v)}$ 는 각 문서에서  $n$ 번째에 등장하는 단어  $w_n$ 를 제외한 나머지 단어들 중  $v$ 번째 단어가  $k$ 번째 주제로 판측되는 빈도수를 의미하며,  $n_{d,-n}^{(k)}$ 는 문서  $d$ 의  $n$ 번째 단어를 제외하고 해당 문서에서 주제가  $k$ 인 단어의 빈도수를 의미한다.

Yuan et al.(2015)이 제안한 것처럼 두 개의 항으로 분리되는  $Z$ 에 대한 조건부 확률분포를 이용하여 새로운 표본을 생성하는 두 가지 제안분포를 사용하였다. 하지만 PWMH 알고리즘은 샘플링 단계에서 제안분포를 기존의 MH within Gibbs 알고리즘처럼 랜덤하게 선택하지 않고 번갈아가며 사용하였다.

LDA에 대한 PWMH 알고리즘을 정리하면 다음과 같다.



## LDA PWMH 알고리즘:

문서집합의 각 문서에 대해서,

(단계 1) 문서집합에 있는 모든 단어에 임의로 토픽을 할당한다.

(단계 2) [샘플링] 모든 단어에 대해서,

(1) 두 가지 제안분포 중 하나를 번갈아가며 선택한다.

- 단어 제안분포
- 문서 제안분포

(2) 선택된 제안분포로부터 새로운 표본  $z^*$ 을 추출한다.

(3) 채택확률을 다음과 같이 구한다.

$$\alpha_{z_t z^*} = \min \left\{ 1, \frac{p(z^*) q(z^*, z_t)}{p(z_t) q(z_t, z^*)} \right\}$$

(4) 채택확률  $\alpha_{z_t z^*}$ 의 확률로  $z_{t+1} = z^*$ 을 채택하고,

$1 - \alpha_{z_t z^*}$ 의 확률로  $z_{t+1} = z_t$ 을 채택한다.

(단계 3) 수렴할 때까지 위 단계를 반복한다.



## 제 4 장 SAMC 알고리즘을 이용한 LDA 근사추론

기존의 마코브 체인 몬테카를로 방법을 이용한 근사추론은 국소트랩의 문제점을 가지고 있다. 이를 해결하기 위해 본 장에서는 Stochastic approximation Monte Carlo(SAMC) 알고리즘을 이용한 LDA 토픽모형 근사추론을 설명한다.

SAMC 알고리즘(Liang et al., 2007)은 표본공간 분할을 통해 국소트랩(local trap) 문제를 해결하는 알고리즘이다.

먼저 SAMC 알고리즘에서 다음과 같은 형태의 분포로부터 표본을 추출하는데에 관심이 있다고 하자.

$$p(Z) = \frac{1}{C} \psi(Z), \quad Z \in \chi \quad (20)$$

$C$ 는 정규화 상수이고,  $\chi$ 는 표본 공간,  $\psi(Z)$ 는 비음(non-negative) 함수이다. 표본 공간은 미리 정해진 상수  $u_0, \dots, u_{m-1}$ 에 따라 함수  $U(Z)$ 가  $m+1$ 개의 분리된 하위영역으로 분할된다.  $E_0 = \{Z: U(Z) \leq u_0\}$ ,  $E_1 = \{Z: u_0 < U(Z) \leq u_1\}, \dots, E_{m-1} = \{Z: u_{m-2} < U(Z) \leq u_{m-1}\}$ ,  $E_m = \{Z: U(Z) > u_{m-1}\}$ 이다. SAMC는 미리 정해진 빈도를 이용하여 각 영역에서 표본을 추출한다.  $m+1$ 개의 하위영역은 non-empty라고 가정한다. 즉,  $i = 0, \dots, m$ 에 대해  $w_i = \int_{E_i} \psi(Z) dZ > 0$ 이다.

$\pi = (\pi_0, \pi_1, \dots, \pi_m)$ 를  $0 < \pi_i < 1$ 와  $\sum_{i=0}^m \pi_i = 1$ 을 만족하는  $m+1$ -벡터인 각 부분 영역에서의 원래의 표본 함수라고 정의한다.  $i = 0, \dots, m$ 에 대해

$\theta_i = \log(\int_{E_i} \psi(Z) dZ / \pi_i) = \log(\frac{w_i}{\pi_i})$ ,  $\theta = (\theta_0, \theta_1, \dots, \theta_m)$ ,  $\Theta$ 는  $\theta$ 의 공간이라 정의한다.  $\psi(Z)$ 의 일반적인 선택을 위해  $\Theta = R^{m+1}$ 이라고 한다.  $\theta^{(t)} = (\theta_0^{(t)}, \theta_1^{(t)}, \dots, \theta_m^{(t)})$



는  $t$ 번째 반복에서 얻어진  $\theta$ 의 추정치이다. 식 (20)을 식 (21)과 같이 다시 표현할 수 있다.

$$f_{\theta^{(t)}}(Z) \propto \sum_{i=0}^m \frac{\psi(Z)}{e^{\theta_i^{(t)}}} I(Z \in E_i) \quad (21)$$

$\{\gamma_t\}$ 는 어떤  $\tau \in (1, 2)$ 에 대해

$$(a) \sum_{t=1}^{\infty} \gamma_t = \infty, \quad (b) \sum_{t=1}^{\infty} \gamma_t^{\tau} < \infty \quad (22)$$

를 만족하는 양수이고 단조 증가하는 수열이라고 정의한다. 본 연구에서는  $t_0 > 1$ 인 어떤 정해진 값에 대해  $\gamma_t = \frac{t_0}{\max(t_0, t)}$ ,  $t = 1, 2, \dots$ 을 사용한다.  $J(Z)$ 는 표본  $Z$ 가 속해있는 하위영역의 인덱스이다.

본 연구에서는 식 (20)의  $\psi(Z)$ 를 다음과 같이 선택하였다.

$$\psi(Z) = \exp\left(-\sum_{d=1}^M \log p(W_d) / \sum_{d=1}^M N_d\right) \quad (23)$$

$$p(W_d) = \sum_{v=1}^V \left( \frac{n_k^{(v)}}{\sum_{v=1}^V n_k^{(v)}} \times \frac{n_d^{(k)}}{\sum_{k=1}^K n_d^{(k)}} \right) \quad (24)$$

여기서  $p(W_d)$ 는  $d$ 번째 문서의 특정 단어가 해당 주제에 부여될 확률을 의미하며,  $N_d$ 는  $d$ 번째 문서의 전체 단어개수를 의미한다.



이와 같은 가정을 이용하여 SAMC 알고리즘을 정리하면 다음과 같다.

### SAMC 알고리즘:

(단계 1) [샘플링] 목표(target)분포 식 (9)을 얻기 위해 한번의

Metropolis-Hastings(MH) 갱신에 의해 표본  $z^{(t+1)}$  추출한다.

(1) 제안분포  $q(z^{(t)}, z^*)$ 에 따라  $z^*$ 를 생성한다.

(2) 다음의 비율을 계산한다.

$$r = e^{\theta_{J_z^{(t)}} - \theta_{J_z^{(t)}}} \frac{\psi(z^*) q(z^*, z^{(t)})}{\psi(z^{(t)}) q(z^{(t)}, z^*)}$$

(3) 확률  $\min(1, r)$  을 가지고 생성된  $z^*$ 를 받아들인다.

만약 받아들이면  $z^{(t+1)} = z^*$ 이고

그렇지 않다면  $z^{(t+1)} = z^{(t)}$ 이다.

(단계 2) [ $\theta$  갱신]  $\theta^* = \theta_t + \gamma_{t+1}(e_{t+1} - \pi)$ 라고 정의한다.

(단계 3) 수렴할 때까지 위 단계를 반복한다.

여기서  $e_{t+1} = (e_{t+1,1}, \dots, e_{t+1,m})$ 이다. 만약  $z^{(t)} \in E_i$  이면  $e_{t+1,i} = 1$ , 그렇지 않으면  $e_{t+1,i} = 0$ 이다. 만약  $\theta^* \in \Theta$  이면  $\theta_{t+1} = \theta^*$ , 그렇지 않으면  $\theta_{t+1} = \theta^* + c^*$ 이다. 여기서  $c^* = (c^*, \dots, c^*)$ 이고  $\theta^* + c^* \in \Theta$ 를 만족하는 임의의 벡터이다. 즉,  $z^*$ 가 뽑힌 지역에 낮은 가중치를 줘서 다른 영역에서  $z^*$ 가 뽑힐 수 있도록  $\theta$ 를 갱신한다.

효율적인 SAMC 알고리즘 이용을 위해 다음을 고려해 볼 필요가 있다.

### 1. 표본분할



본 연구에서는  $m$ 을 2로 정하였다. 표본공간을 함수  $U(x)$ 에 따라 다음과 같이 분할한다.

$$E_0 = \{x : U(x) = 0\}, \quad E_1 = \{x : U(x) = 1\}, \quad E_2 = \{x : U(x) \geq 2\} \quad (25)$$

## 2. 표본분포 $\pi$ 의 선택

본 연구의 목적이  $E_0 = \Omega$ 로부터 표본을 추출하는 것이기 때문에  $\pi$ 는 참조집합에 따라 샘플링이 편의가 되도록 선택될 수 있다. 본 논문에서는 다음과 같이 정한다.

$$\pi_i \propto \frac{1}{(m+1)^\xi}, \quad i = 0, 1, \dots, m \quad (26)$$

여기서  $\xi$ 는 1 또는 2이다. 본 논문에서는  $\xi = 1$ 로 정하였다. 예로  $\xi = 1, m = 2$ 라면,  $(\pi_0, \pi_1, \pi_2) = (0.333, 0.333, 0.333)$ 가 된다.

## 3. gain factor $\gamma_t$ 의 선택과 반복수

조건 (23)을 충족시키기 위해 다음과 같이 정의한다.

$$\gamma_t = \left( \frac{T_0}{\max(T_0, t)} \right)^\eta, \quad t = 0, 1, 2, \dots \quad (27)$$

$T_0 > 1$ 와  $\eta \in (0.5, 1]$ 은 미리 정해진다. 본 논문에서는  $\eta = 1.0$ 으로 정하였다.  $T_0$ 의 큰 값은  $m$ 이 매우 큰 값이라 하더라도 샘플러들이 모든 부분공간에 빠르게 도달하게 해준다.  $N$ 을 반복의 총 횟수라 하자.  $T_0$ 와  $N$ 의 적절한 선택은 정치  $\hat{\theta}$ ,  $\hat{\pi}$ 의 수렴을 검사함으로써 정해진다. 만약 수렴을 하지 않는다면,  $T_0$ 와  $N$ 의 더 큰 값을 통해 다시 분석을 진행한다.



## 제 5 장 모의실험 및 실증분석

### 5.1 토픽모형 성능평가 방법

토픽모형은 문서집합에서 단어가 동시에 출현하는 것을 바탕으로 같은 의미 부류에 속하는 단어들을 하나의 주제로 묶어준다. 이 과정은 수작업이 필요 없어 대용량 문서를 문제없이 처리할 수 있다. 하지만, 흔히 비지도 학습이 그렇듯 자동으로 처리된 결과가 사람이 원하는 결과와 얼마나 일치할지는 모른다. 이러한 점이 비지도 학습인 토픽모형 기법이 가지는 가장 중요한 한계점이다. 따라서 토픽모형 결과가 잘 나왔는지 그 성능을 평가하는 작업이 중요하게 떠오르고 있다.

토픽모형과 같은 클러스터링 기법을 평가하는 방법은 크게 내재적 평가와 외재적 평가로 나뉜다. 내재적 평가는 실제로 잘 분류하는지, 분류된 결과가 사람이 판단하기에 적합한지 등 내부적인 관점에서 평가하는 것이고, 외재적 평가는 해당 기법이 어디에 사용될 수 있는지를 바탕으로 외부적인 관점에서 평가하는 것이다.

본 연구에서는 토픽모형을 내재적으로 평가하는 고전적인 기법인 perplexity (Brown et al., 1992)와 perplexity의 한계를 보완하기 위해 제안된 coherence(Newman et al., 2010)를 사용한다.

#### 5.1.1 Perplexity

Perplexity(Brown et al., 1992)는 정보이론에서 확률모델이 샘플을 얼마나 잘 예측하는지를 측정하는 지표이며, perplexity가 낮을수록 예측력이 좋다. 확률모델이 다른 모델에 비해 얼마나 잘 개선되었는지 평가하거나 동일 모델 내 파라미터에 따른 성능을 평가할 때 주로 사용된다. Perplexity는 다음과 같은 수식



으로 정의된다.

$$perplexity = \exp \left[ -\frac{\sum_{d=1}^M \log p(W_d)}{\sum_{d=1}^M N_d} \right] \quad (28)$$

여기서  $p(W_d)$ 는  $d$ 번째 문서의 특정 단어가 해당 주제에 부여될 확률을 의미하며,  $N_d$ 는  $d$ 번째 문서의 전체 단어개수를 의미한다.

Perplexity 값이 낮다는 것은 학습이 잘되었다는 의미이지 그 결과가 사람이 해석하기에 좋다는 것을 의미하지 않는다. 실제로 Chang et al.(2009)에 따르면 낮은 perplexity 값이 늘 해석에 적절한 결과를 보이지 않는다. 이를 보완하기 위해 Newman et al.(2010)은 coherence를 제시하였다.

### 5.1.2 Coherence

Coherence(Newman et al., 2010)는 토픽이 얼마나 의미론적으로 일관성이 있는지 측정하는 지표로, 단어들이 일관성이 있을수록 coherence가 높아진다. 해당 모델이 실제로 얼마나 의미있는 결과를 도출하는지 확인할 때 주로 사용된다. Coherence를 측정하는 방법은 다양하지만 본 연구에서 사용한 지표는 다음과 같다.

$$coherence = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}, i \neq j \quad (29)$$

여기서  $p(w_i, w_j)$ 는 각 문서에서  $i$ 번째 단어와  $j$ 번째 단어가 동시에 출현할 확률을 의미하며,  $p(w_i)$ 는 전체 문서집합에서  $i$ 번째 단어가 출현할 확률을 의미한다.



## 5.2 PWMH 알고리즘 실증분석

본 장에서는 실증분석을 통해 세 가지 LDA 근사추론 알고리즘을 비교하고자 한다. 비교에 사용된 알고리즘은 Griffiths and Steyvers(2004)가 제안한 C-Gibbs 알고리즘, Yuan et al.(2015)이 제안한 MH within Gibbs 알고리즘, 본 연구에서 제안한 PWMH 알고리즘이다. 모의실험을 위한 통계패키지로는 R(3.5.1)과 Python(3.6)을 사용하였다.

실 데이터는 First Text Retrieval Conference 1992(TRED-1, Harman, 1992)에서 분석되었던 Associated Press(AP) 데이터이다. 해당 데이터는 다양한 컨텐츠 영역에 초점을 맞추고 있으며 토픽모형에서 가장 많이 사용되는 데이터 중 하나이다. Blei의 웹페이지(<http://www.cs.columbia.edu/~blei/>)에서 다운받을 수 있으며 R의 topicmodels 패키지(Grun and Hornik, 2011)에도 내장되어 있다. AP 데이터 형태는 다음과 같다.

<표 1> Associated Press 데이터

	단어1	단어2	단어3	...	총 단어수
문서1	adding	adult	ago	...	186
문서2	able	acknowledged	al	...	174
문서3	able	add	administrator	...	161
...	...	...	...	...	...
문서2245	am	andrew	asked	...	64
문서2246	age	air	aircraft	...	68

문서1은 ‘adding’, ‘adult’, ‘ago’ 단어를 포함하여 총 186개 단어로 구성되어 있으며, 문서2는 ‘able’, ‘acknowledged’, ‘al’ 단어를 포함하여 총 174개 단어로 구성된다. AP 데이터는 총 2,246개 문서와 10,473개 단어로 구성되어 있다. 본 논문은 통계 프로그램 중 Python을 사용하다 보니 표본추출 시 시간이 오래



걸리는 한계점이 존재했다. 따라서, 500개 문서를 랜덤하게 추출해 실증분석을 진행하였다. 분석에 사용된 자료는 총 9,084개 단어로 구성되어 있으며 토픽은 20개로 설정하였다.

PWMH 알고리즘의 적용은 다음과 같다. 실행은  $1 \times 10^4$ 번 반복하며 처음  $5 \times 10^3$ 번은 burn-in 과정으로 제외하고 나머지 반복들을 사용하여 추론을 수행한다. 기존의 C-Gibbs와 MH within Gibbs 알고리즘도 동일하게 적용하였으며, 총 5번의 실행과정을 거쳤다. 종합적으로 C-Gibbs, MH within Gibbs, PWMH 알고리즘을 비교한 결과는 다음과 같다.

<표 2> 실증분석 perplexity 결과

Perplexity	C-Gibbs	MH within Gibbs	PWMH
1	2391.9911	2062.0745	1937.7202
2	2395.2344	2080.1693	1933.0654
3	2375.2535	2081.6343	1932.9606
4	2378.4753	2062.0359	1944.3227
5	2373.4393	2076.9169	1948.4512
평균	2382.8787	2072.5662	1939.3040
표준편차	8.9703	8.7170	6.1729

<표 3> 실증분석 coherence 결과

Coherence	C-Gibbs	MH within Gibbs	PWMH
1	-10.1061	-6.6468	-6.1583
2	-15.5569	-7.0392	-5.8392
3	-11.2264	-7.5499	-5.7829
4	-9.4149	-7.5080	-5.8300
5	-12.4298	-7.9970	-6.2108
평균	-11.7468	-7.3482	-5.9643
표준편차	2.1626	0.4636	0.1817



새롭게 제안한 PWMH 알고리즘이 perplexity와 coherence에서 모두 기존의 MH, C-Gibbs 알고리즘보다 낮은 표준편차를 보여 더 정확한 추정치를 제공하는 것을 확인할 수 있다. PWMH 알고리즘이 LDA 토픽모형의 예측 정확도가 가장 높았으며, 토픽별 단어들의 일관성도 가장 좋은 것으로 나타났다. 실제로 ‘i’, ‘don’t’, ‘can’t’와 같은 단어들의 가중치가 0으로 계산되어 토픽별 단어분포에서 의미없는 단어추출이 적어 PWMH 알고리즘의 성능이 훨씬 좋은 결과를 보인 것으로 판단된다.

<표 4>는 PWMH 알고리즘의 토픽별 단어분포이다. 20개의 토픽 중 4개의 토픽을 가지고 왔다. Topic1은 earth, flight, NASA, shuttle, orbit 등의 단어로 보아 ‘Universe’ 관련 주제라는 것을 짐작할 수 있다. 이와 동일하게 Topic2는 ‘Company’, Topic3은 ‘Stock’, Topic4는 ‘Election’임을 알 수 있었다.

<표 4> PWMH 알고리즘 토픽별 단어분포

Topic1	Topic2	Topic3	Topic4
earth	new	prices	republican
flight	company	percent	campaign
just	Inc	fell	democratic
make	billion	stock	president
mission	Corp	average	i
NASA	firm	market	governor
shuttle	executive	new	years
orbit	million	year	Sen
space	president	rates	presidential
two	offer	price	race



### 5.3 SAMC 알고리즘 모의실험

기존의 마코브 체인 몬테카를로 방법을 이용한 근사추론은 국소트랩의 문제점을 가지고 있었다. 이를 해결하기 위해 표본공간의 분할을 이용하는 SAMC 알고리즘을 제안하였고 모의실험을 통해서 기존의 근사추론 방법과 비교 분석해보았다. 모의실험에 사용된 데이터는 임의로 5개의 토픽을 가지는 문서를 생성하여 사용하였다. 모의실험 데이터는 총 5개 문서와 49개 단어로 구성되어 있다.

<표 5> 모의실험 데이터

	Stock	Education	Game	Statistics	Art
단어1	percent	school	game	machine learning	film
단어2	million	student	team	statistic	show
단어3	market	education	win	R	music
단어4	price	teacher	player	Python	movie
단어5	tax	high	season	data	play
단어6	spending	public	victory	regression	musical
단어7	save	elementary	item	Hadoop	actor
단어8	money	assignment	champion	probability	opera
단어9	budget	university	computer	MongoDB	theater
단어10	government		character	SVM	actress

SAMC 알고리즘의 적용은 다음과 같다. 실행은  $1 \times 10^5$ 번 반복하며 총 5번의 실행과정을 거쳤다. 기존의 C-Gibbs와 MH within Gibbs 알고리즘도 동일하게 적용하였다. 종합적으로 C-Gibbs, MH within Gibbs, SAMC 알고리즘을 비교한 결과는 다음과 같다.



<표 6> 모의실험 perplexity 결과

Perplexity	C-Gibbs	MH within Gibbs	SAMC
평균	9.8083	13.2538	9.8083
표준편차	0	2.2551	0

<표 7> 모의실험 coherence 결과

Coherence	C-Gibbs	MH within Gibbs	SAMC
평균	-40	-275.5556	-40
표준편차	0	146.7071	0

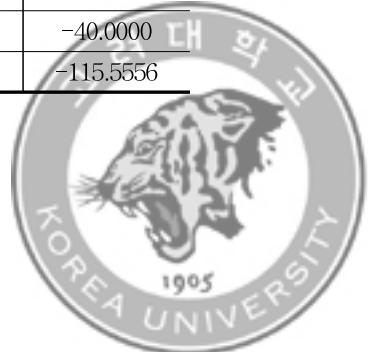
한 번 실행할 때마다 최소 perplexity 상위 10개에 따른 coherence를 계산하였고, <표 6>과 <표 7>은 총 5번 실행의 평균과 표준편차 결과이다. 새롭게 제안한 SAMC 알고리즘의 최소 perplexity와 coherence는 기존의 MH within Gibbs 알고리즘이보다는 좋고 C-Gibbs 알고리즘과는 동일한 결과를 보이지만 <표 8>과 <표 9>를 살펴보면 평균적으로 SAMC 알고리즘의 coherence의 결과가 더 좋은 것으로 확인할 수 있다.

<표 8> SAMC 결과

SAMC	Perplexity	Coherence
1	9.8083	-40.0000
2	10.4811	-40.0000
3	10.5026	-40.0000
4	10.5026	-111.1111
5	10.5026	-40.0000
6	10.5026	-40.0000
7	10.5026	-40.0000
8	10.5026	-40.0000
9	10.9099	-40.0000
10	10.9099	-40.0000

<표 9> C-Gibbs 결과

C-Gibbs	Perplexity	Coherence
1	9.8083	-40.0000
2	10.4811	-80.0000
3	10.5026	-40.0000
4	10.5026	-40.0000
5	10.5026	-40.0000
6	10.5026	-40.0000
7	10.5026	-40.0000
8	10.5026	-40.0000
9	10.9099	-40.0000
10	10.9099	-115.5556



본 논문에서는 Python을 사용하여 시간이 오래 걸리는 한계점이 존재하기 때문에 SAMC 알고리즘의 모의실험만 진행하였다. 하지만 C나 Java와 같은 프로그램을 사용해 구현한다면 대용량 자료에서 활용해 효율적인 토픽모델 결과를 도출할 것으로 기대된다.



## 제 6 장 결론

LDA 토픽모형은 통계적 추론에서 사후분포의 정확추론이 불가능해 근사추론이 사용된다. 기존의 LDA 근사추론 방법인 VEM, C-Gibbs, MH within Gibbs 알고리즘은 Bag-of-word 기반으로 각각의 단어를 모두 동등하게 보았다. 본 논문에서는 불용어 및 빈번하게 사용되는 단어 제거로 LDA 토픽모형 성능을 높이기 위해서 점별 상호정보량(PMI) 가중치를 적용한 PWMH 알고리즘을 제안하였다. 실제로 ‘i’, ‘don’t’, ‘can’t’와 같은 단어들의 가중치가 0으로 계산되어 토픽별 단어분포에서 의미없는 단어추출이 적었고, 이로 인해 PWMH 알고리즘의 perplexity와 coherence가 기존의 근사추론 방법들보다 더 좋은 결과를 보였다. 또한, 더 낮은 표준편차를 제공하기 때문에 PWMH 알고리즘이 더 정확한 추정치를 제공한다.

기존의 마코브 체인 몬테카를로 방법을 이용한 근사추론은 국소트랩의 문제점이 발생한다. 추가적으로 본 논문에서는 전체 표본공간 분할을 통해 국소트랩의 문제를 해결하는 SAMC 알고리즘을 이용하여 LDA 토픽모형 근사추론을 수행해보았다. SAMC 알고리즘의 최소 perplexity와 coherence는 기존의 MH within Gibbs 알고리즈다는 좋고 C-Gibbs 알고리즘과는 동일한 결과를 보이지만, SAMC 알고리즘이 더 좋은 결과를 생성시킬 확률이 높다. SAMC 알고리즘은 기존의 근사추론 알고리즘의 문제점을 개선한 알고리즘으로, 편향된 자료이거나 불균형자료 또는 표본의 크기에 상관없이 국소트랩의 문제점을 가지지 않아 LDA 근사추론에 대한 보다 정확한 추정치를 제공할 것으로 판단된다.

하지만 SAMC 알고리즘의 경우 Python을 사용하여 시간이 오래 걸리는 한계점이 존재하기 때문에 C나 Java와 같은 프로그램을 사용해 구현한다면 대용량 자료에도 활용해 효율적인 LDA 토픽모형 결과를 도출할 것으로 기대된다.



## 참 고 문 헌

- [1] Abramowitz, M., Stegun, I. (1970). *Handbook of mathematical functions*. Dover, New York.
- [2] Andrieu, C., Freitas, N. D., Doucet, A., Jordan, M. (2003). An Introduction to MCMC for Machine Learning, *Machine Learning*, 50(1), 5–43.
- [3] Blei, D. M. (2012). Probabilistic topic models, *Communications of the ACM*, 55(4), 77–84.
- [4] Blei, D. M., McAuliffe, J. D. (2010). Supervised topic models, *Submitted to the Statistical Science*.
- [5] Blei, D. M., Ng, A. Y., Jordan, M. I. (2003). Latent dirichlet allocation, *Journal of Machine Learning Research*, 3, 993–1022.
- [6] Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Lai, J. C., Mercer, R. L. (1992). An estimate of an upper bound for the entropy of English, *Computational Linguistics*, 18(1), 31–40.
- [7] Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*. 288–296.
- [8] Deerwester, S., Dumais, S., Landauer, T., Furnas, G., Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6), 391–407.
- [9] Dickey, J. (1983). Multiple hypergeometric functions: Probabilistic interpretations and statistical uses. *Journal of the American Statistical Association*, 78, 628–637.
- [10] Geigle, C. (2016). Inference methods for latent Dirichlet allocation. *Techincal Report*, Department of Computer Science, Illinois University.
- [11] Geman, S., Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6( 6), 721–741.



- [12] Griffiths, T., Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*. April 6, 101, 5228–5235.
- [13] Grun, B., Hornik, K. (2011). Topicmodels: an R package for fitting topic models. *Journal of Statistical Software*, 40(13), 1–30.
- [14] Harman, D. (1992). Overview of the first text retrieval conference (TREC-1). In *Proceedings of the First Text Retrieval Conference (TREC-1)*, 1–20.
- [15] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications, *Biometrika*, 57(1), 97–109.
- [16] Hofmann, T. (1999). Probabilistic latent semantic indexing. *Proceedings of the Twenty-S second Annual International SIGIR Conference*.
- [17] Li, A., Ahmed, A., Ravi, S., Smola, A. J. (2014). Reducing the sampling complexity of topic models. *Proceedings of the 20th ACM SIGKDD, International Conference on Knowledge Discovery and Data Mining*, 891–900.
- [18] Liang, F., Liu, C., Carroll, R. (2007). Stochastic approximation in Monte Carlo computation, *Journal of the American Statistical Association*, 102(447), 305–320.
- [19] Marsaglia G., Tsang W. W., Wang J. (2004). Fast generation of discrete random variables. *Journal of Statistical Software*, 11(3), 1–11.
- [20] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E. (1953). Equations of state calculations by fast computing machines, *Journal of Chemical Physics*, 21(6), 1087–1092.
- [21] Muller, P. (1993). Alternatives to the Gibbs sampling scheme. *Technical Report*, Institute of Statistics and Decision Sciences, Duke University.
- [22] Newman, D., Lau, J. H., Grieser, K., Baldwin, T. (2010). Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 100–108.
- [23] Papadimitriou, C., Tamaki, H., Raghavan, P., Vempala, S. (2000). Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, 61(2), 217–235.



- [24] Park, J. H., Song, M. (2013). A Study on the Research Trends in Library & Information Science in Korea using Topic Modeling, *Journal of the Korean Society for Information Management*, 30(1), 7–32.
- [25] Salton. G., McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.
- [26] Sung, J. J. (2014). Comparative analysis study of inference techniques for probabilistic model LDA. *Korea Aerospace University, Master thesis*.
- [27] Teh, Y. W., Newman, D., Welling, M. (2007). A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, 19, 1353–1360.
- [28] Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22, 1701 - 1762.
- [29] You, E. S., Choi, G. H., Kim, S. H. (2015). Study on extraction of keywords using TF-IDF and text structure of novel, *Journal of The Korea Society of Computer and Information*, 20(2), 121–129.
- [30] Yuan, J., Gao, F., Ho, Q., Dai, W., Wei, J., Zheng, X., Xing, E. P., Tie-Yan Liu, Wei-Ying Ma, (2015). LightLDA: Big topic models on modest compute clusters. In *Proceedings of the Annual International Conference on World Wide Web(WWW)*, 1351 - 1361.

