論　文

特徴選択を導入した低・ゼロ頻度 N-gram の効率的なゆう度比推定法*

菊地　真人[†a)]　吉田　光男[††]　梅村　恭司[†††]　大囿　忠親[†]

Efficient Likelihood Ratio Estimation Method for Low- and Zero-Frequency N-grams Using Feature Selection*

Masato KIKUCHI[†a)], Mitsuo YOSHIDA[††], Kyoji UMEMURA[†††], and Tadachika OZONO[†]

あらまし　自然言語処理 (NLP) では，N-gram のゆう度比を頻度情報から推定することがある．コーパスが含む N-gram はごく一部であり，そのほとんどは出現頻度が低い．このとき頻度による単純なゆう度比推定法は，低頻度から計算される推定値を不当に高く見積もり，コーパスで未観測のゼロ頻度 N-gram には有用な推定値を算出できない．ゼロ頻度 N-gram への対策として，N-gram を文字や単語などの離散値に分解し，それらのゆう度比の積を取る方法が考えられる．更に，頻度に応じてゆう度比を低めに見積もる推定法を，個々のゆう度比推定へと適用し，低頻度に対する過大推定を抑制できる．しかしこの方法では多くの離散値を扱うため，推定に要する実行時間やメモリ使用量が増加する．加えて不要な離散値を用いると推定精度が低下する．そこで本論文では，先述の方法と文書分類で用いられる特徴選択法を組み合わせる．有用な離散値のみを選択して用い，推定精度の低下を抑制しつつ推定効率の向上を図る．コーパスから固有表現の文脈をゆう度比で予測する実験を行い，提案する推定法が低頻度及びゼロ頻度 N-gram に対し，効果的かつ効率的な推定結果を提供することを示す．

キーワード　ゆう度比推定，低頻度，ゼロ頻度，N-gram，特徴選択法

1. まえがき

ゆう度比は確率分布の比で定義される統計量であり，ゆう度比を利用したアプリケーションは多岐にわたる．ゆう度比を実際に用いるには推定が必要で，その推定性能はアプリケーションの有効性を決定づける重大な因子になり得る．自然言語処理 (NLP) では，文字列や単語列に対するゆう度比を，コーパスから得た頻度情報を使用して推定することがある [1]．このとき，ゆう度比を推定する一つの方法は，次式のように各々の確率分布を相対頻度 $\widehat{p}_*(x)$ で求めてその比を取ることである．

$$\widehat{p}_*(x) = \frac{f_*(x)}{n_*}, \quad * \in \{\mathrm{de}, \mathrm{nu}\}$$

$$r_{\mathrm{MLE}}(x) = \frac{\widehat{p}_{\mathrm{nu}}(x)}{\widehat{p}_{\mathrm{de}}(x)}$$

ここで de と nu はゆう度比の分母・分子を表す添え字であり，離散型確率変数 X がとる値を x とする．$x = \langle t_1, t_2, \ldots, t_{\mathrm{N}} \rangle$ は N 個の文字や単語が連なる系列を表し，NLP では N-gram と呼ばれる．t_k, $k = 1, 2, \ldots, \mathrm{N}$ は x を構成する k 番目の文字や単語である．X の分布状況を表現する確率関数を $p_*(x)$ とするとき，$f_*(x)$ は $p_*(x)$ に基づいて得た x の観測頻度であり，$n_* = \sum_x f_*(x)$ である．上記の推定法は単純だが，推定に用いる頻度が低い場合は，推定値が不当に大きくなったり，少しの頻度差で推定値が大幅に変動したりする問題が生じる．N-gram などの言語要素は多くの種類があり，低頻度のものが多数を占める．ゆえに，言語要素の観測頻度からゆう度比を推定する際は，前述の問題に直面することがある．

これまでに低頻度に起因する問題の対処法 [2] が提案されており，その推定量は

† 名古屋工業大学大学院情報工学専攻，名古屋市
Department of Computer Science, Graduate School of Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya-shi, 466–8555 Japan
†† 筑波大学ビジネスサイエンス系，東京都
Faculty of Business Sciences, University of Tsukuba, 3–29–1 Otsuka, Bunkyo-ku, Tokyo, 112–0012 Japan
††† 豊橋技術科学大学情報・知能工学系，豊橋市
Department of Computer Science and Engineering, Toyohashi University of Technology, 1–1 Hibarigaoka, Tempaku-cho, Toyohashi-shi, 441–8580 Japan
a) E-mail: kikuchi@nitech.ac.jp
* 本論文は，FIT2021 推薦論文である．
DOI:10.14923/transinfj.2022JDT0001

表 1　ゆう度比の推定例．$\widehat{r}(x)$ の λ は 10^{-5} とした．
Table 1　Examples of likelihood ratio estimation. We set λ of $\widehat{r}(x)$ to 10^{-5}.

N-gram	観測頻度				$r_{\mathrm{MLE}}(x)$	$\widehat{r}(x)$
x	n_{de}	$f_{\mathrm{de}}(x)$	n_{nu}	$f_{\mathrm{nu}}(x)$		
x_{a}	10^7	2,000	10^4	100	50	47.6
x_{b}	10^7	20	10^4	1	50	8.3
x_{c}	10^7	20	10^4	2	100	16.7
x_{d}	10^7	6	10^4	0	0	0

$$\widehat{r}(x) = \left(\frac{f_{\mathrm{de}}(x)}{n_{\mathrm{de}}} + \lambda \right)^{-1} \times \frac{f_{\mathrm{nu}}(x)}{n_{\mathrm{nu}}}$$

と定義される．ここで $\lambda\ (\geq 0)$ は正則化パラメータである．なお，本推定量については導出過程も含めて **3.1** で詳説する．確率分布の推定を介してゆう度比を間接的に求める方法と異なり，この手法は 2 乗損失の最小化問題を解くことでゆう度比を直接推定する．その最小化問題で導入される正則化パラメータ λ が，低頻度の問題を緩和する役割を果たす．このことを表 1 に示した推定例で確認する．まず x_{a} と x_{b} に着目すると，$f_*(x_{\mathrm{a}})$ と $f_*(x_{\mathrm{b}})$ にかなりの差があるのに対し，$r_{\mathrm{MLE}}(x_{\mathrm{a}})$ と $r_{\mathrm{MLE}}(x_{\mathrm{b}})$ は共に 50 と大きな値になる．ここで，$f_{\mathrm{nu}}(x_{\mathrm{b}}) = 1$ は偶然の出現であることが十分に考えられる．このことから，頻度に応じた “推定の信頼性” が推定値に反映されることが望ましい．一方で，$\widehat{r}(x_{\mathrm{a}})$ は 47.6 となり $r_{\mathrm{MLE}}(x_{\mathrm{a}})$ の 50 に近く，$\widehat{r}(x_{\mathrm{b}})$ は 8.3 となり $r_{\mathrm{MLE}}(x_{\mathrm{b}})$ の 50 よりもはるかに低い．したがって，$\widehat{r}(x)$ は頻度による信頼性を反映した推定量となっている．次に x_{b} と x_{c} に着目すると，$f_{\mathrm{nu}}(x_{\mathrm{b}})$ と $f_{\mathrm{nu}}(x_{\mathrm{c}})$ の差がわずか 1 にもかかわらず，$r_{\mathrm{MLE}}(x_{\mathrm{c}})$ は 50 から 100 へと大きく変動している．しかしながら，$f_*(x_{\mathrm{b}})$ と $f_*(x_{\mathrm{c}})$ は共に低頻度のため，この状況では推定値はロバストになることが望ましい．$\widehat{r}(x)$ は低頻度に対する推定値を低く見積もるから，$\widehat{r}(x_{\mathrm{b}})$ と $\widehat{r}(x_{\mathrm{c}})$ との差が小さくなる（それぞれ 8.3，16.7）．よって，$\widehat{r}(x)$ は低頻度に対するロバスト性を備えた推定量である．一方で x_{d} に着目すると，$f_{\mathrm{nu}}(x_{\mathrm{d}})$ がゼロのため，$r_{\mathrm{MLE}}(x_{\mathrm{d}}) = \widehat{r}(x_{\mathrm{d}}) = 0$ となる．これは $\widehat{r}(x)$ であっても，コーパス中で観測されないゼロ頻度の N-gram には有益な推定値を算出できないことを意味する．

先述のように，コーパスが含む言語要素には多くの種類が存在し，その大半はまれにしか出現しない．更に，コーパスのサイズは有限のため，コーパスに含まれないゼロ頻度の言語要素も多いと考えられる．NLP に関するアプリケーション（例えば，機械翻訳システムや情報検索システム）では，学習データに存在しないゼロ頻度の文字列や検索クエリが入力として与えられることがあり，そのときでさえ情報のある推定値を返すことが求められる．以上を踏まえると，低頻度に加えゼロ頻度に対しても，有益な推定値を算出できるゆう度比の推定法が必要と考える．そこで本論文では，低頻度への対処法 [2] を応用して，ゼロ頻度にも対処可能なゆう度比の推定法を提案する．

ゼロ頻度に対する簡単な対処法は，x をなす離散値 t_k の間に確率的な独立性を仮定し，$r(x)$ を $r(t_k)$ の積 $\prod_{k=1}^{\mathrm{N}} r(t_k)$ で近似することである．この t_k の扱いは，ナイーブベイズ分類器でも適用される一般的なものである．また，$r(t_k)$ の推定に低頻度への対処法 [2] を適用すれば，低頻度 N-gram のゆう度比にも安定した推定値を付与できる．しかし，前述した t_k の扱いには以下の問題がある．まず t_k 間の確率的な独立性は，実際には成立しないことが多く，ゆう度比の推定精度を低下させる原因になる．次に x を t_k 単位に分解することで，膨大な種類の t_k を扱わなければならない．膨大な t_k の中には，ゆう度比推定に不要なものや悪影響を及ぼすものが多く存在する．そのため，全ての t_k を闇雲に扱うことは，推定の精度と効率を低下させる．そこで，文書分類タスクで用いられる特徴選択法を，ゆう度比推定に組み合わせる．つまり，利用できる t_k の中から推定に有益なものを選択し，それらを用いて効率良くゆう度比推定することを提案する．実験では，コーパスから固有表現の出現文脈をゆう度比で予測することを試みる．そして全語彙を推定に用いた手法と比較し，提案手法が同等以上の予測精度を維持しつつ，実行時間とメモリ使用量の観点から効率良くゆう度比推定できることを報告する．

2. 関 連 研 究

NLP における確率推定では，低頻度及びゼロ頻度への対処法が提案されてきた [3]．それらはスムージング法と呼ばれ，観測された事象の確率推定量から一定量を割り引き，観測されない事象の確率推定量へと分配する枠組みである．この枠組みにより，低頻度の事象に対する確率は低めに推定され，ゼロ頻度の事象に対する確率はゼロより大きく推定される．スムージング法はゆう度比推定にそのまま応用できないが，ゆう度比を構成する確率分布をスムージング法で求めてその比を取ることはできる．しかし，この方法で求めたゆう度比は，実用性が低いことが明らかになってい

る [2]．よって，ゆう度比の推定法は新たに考案する必要がある．

確率分布の推定を介してゆう度比を間接的に求める手法は，大きな推定誤差を生むことが知られている [4]．ゆえに，確率分布の推定を経由せずにゆう度比を直接推定する手法が提案されてきた [5]〜[8]．これらの直接推定法は，連続空間上で定義されるゆう度比を推定対象とする．対して我々が扱うのは離散的な標本空間から得た文字列や単語列であり，それらの観測頻度から推定されるゆう度比も離散空間上に定義される．そこで菊地ら [2] は，直接推定法の一つである uLSIF [8] を離散的なゆう度比の推定にも適用可能にした．この手法では，最適化で導入される正則化パラメータが，低頻度から求まるゆう度比にも安定した推定値を与える．しかし菊地らの手法でも，ゼロ頻度からは有益な推定値を算出できない．

文書は膨大な種類の単語を含むため，文書分類では単語の扱いがしばしば問題となる．単語の大半はまれにしか出現せず分類に貢献しない．それどころか，幾つかの単語は分類誤りを誘発する．そのため，ノイズであり情報の少ない，冗長な単語を排する特徴選択法が提案されてきた [9]．特徴選択法は特徴（単語）の部分集合を生成する手段の違いにより，フィルタ，ラッパー，埋め込み，ハイブリッド方式の 4 種類に大別される．フィルタ法 [10]〜[12] は，何らかのスコア関数を用いて，語彙 V から学習に用いる単語の部分集合 Θ を選択する．ラッパー法 [13], [14] は，任意の集合 $\Theta \subset V$ を分類器に与え，得た分類性能を頼りに最適な部分集合を決定する．埋め込み法 [15] では，学習が始まる前に特徴選択が実行されない代わりに，特徴選択が学習プロセスに組み込まれる．ハイブリッド法 [16], [17] は，特徴選択のプロセス中でフィルタ法とラッパー法を組み合わせる．なお特徴選択法の大半は，効果的かつ効率的なフィルタ方式に属する．本論文で提案するゆう度比の推定量は，ハイパーパラメータである正則化パラメータをもつ．ゆえに，ゆう度比推定に組み合わせる特徴選択法としても，計算量が軽量なフィルタ方式を採用する．

3. 前提知識

本節では，提案手法の導入に必要となる，ゆう度比の直接推定法と文書分類のための特徴選択法を説明する．

3.1 ゆう度比の直接推定法

あるデータが含む要素 x の集合を $D \subset \mathbb{U}$ とする．$\mathbb{U}$ は存在し得る全 v 種類の要素を含む集合であり，情報理論では有限アルファベットとも呼ばれる．ここでいうデータとは要素がサンプリングされるデータを意味する．今，二つの i.i.d. 標本

$$\{x_i^{\mathrm{de}}\}_{i=1}^{n_{\mathrm{de}}} \overset{\text{i. i. d.}}{\sim} p_{\mathrm{de}}(x), \quad \{x_j^{\mathrm{nu}}\}_{j=1}^{n_{\mathrm{nu}}} \overset{\text{i. i. d.}}{\sim} p_{\mathrm{nu}}(x)$$

を得たとする．de と nu はゆう度比の分母・分子を表す添え字である．ここで x は文字列や単語列といった離散要素を表し，$p_{\mathrm{de}}(x)$ と $p_{\mathrm{nu}}(x)$ は D が含む x の分布状況を表す確率関数とする．これまでの先行研究と同様に，確率関数 $p_{\mathrm{de}}(x)$ が条件

$$p_{\mathrm{de}}(x) > 0 \quad \text{for all } x \in D$$

を満足すると仮定する．これにより，全ての x に対してゆう度比を定義できる．本節では，二つの標本 $\{x_i^{\mathrm{de}}\}_{i=1}^{n_{\mathrm{de}}}$，$\{x_j^{\mathrm{nu}}\}_{j=1}^{n_{\mathrm{nu}}}$ からゆう度比

$$r(x) = \frac{p_{\mathrm{nu}}(x)}{p_{\mathrm{de}}(x)}$$

を確率分布の推定を介さずに直接推定する．

unconstrained Least-Squares Importance Fitting (uLSIF) [8] は，2 乗損失の最小化によりゆう度比を直接推定する手法である．uLSIF では $r(x)$ を線形和

$$\widehat{r}(x) = \sum_{l=1}^{b} \beta_l \varphi_l(x)$$

でモデル化する．$\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_b)^{\mathrm{T}}$ は標本から学習されるパラメータ，$\{\varphi_l\}_{l=1}^{b}$ は非負値を取る基底関数である．本来の uLSIF は，ガウスカーネルに基づく基底関数によって，連続的な標本空間の構造をゆう度比推定に活用する．しかし，本論文で扱う標本空間は離散のため，ガウスカーネルが効力を発揮しない．そこで，菊地ら [2] が提案した基底関数 $\{\delta_l\}_{l=1}^{v}$

$$\delta_l(x) = \begin{cases} 1 & (x = x_{(l)}) \\ 0 & (x \neq x_{(l)}) \end{cases} \tag{1}$$

を代用する．このとき，b は存在し得る離散要素の種類数 v に置換される．添え字 l は v 種類の要素から特定の要素を指定する．すなわち，$x_{(l)}$ は v 種類の要素のうち，l 種類目の要素を意味する．$\{\delta_l\}_{l=1}^{v}$ は要

素間の関係性を捉えられないが，uLSIF へ導入すると解が解析的に求まる利点がある．式 (1) を推定モデル $\widehat{r}(x_{(m)}),\ m = 1, 2, \ldots, v$ へ代入すると，

$$\widehat{r}(x_{(m)}) = \sum_{l=1}^{v} \beta_l \delta_l(x_{(m)}) = \beta_m \tag{2}$$

が得られる．uLSIF では，推定モデル $\widehat{r}(x_{(m)})$ と真のゆう度比 $r(x_{(m)})$ の 2 乗損失を最小化するように，パラメータ $\boldsymbol{\beta}$ を学習する．その最適化問題は

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^v} \left[\frac{1}{2} \boldsymbol{\beta}^{\mathrm{T}} \widehat{\boldsymbol{H}} \boldsymbol{\beta} - \widehat{\boldsymbol{h}}^{\mathrm{T}} \boldsymbol{\beta} + \frac{\lambda}{2} \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\beta} \right] \tag{3}$$

として与えられる(注1)．$\mathbb{R}^v$ は実 v 次元空間である．上式では，$\boldsymbol{\beta}$ に対する正則化のためにペナルティ項 $\frac{\lambda}{2} \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\beta}$ が導入される．$\lambda\ (\geq 0)$ は正則化パラメータ，$\boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{\beta}/2$ は ℓ_2-正則化項である．$\widehat{\boldsymbol{H}}$ は $v \times v$ 行列であり，その (l, l') 番目の要素は

$$\begin{aligned} \widehat{H}_{l,l'} &= \frac{1}{n_{\mathrm{de}}} \sum_{i=1}^{n_{\mathrm{de}}} \delta_l(x_i^{\mathrm{de}}) \delta_{l'}(x_i^{\mathrm{de}}) \\ &= \begin{cases} \frac{f_{\mathrm{de}}(x_{(l)})}{n_{\mathrm{de}}} & (l = l') \\ 0 & (l \neq l') \end{cases} \end{aligned} \tag{4}$$

と定義される．$f_*(x_{(l)}), * \in \{\mathrm{de}, \mathrm{nu}\}$ は確率関数 $p_*(x_{(l)})$ に基づいて得た $x_{(l)}$ の頻度である．上式から明らかなように，$\widehat{\boldsymbol{H}}$ は対角行列となる．$\widehat{\boldsymbol{h}}$ は v 次元ベクトルであり，その l 番目の要素は

$$\widehat{h}_l = \frac{1}{n_{\mathrm{nu}}} \sum_{j=1}^{n_{\mathrm{nu}}} \delta_l(x_j^{\mathrm{nu}}) = \frac{f_{\mathrm{nu}}(x_{(l)})}{n_{\mathrm{nu}}} \tag{5}$$

と定義される．式 (3) は拘束なし二次計画問題であり，その解は次式で解析的に求められる．

$$\widetilde{\boldsymbol{\beta}}(\lambda) = (\widehat{\boldsymbol{H}} + \lambda \mathbf{1}_v)^{-1} \widehat{\boldsymbol{h}}$$

$\mathbf{1}_v$ は要素が全て 1 の v 次元ベクトルである．式 (2), (4), (5) より，式 (3) の解は次式となる．

$$\begin{aligned} \widehat{r}(x_{(m)}) &= \widetilde{\beta}_m(\lambda) \\ &= \left(\frac{f_{\mathrm{de}}(x_{(m)})}{n_{\mathrm{de}}} + \lambda \right)^{-1} \times \frac{f_{\mathrm{nu}}(x_{(m)})}{n_{\mathrm{nu}}} \end{aligned} \tag{6}$$

本来の uLSIF は解が負値となる場合があり，ゆう度比の非負性を考慮して負値をゼロに丸める．しかし上式は常に非負のため，$\widetilde{\beta}_m(\lambda)$ がそのまま解となる．

式 (6) では，正則化パラメータ λ が頻度に応じて推定値を低めに見積もりロバストにする．また，この式はゆう度比の分母のみを補正する形式になる．なお $\lambda = 0$ のとき，この式はそれぞれの確率分布を相対頻度で求め，その比を取った結果と等しい．

3.2 文書分類のための特徴選択法

代表的な機械学習アルゴリズムの一つであるナイーブベイズ分類器は，強力な独立性仮定とベイズの定理に基づく確率的分類器である．ある単語ベクトル $\langle t_1, t_2, \ldots, t_{\mathrm{N}} \rangle$ で表される文書 d が与えられたとする．ここで $t_k,\ k = 1, 2, \ldots, \mathrm{N}$ は文書の先頭から k 番目に位置する単語を意味する．C をクラス変数，c を C の取る値とすると，d を適切なクラスへ分類する問題を解くナイーブベイズ分類器は

$$\widehat{c}(d) = \arg\max_{c \in C} p(c) \prod_{k=1}^{\mathrm{N}} p(t_k \mid c) \tag{7}$$

と定式化される．$\widehat{c}(d)$ は d が分類されるクラスラベルである．この分類器では，d に出現する各単語 t_k を個別に扱い，条件付き確率 $p(d \mid c)$ を $p(t_k \mid c)$ の積で近似する．この近似は，t_k の出現がクラス c のもとで他の単語と条件付き独立という仮定に基づく．しかしながら，この仮定は実際に成立しないことが多く，ナイーブベイズ分類器の分類精度を低下させる要因の一つとして知られている．加えて，多数の単語を扱うことで分類の効率も低下する．

上記の問題を緩和する方法として，特徴選択法が提案されてきた [9]．この方法では，学習データの語彙 V から分類に有用な単語の部分集合 Θ を選択し，それを学習に用いることで分類精度と分類効率の向上を試みる．単語に対するスコア関数を定義し，計算されたスコアを Θ が含む単語の選択基準として用いる．ここで肝心なのは，どのようなスコア関数を定義し，どのように用いるかということである．本節では，有名な三つのスコア関数を紹介する．以下の関数を用いた場合，スコアの降順から単語を選択する．

一つ目は交差エントロピー (CET; expected cross entropy for text) [10] であり

$$\begin{aligned} CET_{m,c} =& p(t_{(m)}, c) \log \frac{p(t_{(m)}, c)}{p(t_{(m)}) p(c)} + \\ & p(t_{(m)}, \bar{c}) \log \frac{p(t_{(m)}, \bar{c})}{p(t_{(m)}) p(\bar{c})} \end{aligned} \tag{8}$$

(注1)：式 (3) の導出については uLSIF の原論文 [8] を参照のこと．

と定義される．$m = 1, 2, \ldots, v$ は単語の種類を指定する添え字である．すなわち，$t_{(m)}$ は v 種類ある単語のうち，m 種類目の単語を意味する．$p(t_{(m)}, c)$ は単語 $t_{(m)}$ がある文書に出現し，その文書がクラス c に属する確率である．$p(t_{(m)})$ は $t_{(m)}$ がある文書に出現する確率で，$p(c)$ はある文書が c に属する確率である．一方で $p(t_{(m)}, \bar{c})$ は $t_{(m)}$ がある文書に出現し，その文書が c に属さない確率である．$p(\bar{c})$ はある文書が c に属さない確率である．$t_{(m)}$ が c 及び $\bar{c}$ と独立であるとき，つまりある文書が c に属するか否かの分類に $t_{(m)}$ が全く影響を及ぼさないとき，CET はゼロになる．

二つ目はカイ 2 乗統計量 [11] である．この統計量は単語 $t_{(m)}$ とクラス c の関連の度合いを測定するために用いられる．文書内の単語に対する否定的証拠を用いるカイ 2 乗統計量は

$$\chi^2_{m,c} = \frac{\left[p(t_{(m)}, c)p(\bar{t}_{(m)}, \bar{c}) - p(t_{(m)}, \bar{c})p(\bar{t}_{(m)}, c)\right]^2}{p(t_{(m)}, c)p(t_{(m)}, \bar{c})p(\bar{t}_{(m)}, c)p(\bar{t}_{(m)}, \bar{c})} \tag{9}$$

と定義される．$p(\bar{t}_{(m)}, \bar{c})$ は $t_{(m)}$ がある文書に出現せず，その文書が c に属さない確率であり，$p(\bar{t}_{(m)}, c)$ は $t_{(m)}$ がある文書に出現せず，その文書が c に属する確率である．

三つ目は *GSS coefficient* [12] である．この関数も単語に対する否定的証拠を用いて

$$GSS_{m,c} = p(t_{(m)}, c)p(\bar{t}_{(m)}, \bar{c}) - p(t_{(m)}, \bar{c})p(\bar{t}_{(m)}, c) \tag{10}$$

と定義される．幾つかのデータセットにおいて，この関数はカイ 2 乗統計量よりも優れた性能を示すことが報告されている [12]．

上記の関数は，文書に単語が出現するか $(t_{(m)})$ しないか $(\bar{t}_{(m)})$ を考慮するが，単語の出現位置 k は考慮しない．その理由として，一般に文書は出現位置の定まった特徴からは構成されず，文書の長さ（つまり文書を構成する単語数）も各文書で異なることが挙げられる．それに対して本論文では，固有表現の左（単語）N-gram を予測する問題を扱う．**4.** で述べるように，この問題では単語の種類に加えて出現位置も考慮することに意味がある．また，特徴選択法は学習データを洗練する枠組みのため，ゆう度比の推定にも自然に応用できる．以上を踏まえて我々は，単語の出現位置 k も考慮した特徴選択法をゆう度比推定に応用する．

4. 提案手法

特徴ベクトル $x = \langle t_1, t_2, \ldots, t_{\mathrm{N}} \rangle$ に対する次のゆう度比を推定する問題を考える．

$$r(x) = \frac{p_{\mathrm{nu}}(x)}{p_{\mathrm{de}}(x)}$$

ここで t_k, $k = 1, 2, \ldots, \mathrm{N}$ は文字や単語などの離散値であり，N 個の離散値の連なりとなる x は N-gram と呼ばれる．$r(x)$ の単純な推定法は，二つの確率分布を x の相対頻度でそれぞれ求め，その比を取ることである．しかし N-gram は言語要素であるため，低頻度やゼロ頻度のものが多い．特に N が大きくなると，分母分子の相対頻度のどちらか（あるいは両方）がゼロになることが増え，有用な推定値の算出が難しい．

これに対する素朴な対策として，t_k の出現が他の離散値の出現と確率的に独立という仮定を置き，ゆう度比 $r(x)$ を $r(t_k)$ の積で近似することが考えられる．x を構成する t_k を個別に扱うことで，個々の t_k が出現すれば $r(x)$ を推定できる．また，$r(t_k)$ の推定に **3.1** の手法を活用することで，低頻度の N-gram に対しても安定した推定値を算出できる．しかし，独立性の仮定は実際に成立しないことが多い．成立しない独立性を強引に仮定すると，他の離散値と共起することで $r(x)$ の推定に寄与していた t_k が，$r(x)$ の推定にほぼ無関係になったり悪影響を及ぼしたりすることが想定される．そして，そのような t_k のゆう度比の積を取ることで，$r(x)$ の推定値が真値から大きく変化してしまう可能性がある．したがって独立性を仮定した場合は，t_k の扱いに注意を払う必要がある．加えて，t_k を扱うと膨大な計算が必要となるため，ゆう度比推定にかかる時間やメモリ使用量の増加も想定される．

そこで特徴選択法を導入し，推定に必要な識別力のある離散値のみを学習データから選択する．これによって，独立性の仮定で $r(x)$ の推定に悪影響を与えるようになった t_k を排除し，推定精度の向上及び推定に要する時間とメモリ使用量の効率化を試みる．提案手法は

$$r_{\mathrm{ours}}(x) = \prod_{k=1}^{\mathrm{N}} \widetilde{r}(t_k)^{w_{k(m)}} \tag{11}$$

$$\widetilde{r}(t_k) = \left\{ \frac{f_{\mathrm{de}}(t_k) + 1}{n_{\mathrm{de}} + 2} + \lambda \right\}^{-1} \frac{f_{\mathrm{nu}}(t_k) + 1}{n_{\mathrm{nu}} + 2}$$

と定義される．ここで λ (≥ 0) は正則化パラメータで

ある．なお一般に，λ の値は過学習を防ぐという観点から決定されるが，**5.** の実験では F1 値を最大化する値を探索するという特殊なパラメータの決定方法を採用している．また，t_k の頻度をそのまま使用すると，$f_{\mathrm{nu}}(t_k) = 0$ となる t_k を一つでも含む x の推定値がゼロになってしまう．この問題を回避するため，上式では $f_*(t_k)$ と n_* にそれぞれ 1 と 2 を加算した補正頻度を用いる．$\frac{f_*(t_k)+1}{n_*+2}$ は，確率関数 $p_*(t_k), * \in \{\mathrm{de}, \mathrm{ne}\}$ に対応する離散確率分布を t_k が出現するか否かのベルヌーイ試行による確率分布と捉え，事前分布を一様分布としたときの事後期待値と等しい．他の補正法として n_* に t_k の種類数を加算する方法もあるが，コーパスといった言語要素を含むデータでは t_k の種類数が n_* とほぼ等しくなり，補正前後の値が大きく異なる．そのため n_* に 2 を加算し，補正による影響を小さく抑える．そして，離散値の種類 m と出現位置 k を考慮した重み

$$w_{k(m)} = \begin{cases} 1 & (t_{k(m)} \in \Theta_k) \\ 0 & (t_{k(m)} \notin \Theta_k) \end{cases} \tag{12}$$

により特徴選択を実現する．ここで $t_{k(m)}$ は N-gram の k 番目にある m 種類目の離散値であり，Θ_k は位置 k で出現し得る離散値の集合 V_k から **3.2** で述べた選択方法で選ばれた部分集合である．部分集合のサイズ $|\Theta_k|$ はハイパーパラメータである．式 (11)，(12) で示されるように，離散値に対する重みが 1 のときは推定に利用し，重みが 0 のときは無視する．

特徴選択の基準とするスコア関数は，離散値の種類 m に加え出現位置 k も考慮して離散値のスコアを測る．そして，スコアをもとに選択した $|\Theta_k|$ 個の離散値をゆう度比推定に利用する．そのため同じ種類の離散値であっても，N-gram での出現位置によって，選択される場合とされない場合があり得る．例えば，次の単語 2-gram x_{A} 及び x_{B} が人名の左文脈か否か，つまり人名の左に現れるか否かを二値分類することを考える．

$$x_{\mathrm{A}} = \langle \mathrm{Prime}, \underline{\mathrm{Minister}} \rangle, \quad x_{\mathrm{B}} = \langle \underline{\mathrm{Minister}}, \mathrm{Margaret} \rangle$$

ここで x_{A} は人名の左文脈だが，x_{B} は人名の一部を含むため人名の左文脈ではないことに注意する．今，m 種類目の単語 $t_{(m)}$ を “Minister” としよう．一般に “Minister” は人名の直前 ($k = 2$) によく出現する．そのため，$k = 2$ の “Minister” は $k = 1$ のそれよりも識別力が高く，分類に有用なことが推測される．しかし，もし **3.2** の関数を用いると，x_{A} と x_{B} の（出現位置が異なる）“Minister” に同じスコアを与えてしまう．そこで我々は $t_{k(m)}$ に対してスコアを計算する．すなわち提案手法では，位置 k も考慮した CET，カイ 2 乗統計量，GSS coefficient をスコア関数として用い，実験にて各々を用いた際のふるまいを比較する．各関数の詳細は **5.4** で述べる．これによって，x_{A} と x_{B} の “Minister” はそれぞれ $t_{2(m)}$，$t_{1(m)}$ と区別され，異なる重みを得ることができる．

5. 評価実験

固有表現（地名あるいは人名）の左にある単語 N-gram をゆう度比を用いて予測する．その理由は次の三点である．第一に N-gram を構成する単語は，その種類が豊富な反面，まれにしか出現しない不要なものが多いためである．この状況では特徴選択の効果を確認しやすい．第二に，地名の左 N-gram と人名の左 N-gram では特徴選択の難易度が違うためである．人名のケースでは，“Mr.” や “Mrs.” などの敬称，“President” や “Minister” などの役職・職業を意味する名詞が人名の直前に現れやすい．よって，これらが選択されるか否かが予測性能を大きく左右する．一方，地名のケースでは “in” や “at” などの前置詞が地名の前に現れやすいが，これらは他の文脈にも現れやすい．それゆえ，これら単独では地名の文脈を特定できず，特定のためには他の位置の単語も適切に考慮される必要がある．性質の異なる二種類の左 N-gram を予測することは提案手法のふるまいを解明する手がかりになる．第三に，固有表現の左 N-gram は一意に定まり，手法の定量評価ができるためである．以上を踏まえて提案手法により，予測精度の向上及び実行時間とメモリ使用量の低減がどの程度可能かを検証する．更に，単語の出現位置を考慮した複数のスコア関数を用意し，各々を用いたときのふるまいも比較する．

5.1 実験環境

実験環境を以下に示す．

- OS：Windows 10 Pro
- プロセッサ：Intel Xeon W3520 @ 2.67GHz
- メモリ：16.0GB
- Perl：v5.30.2

5.2 実験データと実験条件

実験データの作成手順を述べる．実験データはウォー

表 2　実験で用いるデータセット
Table 2　Descriptions of experimental datasets.

データ	全体の 10-gram		地名の左 10-gram	
	種類数	総頻度	種類数	総頻度
学習	3,906,050	3,922,930	62,228	62,532
開発	392,746	393,445	5,950	5,957
評価	394,850	395,145	5,713	5,716

データ	人名の左 10-gram	
	種類数	総頻度
学習	66,667	66,766
開発	7,348	7,350
評価	7,520	7,522

ル・ストリート・ジャーナルコーパス(注2)の 1987 年版をもとに作成した．まず，コーパスに含まれる記事を学習，開発，評価データへとランダムに分配した．各データサイズは学習データから順に 10,000 記事，1,000 記事，1,000 記事とした．次に各データに対し，Stanford Named Entity Recognizer (Stanford NER) [18] を用いて固有表現タグ（地名と人名）を付与した(注3)．なお N-gram の次数 N は 10 に固定した(注4)．データセットの情報を表 2 に示す．この表が示すように，10-gram の種類数は総頻度に近く，10-gram の大半は低頻度なことが分かる．また，評価データが含む 10-gram の種類のうち，99% 以上が学習データに含まれない．

実験条件は二つある．第一の条件は固有表現を地名にするか人名にするかである．第二の条件は選択される単語数 $|\Theta_k|$ である．実験では，10-gram における出現位置 k ごとの語彙 V_k から，k によらず一定サイズの部分集合 Θ_k を選択する．すなわち，学習データから選択される単語の総数は $10 \times |\Theta_k|$ 個となる．10^2，5×10^2，10^3，5×10^3，10^4，5×10^4，10^5 のうち，一つを k ごとの単語数 $|\Theta_k|$ として設定する．上記の二条件から選択肢を一つずつ選んだ組み合わせで実験する．

5.3　実 験 手 順

前節の実験条件を事前に定め，次の手順で実験を行う．まず学習データが含む全ての N-gram を単語に分解し，固有表現の左，学習データ全体における頻度を計数する．特徴選択する場合は，ここで語彙 V_k から部分集合 Θ_k を選択する．そして，評価データに含まれる N-gram x に対してゆう度比

$$r(x) = \frac{p(x \mid c_{\mathrm{NE}})}{p(x \mid \bar{c}_{\mathrm{NE}})}$$

を推定する．c_{NE} は固有表現の左に出現する x に付与されるクラスラベルであり，$\bar{c}_{\mathrm{NE}}$ は固有表現の左以外に出現する x に付与されるクラスラベルである．$r(x)$ は学習データの全語彙，あるいはそこから選択した単語を用いて推定され，推定値が大きいほど x は固有表現の左に出現しやすいと判断する．

手法の性能評価を行う．まず，x を $r(x)$ の推定値の降順に並べて順位づけて，上位から順に 8,000 件を正誤判定する．評価データで一度でも固有表現の左に出現した x は正解，それ以外の x は不正解とする．ここで上位 n 件での適合率，再現率をそれぞれ

$$\mathrm{Precision}_n = \frac{|\{x \mid x \in R \cap X_n\}|}{n}$$

$$\mathrm{Recall}_n = \frac{|\{x \mid x \in R \cap X_n\}|}{|R|}$$

と定義する．ただし，X_n は上位 n 件の N-gram を含む集合であり，$1 \le n \le 8,000$ である．R は評価データにおいて固有表現の左に出現する N-gram の集合，すなわち正解集合を意味する．そして，上位 8,000 件の N-gram を用いて F1 値

$$\mathrm{F1}_{8,000} = \frac{2 \times \mathrm{Precision}_{8,000} \times \mathrm{Recall}_{8,000}}{\mathrm{Precision}_{8,000} + \mathrm{Recall}_{8,000}}$$

を計算する．加えてスコア関数のうち，F1 値が最高のものについてはランク—再現率曲線も描画する．この曲線は横軸を x の順位 n，縦軸を n での再現率 Recall_n としたグラフ上に描かれる．グラフの原点から曲線上のある点を結んだ直線の傾きが，その点での順位 n における適合率 $\mathrm{Precision}_n$ に比例する．なお本実験では，固有表現の左に出現するか否かで N-gram を二値分類するのではなく，ゆう度比によって N-gram を順位付けする．このとき評価データの全件数を手法の評価に用いると，手法の種類によらず再現率は 1 となり，適合率は 0 に近くなるため，それらを用いて算出した F1 値は評価指標として意味のない値となってしまう．また，正則化パラメータの有無による予測精度の差は，手法間の上位でのふるまいの違いから観測されると考える．一方で有望なスコア関数を用いた提案手法では，上位に位置する N-gram はほぼ同じになると予想され，その性能差は下位でのふるまいの違いから観測されると考える．よって，次の 3 条件：1）F1

（注2）：https://catalog.ldc.upenn.edu/LDC2000T43
（注3）：Stanford NER は文脈上の長距離構造を活用しており，まれな出現文脈をもつ固有表現にもタグ付けできると考える．更に，Stanford NER は他の NER モデルと比べて良好な性能をもつこと示されている [19].
（注4）：N=2 のケースでも実験したが，N=10 のケースと同様の結果が得られたため，N=10 の結果のみ掲載する．

値で意味のある比較が可能，2）人手で実験結果の観察・分析が可能な件数，3）上位と下位での変化が予測精度に反映される適度な件数，を満たす上位 8,000 件を予測精度の評価に用いる．上述の評価に加えて，選ばれた単語のみを用いた場合の実行時間，それらの単語のみを保持したときのメモリ使用量も測定し，特徴選択しない場合（すなわち，学習データにある全語彙を使用した場合）との差を比較する．

5.4 比較手法

特徴選択の有効性を検証するため，特徴選択しない次の二手法をベースラインとする．

手法 1：All used ($\lambda = 0$) 正則化パラメータ，特徴選択法の両方を使用しない推定法である．この手法は式 (11) の正則化パラメータ λ を 0，重み $w_{k(m)}$ を 1 とした場合と等しい．

手法 2：All used (λ^*) 特徴選択法を使用しない推定法である．この手法は式 (11) の $w_{k(m)}$ を 1 とした場合と等しい．λ^* は正則化パラメータの最適値であり，これを決定する方法は後述する．

特徴選択の基準として以下のスコア関数を用意し，各々を用いた場合の提案手法のふるまいを比較する．

手法 3：Random 出現位置 k ごとの語彙 V_k から，単語をランダムに $|\Theta_k|$ 個選び，式 (12) に示した重みの決定に用いる．

手法 4：TF 語彙 V_k から，高頻度の単語を $|\Theta_k|$ 個選んで重みの決定に用いる．

手法 5：CET 特徴選択の基準として交差エントロピーを使用する．ただし提案手法では，単語の種類 m と出現位置 k の両方を考慮して単語のスコアを測定する．このとき，式 (8) で示された交差エントロピーは

$$\begin{aligned} CET_{k,m,c} =& p(t_{k(m)}, c) \log \frac{p(t_{k(m)}, c)}{p(t_{k(m)})p(c)} + \\ & p(t_{k(m)}, \bar{c}) \log \frac{p(t_{k(m)}, \bar{c})}{p(t_{k(m)})p(\bar{c})} \end{aligned} \tag{13}$$

と置き換えられる．$t_{k(m)}$ は N-gram の k 番目に位置する m 種類目の単語である．各々の確率は相対頻度で推定される．語彙 V_k から，$CET_{k,m,c}$ の高い単語を $|\Theta_k|$ 個選んで重みの決定に用いる．

手法 6：χ^2 特徴選択の基準として，$t_{k(m)}$ に対するカイ 2 乗統計量を計算する．このとき，式 (9) は

$$\chi^2_{k,m,c} = \frac{\left[p(t_{k(m)}, c)p(\bar{t}_{k(m)}, \bar{c}) - p(t_{k(m)}, \bar{c})p(\bar{t}_{k(m)}, c)\right]^2}{p(t_{k(m)}, c)p(t_{k(m)}, \bar{c})p(\bar{t}_{k(m)}, c)p(\bar{t}_{k(m)}, \bar{c})}$$

と置き換えられる．上式の確率は相対頻度で推定される．語彙 V_k から，$\chi^2_{k,m,c}$ の高い単語を $|\Theta_k|$ 個選んで重みの決定に用いる．

手法 7：GSS 特徴選択の基準として，$t_{k(m)}$ に対する GSS coefficient を計算する．このとき，式 (10) は

$$\begin{aligned} GSS_{k,m,c} =& p(t_{k(m)}, c)p(\bar{t}_{k(m)}, \bar{c}) - \\ & p(t_{k(m)}, \bar{c})p(\bar{t}_{k(m)}, c) \end{aligned}$$

と置き換えられる．上式の確率は相対頻度で推定される．語彙 V_k から，$GSS_{k,m,c}$ の高い単語を $|\Theta_k|$ 個選んで重みの決定に用いる．

手法 2 から手法 7 では，あらかじめ正則化パラメータの最適値 λ^* を設定する必要がある．そこで開発データを評価データとみなし，λ を 10^{-9}, 10^{-8}, …, 10^{-1} と変化させるごとに，推定値の降順 8,000 件までの x を用いて F1 値を計算する．そして，F1 値が最高となった λ の値を λ^* とする．

5.5 実験結果

各手法に対する F1 値を図 1 と図 2 に示す．各グラフの横軸は語彙 V_k から選択された単語数 $|\Theta_k|$，縦軸はそのときの F1 値である．最高の F1 値をもつ手法が，N-gram の予測精度に関して最良の手法となる．また図 1 と図 2 のみでは，All used (λ^*)，CET，GSS の F1 値の差異が読み取りにくいため，これら 3 手法の F1 値を表 3 と表 4 にまとめた．まず，学習データの全語彙を用いる二つのベースラインに注目すると，正則化

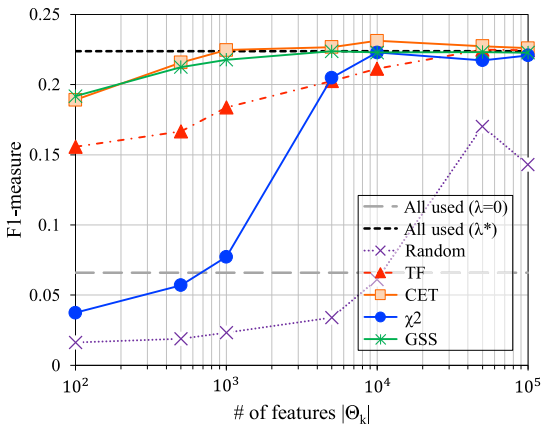


図 1 各手法に対する F1 値（地名）
Fig. 1 F1 values for each method (location names).

表 3　All used (λ^*)，CET，GSS に対する F1 値（地名）
Table 3　F1 values for All used (λ^*), CET, and GSS (location names).

手法	# of features $\lvert\Theta_k\rvert$						
	10^2	5×10^2	10^3	5×10^3	10^4	5×10^4	10^5
All used (λ^*)	0.224	0.224	0.224	0.224	0.224	0.224	0.224
CET	0.189	0.216	0.225	0.227	0.231	0.227	0.226
GSS	0.192	0.213	0.218	0.224	0.223	0.223	0.223

表 4　All used (λ^*)，CET，GSS に対する F1 値（人名）
Table 4　F1 values for All used (λ^*), CET, and GSS (person names).

手法	# of features $\lvert\Theta_k\rvert$						
	10^2	5×10^2	10^3	5×10^3	10^4	5×10^4	10^5
All used (λ^*)	0.454	0.454	0.454	0.454	0.454	0.454	0.454
CET	0.440	0.454	0.460	0.459	0.459	0.453	0.453
GSS	0.440	0.454	0.452	0.453	0.455	0.454	0.454

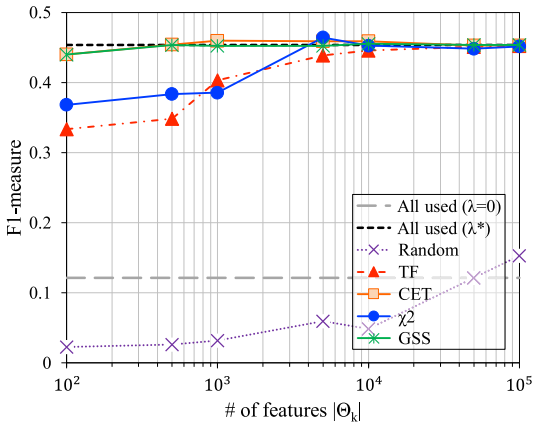


図 2　各手法に対する F1 値（人名）
Fig. 2　F1 values for each method (person names).

パラメータを使用する手法 All used (λ^*) が使用しない手法 All used ($\lambda = 0$) よりも大きな F1 値をもつ．したがって，正則化パラメータが有効なことが示唆された．また図 1 と図 2 を見比べると，人名の F1 値は地名の F1 値の倍近くあり，人名の文脈は地名の文脈よりも予測しやすいことが観察できた．次に，特徴選択の方法が異なる五つの提案手法に注目する．これらのうち，F1 値の低さが際立つのが Random である．高頻度の単語を選ぶ TF でも $\lvert\Theta_k\rvert$ が小さいときには F1 値の低さが目立つ．よってゆう度比推定でも，分類に寄与する単語を選ぶことが重要と考えられる．他の三つの提案手法 CET，χ^2，GSS は特徴選択に有用とされるスコア関数を用いた手法である．$\lvert\Theta_k\rvert$ が小さいとき χ^2 は低い F1 値をもつが，図 2 に示すように $\lvert\Theta_k\rvert = 10^4$ ときは人名のケースで最高の F1 値をもつ．これは χ^2 がスコア関数として有用な可能性を残すものの，$\lvert\Theta_k\rvert$ が小さい場合は低頻度による悪影響を受けやすい課題を浮き彫りにした．それに対して表 3 と表 4 から分かるように，CET と GSS は $\lvert\Theta_k\rvert$ の大小によらず安定した性能を保ち，$\lvert\Theta_k\rvert = 10^3$ 以上では全語彙を用いた All used (λ^*) と同等以上の F1 値を示した．したがって，CET と GSS が効果的なスコア関数とみなすことができる．

特徴選択の利点は，ゆう度比推定に要する実行時間とメモリ使用量の低減である．そこで，図 1 と図 2 で最高の F1 値を達成した手法について，実行時間とメモリ使用量の観点から All used (λ^*) と比較する．実行時間とは，学習データにある全語彙あるいはその部分集合を用いて，評価データの全 10-gram に対してゆう度比を推定するまでにかかる時間である．なお，学習データから部分集合を得るのにかかる時間は，その処理が一度で済むこと，及び部分集合が推定に使い回せることを踏まえて実行時間には含めていない．メモリ使用量とは，推定に用いる語彙とその頻度を学習データから全て格納した際のメモリ使用量である．加えて上位 8,000 件までのランク—再現率曲線も描く．F1 値が上位 8,000 位という一点での予測精度を示すのに対し，ランク—再現率曲線では上位 8,000 位に至るまでの予測精度を示す．この曲線では，上位で適合率が高く，下位で高い再現率を保つ手法が優れているとみなす．

特徴選択の有無に対する予測精度，実行時間，メモリ使用量の比較結果を図 3 と図 4 に示す．ランク—再現率曲線を描くには $\lvert\Theta_k\rvert$ を固定する必要がある．そこで，図 1 と図 2 で最高の F1 値を示す $\lvert\Theta_k\rvert$ をそれぞれ選択した．実行時間とメモリ使用量の両グラフでは，曲線に対応する $\lvert\Theta_k\rvert$ の点を矢印で指示している．な

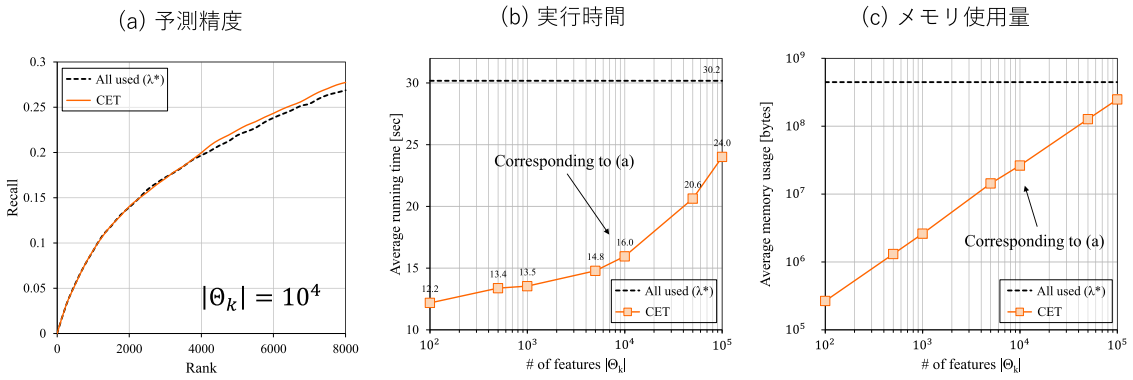


図 3　特徴選択の有無に対する予測精度，実行時間，メモリ使用量の比較結果（地名）
Fig. 3　Comparison results of prediction accuracy, running time, and memory usage with and without feature selection (location names).

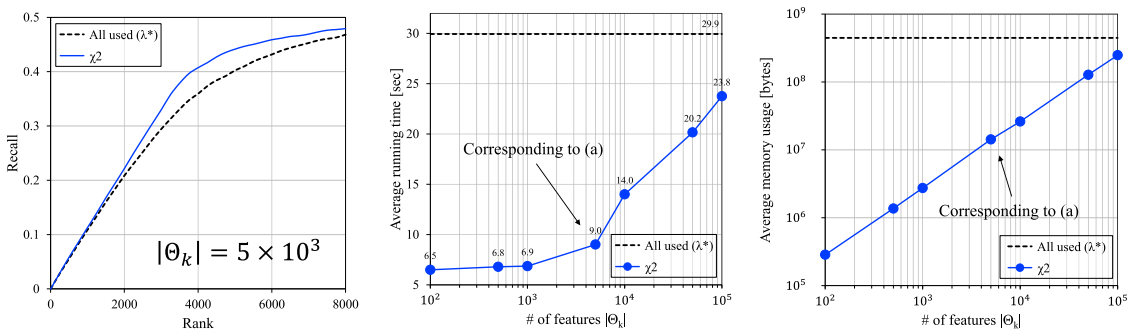


図 4　特徴選択の有無に対する予測精度，実行時間，メモリ使用量の比較結果（人名）
Fig. 4　Comparison results of prediction accuracy, running time, and memory usage with and without feature selection (person names).

お，実行時間とメモリ使用量はプログラム実行ごとにわずかに異なるため，10 回実行した際の算術平均をプロットした．まず固有表現が地名の結果に注目する．図 3(a) から分かるように，CET のランク—再現率曲線は All used (λ^*) とほぼ一致する．そして図 3(b) と図 3(c) の対応点を見ると，CET の実行時間は All used (λ^*) の 1/2，メモリ使用量は 1/10 程度に削減できた．次に固有表現が人名の結果に注目する．図 4(a) から分かるように，χ^2 の曲線は All used (λ^*) と比較して上位 4,000 位程度まで傾きが大きい．これは上位で χ^2 が All used (λ^*) よりも高い適合率をもつことを意味する．更に 4,000 位以降でも χ^2 は高い再現率を維持した．よって χ^2 の優位性を確認できた．また，図 4(b) と図 4(c) の対応点を見ると，χ^2 の実行時間は All used (λ^*) の 1/3，メモリ使用量は 1/10 程度に削減できた．以上から提案手法の有効性を確認した．

6. 考　　察

正則化パラメータ λ と特徴選択法の相互作用を議論する．スコア関数として式 (13) の CET を用い，λ の値が異なる二手法の F1 値を図 5 と図 6 に示す．CET ($\lambda = 0$) は正則化パラメータを用いない手法である．CET (λ^*) は正則化パラメータに最適値 λ^* を設定した手法であり，実験で用いた手法 5 と同じである．まず CET ($\lambda = 0$) に着目すると，どの $|\Theta_k|$ でも図 1 と図 2 に示した All used ($\lambda = 0$) より F1 値が高いことが分かる．これは推定に悪影響を及ぼす低頻度語が特徴選択によって除かれたためと考えられ，特徴選択法のみでもゆう度比推定への有効性を確認できた．しかし $|\Theta_k|$ が大きくなると，低頻度語も推定に用いることになり，それらを適切に扱えない CET ($\lambda = 0$) は F1 値が急激に低下してしまう．次に CET (λ^*) に着目すると，F1

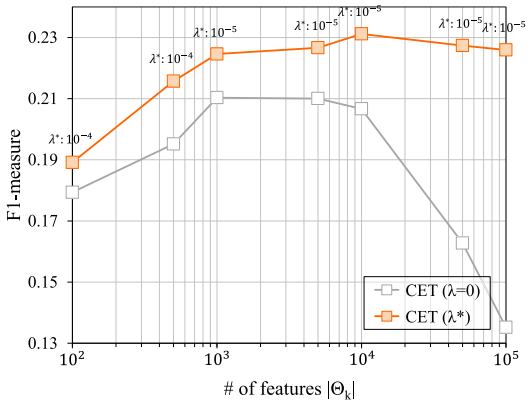


図 5　λ の有無に対する F1 値の比較結果（地名）
Fig. 5　Comparison results of F1 values with and without λ (location names).

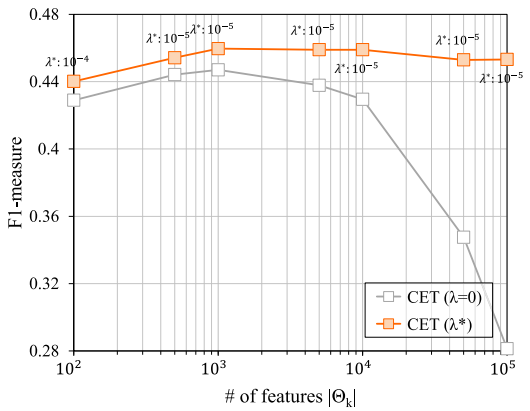


図 6　λ の有無に対する F1 値の比較結果（人名）
Fig. 6　Comparison results of F1 values with and without λ (person names).

値は常に高く，急激な低下も見られない．ここで注目すべき点は，$|\Theta_k|$ が 10^2 と小さいときでさえ，F1 値が高いことである．特徴選択の語彙のサイズが小さいときは，頻度が富んでいて識別力もあるごく少数の単語のみを推定に用いる．前述の結果は，そのような状況でも正則化パラメータが有効なことを示唆した．また，図中では $|\Theta_k|$ が指数関数的に変化するのに対し，最適値 λ^* は 10^4 か 10^5 で安定している．このことは，$|\Theta_k|$ の変化に対して λ がロバストであることを意味する．この性質は λ^* を決定する際に役立つと考えられる．以上から，正則化パラメータと特徴選択法は互いに有効に作用することを確認した．

7. むすび

本論文では，低頻度及びゼロ頻度 N-gram の両方に対処できるゆう度比の推定法を提案した．ゼロ頻度を有効に扱う一方法は，N-gram をなす離散値 t_k の出現に確率的な独立性を仮定し，それらのゆう度比の積を取ることである．また，t_k のゆう度比を推定するのに低頻度への対処法 [2] を適用すれば，低頻度の N-gram に対しても安定した推定値を付与できる．しかし，t_k 間に仮定した独立性は実際にはあまり成立しない．また，膨大な数の t_k を扱うため，推定の精度と効率が低下してしまう．これらの問題を避ける目的で提案手法では，文書分類のための特徴選択法を前述の方法に組み合わせた．実験では，固有表現の左に出現する単語 N-gram をゆう度比で予測した．そして提案手法が，コーパス中の全語彙を推定に用いた手法と同等以上の予測精度を保ち，かつ効率良くゆう度比推定できることを確認した．また特徴選択に有望なスコア関数として，単語の種類と出現位置の両方を考慮した CET，χ^2，GSS を用い，それぞれのふるまいを比較した．結果として，χ^2 は低頻度語の扱いに問題があるものの，CET と GSS は安定した良い性能を示すことが分かった．今回は提案手法の有効性を適切に評価するため，種類が豊富でそのほとんどがまれにしか出現しない N-gram をゆう度比の推定対象とした．今後は，推定対象が同様の性質をもつ実用的なタスクを探し，ゆう度比推定を介してそれを解くことで提案手法の実用性も検証したい．

謝辞　本研究の一部は JSPS 科研費 JP22K18006 の助成を受けたものです．

文　献

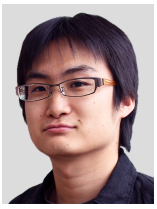
[1] C.D. Manning and H. Schütze, Foundations of statistical natural language processing, MIT press, 1999.
[2] 菊地真人，川上賢十，吉田光男，梅村恭司，“観測頻度に基づくゆう度比の保守的な直接推定，”信学論（D），vol.J102-D, no.4, pp.289–301, April 2019.
[3] S.F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” Computer Speech & Language, vol.13, no.4, pp.359–394, 1999.
[4] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz, Nonparametric and semiparametric models, Springer Science & Business Media, 2012.
[5] J. Huang, A.J. Smola, A. Gretton, K.M. Borgwardt, and B. Schölkopf, “Correcting sample selection bias by unlabeled data,” NIPS, pp.601–608, 2007.
[6] S. Bickel, M. Brückner, and T. Scheffer, “Discriminative learning for differing training and test distributions,” ICML, pp.81–88,

2007.
[7] M. Sugiyama, S. Nakajima, H. Kashima, P. vonBünau, and M. Kawanabe, “Direct importance estimation with model selection and its application to covariate shift adaptation,” NIPS, pp.1433–1440, 2008.
[8] T. Kanamori, S. Hido, and M. Sugiyama, “A least-squares approach to direct importance estimation,” J. Mach. Learn. Res., vol.10, pp.1391–1445, July 2009.
[9] X. Deng, Y. Li, J. Weng, and J. Zhang, “Feature selection for text classification: A review,” Multimedia Tools and Applications, vol.78, no.3, pp.3797–3816, 2019.
[10] D. Mladenić and M. Grobelnik, “Feature selection for unbalanced class distribution and naive bayes,” ICML, pp.258–267, 1999.
[11] Y. Yang and J.O. Pedersen, “A comparative study on feature selection in text categorization,” ICML, pp.412–420, 1997.
[12] L. Galavotti, F. Sebastiani, and M. Simi, “Experiments on the use of feature selection and negative evidence in automated text categorization,” TPDL, pp.59–68, 2000.
[13] J.G. Dy and C.E. Brodley, “Feature selection for unsupervised learning,” J. Mach. Learn. Res., vol.5, pp.845–889, Aug. 2004.
[14] P. Pudil, J. Novovičová, and J. Kittler, “Floating search methods in feature selection,” Pattern Recognit. Lett., vol.15, no.11, pp.1119–1125, 1994.
[15] L.C. Molina, L. Belanche, and À. Nebot, “Feature selection algorithms: A survey and experimental evaluation,” ICDM, pp.306–313, 2002.
[16] S. Das, “Filters, wrappers and a boosting-based hybrid for feature selection,” ICML, pp.74–81, 2001.
[17] E.P. Xing, M.I. Jordan, and R.M. Karp, “Feature selection for high-dimensional genomic microarray data,” ICML, pp.601–608, 2001.
[18] J.R. Finkel, T. Grenager, and C. Manning, “Incorporating non-local information into information extraction systems by Gibbs sampling,” ACL, pp.363–370, 2005.
[19] R. Jiang, R.E. Banchs, and H. Li, “Evaluating and combining name entity recognition systems,” NEWS, pp.21–27, 2016.

（2022 年 2 月 26 日受付，7 月 28 日再受付，
9 月 28 日早期公開）

菊地　真人　（正員）

2021 豊橋技術科学大学大学院工学研究科情報・知能工学専攻博士後期課程了．同年より名古屋工業大学大学院工学研究科（情報工学専攻）助教，現在に至る．博士（工学）．自然言語処理，データ工学，Web インテリジェンスに関する研究に従事．2021 SMASH21 Summer Symposium 奨励賞，2022 FIT 論文賞各受賞．情報処理学会，言語処理学会，電子情報通信学会，IEEE 各会員．



吉田　光男

2014 筑波大学大学院システム情報工学研究科博士後期課程了．同年より豊橋技術科学大学大学院工学研究科（情報・知能工学系）助教．2021 筑波大学ビジネスサイエンス系准教授，現在に至る．博士（工学）．ウェブ工学，自然言語処理，計算社会科学に関する研究に従事．情報処理学会，言語処理学会，人工知能学会，日本データベース学会各会員．

梅村　恭司　（正員）

1983 東京大学大学院工学系研究科情報工学専攻修士課程了．同年日本電信電話公社電気通信研究所入所．1995 豊橋技術科学大学工学部情報工学系助教授，2003 同教授，現在に至る．博士（工学）．自然言語処理，システムプログラム，記号処理に関する研究に従事．情報処理学会，IEEE，電子情報通信学会，日本ソフトウェア科学会，言語処理学会，計量国語学会，ACM 各会員．

大囿　忠親　（正員）

2000 名古屋工業大学大学院工学研究科電気情報工学専攻博士後期課程了．同年より同大学工学部知能情報システム学科助手．2006 同大学助教授．2007 同大学准教授．2019 同大学教授，現在に至る．博士（工学）．Web インテリジェンス，マルチエージェント，リアルタイム協調支援の研究に従事．2003〜2004 にかけてマレーシア・マルチメディア大学客員研究員．2004（株）ウィズダムウェブ設立創業者最高技術責任者．2004 情報処理学会全国大会優秀賞，2012 年度人工知能学会研究会優秀賞，2013 LOD チャレンジ Japan 2013 データセット部門優秀賞，2018 FIT 論文賞各受賞．日本ソフトウェア科学会（理事），電子情報通信学会（和文論文誌編集委員），情報処理学会（シニア会員），人工知能学会，AAAI，ACM 各会員．