Topic Model

- Topic Model

  문서의 묶음(collection)에 나타나는 토픽들(또는 토픽들의 분포)을 찾는 통계 모델.
  문서는 여러 토픽의 조합, 토픽은 여러 단어의 조합이라 가정.

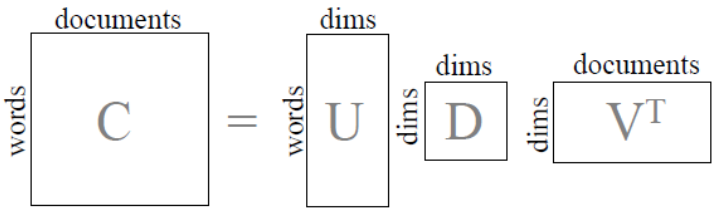
- 활용예시 : CRM

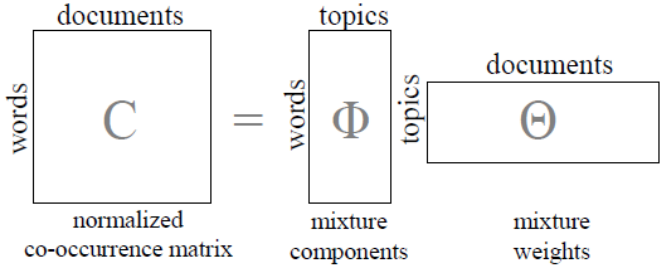
  이미 갖고 있는 고객리뷰 데이터에서 토픽 추출
  → 각 토픽에 대한 대응 매뉴얼 제작
  → 새로운 리뷰가 들어왔을 때 훈련된 모델에 넣어 토픽 분류, 피드백 자동화

- Model

  Latent Dirichlet Allocation, Singular Value Decomposition(Latent Semantic Indexing)

Topic Model





SVD	LDA
Projection to lower dimension(t)	
문서를 가장 잘 대표하도록 t 차원 공간에 projection	t개의 토픽이 확률적으로 문서를 구성 ($\emptyset_{1,1}$ = 토픽1에서 단어1이 차지하는 비중)
Orthonormal basis	토픽분포와 단어분포가 독립이고 not orthogonal -> 모델추론을 따로 해야 함

Topic Model - LDA

- LDA(Latent Dirichlet Allocation)

  Given : D docs, T topics, W words

  Observation : $w_{m,n}$ = doc m에서 word n에 대한 identity vector = [0, 0, … , 1, …, 0]

  doc m에서 word n이 나타나면
  n_{th}index에서 1의 관측치를 얻음

  Assumption :

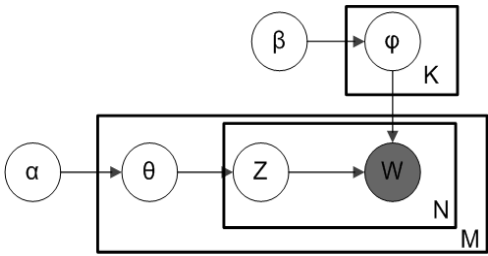
  $\varphi_k = topic\ k$의 단어 분포 $\sim Dir(\beta)$ $\quad ex)\ [0.2, 0.1, 0, \dots]$

  $\theta_m = doc\ m$의 토픽 분포 $\sim Dir(\alpha)$ $\quad ex)\ [0.3, 0.1, \ \dots]$

  $z_{m,n} = doc\ m\ word\ n$의 토픽 $\sim Multi(\theta_m)$ $\quad ex)\ [0, 1, 0, \dots]$

  $w_{m,n} = doc\ m\ word\ n$의 $identity\ vector \sim Multi(\varphi_{z_{m,n}})$ $\quad ex)\ [0, 0, 1, \dots]$

  Goal : 각 문서에서 토픽들, 각 토픽에서 단어들의 분포를 알고 싶다!
  즉, φ_k 와 θ_m를 추정하자.



LDA - Inference φ_k, θ_m

Using Gibbs sampling

$$(\varphi, \theta) = argmax\ P(z, w, \theta, \varphi)$$

$$= argmax\ \pi[P(\varphi_i; \beta)]\ \pi[P(\theta_j; \alpha)\ \pi(P(z_{j,t}|\theta_j)P(w_{j,t}|\varphi_{z_{j,t}}))]$$

$P(z_{j,t}|\theta_j)$: doc j word t의 토픽분포

$P(w_{j,t}|\varphi_{z_{j,t}})$: doc j word t의 토픽분포가 주어졌을 때 doc j에 word t가 나올 확률

$$\propto argmax\ P(z, w; \alpha, \beta) = \prod_{j=1}^{M} \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \frac{\prod_{i=1}^{K} \Gamma(n_{j,(\cdot)}^{i} + \alpha_i)}{\Gamma\left(\sum_{i=1}^{K} n_{j,(\cdot)}^{i} + \alpha_i\right)} \times \prod_{i=1}^{K} \frac{\Gamma\left(\sum_{r=1}^{V} \beta_r\right)}{\prod_{r=1}^{V} \Gamma(\beta_r)} \frac{\prod_{r=1}^{V} \Gamma(n_{(\cdot),r}^{i} + \beta_r)}{\Gamma\left(\sum_{r=1}^{V} n_{(\cdot),r}^{i} + \beta_r\right)}$$

α, β, 단어출현회수로 결정되는 식

$$\propto argmax\ P(Z_{(m,n)} = k \mid \boldsymbol{Z}_{-(\boldsymbol{m},\boldsymbol{n})}, \boldsymbol{W}; \alpha, \beta)$$ = doc m word n의 토픽이 k일 확률

$$\propto argmax\ \left(n_{m,(\cdot)}^{k,-(m,n)} + \alpha_k\right) \frac{n_{(\cdot),v}^{k,-(m,n)} + \beta_v}{\sum_{r=1}^{V} n_{(\cdot),r}^{k,-(m,n)} + \beta_r}$$

Smoothing of Dirichlet par

1. Given $z_{-(m,n)}$ of previous iteration, calculate $P(z_{(m,n)})$
2. Sample $z_{(m,n)}$ from updated multinomial distribution $P(z_{(m,n)})$
3. Update θ, φ using Z until convergence

Gibbs sampling

0. Initiate $\boldsymbol{z}$ randomly

LDA - Inference φ_k, θ_m

Using Gibbs sampling

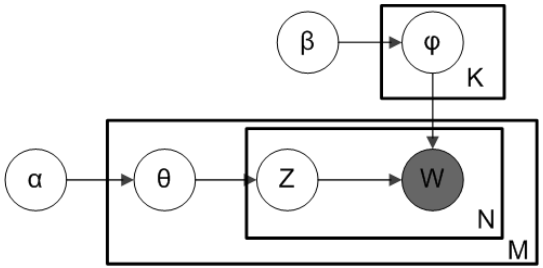
- 3단계에서 update θ, φ by $\hat{\phi}_j^{(w)} = \frac{n_j^{(w)} + \beta}{n_j^{(\cdot)} + W\beta}$, $\hat{\theta}_j^{(d)} = \frac{n_j^{(d)} + \alpha}{n_\cdot^{(d)} + T\alpha}$

- 깁스샘플링의 장점
  - θ, φ의 MLE를 직접 구하지 않음(local maxima)
  - 이전 단계의 $z_{-(m,n)}$을 사용하여 본 단계에서는 $z_{(m,n)}$만 계산 -> 계산량↓

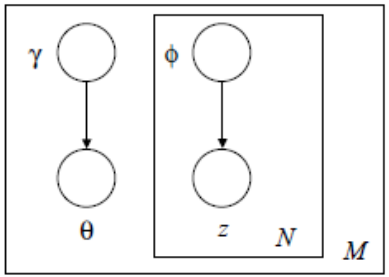
- Hyperparameter K(토픽의 개수)
  - Maximize $Perplexity(w) = exp\left[-\frac{log\{p(w)\}}{\sum_{d=1}^{D}\sum_{j=1}^{V} n^{jd}}\right]$

LDA - Inference φ_k, θ_m

Using Variational Bayes



$\varphi \sim Dir(\beta), w \sim Multi(\varphi)$를 없애고



Dirichlet par γ와 Multinomial par $\emptyset$를 도입
(called *variational parameter*)

1. Get lower bound of log-likelihood by Jensen inequality.

$$\log p(w|\alpha,\beta) \geq \mathrm{E}_q[\log p(\theta,\mathbf{z},\mathbf{w}\,|\,\alpha,\beta)] - \mathrm{E}_q[\log q(\theta,\mathbf{z})]$$

2. $(\hat{\gamma}, \hat{\emptyset})$ = argmin{ KL diverg of posterior prob from $(\gamma, \emptyset)$ and true par(**w**, a, b) }

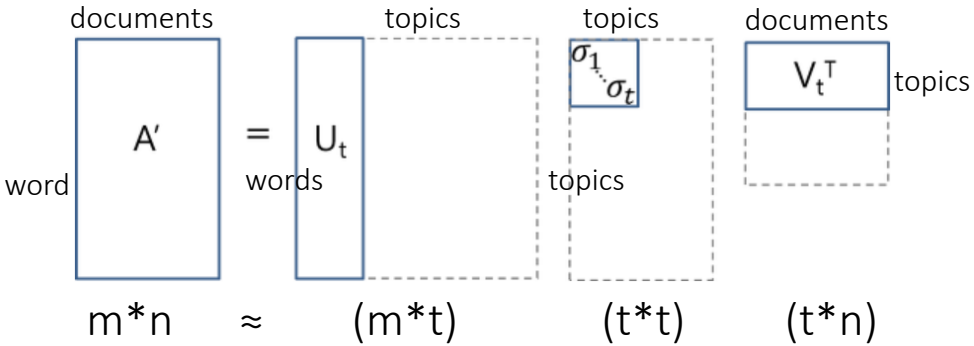
Maximize lower bound ⇔ Min $\mathrm{D}(q(\theta,\mathbf{z}\,|\,\gamma,\phi) \parallel p(\theta,\mathbf{z}\,|\,\mathbf{w},\alpha,\beta))$

3. *variational EM* procedure that max lower bound

E-step : $(\hat{\gamma}, \hat{\emptyset}) = argmin\ (KL\ div; \alpha, \beta)$

M-step: $(\hat{\alpha}, \hat{\beta}) = argmax\ (lower\ bound\ of\ likelihood)$

Topic Model - SVD



col of U = 어떤 토픽으로 나타나는 단어들
row of V = 어떤 문서에 나타나는 토픽들

Projection of a doc(1 row of A) to t dim space = row of A * V_t
Projection of a word(1 col of A) to t dim space = col of A * U_t

⇒ 문서 유사성, 단어 유사성, 단어와 문서의 연관성

SVD의 단점

1. Uninterpretable embedding(rotation으로 해결)
2. Need large dataset for accuracy

Topic Modeling Example

- Data
  - PUBLIC.NEWS_ARTICLE
  - 4000 rows × 3 cols
  - 4000개 뉴스 기사 및 제목

docID	title	content
9e7e0b5b7c1e03a7	근로자 면세비율 축소…세법개정 핵심과제로 부각되나	국민의당, 9월초 구체방안 제시…새누리·더민주 '미지근' 정부, 정치권 논의 방향 예의주시 근로소득세 면세비율을 낮추는 방안이 오는 9월 이후 본격화될 20대 국회 첫 세법개정의 핵심과제로 부각될 전망이다. 근로소득세
42058411b079781e	태영건설 '에코시티 데시앙 2차' 1순위 마감…최고경쟁률 4.15대 1	[헤럴드경제=정찬수 기자] 태영건설이 전주 에코시티 7 · 12블록에 짓는 에코시티 데시앙 2차이 전타입 1순위 당해 마감했다고 4일 밝혔다. 지난 3일 진행한 에코시티 데시앙 2차 1순위 청약접수 결과 7블록은 625가구(특별공
307d96ba740129a4	투자자 몰리는 섹션오피스, '세종 한림프라자'	정부의 집단대출 규제가 주택 위주로 집중되면서 섹션 오피스와 상가 등 일부 수익형 상품이 반사이익을 얻고 있다. 특히 수익형부동산의 한 종류로 인기를 끌던 오피스텔도 규제에 자유로울 수 없고 초과공급 우려가 제기되면

Python LDA

konlpy, genism package

0. data = PUBLIC.NEWS_ARTICLE

1. Parsing and Part-of-Speech Tagging

```
'분야/Noun', '에서/Josa', '우수하다/Adjective', '연구/Noun', '성/Suffix'
'지원/Noun', '하다/Verb', '위해/Noun', '지난/Noun', '2011년/Number', '제
```

2. Bag-of-Word transformation

3. tf-idf transformation

```
# print first 10 elements of first document's tf-idf vector
print(tfidf_ko.corpus[0][:10])
```

```
[(0, 1), (1, 4), (2, 2), (3, 2), (4, 3), (5, 4), (6, 3), (7, 1), (8, 1), (9, 1)]
```

4. LDA

Python LDA

konlpy, genism package

```
0.002*"갤럭시/Noun" + 0.002*"노트/Noun" + 0.002*"7/Number" + 0.002*"전자/Noun" + 0.001*"'/Punctuation" + 0.001*" '/Foreign" + 0.001*"삼/Mod
ifier" + 0.001*""/Punctuation" + 0.001*"V/Alpha" + 0.001*"' /Punctuation"

0.001*"'/Punctuation" + 0.001*"분양/Noun" + 0.001*"대출/Noun" + 0.001*"㎡/Foreign" + 0.001*"농협/Noun" + 0.001*"금융/Noun" + 0.001*"가구/No
un" + 0.001*"아파트/Noun" + 0.001*" '/Foreign" + 0.001*"' /Punctuation"

0.001*"'/Punctuation" + 0.001*"보험/Noun" + 0.001*""/Punctuation" + 0.001*"심사/Noun" + 0.001*"요르단/Noun" + 0.001*" '/Foreign" + 0.001
*"' /Punctuation" + 0.001*"단지/Noun" + 0.001*"수주/Noun" + 0.001*"건설/Noun"

0.001*""/Punctuation" + 0.001*"'/Punctuation" + 0.001*"금리/Noun" + 0.001*"분양/Noun" + 0.001*"아파트/
지/Noun" + 0.001*"KB/Alpha" + 0.001*" · /Punctuation" + 0.001*" '/Foreign"

0.002*""/Punctuation" + 0.002*"터키/Noun" + 0.002*"'/Punctuation" + 0.001*"시리아/Noun" + 0.001*"IS/Al
*"대통령/Noun" + 0.001*"테러/Noun" + 0.001*"명/Noun" + 0.001*"뉴시스/Noun"
```

→ SAS VTA LDA에 비해 무의미한 단어 제거

1	터키	0.0075021901
1	명	0.0073163355
1	로	0.0064884377
1	것	0.0062434476
1	일	0.0060744888
1	과	0.0059731136
1	도	0.0056816598
1	=	0.0054662374
1	있다	0.0050100488
1	와	0.0049340174
1	·	0.004857986
1	is	0.0044144693
1	시리아	0.0043849015
1	고	0.0043342139
1	대통령	0.0042835263
1	쿠데타	0.0041568072

Python LDA

konlpy, genism package

- TF-IDF(Term Frequency – Inverse Document Frequency)

$$TF\text{-}IDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

= 문서d에서 단어t의 빈도

× 전체문서에서 단어t가 얼마나 공통적으로 나타나는지

→ 특정 문서에서 빈도수가 높고, 전체문서 중 그 단어를 포함한 문서 수가 적을 수록 높음

→ 모든 문서에서 흔하게 나타나는 단어를 거를 수 있음

D.Agarwal and B.Chen. fLDA : Matrix Factorization through Latent Dirichlet Allocation (http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.564.5299&rep=rep1&type=pdf)

fLDA

Matrix Factorization through LDA

- Goal

  추천 시스템에서 아이템에 대한 유저의 레이팅을 예측

- Observation

  User Feature, Item Feature, User-Item Feature,

  Item = word1_word2_word3

- Key Idea

  Item = word1_word2_word3

  → 아이템을 이루는 단어들의 토픽정보를 활용하자!

  → $\widehat{\text{레이팅}} = f$(유저 정보, 아이템 정보, 유저-아이템 정보, 아이템의 토픽)

- Example

  기사를 추천하는 신문사(아이템 = 기사)

  기사=(Obama, Trump, …)에 대한 유저의 레이팅 예측률을 높이는 추론

  → 기사1=(Obama, …)과 기사2=(Trump, …)이 다른 토픽을 갖게 됨

- By-Product

  아이템 추천 시 아이템의 토픽 정보 이용

  (‘낮’에 뉴스를 보는 사람들은 ‘스포츠’에 관심이 있다 → 기사추천, 광고 타겟팅)

fLDA

Matrix Factorization through LDA

- Model

$$
\begin{aligned}
&\textbf{Rating:} && y_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma^2), \text{or} && \text{(Gaussian)} \\
& && y_{ij} \sim \text{Bernoulli}(\mu_{ij}) && \text{(Logistic)} \\
& && l(\mu_{ij}) = x'_{ij}\, b + \alpha_i + \beta_j + s'_i\, \bar{z}_j \\
&\textbf{User factors:} && \alpha_i = g'_0 x_i + \epsilon_i^{\alpha}, && \epsilon_i^{\alpha} \sim \mathcal{N}(0, a_\alpha) \\
& && s_i = H x_i + \epsilon_i^{s}, && \epsilon_i^{s} \sim \mathcal{N}(\mathbf{0}, A_s) \\
&\textbf{Item factors:} && \beta_j = d'_0 x_j + \epsilon_j^{\beta}, && \epsilon_j^{\beta} \sim \mathcal{N}(0, a_\beta) \\
& && \bar{z}_j = \textstyle\sum_n z_{jn} \;/\; W_j \\
&\textbf{Topic model:} && \theta_j \sim \text{Dirichlet}(\lambda) \\
& && \Phi_k \sim \text{Dirichlet}(\eta) \\
& && z_{jn} \sim \text{Multinom}(\theta_j) \\
& && w_{jn} \sim \text{Multinom}(\Phi_{z_{jn}})
\end{aligned}
$$

x_i = user feature = (age, gender, location)

x_j = item feature = (publisher, category)

Observed = y_{ij}, x_{ij}, w_{jn} = (rating, user·item feature, words in item)

Want to know = $\Theta = [b, g_0, d_0, H, \sigma^2, a_\alpha, a_\beta, A_s, \lambda, \eta]$

→ $\hat{\Theta} = \arg\max_{\Theta} \Pr[y, w \mid \Theta, X]$

Note : θ의 empirical distribution인 z를 이용 → 유저레벨 회귀분석과 빠른 수렴

fLDA

Matrix Factorization through LDA

- Inference

  E-step : Compute $E_{\Delta,z}[LL(\Theta;\Delta,z,y,w,X)\,|\,\hat{\Theta}^{(t)}]$

  where $\Delta_{ij}=[\alpha_i,\beta_j,s_i]$, $(\Delta,z\,|\,\hat{\Theta}^{(t)},y,w,X)$

  M-step : $\hat{\Theta}^{(t+1)}=\arg\max_{\Theta} E_{\Delta,z}[LL(\Theta;\Delta,z,y,w,X)\,|\,\hat{\Theta}^{(t)}]$

  E-step done by Gibbs sampling.

  M-step done by some regression problems on each variables.

Rating:
$y_{ij}\sim\mathcal{N}(\mu_{ij},\sigma^2)$, or (Gaussian)
$y_{ij}\sim\text{Bernoulli}(\mu_{ij})$ (Logistic)
$l(\mu_{ij})=x'_{ij}\,b+\alpha_i+\beta_j+s'_i\,\bar{z}_j$

User factors:
$\alpha_i=g'_0x_i+\epsilon^{\alpha}_i$, $\epsilon^{\alpha}_i\sim\mathcal{N}(0,a_\alpha)$
$s_i=Hx_i+\epsilon^{s}_i$, $\epsilon^{s}_i\sim\mathcal{N}(\mathbf{0},A_s)$

Item factors:
$\beta_j=d'_0x_j+\epsilon^{\beta}_j$, $\epsilon^{\beta}_j\sim\mathcal{N}(0,a_\beta)$
$\bar{z}_j=\sum_n z_{jn}\,/\,W_j$

Topic model:
$\theta_j\sim\text{Dirichlet}(\lambda)$
$\Phi_k\sim\text{Dirichlet}(\eta)$
$z_{jn}\sim\text{Multinom}(\theta_j)$
$w_{jn}\sim\text{Multinom}(\Phi_{z_{jn}})$