混合ディリクレ分布を用いたトピックに基づく言語モデル

貞光　九月[†]　　三品　拓也[††]　　山本　幹雄[†]

Topic-based language models using Dirichlet mixtures

Kugatsu SADAMITSU[†], Takuya MISHINA[††], and Mikio YAMAMOTO[†]

あらまし　混合ディリクレ分布を多項分布パラメータの事前分布とした (合成分布は混合 Polya 分布), 生成文書モデルを提案し, 統計的言語モデルへの応用という面で高い性能を持つことを示す. 本稿では, 混合ディリクレ分布のパラメータ推定法および適応時に必要な事後分布の期待値推定法をいくつか述べた後に, 2 つの代表的な従来の文書モデルと比較する. 1 つ目の従来モデルは, 統計的言語モデルにトピックを取り込むときによく使われる Mixture of Unigrams である. 2 つ目は代表的な生成文書モデルである LDA(Latent Dirichlet Allocation) である. 新聞記事を用いた文書確率および動的に適応する $n$gram モデルを用いた実験で, 提案モデルは従来の 2 つのモデルと比べて低い混合数で低いパープレキシティを達成できることを示す.

キーワード　混合ディリクレ分布，確率的 LSA，統計的言語モデル，EM アルゴリズム，LDA，アスペクトモデル

1. まえがき

文書のトピックをモデル化する方法として特異値分解を利用した LSA(Latent Semantic Analysis) [1] が有名であるが，これを確率的な意味で再構築したモデルとして Probabilistic LSA(以下，PLSA) [2] が提案されている．PLSA は単語の出現確率を数十から数百のユニグラムモデル (多項分布のパラメータ) の混合としてモデル化し，動的適応時に，観測された単語の履歴を用いて各ユニグラムモデルの混合比を計算する．PLSA は情報検索 [2] や統計的言語モデルの性能向上に有効であることが明らかにされている [3]〜[5]. また，モデル推定時に特異値分解等の行列演算が必要ないので効率的であり，比較的大規模な学習データを用いることができる．ただし，推定すべきパラメータ数が多く，最ゆう推定によるパラメータ学習・動的適応では過適応しやすいという問題点があった．これを解決するために，一般に過適応しにくいと考えられるベイズ学習 [6] を利用した LDA (Latent Dirichlet Allocation) が提案されている [7]．LDA は，PLSA のユニグラムモデルを混合する重みの事前分布として，多項分布の共役事前分布であるディリクレ分布を仮定し，事後分布の期待値を計算するために必要な積分を変分法によって近似するものである．

本稿でも，ベイズ学習を利用するために事前分布を導入することを試みるが，PLSA を拡張した LDA とは異なり，単語の出現確率そのものをディリクレ分布でモデル化する．すなわち，混合されるユニグラムモデルの「混合比」を確率変数とする方法 (LDA) ではなく，最終的に求めたい出現確率そのものを確率変数とする方法となる．ただし，ディリクレ分布は共分散構造を柔軟にモデル化できない．このため, ある単語 A が出現した場合, 単語 B は同じ文書に出現しやすいが, 単語 C は出現しにくい等といった単語の共起性をうまくモデル化できない．この問題は, 履歴から次の単語を予測することを本質とする統計的言語モデルへの応用において，大きな弱点となりえる．そこで, 本稿ではディリクレ分布の混合分布 (以下, 混合ディリクレ分布) を利用する方法を検討する．混合分布によって, 各要素分布があるトピックを表現し, 各トピック毎に異なった単語の共起性をモデル化することが可能となる．さらに，PLSA のようにユニグラムモデルの混

[†] 筑波大学大学院システム情報工学研究科，つくば市
Graduate School of Systems and Information Engineering, University of Tsukuba, 1-1-1 Tennodai, Tsukuba-shi, 305-8573 Japan
[††] 日本アイ・ビー・エム（株）東京基礎研究所，大和市
IBM Research, Tokyo Research Laboratory, 1623-14 Shimotsuruma, Yamato-shi, Kanagawa-ken, 242-8502 Japan

合の場合，合成できない単体 (変数の和が 1 になる領域) 上の確率領域がありえるが，本方式はすべての確率領域をモデル化しているため，あらゆる未知の状況にも対応できるという利点がある．

混合ディリクレ分布はアミノ酸の分布モデルとしてバイオ分野で使われているが [8]，テキストのモデルとして利用する場合と比べて規模が小さい．例えば，各ディリクレ分布のパラメータは出現する記号種類数と同じであるが，アミノ酸の場合は 20 であるのに対し，テキストの場合は少なくとも 2 万 (単語数) 程度は必要である．また，混合数も文献 [8] では 10 以下であるが，本稿では後で述べるように数十から数百 (最大 500) を考える．このようにパラメータ数で 4 桁程度違うため，文献 [8] で提案されている推定方法では多くの場合モデルが求まらない．また計算時間も膨大となり，混合ディリクレ分布をそのまま言語モデルに応用することは難しかった．本稿では桁違いのパラメータ数でも安定してモデル推定が可能で，かつ高速な混合ディリクレ分布の推定方法を述べる．

以下，混合ディリクレ分布の最ゆう推定手法を 2 種類述べた後に，履歴を与えられた場合の予測分布式を導出する．最初のパラメータ推定方法は，学習データ中の文書ごとに求めた単語のユニグラム確率をデータとして，直接，混合ディリクレ分布のパラメータを推定する方法である．2 つ目の方法は，混合ディリクレ分布をパラメータの事前分布とした多項分布 (合成分布は混合 Polya 分布) を仮定し，各文書の単語出現頻度から推定する方法である．3. で述べる予測分布は，2 つ目のパラメータ推定方法と同様に混合 Polya 分布を仮定して履歴中の単語出現頻度から直接導出する．また，実験の節で示すように，提案モデルは低い混合数で高い性能を達成するが，高い混合数になると性能が悪化してしまう．そこで，これを改善するモデル平均の手法を導入したモデルについても 4. で述べる．5. では各種テキストモデル間の関係についてグラフィカルモデル表現を示しながら考察し，最後の節では，履歴情報によって動的に適応を行う $n$gram モデルのパープレキシティを用いた比較実験を報告する．

2. 混合ディリクレ分布とパラメータ推定

2.1 混合ディリクレ分布を利用した文書モデル

文書はジャンル，著者，文書のスタイル，トピック等の様々な変動要因を持っており，それらによってその文書中の単語出現確率が変化する [9]．本論文では，それらの変動要因を一まとめにして「トピック」と呼ぶことにする．

トピックによる出現確率の変動を捕らえて言語モデルの性能を改善するモデルの代表は Mixture of Unigrams である [10], [11]．このモデルは決められた個数のトピック (以下「代表的トピック」と呼ぶ) を設定し，各代表的トピック毎の単語出現確率を unigram モデルでモデル化する．しかし，実際にはトピックは無数に考えられ，各 unigram モデルは代表的トピックの周辺をモデル化していると考えられる．Mixture of Unigrams モデルをさらに改良するために，代表的トピック毎に周辺のトピックの広がりまでをモデル化する方法が本論文で提案する混合ディリクレ分布である．

Mixture of Unigrams における各要素の unigram モデルは代表的トピックの周辺の文書に対する平均の単語出現確率を与える．しかし，代表的トピックによって単語出現確率が変動するように，代表的トピックによって周辺のトピックの広がりが異なる．例えば，新聞記事に対して代表的トピックとして「野球」と「国際関係」を考えた場合，国際関係の方がはるかに大きいトピックの広がりを持っている（すなわち単語の出現確率の変動が大きい）ことが予想される．ディリクレ分布は，こうした単語出現確率の変動を直接モデル化することができる．さらに，各代表的トピックの周辺について単語出現確率の変動を一つのディリクレ分布でモデル化すれば，全体としては次節から詳しく述べる混合ディリクレ分布となる．

もう一つ別の観点から，混合ディリクレ分布を解釈すると次のようになる．単語出現頻度の文書をまたぐ変動は負の二項分布でよくモデル化できることが知られている [9]．二項分布のパラメータ (単語の出現確率) の事前分布としてベータ分布を仮定した場合の合成分布である負の超幾何分布 (ベータ二項分布) は，観測数 n を大きくしたとき負の二項分布に分布収束する．これらのことから単語出現頻度の文書をまたぐ変動 (トピックの分布によると仮定できる) は，二項分布のパラメータである単語の出現確率がベータ分布 (トピックの分布) によって変動するためと解釈できる．これは各単語を個別に考えた場合のモデルであるが，この状況を多変数，すなわち複数の単語出現を同時に考慮したモデルに拡張する場合，ベータ分布の多変数版であるディリクレ分布を導入するのが自然である．ただし，単一のディリクレ分布では共分散構造を柔軟にモデル化できないため，これを改善するために混合分布

とすれば統計的言語モデルとしての性能をさらに上げることができると予想できる．

2.2 混合ディリクレ分布

V 次元の単体 $\Delta(V)$ 上の確率変数 $\boldsymbol{p} = (p_1, p_2, ..., p_V)$ に対するディリクレ分布の確率密度関数 $P_D(\boldsymbol{p}; \boldsymbol{\alpha})$ は次のように定義される．$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_V), \alpha_v > 0$ であり，ディリクレ分布のパラメータである．

$$P_D(\boldsymbol{p}; \boldsymbol{\alpha}) = \frac{\Gamma(\alpha)}{\prod_{v=1}^{V} \Gamma(\alpha_v)} \prod_{v=1}^{V} p_v^{\alpha_v - 1} \tag{1}$$

ここで，$\alpha = \sum_{v=1}^{V} \alpha_v$ である．また，文書をモデル化している場合，V は対象としている単語の語彙サイズ，p_v は v 番目の単語の出現確率の値を表す確率変数と解釈される．M 個のディリクレ分布 $P_D(\boldsymbol{p}; \boldsymbol{\alpha}_m)$ を $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_M)$ で重み付けした混合ディリクレ分布 $P_{DM}(\boldsymbol{p}; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M)$ は次のようになる．

$$\begin{aligned} P_{DM}(\boldsymbol{p}; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M) &= \sum_{m=1}^{M} \lambda_m P_D(\boldsymbol{p}; \boldsymbol{\alpha}_m) \\ &= \sum_{m=1}^{M} \lambda_m \frac{\Gamma(\alpha_m)}{\prod_{v=1}^{V} \Gamma(\alpha_{mv})} \prod_{v=1}^{V} p_v^{\alpha_{mv} - 1} \end{aligned} \tag{2}$$

ここで，M は混合数，$\boldsymbol{\alpha}_1^M = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_M)$，$\boldsymbol{\alpha}_m$ は第 m コンポーネントのディリクレ分布のパラメータ，$\alpha_m = \sum_v \alpha_{mv}$ である．

2.3 混合ディリクレ分布の最ゆう推定

今，i 番目の文書中での単語出現確率分布を $\boldsymbol{p}_i = (p_{i1}, p_{i2}, ..., p_{iV})$，$n$ 個の文書集合の単語出現確率分布の集合を $D = (\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_n)$ としたときの混合ディリクレ分布の対数ゆう度 $\mathcal{L}(D)$ は以下のようになる．

$$\mathcal{L}(D; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M) = \sum_{i=1}^{n} \log \sum_{m=1}^{M} \lambda_m P_D(\boldsymbol{p}_i; \boldsymbol{\alpha}_m) \tag{3}$$

ゆう度を最大とするパラメータ $(\boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M)$ を求めるには，$\boldsymbol{p}_i$ を出力した混合コンポーネントを隠れ変数 z_i とする EM アルゴリズムによって，$\boldsymbol{\lambda}$ と $\boldsymbol{\alpha}_1^M$ を交互に更新すればよい．ただし，この方法は各文書毎に単語出現確率分布を求めなければならないが，各文書一つ一つは数万単語の単語出現確率分布を精度よく推定できるほどには大きくないため，学習データそのものに精度の問題が生じる．この推定方法は直接的かつ基本的ではあるが，上記の理由によって本稿の実験では用いていないため，紙面の節約のため省略する．

2.4 混合 Polya 分布を用いたパラメータ推定

本節では，混合 Polya 分布 (多項分布のパラメータが混合ディリクレ分布に従う場合の合成分布) を仮定し，文書中の単語の出現頻度を用いてパラメータを推定する方法を導出する．基本的には混合分布でない Polya 分布に対する推定方法 [12] に EM アルゴリズムを適用した方法である．

$\boldsymbol{y}$ を文脈または文書，$y_v (v = 1, 2, ..., V)$ を $\boldsymbol{y}$ に出現する各単語の出現頻度，$P_{Mul}(\boldsymbol{y}|\boldsymbol{p})$ を $\boldsymbol{p} \in \Delta(V)$ がパラメータである多項分布とすると，混合 Polya 分布 $P_{PM}(\boldsymbol{y}; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M)$ は次のように定義される．なお，式中 2 行目から 3 行目への変形は [6] 参照．

$$\begin{aligned} P_{PM}(\boldsymbol{y}; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M) &= \int P_{Mul}(\boldsymbol{y}|\boldsymbol{p}) P_{DM}(\boldsymbol{p}; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M) d\boldsymbol{p} \\ &= \sum_{m=1}^{M} \lambda_m \int P_{Mul}(\boldsymbol{y}|\boldsymbol{p}) P_D(\boldsymbol{p}; \boldsymbol{\alpha}_m) d\boldsymbol{p} \\ &= \sum_{m=1}^{M} \lambda_m \frac{\Gamma(\alpha_m)}{\Gamma(\alpha_m + y)} \prod_{v=1}^{V} \frac{\Gamma(y_v + \alpha_{mv})}{\Gamma(\alpha_{mv})} \end{aligned} \tag{4}$$

ここで，$y = \sum_v y_v$ である．$\boldsymbol{D} = (\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N)$ とすると，対数ゆう度関数 $\mathcal{L}(\boldsymbol{D}; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M)$ は次のようになる．

$$\mathcal{L}(\boldsymbol{D}; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M) = \sum_{i=1}^{N} \log P_{PM}(\boldsymbol{y}_i; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M) \tag{5}$$

$\boldsymbol{Z} = (z_1, z_2, ..., z_N)$，$z_i$ は $\boldsymbol{y}_i$ を生成したコンポーネントを表す隠れ変数とすると，完全データの対数ゆう度は次のようになる．

$$\mathcal{L}(\boldsymbol{D}, \boldsymbol{Z}; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M) = \sum_{i=1}^{N} \log P(\boldsymbol{y}_i, z_i; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M) \tag{6}$$

EM アルゴリズムを導出するための Q 関数 (「完全データの対数ゆう度」の「隠れ変数の確率」による期待値) は次のようになる．ここで，$P_{im} = P(z_i = m|\boldsymbol{y}_i; \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\alpha}}_1^M)$ とする．$\bar{\lambda}, \bar{\alpha}_1^M$ は現在のパラメータ値である．

$$
\begin{aligned}
Q(\theta|\bar{\theta}) &= \sum_i \sum_m P_{im} \log P(\boldsymbol{y}_i, z_i = m; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M) \\
&= \sum_i \sum_m P_{im} \log \lambda_m \\
&+ \sum_i \sum_m P_{im} \log \frac{\Gamma(\alpha_m)}{\Gamma(\alpha_m + y_i)} \prod_{v=1}^{V} \frac{\Gamma(y_{iv} + \alpha_{mv})}{\Gamma(\alpha_{mv})}
\end{aligned} \tag{7}
$$

ここで，$y_i = \sum_v y_{iv}$ である．右辺の第 1 項と第 2 項をそれぞれ λ_m と α_{mv} ごとに最大化する．λ_m は以下のように更新する．

$$
\lambda_m \propto \sum_i P_{im} \tag{8}
$$

α_{mv} に関しては Minka の fixed-point iteration 法 [12] による単一ディリクレ分布の推定法に EM アルゴリズムを適用し，以下のように最大化する (付録 1).

$$
\alpha_{mv} = \bar{\alpha}_{mv} \frac{\sum_i P_{im} \{\Psi(y_{ik} + \bar{\alpha}_{mv}) - \Psi(\bar{\alpha}_{mv})\}}{\sum_i P_{im} \{\Psi(y_i + \bar{\alpha}_m) - \Psi(\bar{\alpha}_m)\}} \tag{9}
$$

ここで $\bar{\alpha}_{mv}, \bar{\alpha}_m$ は現在の α_{mv}, α_m の値，$\Psi(x)$ はディガンマ関数で，対数ガンマ関数の一階微分である．さらに，高速化する方法として Minka の leaving-one-out 法を用いた推定方法 [12] を混合分布に適用したものが以下である (付録 2).

$$
\alpha_{mv} = \bar{\alpha}_{mv} \frac{\sum_i P_{im} \{y_{iv} / (y_{iv} - 1 + \bar{\alpha}_{mv})\}}{\sum_i P_{im} \{y_i / (y_i - 1 + \bar{\alpha}_m)\}} \tag{10}
$$

上記の推定方法は，特殊関数 ($digamma$ 関数) の計算が必要ないため，高速に推定が行なえる (5 倍以上). 6. の実験で用いたモデルは，すべて leaving-one-out 法を用いて推定した．

3. 予測分布

前節の推定方法によって混合ディリクレ分布を用いた単語出現確率の事前確率が設定された．本節では，文書の前半 (文脈) を観測した状態で，後半の単語出現確率を予測するために，文脈中の単語の出現頻度を多項分布でモデル化する場合のベイズ学習 (適応) を定式化する．混合ディリクレ分布をパラメータの事前分布とする多項分布の事後分布は以下のようになる．ここで，$\boldsymbol{p}$ は多項分布のパラメータ，$\boldsymbol{y}$ は文脈，$y_v (v = 1, 2, ..., V)$ は文脈中の各単語の出現頻度である．

$$
\begin{aligned}
P(\boldsymbol{p}|\boldsymbol{y}) &= \frac{P(\boldsymbol{y}|\boldsymbol{p}) P(\boldsymbol{p})}{\int P(\boldsymbol{y}|\boldsymbol{p}) P(\boldsymbol{p}) d\boldsymbol{p}} \\
&= \frac{P_{Mul}(\boldsymbol{y}|\boldsymbol{p}) P_{DM}(\boldsymbol{p}; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M)}{\int P_{Mul}(\boldsymbol{y}|\boldsymbol{p}) P_{DM}(\boldsymbol{p}; \boldsymbol{\lambda}, \boldsymbol{\alpha}_1^M) d\boldsymbol{p}}
\end{aligned} \tag{11}
$$

これに具体的な関数を代入して計算すると以下のようになる．

$$
P(\boldsymbol{p}|\boldsymbol{y}) = \frac{\sum_{m=1}^{M} B_m \prod_v p_v^{\alpha_{mv} + y_v - 1}}{\sum_{m=1}^{M} C_m}, \tag{12}
$$

ここで

$$
\begin{aligned}
B_m &= \lambda_m \frac{\Gamma(\alpha_m)}{\prod_{v=1}^{V} \Gamma(\alpha_{mv})}, \\
C_m &= B_m \frac{\prod_{v=1}^{V} \Gamma(\alpha_{mv} + y_v)}{\Gamma(\alpha_m + y)}, \\
\alpha_m &= \sum_{v=1}^{V} \alpha_{mv}, \\
y &= \sum_{v=1}^{V} y_v.
\end{aligned}
$$

単語出現確率の事後分布から，単語 w の出現確率の期待値 (予測分布)$P(w^*|\boldsymbol{y})$ を計算すると以下のようになる．ここで，$\delta(k)$ はクローネッカーのデルタ関数で，引数が 0 のとき 1，引数が 0 でない場合に 0 を値として返す．

$$
\begin{aligned}
P(w^*|\boldsymbol{y}) &= \int p_w P(\boldsymbol{p}|\boldsymbol{y}) d\boldsymbol{p} \\
&= \frac{\sum_{m=1}^{M} B_m \int \prod_{v=1}^{V} p_v^{\alpha_{mv} + y_v + \delta(v-w) - 1} d\boldsymbol{p}}{\sum_{m=1}^{M} C_m} \\
&= \frac{\sum_{m=1}^{M} B_m \prod_{v=1}^{V} \frac{\Gamma\{\alpha_{mv} + y_v + \delta(v-w)\}}{\Gamma(\alpha_m + y + 1)}}{\sum_{m=1}^{M} C_m} \\
&= \frac{\sum_{m=1}^{M} C_m \frac{\alpha_{mv} + y_v}{\alpha_m + y}}{\sum_{m=1}^{M} C_m}
\end{aligned} \tag{13}
$$

ちなみに，$\sum_w P(w^*|\boldsymbol{y}) = 1$ であり，動的適応時に C_m を計算するだけで任意の単語の予測分布は高速に求まる．以上のように，予測分布は閉じた式で求めることができる．PLSA にベイズ学習を導入した LDA が，期待値を求めるために繰り返しによる積分近似が必要な点と対照的である．

4. モデル平均

6. の実験で示すように，混合ディリクレ分布を用いた言語モデルは低い混合数で PLSA 等の他のモデルに比べて高い性能 (低いパープレキシティ) を達成できるが，混合数を増やしても性能が向上せず，むしろ悪化する．これは過学習が原因であるので，本節ではモデル平均を用いた方法について述べる．

モデル平均は，混合数等の異なるモデルによる予測分布をその信頼性に応じた重み付けで平均する方法である．同じデータを用いて学習した混合数の異なる N 個の混合ディリクレ分布を考え，文脈 $\boldsymbol{y}$ が与えられると前節で述べた方法で各モデルごとに適応し，N 個の予測分布 $P^i(w^*|\boldsymbol{y}), i = 1, 2, ..., N$ を求める．この際，各モデルによる文脈 $\boldsymbol{y}$ に対する確率 (混合 Polya 分布:$P^i_{PM}(\boldsymbol{y}; \boldsymbol{\lambda}, \boldsymbol{\alpha}), i = 1, 2, ..., N$) が同時に求まる (前節の $\sum_m C_m$)．この確率を各モデルの信頼性と考え，予測分布をこの確率によって重みつき平均したものを方法 1 と呼ぶ．具体的には以下のように定義する．

$$P_{ma1}(w^*|\boldsymbol{y}) = \sum_i \frac{P^i_{PM}(\boldsymbol{y}; \boldsymbol{\lambda}, \boldsymbol{\alpha})}{\sum_j P^j_{PM}(\boldsymbol{y}; \boldsymbol{\lambda}, \boldsymbol{\alpha})} P^i(w^*|\boldsymbol{y}) \tag{14}$$

また，より単純な方法として，以下で定義するように各予測分布を単に算術平均する方法を方法 2 と呼ぶ．

$$P_{ma2}(w^*|\boldsymbol{y}) = \frac{1}{N} \sum_i P^i(w^*|\boldsymbol{y}) \tag{15}$$

5. その他のテキストモデルとの比較

実験結果を述べる前に，混合ディリクレ分布を用いたモデルとその他 2 種 (Mixture of Unigrams [11] と LDA [7]) のモデルとの違いを述べる．ここで，y_v は $\boldsymbol{y}$ 中の単語 w_v の出現回数である．

Mixture of Unigrams $P(\boldsymbol{y}) = \sum_z p(z) P_{Mul}(\boldsymbol{y}|z)$.

LDA $P(\boldsymbol{y}) = \int P_D(\boldsymbol{\theta}|\boldsymbol{\alpha}) \prod_v P(w_v|\boldsymbol{\theta})^{y_v} d\boldsymbol{\theta}$

ここで $P(w_v|\boldsymbol{\theta}) = \sum_z p(z|\boldsymbol{\theta}) p(w_v|z)$

DM $P(\boldsymbol{y}) = P_{PM}(\boldsymbol{y})$.

Mixture of Unigrams と LDA の z はトピックを表す潜在変数である．LDA の $\boldsymbol{\theta}$ は各 unigram モデルの確率的重みであり，ディリクレ分布 $P_D(\boldsymbol{\theta}|\boldsymbol{\alpha})$ によってモデル化されている．混合ディリクレ分布の場合は，2. 4 で述べたように 文書の確率は混合 Polya 分布となる．

図 1 は PLSA を含む 4 種テキストモデルのグラフィカルモデル表現である．各表現の大きな 2 つの四角は，外側が N 個の文脈／文書の集合，内側が各文脈／文書ごとの L 個の単語を表現する．丸は確率変数，二重丸はモデルパラメータ，w は単語，d は文書データ，矢印は変数 (データ) 間の依存関係 (条件付確率) を表す (矢の羽側が条件)．

最も単純なモデルは Mixture of Unigrams である．Mixture of Unigrams は複数のトピックを考え，文書はその中のいずれか一つのトピックから生成されたとする (文書を表す内側の四角の外にトピック変数 z がある)．複数のトピックを同時に含むような文書をモデル化していないため，学習時に過学習しやすい [7]. これを緩和するために，複数トピックから生成された文書をモデル化する手法が PLSA である．しかし，PLSA は文書 d 毎に文書に対する重みを設定してしまうので，未知の文書に対する真の生成モデルとはなっていない．文書 d の分布まで考慮に入れたモデルが LDA である．これは複数トピックを含む文書をモデル化すると共に (z が文書の四角の内側にある)，未知の文書に対する確率を付与できる．しかし，上記の式を見れば分かるように計算は複雑な積分を必要とし，変分法等による積分近似を必要とする [7], [13].

一方，混合ディリクレ分布は，PLSA や LDA とは異なる視点による Mixture of Unigrams の拡張である．PLSA は複数のトピックを含む文書をモデル化することによりモデルの精密さを改良した．これに対し，混合ディリクレ分布は単一トピックモデルのままではあるが，各トピックを単なる unigram モデルではなく，ディリクレ分布でより柔軟にモデル化することにより，モデルの精密さを改良している．

6. 実　　験

文脈に対する動的適応を行なう $n$gram 言語モデルのテストセットパープレキシティによって，混合ディリクレ分布を用いたモデルと LDA を比較した．

学習データは 1999 年版毎日新聞記事 98211 記事，テストセットは 1998 年版毎日新聞から 40 単語以上を含む 495 記事をランダムに選択した．語彙は学習データ中の高頻度 20000 語とした (カバー率 97.1%).

LDA のパラメータ推定には変分 EM 法を用いた [7], [13]. ただし，ディリクレ分布のモデルパラメータ $\boldsymbol{\alpha}$ は，[12] の fixed-point iteration 法を用いた．学習の終了条件は，パープレキシティの減少率が 0.1%以下になった時点とした．

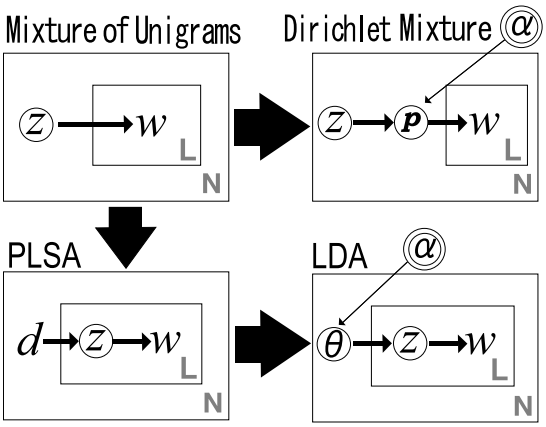


図 1 各種 generative なテキストモデルのグラフィカルモデル表現

Fig. 1 Graphical model expressions of generative text models

混合ディリクレ分布のパラメータ推定には 2.4 で述べた leaving-one-out 法による推定式を用いた．学習の終了条件は LDA と同じく，パープレキシティの減少率が 0.1%以下になった時点とした．混合数は 1,2,5,10,20,50,100,200,500 のモデルを作成した．学習時間は，XEON 2.4GHz の Linux マシンを用いた場合，混合数 500 のモデルで 5 日程度の時間を要した．モデル平均の実験では，混合数 10 から始めて混合数のより多いモデルを 1 つずつ追加し，平均するモデルの数を増加させながらパープレキシティを測定した．例えば，混合数 10,20 の平均，混合数 10,20,50 の平均…である．

はじめに，[7] でとりあげられている文書確率について実験を行った．文書確率は生成的なモデルから計算できる文書全体の確率 $P(\boldsymbol{y})$ である．混合ディリクレ分布の場合は，混合 Polya 分布の確率 $P_{PM}(\boldsymbol{y})$ となる．また，文書確率によるパープレキシティは，文書長を N として，$P(\boldsymbol{y})^{-1/N}$ で計算される．実験の結果を図 2 に示す．'DM' とマークしているのが提案手法である混合ディリクレモデル，'LDA'，'Mixture of Unigrams' とマークされたものが，比較手法である LDA と Mixture of Unigrams である．混合ディリクレ分布を用いたモデルの性能は，Mixture of Unigrams や LDA の性能を大きく上回っていることが確認できる．しかし，混合数を固定した単一の混合モデルでは，混合数 50 で最高性能となり，さらに混合数を増やすとパープレキシティが増加してしまった．これは学習における過適応が原因であると考えられるが，低い混合数の場合で，既に他のモデルより高い性能を示している．

また，ヒストリ適応確率についても実験を行った．ヒストリ適応確率は，文脈が 20 単語増える毎にそこまでの全単語を用いて適応を行ない，次の 20 単語の確率を予測することを繰り返すことで計算することとした．ヒストリ適応確率によるパープレキシティは，

$$pp = \exp\{-\sum_{d=1}^{D}\sum_{i=1}^{N_d}\log P(w_{di}) / \sum_{d=1}^{D} N_d\} \quad (16)$$

ここで D は総文書数，N_d は文書 d 中の総単語数，w_{di} は文書 d 中の i 番目の単語，$P(w_{di})$ は文脈適応を行った w_{di} の予測分布である．文書確率の実験結果からも分かる通り，単一トピックモデルとしての性能として，混合ディリクレモデルが Mixuture of Unigrams を上回っていることは明らかであるため，ヒストリ適応確率の実験では，LDA と混合ディリクレモデルの 2 つのモデルの対比のみを行うこととした．なお，ヒストリ適応確率における，単純なユニグラムモデルのテストセットパープレキシティは 666.64 である。

トライグラム確率は標準のトライグラム確率を文脈モデルで求めたユニグラム確率に比例して増減させる unigram-rescaling 法 [3] を用いて計算した．標準となるトライグラムモデルは毎日新聞 1999 年版の記事を学習データとし，CMU/Cambridge SLM toolkit で構築した back-off (Good-Turing discount) モデルである．なお、標準のトライグラムモデルのテストセットパープレキシティは 75.55 である．

図 3 がユニグラムの比較，図 4 が unigram-rescaling を行ったトライグラムの比較である．'DM' の後に 'ave.1' と書いてあるのがモデル平均の手法 1，'ave.2' と書いてあるのがモデル平均の手法 2 の結果である．モデル平均のグラフの横軸は，使用したモデル集合の中で最も大きな混合数をモデル平均の場合の混合数とした．例えば，モデル平均の場合の混合数 100 の点は混合数 10,20,50,100 の 4 つのモデルでモデル平均した場合のパープレキシティである．

混合ディリクレ分布を用いたモデルの性能は，単純な $n$gram モデルや LDA の性能を大きく上回っていることが確認できる．しかし，文書確率の場合と同様に，10 混合という低い混合数で最高性能に達し，それ以上混合数を増やすと，性能は悪化してしまう．モデル平均を用いれば，パラメータの増加にしたがって

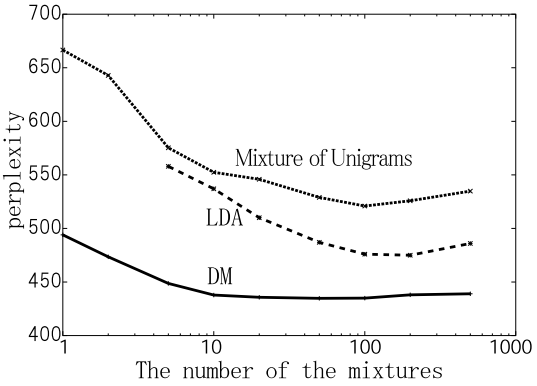


図 2 文書確率によるテストセットパープレキシティの比較

Fig. 2 Comparison of test-set perplexity by document probability

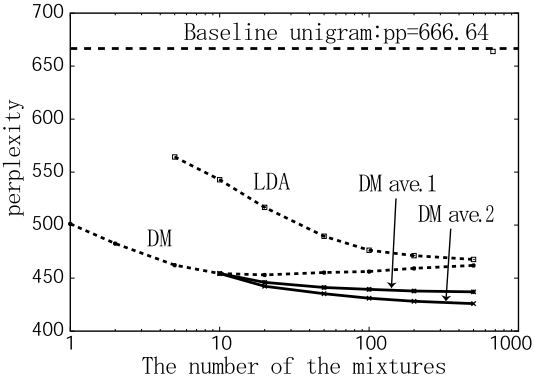


図 3 ヒストリ適応確率によるテストセットパープレキシティの比較 (ユニグラム)

Fig. 3 Comparison of test-set perplexity by history adaptation probability(unigram)

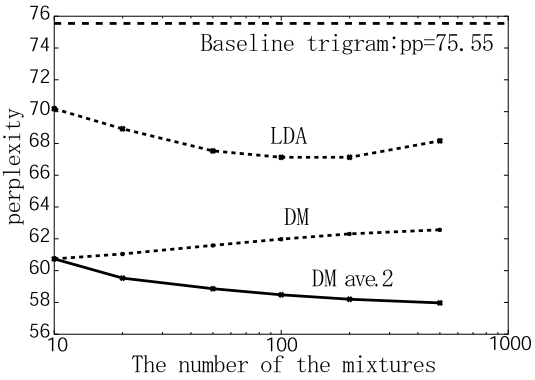


図 4 ヒストリ適応確率によるテストセットパープレキシティの比較 (トライグラム)

Fig. 4 Comparison of test-set perplexity by history adaptation probability(trigram)

単調に性能が向上している．単純な手法 2 が手法 1 よりも高い性能を示した理由は，文脈に対する過適応である。

なお，実験に用いた各モデルのそれぞれの結果のうち，もっとも低いパープレキシティを示した場合について，表 1 にまとめた (表中括弧内はベースラインからのパープレキシティ削減率).

新聞記事を文書データとし, 統計的言語モデルとしての性能を比較した場合ではあるが, 多重トピックモデルである LDA が単一トピックモデルである混合ディリクレ分布よりも, 性能が悪い理由は明らかではない. しかし, 新聞記事の場合, 文書の単語出現確率の分布がある程度固まっており, この分布を事前分布としてうまくモデル化できるか否かによる差が出ている可能性は十分ありえる. すなわち, 複数のディリクレ分布で文書の単語出現確率の分布を直接モデル化する混合ディリクレ分布に対して, LDA は unigram モデルの混合比というやや間接的な変数の分布を単一のディリクレ分布でモデル化するため, 完全に分布をモデル化しきれていない可能性がある. このため, 単語出現確率の分布が定常的でない web データの場合などについては LDA と混合ディリクレ分布の比較についてさらなる検討が必要であるが, 本稿の範囲を超えるため今後の課題としたい.

7. むすび

混合ディリクレ分布を利用した文脈／文書の生成モデルを検討した．2 万単語の語彙で混合数 500 のモデルを約 10 万記事のデータで安定して学習可能なパラメータ推定方法を紹介した．さらにモデル平均の方法を利用することにより混合数 500 まで単調に性能が向上し，表 1 に示すようにテストセット・パープレキシティがユニグラムで約 36%，トライグラムで約 23%，削減できた．これは LDA を用いた場合 (ユニグラムで約 30%, トライグラムで約 11%の削減) よりも高い性能である．今後は提案手法を用いた言語横断検索やスペルチェッカなど，応用分野への展開を検討したい．

文　献

[1] S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. Harshman, “Indexing by latent semantic analysis,” Journal of the American Society for Information Science, vol.41, no.6, pp.391–407, 1990.

[2] T. Hofmann, “Probabilistic latent semantic indexing,” Proc. of the 22nd Annual ACM Conference on Research and Development in Information Retrieval,

表 1 各モデルのパープレキシティ最低値
Table 1 Minimum perplexity of each aspect model

	history adaptation (unigram)	history adaptation (trigram)	document probability
DM	453.06(32.0%)	60.74(19.6%)	434.73
DM ave.2	425.97(36.1%)	57.97 (23.2%)	-
LDA	467.61(29.9%)	67.13 (11.1%)	474.82
Mixture of Unigrams	–	–	520.95

pp.50-57, Berkeley, California, August 1999.

[3] D. Gildea, and T. Hofmann, "Topic-based language models using em," Proc. of the 6th European Conference on Speech Communication and Technology (EUROSPEECH), 1999.

[4] 三品拓也，山本幹雄，"確率的 LSA に基づく ngram モデルの変分ベイズ学習を利用した文脈適応化," 信学技法, NLC-2002-73，pp.13-18，2002.

[5] 高橋力矢，峯松信明，広瀬啓吉，"文脈適応による複数 N-gram の動的補間を用いた言語モデル," 情報処理学会研究報告 NL-155，pp.37–42，2003.

[6] 上田修功，"ベイズ学習 II,IV," 電子情報通信学会誌，vol.85，no.6(pp.421-426),No.8(pp.633-638)，2002.

[7] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," Neural Information Processing Systems, vol.14, 2001.

[8] K. Sjölander, K. Karplus, M. Brown, R. Hunghey, A. Krogh, I.S. Mian, and D. Haussler, "Dirichlet mixtures:a method for improved detection of weak but significant protein sequence homology," Computer Applications in the Biosciences, vol.12, no.4, pp.327–345, 1996.

[9] K.W.Church, and W.Gale, "Poisson mixtures," Natural Language Engineering, vol.1, no.2, pp.163–190, 1995.

[10] R.M. Iyer, and M. Ostendorf, "Modeling long distance dependence in language:topic mixtures versus dynamic cache models," IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, vol.7, no.1, pp.30–39, 1999.

[11] S.T. K. Nigam, A. McCallum, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," Machine Learning, vol.39, no.2/3, pp.103–134, 2000.

[12] T. Minka, "Estimating a Dirichlet distribution," http://www.stat.cmu.edu/ minka/papers/dirichlet/, 2003.

[13] 上田修功，"テキストモデリングの新展開," 言語処理学会第 9 回年次大会チュートリアル資料，pp.1-17，2003.

付　　録

1. fixed-point iteration 法による $\boldsymbol{\alpha}$ の導出

以下は Minka による fixed-point iteration 法を用いた単一のディリクレ分布の推定法 [12] に EM アルゴリズムを適用した導出である．(7) 式の Q 関数における第二項の下限値は，$y_i = \sum_v y_{iv}$, $\alpha_m = \sum_v \alpha_{mv}$, $\bar{\alpha}_m = \sum_v \bar{\alpha}_{mv}$ として，以下の不等式．

$$\frac{\Gamma(\alpha_m)}{\Gamma(\alpha_m + y_i)} \geqq \frac{\Gamma(\bar{\alpha}_m)\exp\left\{(\bar{\alpha}_m - \alpha_m)b_{im}\right\}}{\Gamma(\bar{\alpha}_m + y_i)}, \tag{A·1}$$

$$\frac{\Gamma(\bar{\alpha}_{mv} + y_{iv})}{\Gamma(\bar{\alpha}_{mv})} \geqq c_{imv}\bar{\alpha}_{mv}^{a_{imv}} \quad (if \quad y_{iv} \geqq 1), \tag{A·2}$$

を用いて，次のように与えられる [12].

$$\begin{aligned}
&\sum_i \sum_m P_{im} \log \frac{\Gamma(\bar{\alpha}_m)}{\Gamma(\bar{\alpha}_m + y_i)} \prod_{v=1}^{V} \frac{\Gamma(y_{iv} + \bar{\alpha}_{mv})}{\Gamma(\bar{\alpha}_{mv})} \\
&\geqq \sum_i \sum_m P_{im} \left[\log \frac{\Gamma(\bar{\alpha}_m)\exp\left\{(\bar{\alpha}_m - \alpha_m)b_{im}\right\}}{\Gamma(y_i + \bar{\alpha}_m)} \right. \\
&\quad \left. + \sum_v \log c_{imv}\alpha_{mv}^{a_{imv}} \right] \equiv Q'(\boldsymbol{\alpha})
\end{aligned} \tag{A·3}$$

ここで，

$$\begin{aligned}
a_{imv} &= \left\{\Psi(\bar{\alpha}_{mv} + y_{iv}) - \Psi(\bar{\alpha}_{mv})\right\}\bar{\alpha}_{mv}, \\
b_{im} &= \Psi(\bar{\alpha}_m + y_i) - \Psi(\bar{\alpha}_m), \\
c_{imv} &= \frac{\Gamma(\bar{\alpha}_{mv} + y_{iv})}{\Gamma(\bar{\alpha}_{mv})}\bar{\alpha}_{mv}^{-a_{imv}}.
\end{aligned}$$

この下限値を最大化する α_{mv} の更新式は以下のようになる．

$$\frac{\partial Q'(\boldsymbol{\alpha})}{\partial \alpha_{mv}} = -\sum_i P_{im}b_i + \frac{1}{\bar{\alpha}_{mv}}\sum_i P_{im}a_{imv} = 0 \tag{A·4}$$

$$\alpha_{mv} = \bar{\alpha}_{mv}\frac{\sum_i P_{im}\{\Psi(y_{ik} + \bar{\alpha}_{mv}) - \Psi(\bar{\alpha}_{mv})\}}{\sum_i P_{im}\{\Psi(y_i + \bar{\alpha}_m) - \Psi(\bar{\alpha}_m)\}} \tag{A·5}$$

2. leaving-one-out 法による $\boldsymbol{\alpha}$ の導出

ある単語 v を 1 単語取り除いた文書 $\boldsymbol{y}_i^{-v}$ がデータとして与えられたときの単語 v の予測分布 $p(v^*|\boldsymbol{y}_i^{-v})$ は，(13) 式より，以下のようになる．

$$p(v^*|\boldsymbol{y}_i^{-v}) \fallingdotseq \sum_m P_{im} \frac{\alpha_{mv} + y_{iv} - 1}{\alpha_m + y_i - 1} \qquad \text{(A·6)}$$

P_{im} の値については，1 単語を除外しても影響は少ないため，$P_{im}^{-v} \fallingdotseq P_{im}$ の近似を用いている．この予測分布の対数を，文書集合 $\boldsymbol{y}$ 中のすべての単語に関して求め，和をとった値が LOO 対数ゆう度 $\mathcal{L}_{LOO}$ である．

$$\mathcal{L}_{LOO}(\boldsymbol{y}|\boldsymbol{\alpha}) \fallingdotseq \sum_i \sum_v y_{iv} \log \sum_m P_{im} \frac{\alpha_{mv} + y_{iv} - 1}{\alpha_m + y_i - 1} \qquad \text{(A·7)}$$

ここで，P_{im} はほとんどの場合，1 または 0 の値であるから，LOO 対数ゆう度 $\mathcal{L}_{LOO}$ は次のように変形することができる．

$$\mathcal{L}_{LOO}(\boldsymbol{y}|\boldsymbol{\alpha}) \fallingdotseq \sum_i \sum_v \sum_m y_{iv} P_{im} \log \left(\frac{\alpha_{mv} + y_{iv} - 1}{\alpha_m + y_i - 1} \right) \qquad \text{(A·8)}$$

上式を用いて，m 番目のディリクレ分布に対し，独立に α_{mv} を最大化すればよいことになる．よって以下の不等式 [12]

$$\begin{aligned} \log(n+x) &\geqq q \log x + (1-q)\log n \\ &\quad -q\log q - (1-q)\log(1-q), \qquad \text{(A·9)} \\ q &= \frac{\hat{x}}{n+\hat{x}}, \\ \log(x) &\leqq ax - 1 + \log \hat{x}, \qquad \text{(A·10)} \\ a &= 1/\hat{x}, \end{aligned}$$

から，各 m のコンポーネントにおける LOO 対数ゆう度 $\mathcal{L}_{LOO}^m$ の下限値を得ることができる．

$$\begin{aligned} \mathcal{L}_{LOO}^m &\geqq \sum_i \Big(\sum_v y_{iv} P_{im} q_{imv} \log \alpha_{mv} - y_i P_{im} a_{im} \alpha_m \Big) \\ &\quad + (const.) \qquad \text{(A·11)} \end{aligned}$$

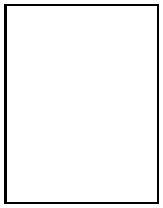
ここで

$$\begin{aligned} q_{imv} &= \frac{\bar{\alpha}_{mv}}{\bar{\alpha}_{mv} + y_{iv} - 1}, \\ a_{im} &= \frac{1}{\bar{\alpha}_m + y_i - 1}. \end{aligned}$$

である．得られた LOO 対数ゆう度の下限値に対し，fixed-point iteration 法を用いることで，以下の更新式が得られる．

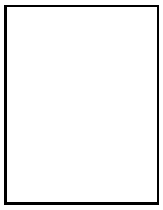
$$\alpha_{mv} = \bar{\alpha}_{mv} \frac{\sum_i P_{im}\{y_{iv}/(y_{iv} - 1 + \bar{\alpha}_{mv})\}}{\sum_i P_{im}\{y_i/(y_i - 1 + \bar{\alpha}_m)\}} \qquad \text{(A·12)}$$

（平成 x 年 xx 月 xx 日受付）



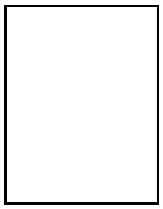
貞光　九月　（学生員）

平成 16 年筑波大学第三学群情報学類卒業．現在筑波大学大学院システム情報工学研究科在学中．



三品　拓也

平成 16 年筑波大学大学院修士課程修了．現在日本アイ・ビー・エム (株) 東京基礎研究所勤務．



山本　幹雄　（正員）

昭和 61 年豊橋技術科学大学大学院修士課程修了．同年 (株) 沖テクノシステムズラボラトリ研究開発員．昭和 63 年豊橋技術科学大学情報工学系教務職員．平成 4 年同助手．平成 7 年筑波大学電子・情報工学系講師．平成 10 年同助教授．博士 (工学)．自然言語処理，音声言語情報処理の研究に従事．情報処理学会，言語処理学会，人工知能学会，音響学会，ACL 各会員．