階層 Pitman-Yor 過程に基づく可変長 n-gram 言語モデル

持橋大地 †,☆ 隅田 英一郎 †

本論文では, n-gram 分布の階層的生成モデルである階層 Pitman-Yor 過程を拡張することで, 各単語の生まれた隠れたマルコフ過程のオーダーを自動的に推定し, 適切な文脈を用いる可変長 n-gram 言語モデルを提案する. 無限の深さをもつ予測接尾辞木上の確率過程を考えることにより, 句を確率的に発見し, 適切な文脈長を学習することができる. これにより, 従来不可能だった高次 n グラムの学習が可能になる. 本手法は言語モデルだけでなく, マルコフモデル一般について, そのオーダーをデータから推定できる可変長生成モデルとなっている. 英語および日本語の標準的なコーパスでの実験により, 提案法の有効性を確認した.

Bayesian Variable Order n-gram Language Model based on Hierarchical Pitman-Yor Processes

Daichi Mochihashi ? and Eiichiro Sumita ?

This paper proposes a variable order n-gram language model by extending a recently proposed model based on the hierarchical Pitman-Yor processes. Introducing a stochastic process on an infinite depth prediction suffix tree, we can infer the hidden n-gram context from which each word originated. Experiments on standard large corpora showed validity and efficiency of the proposed model. Our architecture is also applicable to general Markov models to estimate their variable orders of generation.

1. はじめに

単語間のマルコフ過程によって文の確率を計算する n グラムモデルは, Shannon の歴史的な論文[1] で最初に導入されて以来, 自然言語処理の様々な場面に適用され, その有効性が示されてきた, 基礎的で重要な方法である.

n グラムモデルは, 直前の $(n-1)$ 個の単語列を状態とした $(n-1)$ 次のマルコフモデルにより, 次の単語の条件つき確率を計算していく. このとき, 状態数は単語の総数を V とすると V^{n-1} のオーダーとなり, n を 1 増やすと総パラメータ数は通常数万倍となり, 指数的に爆発する. このため, 様々なスムージング法を用いた場合でも, 通常 $n=3$ (トライグラム) から最大でも $n=4,5$ 程度が限界であり, それ以上の長い相関は実際上取り扱えないという問題があった.

しかしながら, 現実の言語データには “The United States of America” のように, トライグラムを超える長いフレーズや固有名詞が頻出する. これらをチャンクとして分類し, 一単語とみなす方法もあるが, これには教師データが必要なために主観に依存する上, 慣用句のような系列を全てカバーすることは難しいという問題がある. ☆☆ 特に, 日本語や中国語のように単語境界が曖昧な言語の場合, 品詞体系によっては短い助詞の連続などによる長い n グラムが頻出する可能性があり, 「単語分割の粒度に依存しない言語モデル」が特に重要だと考えられる.

また逆に, “longer than” のように短い n グラムで充分な文法的関係も多いことを考えると, n を常に 3 などの固定値とするのではなく, 文脈に応じて必要なだけの長さを用いる「可変長 n グラム言語モデル」の意義は, 言語学的にみても高いと考えられる.

しかしながら, これまで提案されてきた “可変長 n グラムモデル” はいずれも, 実際には最大オーダーの n グ

† ATR 音声言語コミュニケーション研究所 / (独) 情報通信研究機構
ATR Spoken Language Communication Research Laboratories / National Institute of Communications Technology
☆ 現在, NTT コミュニケーション科学基礎研究所
Presently with NTT Communication Science Laboratories

☆☆ 何が句であるかには正解はなく, 確率的にしかとらえられないと考えている[2]. さらに, この区切りは固定ではなく, 一般に文脈にも依存する.

ラムを最初に作り, それを枝刈りするか[3)4)], または頻度閾値でカットオフを行うもの[5)] であった. ☆ この際の閾値や MDL などの基準はモデルとは別に外部から与えるものであり, さらに, 指数的に大きくなる最大モデルを事前に作る必要がある点で, 可変長モデルの構成意図と矛盾していた. 可変長マルコフモデルはバイオインフォマティクス[6)] および統計学[7)] の分野でも最近提案されているが, これらも同じ問題点を持っている.

これまで, この問題に理論的な解決が存在しなかった理由は, nグラム分布を階層的に生成する確率モデルが一般的でなかったためだと考えられる. ☆☆ しかし, 最近 10), 11) により, 階層 Pitman-Yor 過程とよばれるノンパラメトリックな確率過程によって, 高精度にスムージングされたnグラム分布を 0-gram→1-gram→2-gram→ $\cdots$ とベイズ統計の枠組から階層的に生成および推定できることが明らかになった.

そこで本論文では, この階層 Pitman-Yor 過程によるnグラムモデルを拡張し, データ中の各単語が生成されたnグラム長を隠れ変数とみなしてベイズ推定を行うことで, 文脈により様々にオーダーの異なる可変長nグラムの生成モデルを提案する. この方法により, nグラムモデルの n を指定せず, 原理的には無限長とすることができ, 従来不可能だった高次nグラムの推定が可能となる. また, 言語モデルの副産物として, 上で述べた可変長の「句」を教師なし学習の枠組から, 確率的に推定することができる.

以下ではまず 2 節で, 階層 Pitman-Yor 過程によるnグラム言語モデルについて説明し, 3 節で Suffix Tree 上に確率過程を考えることにより, これを可変長に拡張する. 4 節で LDA を用いたそのトピック適応化について述べ, 5 節で実験結果を示す. 6 節で考察および関連研究について述べ, 7 節で全体のまとめと将来の展望についてふれる.

2. 階層 Pitman-Yor 過程と n-gram 言語モデル

階層 Pitman-Yor 過程によるnグラム言語モデル (HPYLM)[10)11)] は, 最近提案されたnグラム分布の高精度な階層的生成モデルであり☆☆☆, 後に説明するように, 現在最高性能といわれる言語モデルの Kneser-Ney スムージング[12)] は, この確率過程の近似となっている.

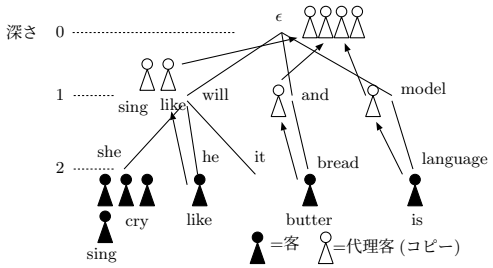


図 1 階層的 CRP の Suffix Tree 上の表現. 客一人一人が, モデル上のカウントを表している.

Fig. 1 Suffix tree representation of a hierarchical Chinese Restaurant Process. Each customer corresponds to a count in the model.

階層 Pitman-Yor 過程は, 確率論の分野では 2-パラメータポアソン=ディリクレ過程[13)] とよばれている Pitman-Yor 過程を階層化したものであり, 階層ディリクレ過程[14)] の拡張と考えることができるが, ここでは直接測度論に基づく式ではなく, それと等価な階層的な CRP (中華料理店過程)[15)] を使って, 直感的に説明する.

例として, トライグラムの言語モデルを考えよう. これは**図 1** のように, 深さ 2 の Prediction Suffix Tree (予測接尾辞木)[25)] で表すことができる. いま, 文脈 'she will' に続いて 'sing' を予測したいとき, この木をユニグラムに対応する根 ϵ から, $\epsilon \rightarrow \text{will} \rightarrow \text{she}$ の順に枝をたどり, 到達した 'she will' に対応するノードの持つカウント分布を用いて, $p(\text{sing}|\text{she will})$ を計算する. 予測接尾辞木はコーパス上の接尾辞に対する木であるが, 通常の接尾辞木 ($\epsilon \rightarrow \text{she} \rightarrow \text{will}$) と異なり, シンボルを後ろからたどるために, ラベルの順番が逆になっていることに注意されたい. 簡単のため, 以下では Prediction Suffix Tree のことをすべて Suffix Tree と表記する.

HPYLM の学習を行う際には, 根だけの初期状態の木から始め, 学習データのトライグラムについて, その 1 つ 1 つのカウント (CRP では, 可算無限個のテーブル=語彙数を持つ中華料理店のメタファーから, このカウントを「客」と呼ぶ) を図 1 のように, 深さ 2

☆ 頻度による一様なカットオフは, 性能がきわめて悪いことが示されている[4)].

☆☆ 先行研究として, nグラムの事前分布に階層的なベータ分布を考える研究[8)] や, 階層ディリクレ過程を用いる研究[9)] があったが, これらによって得られる加法的なスムージングは 10) で示されているように, 他の高精度なスムージング法に比べて精度が低いという実用上の問題を持っていた.

☆☆☆ これまでのnグラムの確率モデルは各文脈での頻度をディスカウントした後, 和を 1 に正規化して確率とするものであり, その頻度の由来を問うものではなかった. また, 各 1-グラム, 2-グラム,$\cdots$ のカウントとその確率は, 基本的に独立に計算されていた.

の対応するノードに, 必要に応じて新しくノードを作りながら追加してゆく.

ここで一般には, どのノードにも全ての種類の単語のカウントがあるわけではない. そこで, 客をノードに追加する時, ある確率で☆ その客のコピー (代理客) が親ノードに客として送られる. たとえば, ノード 'she will' に客 'like' がいなくても, 姉妹ノード 'he will' に 'like' がいれば, そのコピーが共通の親ノード 'will' に送られているため, 確率 $p(\text{like}|\text{she will})$ は 3-gram 確率 $p(\text{like}|\text{she will})$ (これは 0) と, 親ノードの持つ 2-gram 確率 $p(\text{like}|\text{will})$ を適切に補間して計算される. この過程は再帰的に行われ, 親ノードに客 'like' が追加される際にもコピーがその親ノードに送られ, 2-gram 確率は 1-gram 確率との補間で計算されるため☆☆, 直接親ノードにない語についても, 必ず確率を求めることができる.

結果として HPYLM では, 文脈 $h = w_{t-n} \cdots w_{t-1}$ に続く単語 w の確率は,

$$p(w|h) = \frac{c(w|h) - d \cdot t_{hw}}{\theta + c(h)} + \frac{\theta + d \cdot t_{h\cdot}}{\theta + c(h)} p(w|h') \quad (1)$$

として, 再帰的に計算される.

ここで $c(w|h)$ はノード h での w のカウント, $c(h) = \sum_w c(w|h)$ はその総和であり, $h' = w_{t-n+1} \cdots w_{t-1}$ は 1 つオーダーを落とした文脈である.

t_{hw} は w が $p(w|h)$ からではなく, 親ノード $p(w|h')$ から生成されたと推定された回数であり, $\sum_w t_{hw} = t_{h\cdot}$ と書いた. d, θ は Pitman-Yor 過程のパラメータであり, Suffix Tree 上のすべての客の分布から, それぞれベータ事後分布, ガンマ事後分布によって推定できる[10]. これらの統計量はすべて, 末端ノードに追加されるカウント (客) から, 再帰的に親ノードに送られる代理客の数とその配置に依存するが, それらは相互依存関係にあるため, 推定には MCMC 法[16] の一種であるギブスサンプリングを用いる. ランダムに選んだ客をその代理客を含めていったん削除し, 新しく追加し直してその代理客の配置を再サンプリングすることを繰り返すことで, 代理客の配置が真の分布からのサンプルに収束する. 図 4 を参照されたい.

式 (1) は, カウント $c(w|h)$ をディスカウントした分布を 1 つ低いオーダーの分布と補間した確率になっているが, $t_{hw} \equiv 1$ とすると, この式は Kneser-Ney スムージングと一致し, Kneser-Ney スムージングはこの確率過程のヒューリスティックな近似であることがわかる.

☆ (1) 式において, 単語が親ノードである第二項から生成されたと判断された場合.

☆☆ 1-gram 確率はさらに, 0-gram 確率 (全ての確率が $1/V$) と補間される.

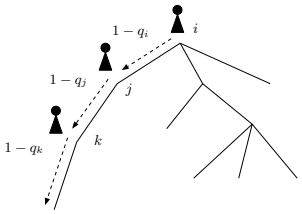


図 2 Suffix Tree に客を追加する確率過程. $(1-q_i)$ はノード i の持つ「通過確率」を表す.

Fig. 2 A stochastic process to add a customer to the suffix tree. $(1-q_i)$ means a "penetration probability" of node i.

さて, ここでの問題は, 図 1 において客がすべて, 深さ $(n-1)$ のノードに到着するとしていることである. 実際には, 自然言語の持つ Suffix Tree はある所で浅く, ある所は非常に深い, **図 2** のような形をしており, 言語はさまざまに長さの違う n グラム文脈から生成されていると考えられる. それでは, 自然言語の持つこのような木をどのようにして推定すればよいのだろうか.

3. 可変長階層 Pitman-Yor 過程

3.1 Suffix Tree 上の確率過程

そこで我々は, Suffix Tree の各ノード i に, 木を根からたどる時にそこで止まる確率 q_i (すなわち, $(1-q_i)$ はノード i の「通過確率」を表す) があると考え, 各 q_i は $[0,1]$ 上の最も自然な確率分布として, 共通のベータ事前分布

$$q_i \sim \text{Be}(\alpha, \beta) \quad (2)$$

から生成されていると仮定する.

ここで, $\text{Be}(\alpha,\beta) = \Gamma(\alpha+\beta)/(\Gamma(\alpha)+\Gamma(\beta)) \cdot q^{\alpha-1}(1-q)^{\beta-1}$ は二項確率 q の確率分布であるベータ分布の確率密度関数であり, 期待値は $E[q] = \alpha/(\alpha+\beta)$ である. **図 3** に, ハイパーパラメータ (α, β) の違いによるベータ分布の例をいくつか示す.

文脈 $h = w_{t-\infty} \cdots w_{t-2} w_{t-1}$ に続いて単語 w_t が観測されたとき, 我々は Suffix Tree を根 ϵ から始めて,

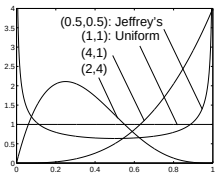


図 3 ベータ分布の例とハイパーパラメータ.

Fig. 3 Examples of Beta distributions with different hyperparameters.

$\epsilon \to w_{t-1} \to w_{t-2} \to \cdots$ の順にノードをたどって降りていくが, この時, 必ず深さ $(n-1)$ で止まるのではなく, このパス上の q_i をそれぞれ $q_0, q_1, q_2, \ldots$ として, 確率

$$p(n=l|h) = q_l \prod_{i=0}^{l-1} (1-q_i) \tag{3}$$

に従って深さ l で停止し, そこに客を追加する (図 2).

この式からわかるように, 非常に深いノードであっても, そのパスに沿った q_i が小さければ (すなわち, 「通過確率」が高ければ) 深い n グラムが使われ, 逆に浅いノードでも, q_i が大きければ (「通過確率」が低ければ) そこで止まる確率が大きくなる. 式 (3) より, 深さ n のノードに到達する確率は n に従っておおよそ指数的に減少するが, その度合は木の枝によって異なり, 高頻度の長い系列に対応する深いノードを許すことのできるモデルとなっている. ☆

3.2 可変長ベイズ n-gram 言語モデル

もちろん, われわれは言語の Suffix Tree のノードが持つ真の q_i の値は知らない. それでは, どうやって q_i を推定すればよいだろうか.

ここで, 上の可変長生成モデルでは, データ $\mathbf{w} = w_1 w_2 \cdots w_T$ について, それぞれの単語が生成された隠れた n グラム長 $\mathbf{n} = n_1 n_2 \cdots n_T$ が存在していると仮定していることに注意したい. したがって, われわれのモデルでは, データ $\mathbf{w}$ の確率は

$$p(\mathbf{w}) = \sum_{\mathbf{n}} \sum_{\mathbf{s}} p(\mathbf{w}, \mathbf{n}, \mathbf{s}) \tag{4}$$

と表すことができる.

ここで $\mathbf{s} = s_1 s_2 \cdots s_T$ であり, s_t はそれぞれ, 単語 w_t から送られた代理客の配置を表す隠れ変数である[10]. この $\mathbf{n}, \mathbf{s}$ は, 代理客の配置を確率的に最適化する, HPYLM のギブスサンプリングを拡張することで推定することができる.

具体的には, 単語 w_t の持つ隠れた n グラムオーダー n_t を,

$$n_t \sim p(n_t|\mathbf{w}, \mathbf{n}_{-t}, \mathbf{s}_{-t}) \tag{5}$$

のようにサンプリングして更新していく. (s_t も, この後同時にサンプリングを行う.) ここで $\mathbf{n}_{-t}$, $\mathbf{s}_{-t}$ はそれぞれ, $\mathbf{n}, \mathbf{s}$ から n_t, s_t を除いたベクトルである.

式 (5) から n_t をサンプリングする際, われわれは他の客 (単語) が Suffix Tree をたどった深さ $\mathbf{n}_{-t}$ を全て知っていることに注意されたい. したがって, 他の客がノード i で止まった回数を a_i, 通過した回数を b_i とおくと, q_i の期待値は, ベータ事後分布の期待値として

$$E[q_i] = \frac{a_i + \alpha}{a_i + b_i + \alpha + \beta} \tag{6}$$

と推定される. また, 式 (5) はベイズの定理から

$$\begin{aligned} &p(n_t|\mathbf{w}, \mathbf{n}_{-t}, \mathbf{s}_{-t}) \\ &\propto p(w_t|\mathbf{w}_{-t}, \mathbf{n}, \mathbf{s}_{-t})\, p(n_t|\mathbf{w}_{-t}, \mathbf{n}_{-t}, \mathbf{s}_{-t}) \end{aligned} \tag{7}$$

と展開できるが, この第一項は n グラムオーダーが n_t と決まったときの w_t の n グラム確率であり, 式 (1) から求められる. 第二項は, この文脈で深さ n_t のノードに到達する事前確率であり, 式 (3) および式 (6) から,

$$\begin{aligned} &p(n_t = l|\mathbf{w}_{-t}, \mathbf{n}_{-t}, \mathbf{s}_{-t}) \\ &= \frac{a_l + \alpha}{a_l + b_l + \alpha + \beta} \prod_{i=0}^{l-1} \frac{b_i + \beta}{a_i + b_i + \alpha + \beta} \end{aligned} \tag{8}$$

のように計算することができる.

これらの確率を用いて, 各単語の持つ隠れた n グラムオーダーをサンプリングする, 可変長階層 Pitman-Yor 言語モデル (VPYLM) のギブスサンプラを構成することができる (**図 4**).

このアルゴリズムは HPYLM のギブスサンプリングの拡張であり, 単語 w_t に対応して Suffix Tree の order$[t]$ の深さにいる客を一人, 根からノードをたどって削除し, a_i または b_i をパスに沿って 1 ずつ減らす. 新しい n グラムオーダーをサンプリングした後, 客を新しい深さに追加し直して a_i または b_i を 1 ずつ増やし, この深さを order$[t]$ に記録する. 以上を繰り返す. ☆☆ order$[t]$ をサンプリングせず, 常に定数 $(n-1)$ である場合が, n グラムの HPYLM のギブスサンプリングに相当している.

式 (7) から n_t をサンプリングする際, 計算量的な問題から, 実際にはある最大値 $n_{\max}$ を設定してその中でサンプリングを行うか, または n_t の事前確率 (8) がある小さな値 ϵ 以下になるまで計算を行う. このとき, n グラムモデルにおいて 'n' は必要なくなり, われわれは原理的に, n をベイズ的に積分消去した「∞-グラム言語モデル」を得ることができる.

3.3 予測と文脈長確率分布

予測の際にも, われわれは従来のように n グラム長を固定していないため, n を隠れ変数とみなして, 文脈 $\mathbf{h} = w_{-\infty} \cdots w_{-2} w_{-1}$ に対して

$$\begin{aligned} p(w|\mathbf{h}) &= \textstyle\sum_n p(w, n|\mathbf{h}) && (9) \\ &= \textstyle\sum_n p(w|n, \mathbf{h}) p(n|\mathbf{h}) && (10) \end{aligned}$$

☆ 式 (2)(3) の形からわかるように, これは Suffix Tree 上で定義された, Beta two-parameter process の Stick-breaking process[15] である.

☆☆ このとき, 同時に親ノードの代理客が再帰的に, 確率的に削除/追加される. (s_t のサンプリングに相当する.)

```
For j = 1 ··· N {
  For t = randperm(1 ··· T) {
    if (j > 1) then
      remove_customer (order[t], w_t, w_{1:t−1});
    order[t] = add_customer (w_t, w_{1:t−1});
  }
}
```

図 4 VPYLM のギブスサンプラ.
Fig. 4 The Gibbs sampler of VPYLM.

のように予測を行う．ここで第二項は文脈 $\mathbf{h}$ の持つ n グラム文脈長分布であり, 式 (8) で与えられる．第一項はオーダーを n とした HPYLM の予測確率であり, (1) 式で求められるが, この確率はすでに Kneser-Ney スムージングと同等に階層的にスムージングされていることに注意されたい．すなわち, VPYLM の予測確率は, HPYLM の n グラム確率をさらに $n = 0, 1, \cdots, \infty$ の文脈事後分布で混合したものになっている．実際には (10) をさらに, 訓練データ $\mathbf{w}$ からの $\mathbf{n}, \mathbf{s}$ の N 個の Gibbs 事後サンプルで平均化して予測を行う．☆

3.4 実　　装

ギブスサンプリングの際, Suffix Tree 上で数千から数万にのぼることがある子ノードをたどる操作を高速化するため, われわれは 5) と同様に, 子ノードの管理にスプレー木[18] を使用した．スプレー木は $O(\log n)$ で子ノードを探索可能な二分順序木である．アクセスの際に木を自己組織的に最適化することで, 高頻度なアイテムを高速に探索できるため, 自然言語のように Power Law を持つ系列での動的なデータ構造として特に適している．

われわれはこのスプレー木を子ノードだけでなく, 各ノードの持つ予測単語☆☆の管理にも用いた．**図 5** に, Suffix Tree の各 n グラムノードに使用したデータ構造を示す.

```
struct ngram {          /* n-gram node */
    ngram *parent;      /* parent node */
    splay *children;    /* = (ngram **) */
    splay *words;       /* = (restaurant **) */
    int stop;           /* a_h */
    int through;        /* b_h */
    int ncounts;        /* c(h) */
    int ntables;        /* t_h. */
    int id;             /* word id */
};
```

図 5 n グラムノードのデータ構造.
Fig. 5 Data structure of a n-gram node.

4. 可変長ベイズ n-gram 言語モデルのトピック適応

可変長 n グラムモデルについて, そのベイズ生成モデルが得られたので, 次にそのトピック適応化を自然に考えることができる.

4.1 Latent Dirichlet Allocation (LDA)

このためのモデルとして, 我々は LDA[19] を使用した．LDA では, 各文書に, 隠れたトピック混合分布 $\theta_d = (\theta_1, \theta_2, \ldots, \theta_M)$ があり, θ がディリクレ事前分布

$$\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\gamma}) = \frac{\Gamma(M\gamma)}{\Gamma(\gamma)^M} \prod_{t=1}^{M} \theta_t^{\gamma-1} \qquad (11)$$

から生成されていると仮定する．簡単のため, 上ではディリクレ分布のハイパーパラメータは全て同じ値 γ とした．このとき, 文書 $d = w_1 w_2 \cdots w_N$ の各単語は, 次のようにして生成される.

(1) $\boldsymbol{\theta}_d \sim \mathrm{Dir}(\boldsymbol{\gamma})$ をサンプル.
(2) For $n = 1 \ldots N$,
  a. $t \sim \mathrm{Mult}(\boldsymbol{\theta}_d)$ をサンプル.
  b. $w_n \sim p(w|t)$ をサンプル.

ここで, トピック言語モデル $p(w|t)$ として 19) ではユニグラム分布が使われているが, LDA は混合モデルであるから, これは任意の分布に置き換えることができ, ここではトピックごとの VPYLM とすることができる．しかし予備実験の結果, トピック毎の VPYLM を混合する方法では, トピックを考慮しない時に比べて予測精度が悪化することがわかった.

この理由は, LDA のギブスサンプリングによりデータを分割し, トピック別の n グラム言語モデルを構築すると, n グラムのスパース性がより深刻になるためだと考えられる．トピックを動的に推定することによる精度上昇より, 選択されたトピック言語モデルのカバレージ不足による精度悪化の方が問題となるからである.

4.2 ギブスサンプリングによるモデル推定

そこでわれわれは, VPYLM において, データの多いユニグラム分布のみを混合分布 (混合 Pitman-Yor 測度) とすることとした．LDA によって決まるトピック分布 $\boldsymbol{\theta}_d$ に従い, 単語ごとに異なるユニグラム分布を用いた可変長 n グラム分布から単語が生成される.

このとき, 階層 Pitman-Yor 過程によって生成される n グラムカウントは, テーブル☆☆☆ごとに生成され

☆ これは, Gibbs サンプリングが事後分布からの正確なサンプリングであり, 事後確率最大化を行うものではないからである．ディリクレ過程混合モデルに対して最近提案されたモデル探索法[17] は, この意味で興味深い.
☆☆ CRP においては,「レストラン」と呼ばれている.
☆☆☆ 各 n グラムノード h でのカウントは, 単語ごとのテーブルの一つに加算される．このテーブルは, 単語 w ごとに t_{hw} 個存在する.

たユニグラム分布が異なるため, ギブスサンプリングの際, カウントがどのテーブルに追加され, その親テーブルは何かを明示的に追跡する必要がある.

この情報を用いて, VPYLM の LDA 化 (VPYLDA) のギブスサンプリングは, **図 6** のように行うことができる. ここで draw_topic は, 単語 w_t がトピック k に属する事後確率

$$p(k|w_t, w_{1:t-1}) \propto p(w_t|w_{1:t-1}, k) \cdot p(t|\theta_d) \quad (12)$$
$$\propto p(w_t|w_{1:t-1}, k) \cdot (n^d_{-t,k} + \gamma) \quad (13)$$

から k をサンプルする関数であるが, 各単語のもつ n グラムオーダー order[t] (テーブル table[t] が Suffix Tree の中で存在する深さによって, 陰に表現されている) に加え, その属するトピック topic[t] を同時に隠れ変数として持ち, サンプリングを行う点が VPYLM のギブスサンプリングと異なっている. 上式で, $p(w|h, k)$ はユニグラム分布 k を用いた VPYLM の予測確率 (10), $n^d_{-t,k}$ は文書 d 中でトピック k に割り当てられた単語数 (w_t を除く) を表す.

```
For j = 1 ··· N {
  For t = randperm(1 ··· T) {
    d = document(w_t);
    k = topic[t];
    if (j > 1) then
      remove_customer (table[t]);
    k = draw_topic (w_t, w_{1:t-1}, d);
    table[t] = add_customer (w_t, w_{1:t-1}, k);
    topic[t] = k;
  }
}
```

図 6 VPYLDA のギブスサンプラ.
Fig. 6 The Gibbs sampler of VPYLDA.

5. 実　　験

5.1 VPYLM on Text Corpora

英語および日本語の標準的な大規模コーパスを用いて実験を行った. ☆

5.1.0.1 デ ー タ

英語については, 21), 22) 等で使われている NAB (North American Business News) コーパスの WSJ セットよりランダムに選択した 409,246 文, 10,007,108 語を訓練データ, さらに 10,000 文を評価データとした. 単語はすべて小文字とし, ノイズの可能性の高い, 全体で頻度 10 未満の単語は特別な未知語とみなした.

表 1 英語および日本語のコーパスでのテストセットパープレキシティとモデルの持つノード数. N/A はメモリオーバーフローを表す.
Table 1 Test-set perplexities with the number of nodes in the model in English and Japanese corpora. N/A means a memory overflow.

(a) 英語 (NAB コーパス)

n	HPYLM	VPYLM	Nodes(H)	Nodes(V)
3	113.60	113.74	1,417K	**1,344K**
5	101.08	101.69	12,699K	**7,466K**
7	N/A	**100.68**	N/A	**10,182K**
8	N/A	**100.58**	N/A	**10,434K**
∞	—	117.65	—	**10,392K**

(b) 日本語 (毎日新聞コーパス)

n	HPYLM	VPYLM	Nodes(H)	Nodes(V)
3	78.06	78.22	1,341K	**1,243K**
5	68.36	69.35	12,140K	**6,705K**
7	N/A	68.63	N/A	**9,134K**
8	N/A	68.60	N/A	**9,490K**
∞	—	80.38	—	**9,561K**

総語彙数は 26,497 語である. 日本語については毎日新聞 2000 年度のテキスト (約 150 万文) からランダムに選んだ 520,000 文, 10,079,410 語を訓練データ, 10,000 文を評価データとした. MeCab [23] で分かち書きを行い, 頻度 10 未満の語をまとめて, 総語彙は 32,783 語となった.

5.1.0.2 学 習 設 定

予備実験の結果と計算量的制約から, HPYLM, VPYLM のそれぞれのモデルについて $N = 200$ 回のギブスサンプリングを行い, さらに 50 回の事後サンプルを用いて評価を行った. Suffix Tree の形を決めるベータ事前分布のハイパーパラメータは $(\alpha, \beta) = (4, 1)$ としたが, これはより深い木が得られる $(1, 1)$ の一様分布を用いた場合等とほとんど性能差がなかったためである. (α, β) の値による性能差とその最適化については, 6 節を参照されたい. 実験はすべて Xeon 3.2GHz, メモリ 4GB の Linux 上で行った.

5.1.0.3 結　　果

表 1 に, HPYLM と VPYLM のパープレキシティをモデル中の n グラムノード数とともに示す. モデル次数 n は前者では固定, 後者では最大値 $n_{\max}$ を意味する. ☆☆

この結果から, VPYLM は HPYLM とほぼ同等の性能を40 %以上少ないノード数で達成し, HPYLM では推定できない $n = 7, 8$ のような高次 n グラムにつ

☆ 中国語についても, Sinica バランスドコーパス[20] を用いて漢字単位の言語モデルを構築して実験を行い, 同様の結果を確認している. ($n = 5$ のときパープレキシティ 43.39.)

☆☆ 表 1 には示されていないが, Modified Kneser-Ney スムージングを用いた SRILM [24] による英語のパープレキシティは, $n = 3, 5, 7, 8$ についてそれぞれ 118.91, 107.99, 107.24, 107.21 であり, HPYLM および VPYLM はこれより優れた性能を達成している.

いても, 必要なもののみを選択的にモデルに加えることで推定が可能であり, より高い性能を持つことがわかる.

また, 同じ (最大) オーダーnの場合でも, VPYLM は HPYLM より 20%程度学習が高速である. これは, VPYLM において n グラムオーダーをサンプリングする計算コストよりも, 不必要に深いノードを追加しないことによる計算量の削減が大きいからだと考えられる. **図 7** に, 8-グラム VPYLM において推定された n グラムオーダーの, データ全体での分布を示す. 文脈長を長くするメリットと, 深いノードに到達するコストの間で適切なトレードオフが行われ, $n = 3, 4$ 程度をピークに指数的な減衰が起きていることがわかる.

5.2 Suffix Tree と確率的句

提案法において, 式 (9) の $p(w, n|\mathbf{h})$ は $\mathbf{h} = w_{-\infty} \cdots w_{-2} w_{-1}$ の中で最後の n 語である $w_{-n} \cdots w_{-1}$ を文脈として単語 w が生成された確率であり, すなわち, $w_{-n} \cdots w_{-1} w$ が「句」をなす確率と考えることができる. たとえば, “america” が文脈 “the united states of” から生成されたとき, “the united states of america” は句をなし, これに (必ずしも, 経験確率のように長さに従って減衰しない) 確率を割り当てることができる.

これは木の根からユニグラムで単語 w を生成する代わりに, 子ノードを確率的に $w_{-1} \to w_{-2} \to \cdots \to w_{-n}$ とたどった後で w を出力し, 句 $w_{-n} \cdots w_{-1} w$ が生成されたと考えてもよく, Suffix Tree を深さ優先で探索することで, すべての句とその確率を効率的に計算することができる. **図 8** に, 8-グラム VPYLM から得られた「確率的句」の例を, 上の確率でソートして示す.

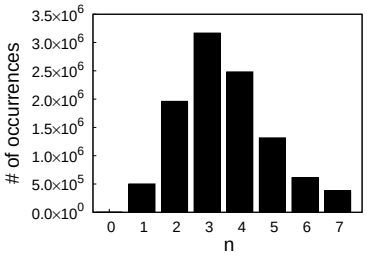


図 7 8 グラム VPYLM で推定されたnグラム文脈長のデータ全体での分布. $n=0$ がユニグラム, $n=1$ がバイグラム,... である. 実際にはさらに式 (1) により, 低次の n グラムと階層的にスムージングされることに注意.

Fig. 7 Global n-gram context length distribution inferred by a 8-gram VPYLM. $n=0$ is a unigram, $n=1$ is a bigram, and so on. Each n-gram is further interpolated by lower-order n-grams recursively through the equation (1).

$p(w, n)$	Stochastic phrases in the suffix tree
0.9784	primary new issues
0.9726	^ at the same time
0.9556	american telephone &
0.9512	is a unit of
0.9394	to # % from # %
0.9026	from # % in # to # %
0.8896	in a number of
0.8831	in new york stock exchange composite trading
0.8696	a merrill lynch & co.
0.7566	mechanism of the european monetary
0.7134	increase as a result of
0.6617	tiffany & co.
:	

図 8 NAB コーパスで学習した 8 グラム VPYLM の Suffix Tree から得られた, 確率的フレーズ.

Fig. 8 *Stochastic phrases* computed from the suffix tree of 8-gram VPYLM that is trained on the NAB corpus.

5.3 VPYLDA on 20news-18828

5.3.0.4 データと学習設定

可変長 n グラム LDA(VPYLDA) の評価データとして, われわれは 20news-18828 データセット[26] を用いた. このうち, 20 個のニュースグループを同等にカバーするように 4,000 文書を訓練データ, 400 文書を評価データとした.

訓練データは NAB コーパスと同様に前処理し, 801,580 語, 語彙数 10,477 語のデータとなった. $N=250$ 回のギブスサンプリングを行った後の単一サンプルを用い, 文脈 $h = w_1 \cdots w_{n-1}$ での単語 w_n の予測確率を, 順に以下のように求めた.

$$p(w_n|h) = \sum_t p(w_n|h, t) \big\langle p(t|h) \big\rangle \quad (n = 1 \ldots N) \tag{14}$$

ここで $p(w_n|h, t)$ はトピック t のユニグラムを用いた可変長 n グラムの予測確率 (10) であり, $\big\langle p(t|h) \big\rangle$ は仮想的な文書と考えた h のもつトピック t の事後期待値である. 言語モデルは固定であるから, 高速化のため, この計算には LDA の変分ベイズ EM アルゴリズム[19] を用いた.

5.3.0.5 実 験 結 果

表 2 に, トピック混合数 M を変えたときの VPYLDA のテストセットパープレキシティを示す. $M=1$ のとき, これは VPYLM と等価であり, 最下段に示した.

予測は改善しているが, その差はわずかであることがわかる. 学習されたモデルを観察したところ, この理由は, 高頻度の単語が各トピックユニグラムに均等に分配されないためであることがわかった. 図 1 の階層的 CRP において, トピック t をもつ客のコピーが親ノードに送られるのは, その客が現在のノードではなく親ノードから生成されたと確率的に判断された場合であるが, 高頻度の語は多くのノードで既に存在する

ため, カウントが再利用されて単に 1 増やされ, トピック別ユニグラムまで確率的に到達しないことが多い.

トピックはユニグラムに限られるものではなく, バイグラムやトライグラム以上にも別の混合数で存在する☆ことを考えると, このような階層的な混合モデルの推定には, また新しい方法を必要とすると思われる.

6. 考察および関連研究

本研究は, データ圧縮における Context Tree Weighting 法[27] を自然言語に応用した, Pereira らの興味深い研究[5] をその動機としている. 彼らの手法も様々な深さの木の事後確率を考え, それらを混合するものであるが, nグラム分布を階層的に推定するモデルは存在しなかったため, 最終的に各ノードでの Witten-Bell スムージングと頻度によるカットオフに頼っており, 性能面でのアドバンテージがないという結果になっていた. これに対し提案手法では, 彼らの手法で定数だった木の生成確率をベイズ化し, 階層 Pitman-Yor 過程でスムージングされたnグラムをさらに混合し, 生成モデルの立場からカウント毎に確率的なプルーニングを行うという点で, その後継となっていると考えることができる.

「無限長」の文脈を表現する方法としては, データ圧縮で用いられる PPM*法が知られているが, PPM*はデータに存在するすべての接尾辞を記憶し, それらの中で文脈に最長一致するノードを用いて, Witten-Bell スムージングとバックオフにより確率を求めるものである. これに対し, 提案法は信頼性の低い最長一致を用いるのではなく, ノードを予測性能によってベイズ的に重みづけ, 確率的なプルーニングを行うことで, 指数的に増加する接尾辞を全て記憶することを避け, 空間計算量を大きく効率化している.

5 節の実験では, Suffix Tree の形を決める事前分布として $(\alpha, \beta) = (4, 1)$ を用いたが, このパラメータは経験ベイズ法により, (6) 式で各 q_i がもつベータ事後分布から最適化することもできる. 28) の Newton 法を用いると, 1M 語の NAB コーパスのサブセットの場合, この値は各繰り返しですべて $(0.85, 0.57)$ に収束した. しかしながら, VPY 言語モデルの性能はハイパーパラメータにはあまり依存しない. **図 9** に, $(\alpha, \beta) \in (0.1 \sim 10) \times (0.1 \sim 10)$ の範囲で変えたときのパープレキシティを示す. $\beta \gg \alpha$ となる場合を除いて, 性能はほぼ一定であることがわかる.

最後に, **図 10** に森鴎外『舞姫』で学習した 5-gram VPYLM から, ランダムに生成した文の例を示す. ☆☆

従来のように文脈と最長一致するモデルのノードを使用すると, オーバーフィットのために $n = 5$ ではほぼ学習データがそのまま再現されてしまうが, ここでは式 (8) にしたがって後ろから確率的に文脈をたどり, 止まったノードから単語を生成することで, 学習データの特徴をとらえたテキストが生成されていることがわかる. モデルには確率的句として, 「エリ ス」(0.856), 「嗚呼 、」(0.803), 「似 たり 。」(0.398), 「二 人 の 間」(0.199) などがあり, 図 10 のテキストを生んでいた.

表 2 20news-18828 データセットにおける VPYLDA のテストセットパープレキシティ.
Table 2 VPYLDA test-set perplexities on the 20news-18828 dataset.

Model	PPL
VPYLDA ($M = 5$)	**104.69**
($M = 10$)	**103.57**
($M = 20$)	**103.28**
VPYLM	105.30

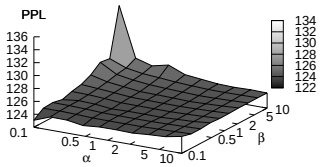


図 9 VPYLM のハイパーパラメータ (α, β) とテストセットパープレキシティの関係.
Fig. 9 Relationship between the VPYLM hyperparameters (α, β) and test-set perplexities.

エリスは打笑ひ玉へ。産れん子の生れむることの我を力になし果てつ。「カイゼルホオフ」へ通ふことの少きをもかくは心の奥に凝り固まりて、一は親族なる某省の官長き猶こゝまで導きつゞきだに事なく済みたらましかば、何事をか叙すべき。わが心はかの合歓といふ木の葉に似たり。余との間にて、余がねの額に印せし面に、鬢の毛の解けてかゝりて、灼くが如く、政治家になるべき。わが心はこの時戸口に入りてこそ紅粉をも粧ひ、折に触れては、故らに知りて、大臣に聞え上げし一諾を知り、言葉にて、その為し難きに、若し真なりせば、詩に詠じ歌によめる後は心地すが／＼しくもなりなむ。

図 10 『舞姫』からランダムに生成したテキスト. VPYLM, $n = 5$.
Fig. 10 Random walk generation from the language model trained on "*Maihime*."

☆ たとえば, 'mixture of Gaussians' と 'mixture of flour' の出現は, 文脈に応じてほぼ排他的だと考えられる.

☆☆ 生成の際にバックオフは行っていないため, 正確なサンプリングとは少し異なる.

7. まとめと展望

本論文では, 最近提案された n グラム分布に対する高精度なノンパラメトリック確率過程である階層 Pitman-Yor 過程をさらに拡張することで, 単語の生まれた n グラム文脈長を適切に推定する可変長 n グラム言語モデルを示した. 無限の深さをもつ確率的な Suffix Tree を考え, それをベイズ推定することで, これまで固定だった n を原理的に不要とし, 可変長とすることができる.

提案法は最大オーダーの従来法と同等の性能を, より少ない空間的および時間的計算量で達成し, 従来不可能だった高次 n グラムの推定も可能にする. また, 言語モデルの副産物として, 学習された Suffix Tree から特徴的な系列を確率つきで取り出せることを明らかにした.

また, 本研究の手法は言語モデルに限られることなく, 可変長マルコフモデルを実現する一般的なベイズ的方法であることに注意されたい. プルーニングに基づく従来の “可変長モデル” と異なり, 提案手法は可変長時系列の完全な生成モデルとなっている.

本研究では Suffix Tree の事前分布として, 柔軟だが単純な共通のベータ分布を用いたが, 29) にみられるような事前分布を考えることで, モデルをより精密にすることができると考えている. また, トピック適応化についても, 階層的な混合モデルのよりよい推定法を考えていきたい.

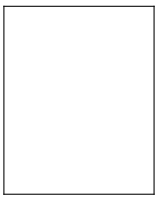
謝辞 本研究は独立行政法人 情報通信研究機構の研究委託「大規模コーパスベース音声対話翻訳技術の研究開発」により実施したものである.

参 考 文 献

1) Shannon, C. E.: A mathematical theory of communication, *Bell System Technical Journal*, Vol.27, pp.379–423, 623–656 (1948).
2) 工藤拓：形態素周辺確率を用いた分かち書きの一般化とその応用，言語処理学会全国大会論文集 NLP-2005 (2005).
3) Siu, M. and Ostendorf, M.: Variable n-grams and extensions for conversational speech language modeling, *IEEE Trans. on Speech and Audio Processing*, Vol.8, pp.63–75 (2000).
4) Stolcke, A.: Entropy-based Pruning of Backoff Language Models, *Proc. of DARPA Broadcast News Transcription and Understanding Workshop*, pp.270–274 (1998).
5) Pereira, F., Singer, Y. and Tishby, N.: Beyond Word N-grams, *Proc. of the Third Workshop on Very Large Corpora*, pp.95–106 (1995).
6) Leonardi, F.G.: A generalization of the PST algorithm: modeling the sparse nature of protein sequences, *Bioinformatics*, Vol.22, No.11, pp.1302–1307 (2006).
7) Buhlmann, P. and Wyner, A. J.: Variable Length Markov Chains, *The Annals of Statistics*, Vol.27, No.2, pp.480–513 (1999).
8) Kawabata, T. and Tamoto, M.: Back-off Method for N-gram Smoothing based on Binomial Posteriori Distribution, *ICASSP-96*, Vol.I, pp.192–195 (1996).
9) Cowans, P.: Probabilistic Document Modelling, PhD Thesis, University of Cambridge (2006). http://www.inference.phy.cam.ac.uk/pjc51/thesis/index.html.
10) Teh, Y. W.: A Bayesian Interpretation of Interpolated Kneser-Ney, Technical Report TRA2/06, School of Computing, NUS (2006).
11) Teh, Y.W.: A Hierarchical Bayesian Language Model based on Pitman-Yor Processes, *Proc. of COLING/ACL 2006*, pp.985–992 (2006).
12) Kneser, R. and Ney, H.: Improved backing-off for m-gram language modeling, *Proceedings of ICASSP*, Vol.1, pp.181–184 (1995).
13) Pitman, J. and Yor, M.: The Two-Parameter Poisson-Dirichlet Distribution Derived from a Stable Subordinator, *Annals of Probability*, Vol.25, No.2, pp.855–900 (1997).
14) Teh, Y.W., Jordan, M.I., Beal, M.J. and Blei, D.M.: Hierarchical Dirichlet Processes, Technical Report 653, Department of Statistics, University of California at Berkeley (2004).
15) Ghahramani, Z.: Non-parametric Bayesian Methods, *UAI 2005 Tutorial* (2005). http://learning.eng.cam.ac.uk/zoubin/talks/uai05tutorial-b.pdf.
16) Gilks, W.R., Richardson, S. and Spiegelhalter, D.J.: *Markov Chain Monte Carlo in Practice*, Chapman & Hall / CRC (1996).
17) Daumé III, H.: Fast search for Dirichlet process mixture models, *AISTATS 2007* (2007).
18) Sleator, D. and Tarjan, R.: Self-Adjusting Binary Search Trees, *JACM*, Vol. 32, No. 3, pp. 652–686 (1985).
19) Blei, D.M., Ng, A.Y. and Jordan, M.I.: Latent Dirichlet Allocation, *Journal of Machine Learning Research*, Vol.3, pp.993–1022 (2003).
20) Huang, C.-R. and Chen, K.-j.: A Chinese Corpus for Linguistics Research, *Proc. of COLING 1992*, pp.1214–1217 (1992).
21) Chen, S. F. and Goodman, J.: An Empirical Study of Smoothing Techniques for Language Modeling, *Proc. of ACL 1996*, pp.310–

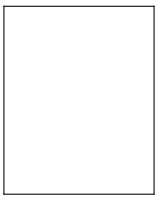
318 (1996).
22) Goodman, J.T.: A Bit of Progress in Language Modeling, Extended Version, Technical Report MSR–TR–2001–72, Microsoft Research (2001).
23) Kudo, T.: MeCab: Yet Another Part-of-Speech and Morphological Analyzer. http://mecab.sourceforge.net/.
24) Stolcke, A.: SRILM – An Extensible Language Modeling Toolkit, *Proc. of ICSLP*, Vol.2, pp. 901–904 (2002).
25) Ron, D., Singer, Y. and Tishby, N.: The Power of Amnesia, *Advances in Neural Information Processing Systems*, Vol.6, pp.176–183 (1994).
26) Lang, K.: Newsweeder: Learning to filter netnews, *Proceedings of the Twelfth International Conference on Machine Learning*, pp.331–339 (1995).
27) Willems, F., Shtarkov, Y. and Tjalkens, T.: The Context-Tree Weighting Method: Basic Properties, *IEEE Trans. on Information Theory*, Vol.41, pp.653–664 (1995).
28) Minka, T. P.: Estimating a Dirichlet distribution (2000). http://research.microsoft.com/~minka/papers/dirichlet/.
29) Pitman, J.: Combinatorial Stochastic Processes, Technical Report 621, Department of Statistics, University of California, Berkeley (2002).

(平成？年？月？日受付)
(平成？年？月？日採録)



持橋　大地（正会員）

1973 年生．1998 年東京大学教養学部基礎科学科第二卒業．2005 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了．博士 (理学)．2003 年 ATR 音声言語コミュニケーション研究所研修研究員/専任研究員．2007 年より, NTT コミュニケーション科学基礎研究所 リサーチアソシエイト．自然言語処理, 機械学習の研究に従事．



隅田英一郎（正会員）

1982 年電気通信大学大学院修士課程修了．1999 年京都大学工学博士．現在,ATR 音声言語コミュニケーション研究所室長． NiCT 知識創成コミュニケーション研究センター研究マネージャ, 神戸大学大学院自然科学研究科連携教授, ATR-Langue 取締役副社長兼務．機械翻訳, e ラーニングの研究に従事．IEICE, NLP, ASJ, ACL, IEEE の会員, ACM/TSLP の Associate Editor．