

大規模テキストを対象にした分散表現に基づくトピック抽出

尾崎 花奈

小林 一郎

お茶の水女子大学 理学部 情報科学科

{g1420513, koba}@is.ocha.ac.jp

1 はじめに

トピックモデルは、文書の中に潜在的に存在するトピックを自動で抽出するためのモデルである。近年最も広く使われているトピック抽出の手法である LDA (Latent Dirichlet Allocation) [1] は、各文書に潜在トピックがあると仮定し、統計的に共起しやすい単語の集合が生成される要因を、潜在トピックという観測できない確率変数で定式化する。近年 LDA を改良した様々なモデルが提案され、その中でも Das ら [2] によって提案された、単語の分散表現 (Word embedding) と LDA を組み合わせた Gaussian LDA が注目を集めている。従来の LDA ではトピックごとの単語分布をカテゴリ分布としていたのに対し、Gaussian LDA では embedding 空間上の多次元ガウス分布としている。単語の意味的関係性を事前知識として持つため、従来の LDA に比べ、Gaussian LDA の方が自己相互情報量 (PMI) の値が高くなったと報告している。また、交差検定をした際に訓練データには無く、評価用データに出現する未知語に対してもトピックを推定できるようになった。これにより、従来の LDA においてトピックの推定を諦めていた未知語に対してもトピックの割り当てができるようになった。

Gaussian LDA における事後分布推定方法では、周辺化ギブスサンプリングを用いているが、本稿では SVI (Stochastic Variational Inference) [3][4] を用いることによって、計算時間の大幅な短縮が可能になり、大規模なコーパスに対して効率的な処理が可能になる。

2 関連研究

2.1 Latent Dirichlet Allocation (LDA)

LDA は、Blei ら [1] によって提案されたテキストの生成モデルである。文書中に含まれる潜在トピックに

対し、単語が出現する確率分布が定義され、そこから単語が抽出され文書が生成されるという生成モデルに基づき、文書中の潜在トピックを推定する。

LDA の生成モデルをまとめると以下のようになる。

1. for $k = 1$ to K
 - (a) Choose topic $\beta_k \sim \text{Dir}(\eta)$
2. for each document d in corpus D
 - (a) Draw topic distribution $\theta_d \sim \text{Dir}(\alpha)$
 - (b) for each word index n from 1 to N_d
 - (a) Draw a topic $z_n \sim \text{Categorical}(\theta_d)$
 - (b) Draw a word $w_n \sim \text{Categorical}(\beta_{z_n})$

ここで θ_d は文書 d のトピック分布、 β_k はトピック k における単語の出現分布を表す。

LDA のグラフィカルモデルを図 1(a) に示す。

2.2 Gaussian LDA

まず、LDA におけるトピックを生成する分布を多次元ガウス分布にするというモデルが Hu ら [5] によって提案された。このモデルに、単語の分散表現を組み合わせたものが Das ら [2] によって提案されたモデルである。Embedding のツールとしては、word2vec [6] を用いている。連続空間に Embedding された単語ベクトルに対し、トピック k を同空間上での多次元ガウス分布としている。これによって、トピックごとの単語分布が連続分布となり、この分布から単語ベクトルが生成される過程がモデル化される。

単語の分散表現を用いることによって、トピック内の意味的結束性が向上し、実験結果として従来の LDA と比較して PMI が上昇することが確認されている。また、トピックごとの単語分布に連続分布を用いることによって、従来の LDA では対応できていなかった未知語に対しても、もう一度モデルでの推定を行うこと

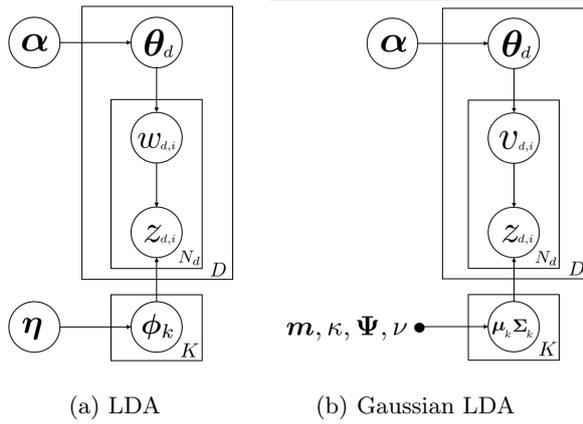


図 1: LDA と Gaussian LDA のグラフィカルモデル

なしに潜在トピックを割り当てることが可能になっている。

Gaussian LDA の生成モデルをまとめると以下のようになる。

1. for $k = 1$ to K
 - (a) Draw topic covariance $\Sigma_k \sim \mathcal{W}^{-1}(\Psi, \mu)$
 - (b) Draw topic mean $\mu_k \sim \mathcal{N}(\mu, \frac{1}{\kappa} \Sigma_k)$
2. for each document d in corpus D
 - (a) Draw topic distribution $\theta_d \sim \text{Dir}(\alpha)$
 - (b) for each word index n from 1 to N_d
 - (a) Draw a topic $z_n \sim \text{Categorical}(\theta_d)$
 - (b) Draw $v_{d,n} \sim \mathcal{N}(\mu_{z_n}, \Sigma_{z_n})$

ここで θ_d は従来の LDA と同じく文書 d のトピック分布を表すが、 μ_k, Σ_k はそれぞれトピック k における多次元ガウス分布の平均と分散を表している。また、 $v_{d,n}$ は単語ベクトルを表す。

Gaussian LDA のグラフィカルモデルを図 1(b) に示す。

3 SVI を用いたトピック推定

Gaussian LDA において、事後分布の推定に用いていたのは周辺化ギブスサンプリングであった。しかし、ギブスサンプリングは実装が簡潔である利点はあるが、計算時間が非常にかかる。

そこで本稿では、確率的変分近似法 (SVI: Stochastic Variational Inference) を用いることによって、計算時間の大幅な減少を実現し、大規模なデータに対して効率的にトピック解析することを目指す。

変分ベイズにおいては、真の事後分布に対してより簡単な近似分布 $q(z, \theta, \mu, \Sigma)$ を考え、対数周辺尤度

$p(v|\alpha, \eta)$ の変分下限を最大にする $q(z, \theta, \mu, \Sigma)$ を求める。

$$\begin{aligned} \log p(v|\alpha, \eta) &\geq L(v, \phi, \gamma, \zeta) \\ &\triangleq \mathbb{E}_q[\log p(v, z, \theta, \mu, \Sigma)] - \mathbb{E}_q[\log q(z, \theta, \mu, \Sigma)]. \end{aligned} \quad (1)$$

平均場近似に基づいて、近似分布 q に対して次のように各確率変数に独立性の仮定をおく。

$$q(z, \theta, \mu, \Sigma) = q(z)q(\theta)q(\mu, \Sigma). \quad (2)$$

単語ごとのトピック割り当て z のパラメータを ϕ 、文書ごとのトピック分布 θ のパラメータを γ 、トピックごとの単語分布の平均と分散 μ, Σ のパラメータを $\zeta = (\mathbf{m}, \kappa, \Psi, \nu)$ とすると、近似分布 q はそれぞれ以下のように表される。

$$\begin{aligned} q(z_{di} = k) &= \phi_{dw_{di}k}; & q(\theta_d) &= \text{Dir}(\theta_d | \gamma_d), \\ q(\mu_k, \Sigma_k) &= \text{NIW}(\mu, \Sigma | \mathbf{m}, \kappa, \Psi, \nu). \end{aligned} \quad (3)$$

また、それぞれのパラメータ ϕ, γ, ζ は以下のように定義される。

$$\begin{aligned} \phi_{dwk} &\propto \exp\{\mathbb{E}_q[\log \theta_{dk}] + \mathbb{E}_q[\log N(v_{dw} | \mu_k, \Sigma_k)]\}, \\ \gamma_{dk} &= \alpha + N_{dk}; & \kappa_k &= \kappa + N_k; & \nu_k &= \nu + N_k, \\ \mathbf{m}_k &= \frac{\kappa \mathbf{m} + N_k \bar{\mathbf{v}}_k}{\kappa_k}, \\ \Psi_k &= \psi + \mathbf{C}_k + \frac{\kappa N_k}{\kappa_k} (\bar{\mathbf{v}}_k - \mathbf{m})(\bar{\mathbf{v}}_k - \mathbf{m})^T. \end{aligned} \quad (4)$$

ここで、

$$\begin{aligned} N_{dk} &= \sum_w n_{dw} \phi_{dwk}; & N_k &= \sum_{d,w} n_{dw} \psi_{dwk}, \\ \bar{\mathbf{v}}_k &= \frac{\sum_d \sum_{i:z_{d,i}=k} (v_{d,i})}{N_k}, \\ \mathbf{C}_k &= \sum_d \sum_{i:z_{d,i}=k} (v_{d,i} - \bar{\mathbf{v}}_k)(v_{d,i} - \bar{\mathbf{v}}_k)^T. \end{aligned} \quad (5)$$

また、 q のもとでの $\log \theta_{dk}$ と $\log N(v_{dw} | \mu_k, \Sigma_k)$ の期待値はそれぞれ

$$\begin{aligned} \mathbb{E}_q[\log \theta_{dk}] &= \Psi(\gamma_{dk}) - \Psi\left(\sum_{i=1}^K \gamma_{di}\right), \\ \mathbb{E}_q[\log N(v_{dw} | \mu_k, \Sigma_k)] &= -\frac{1}{2} \mathbf{v}_{dw}^T \langle \Sigma_k^{-1} \rangle \mathbf{v}_{dw} \\ &\quad + \mathbf{v}_{dw}^T \langle \Sigma_k^{-1} \mu_k \rangle - \frac{1}{2} \langle \mu_k^T \Sigma_k^{-1} \mu_k \rangle \\ &\quad - \frac{1}{2} \langle \log |\Sigma_k| \rangle, \end{aligned} \quad (6)$$

と表される。ただし、 Ψ はディガンマ関数を表し、 $\langle \rangle$ は期待値を表すものとする。

従来の変分近似法における LDA の学習では、文書データ全体に対して繰り返し学習が必要であったが、SVI は文書を逐次的に学習する。 $q(\mathbf{z}_d), q(\boldsymbol{\theta}_d)$ は各文書ごとに学習される近似事後分布であるので逐次学習を行う必要はなく、逐次学習の対象となるのは $q(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ である。よって、パラメータ $\zeta = (\mathbf{m}, \kappa, \boldsymbol{\Psi}, \nu)$ の更新において、確率的自然勾配法を用いた最適化を行う。

n_t 個の単語を含む t 番目の文書において、 ζ は固定して ϕ_t と γ_t の最適化を行う。次に、 ζ の中間パラメータ $\zeta^* = (\mathbf{m}^*, \kappa^*, \boldsymbol{\Psi}^*, \nu^*)$ を以下の式で求める。

$$\begin{aligned} \kappa_k^* &= \kappa + DN_{tk}; & \nu_k^* &= \nu + DN_{tk}, \\ \mathbf{m}_k^* &= \frac{\kappa \mathbf{m} + DN_{tk} \bar{\mathbf{v}}_k}{\kappa_k^*}, \\ \boldsymbol{\Psi}_k^* &= \boldsymbol{\psi} + \mathbf{C}_k + \frac{\kappa DN_{tk}}{\kappa_k^*} (\bar{\mathbf{v}}_k - \mathbf{m})(\bar{\mathbf{v}}_k - \mathbf{m})^T. \end{aligned} \quad (7)$$

ただし、 $N_{tk} = \sum_w n_t \phi_{twk}$ 。

D はコーパスの数を表しており、式 (7) は式 (4) における ζ の計算を文書 t の複製 D 個に対して適用することを意味している。この操作によって、パラメータ ϕ, γ, ζ を更新する各イテレーションにおいてコーパス全体を必要とすることがなくなり、大規模なデータに対して逐次的な計算が可能になる。次のイテレーションに用いる ζ は、 $\rho_t \triangleq (\tau_0 + t)^{-\kappa}$ 、 $\kappa \in (0.5, 1]$ で与えられるステップサイズによって、前回のイテレーションの ζ と更新された ζ^* に対して重みをかけることによって以下の式で求められる。

$$\zeta = (1 - \rho_t)\zeta + \rho_t\zeta^*. \quad (8)$$

アルゴリズムは以下ようになる。

Algorithm 1 SVI for Gaussian LDA

Define $\rho_t \triangleq (\tau_0 + t)^{-\kappa}$

Initialize $\mathbf{m}, \kappa, \boldsymbol{\Psi}, \nu$ randomly.

for $t = 0$ to ∞ **do**

 Estep:

 initialize $\gamma_{tk} = 1$ (The constant 1 is arbitrary.)

repeat

 Set $\phi_{twk} \propto \exp\{\mathbb{E}_q[\log \theta_{tk}] + \mathbb{E}_q[\log N(\mathbf{v}_{dw} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]\}$

 Set $\gamma_{tk} = \alpha + \sum_w n_{tw} \phi_{twk}$

until $\frac{1}{K} \sum_k |\text{change in } \gamma_{tk}| < 0.00001$

 Mstep:

 Compute ζ_k^* with Eq.(7)

 Set $\zeta = (1 - \rho_t)\zeta + \rho_t\zeta^*$

end for

4 まとめと今後の課題

本研究では、単語の分散表現を取り入れたトピックモデルにおいて、大規模テキストに対応できる効率の良い計算方法を導入する提案を行なった。今後は、実験を通じて提案手法の正当性を検証していく。

参考文献

- [1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. (2003) Latent dirichlet allocation, J. Mach. Learn. Res., 3:993-1022, March.
- [2] Rajarshi Das, Manzil Zaheer, and Chris Dyer. (2015) Gaussian LDA for topic models with word embeddings. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics.
- [3] Matthew D. Hoffman, David M. Blei, Francis Bach. (2013) Online Learning for Latent Dirichlet Allocation. In Advances in Neural Information Processing Systems.
- [4] Matthew D. Hoffman, David M. Blei, Chong Wang, John Paisley. (2013) Stochastic variational inference. The Journal of Machine Learning Research, 14(1):1303-1347.
- [5] Pengfei Hu, Wenju Liu, Wei Jiang, and Zhanlei Yang. (2012) Latent topic model based on Gaussian-LDA for audio retrieval. In Pattern Recognition, volume 321 of CCIS, pages 556-563. Springer.
- [6] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. (2013) Linguistic regularities in continuous space word representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 746-751, Atlanta, Georgia, June.