CD-006

# 特徴選択を導入した低・ゼロ頻度 N-gram の効率的な尤度比推定法

# Efficient Likelihood Ratio Estimation Method for Low- and Zero-frequency N-grams Using Feature Selection

菊地 真人[1)]　吉田 光男[2)]　梅村 恭司[2)]　大囿 忠親[1)]
Masato Kikuchi　Mitsuo Yoshida　Kyoji Umemura　Tadachika Ozono

## 1　はじめに

尤度比は確率分布の比で定義される尺度であり，尤度比を利用したアプリケーションは多岐にわたる．尤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うる．自然言語処理（NLP）では，文字列や単語列に対する尤度比を，コーパスから得た頻度情報を用いて推定することがある [1]．このとき，尤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．

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

$x = \langle t_1, t_2, \ldots, t_{\mathrm{N}} \rangle$ は N 個の文字や単語が連なる系列を表し，NLP においては N-gram と呼ばれる．$t_k$, $1 \le k \le \mathrm{N}$ は $x$ を構成する $k$ 番目の文字や単語である．$f_*(x)$ は確率密度 $p_*(x)$ に従う確率分布から得た $x$ 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存在し，そのほとんどがまれにしか出現しない，すなわち低頻度のものが多数を占める特性がある．そのため，言語要素の観測頻度から尤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

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

と定義される．ここで $\lambda$ は後述するパラメータである．確率分布の推定を介して尤度比を間接的に求める方法と異なり，この手法は二乗損失の最小化問題を解くことで尤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 $\lambda$ ($\ge 0$) 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 $x_{\mathrm{a}}$ と $x_{\mathrm{b}}$ 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て，$r_{\mathrm{MLE}}(x_{\mathrm{a}})$ と $r_{\mathrm{MLE}}(x_{\mathrm{b}})$ は共に 50 と大きな値になる．ここで，$f_{\mathrm{nu}}(x_{\mathrm{b}})$ 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 に近いのに対し，$\widehat{r}(x_{\mathrm{b}})$ は 8.3 となり $r_{\mathrm{MLE}}(x_{\mathrm{b}})$ の 50 よりも遥かに低い．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 $x_{\mathrm{b}}$ と $x_{\mathrm{c}}$ に着目すると，$f_{\mathrm{nu}}(x_{\mathrm{b}})$ と $f_{\mathrm{nu}}(x_{\mathrm{c}})$ の差が 1 しかないにもかかわらず，$r_{\mathrm{MLE}}(x_{\mathrm{c}})$ 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 $f_*(x_{\mathrm{c}})$ は共に低頻度であるため，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．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 $x_{\mathrm{d}}$ に着目すると，$f_{\mathrm{nu}}(x_{\mathrm{d}})$ がゼロであるため，$r_{\mathrm{MLE}}(x_{\mathrm{d}}) = \widehat{r}(x_{\mathrm{d}}) = 0$ 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．

**表 1　尤度比の推定例．$\widehat{r}(x)$ の $\lambda$ は $10^{-5}$ とした．**

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

先述のように，コーパスに含まれる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．さらに，コーパスのサイズは有限であるから，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，実用上はそのときでさえ情報のある推定値を返すことが要求される．以上を踏まえると，低頻度のみならずゼロ頻度に対しても，有益な推定値を算出できる尤度比の推定法が必要と考える．そこで本稿では，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尤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 $t_k$ を個別に扱い，$r(x)$ 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 に対する尤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 $t_k$ を個別に扱うには，$t_k$ 間に統計的な独立性を仮定しなければならない．この仮定は実際には成立しないことが多く，尤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，尤度比推定に不要なものや悪影響を及ぼすものが多数存在する．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

1）　名古屋工業大学大学院 情報工学専攻
2）　豊橋技術科学大学 情報・知能工学系

徴選択法を，尤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，それを用いて効率よく尤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尤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よく尤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，低頻度およ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．スムージング法は尤度比推定にそのまま応用できないが，尤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，この方法で求めた尤度比は，実用性が低いことが明らかになっている [2]．したがって，尤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尤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尤度比を直接推定する手法が提案されてきた [5, 6, 7, 8]．これらの直接推定法は，連続空間上で定義される尤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尤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] を離散的な尤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，低頻度から求まる尤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，いく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, 11, 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 $\Theta$ 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，効率的かつ効果的なフィルタ方式に属する．本稿で提案する尤度比の推定量は，ハイパーパラメータである正則化パラメータを持つ．ゆえに，尤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，尤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 尤度比の直接推定法

$D \subset \mathbb{U}$ をデータの定義域とする．$\mathbb{U}$ は $v$ 個の離散要素を持つ集合であり，情報理論では有限アルファベットとも呼ばれる．いま，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．ここで要素 $x$ は単語（列）や文字（列）などの離散値であり，このとき $v$ は存在しうる要素の種類数と等しい．これまでの先行研究と同様に，確率密度 $p_{\mathrm{de}}(x)$ 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 $x$ に対して尤度比を定義できる．本節では，二つの標本 $\{x_i^{\mathrm{de}}\}_{i=1}^{n_{\mathrm{de}}}$, $\{x_i^{\mathrm{nu}}\}_{i=1}^{n_{\mathrm{nu}}}$ から尤度比

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

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，二乗損失の最小化プロセスで尤度比を直接推定する手法である．uLSIF では尤度比 $r(x)$ 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．なお，$b$ と $\{\varphi_l\}_{l=1}^{b}$ は推定に用いる標本 $\{x_i^{\mathrm{de}}\}_{i=1}^{n_{\mathrm{de}}}$, $\{x_i^{\mathrm{nu}}\}_{i=1}^{n_{\mathrm{nu}}}$ と独立である．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活用する．しかし，本稿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．添え字 $l$ は存在しうる $v$ 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，uLISF へ導入すると解が解析的に求められる利点がある．式 (1) を推定モデル $\widehat{r}(x_{(m)})$, $1 \leq m \leq v$ 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．uLISF では，推定モデル $\widehat{r}(x_{(m)})$ と真の尤度比 $r(x_{(m)})$ の二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[1)]．$\mathbb{R}^\nu$ は実 $\nu$ 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 $\ell_2$-正則化項である．$\widehat{\boldsymbol{H}}$ は $\nu \times \nu$ 行列であり，その $(l, l')$ 番目の要素は

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

と定義される．$f_*(x_{(l)}), * \in \{\mathrm{de}, \mathrm{nu}\}$ は確率密度 $p_*(x)$ を持つ確率分布から得られた $x_{(l)}$ 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 $\nu$ 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) は拘束無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．

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

$\mathbf{1}_\nu$ は要素がすべて 1 の $\nu$ 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．

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

本来の uLSIF は解が負値となる場合があり，尤度比の非負性を考慮して負値をゼロに丸める必要がある．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 $\lambda\ (\geq 0)$ が推定値を低めに見積もりロバストにする．また，この式は尤度比の分母のみを補正する形式になる．なお $\lambda = 0$ のとき，この式は確率分布を最尤推定して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．いま，N 個の単語ベクトル $\langle t_1, t_2, \ldots, t_{\mathrm{N}}\rangle$ で表される文書 $d$ が与えられたとする．ここで $t_k,\ 1 \leq k \leq \mathrm{N}$ 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

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

として定式化される．$\widehat{c}(d)$ 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が低下する．

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 $\Theta$ 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，$\Theta$ が含む単語を選ぶ基準として計算されたスコアを用いる．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，これまでによく用いられてきた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

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

と定義される．$1 \leq m \leq \nu$ は単語の種類を指定する添え字である．すなわち，$t_{(m)}$ は $\nu$ 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 $c$ および $\bar{c}$ と独立であるとき，言い換えると，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．

二つ目はカイ二乗統計量 [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，カイ二乗分布によってモデル化される．文書内の単語に対する否定的証拠を用いるカイ二乗統計量は

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

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．カイ二乗統計量と同様に，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．いくつかのデータセットにおいて，この関数はカイ二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，文書は役割の定まった特徴から構成されてはおらず，その長さ（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．それ対し

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

て本稿では，固有表現の左（単語）N-gram を予測する問題を扱う．4 節で述べるように，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，尤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尤度比推定に応用する．

## 4 提案手法

特徴ベクトル $x = \langle t_1, t_2, \ldots, t_{\mathrm{N}} \rangle$ に対する次の尤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 $t_k$, $1 \leq k \leq \mathrm{N}$ 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，二つの確率分布をそれぞれ相対頻度で求めて，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，有益な推定値の算出が困難になる．

これに対する素朴な対策として，$x$ を構成する $t_k$ を個別に扱い，尤度比 $r(x)$ を $r(t_k)$ の積で近似することが考えられる．この近似は，$t_k$ の出現が他の離散値の出現と統計的に独立という仮定に基づく．また，$r(t_k)$ の推定に 3.1 節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，前述の仮定は実際に成立しないことが多く，尤度比の推定精度を低下させる原因になる．加えて，$x$ の代わりに $t_k$ を扱うと膨大な計算が必要となるため，尤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，推定精度の向上およ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 $\lambda$ ($\geq 0$) は正則化パラメータである．なお $t_k$ 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．そして，離散値の種類 $m$ に加えて出現位置 $k$ も考慮した次式の重み $w_{k(m)}$ により特徴選択を実現する．

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

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，$\Theta_k$ は位置 $k$ で出現しうる離散値の集合 $V_k$ から 3.2 節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された $|\Theta_k|$ 個の離散値を尤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 中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うる．例えば，次の単語 2-gram $x_{\mathrm{A}}$ および $x_{\mathrm{B}}$ 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．

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

ここで $x_{\mathrm{A}}$ は人名の左文脈だが，$x_{\mathrm{B}}$ 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．いま，$m$ 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 節の関数を用いると，$x_{\mathrm{A}}$ と $x_{\mathrm{B}}$ 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，カイ二乗統計量，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 節で述べる．これによって，$x_{\mathrm{A}}$ と $x_{\mathrm{B}}$ 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 を尤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，予測精度の向上およ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．さらに，単語の出現位置を考慮した複数の特徴選択法を用意し，各々を用いたときのふるまいの差異も明らかにする．

### 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 @ 2.67GH
- メモリ: 16.0 GB

表 2　実験で用いるデータセット．

| データ | 全体 | | 地名の左 | |
|---|---|---|---|---|
| | 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 | |
|---|---|---|
| | 種類数 | 総頻度 |
| 学習 | 66,667 | 66,766 |
| 開発 | 7,348 | 7,350 |
| 評価 | 7,520 | 7,522 |

• 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ル・ストリート・ジャーナルコーパスの 1987 年版をもとに作成された．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 [18] を用いて固有表現タグ（地名と人名）を付与した．なお N-gram の次数 N は 10 に固定した[2)]．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いため，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は含まれないゼロ頻度のものであった．

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 $k$ 毎の語彙 $V_k$ から，$k$ によらず一定サイズの部分集合 $\Theta_k$ 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．我々は $10^2$，$5\times10^2$，$10^3$，$5\times10^3$，$10^4$，$5\times10^4$，$10^5$ のうち，一つを $k$ 毎の単語数 $|\Theta_k|$ 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 $\Theta_k$ を選択する．そして，評価データに含まれる N-gram $x$ に対して尤度比

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

を推定する．$c_{\mathrm{NE}}$ 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．そして F1 尺度を計算する．またスコア関数のうち，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，縦軸をその点での再現率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，その順位での適合率に比例する．適合率，再現率，F1 尺度はそれぞれ

$$\text{Precision} = \frac{|\{x \mid x \in R\}|}{|\{x\}|}, \quad \text{Recall} = \frac{|\{x \mid x \in R\}|}{|R|},$$
$$\text{F1-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

と定義される．ただし，$R$ 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．さらに，選ばれた単語のみを用いた場合の実行時間，それら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（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 $\lambda$ を 0，重み $w_{k(m)}$ を 1 とした場合と等しい．

**手法 2： All used ($\lambda^*$)**　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．$\lambda^*$ 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

$$CET_{k,m,c} = p(t_{k(m)}, c) \log \frac{p(t_{k(m)}, c)}{p(t_{k(m)})p(c)} + p(t_{k(m)}, \bar{c}) \log \frac{p(t_{k(m)}, \bar{c})}{p(t_{k(m)})p(\bar{c})} \quad (13)$$

と置き換えられる．ここで，$t_{k(m)}$ は N-gram の $k$ 番目に位置する $m$ 種類目の単語である．上式にある各々の確

2)　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．

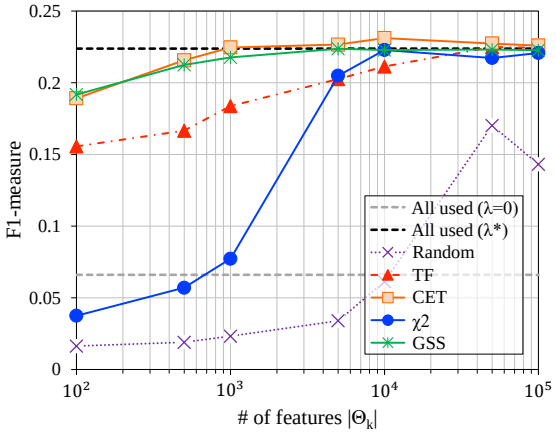


図 1　各手法に対する F1 値（地名）.

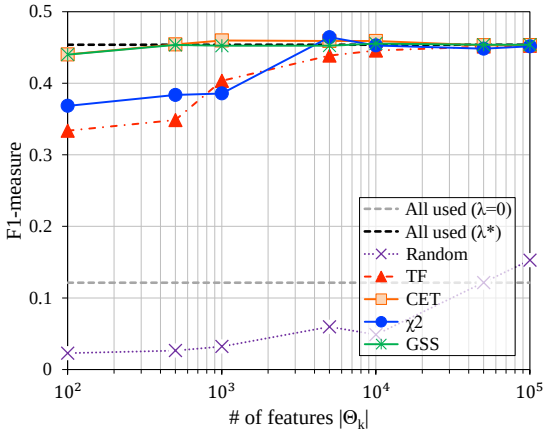


図 2　各手法に対する F1 値（人名）.

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：** $\chi^2$　特徴選択の基準として，$t_{k(m)}$ に対するカイ二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

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

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，性能評価をする前に正則化パラメータの最適値 $\lambda^*$ 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，$\lambda$ を $10^{-9}, 10^{-8}, \ldots, 10^{-1}$ 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 $\lambda$ の値を $\lambda^*$ 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 値を持つ手法が，N-gram の予測精度に関して最良の手法となる．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，正則化パラメータを使用する手法 All used ($\lambda^*$) が使用しない手法 All used ($\lambda = 0$) よりも大きな F1 値を持つ．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 5 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 $|\Theta_k|$ が小さいときには F1 値の低さが目立つ．よって尤度比推定でも，分類に寄与する単語を選ぶことが重要と考えられる．残す 3 つ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．$|\Theta_k|$ が小さいとき $\chi^2$ は低い F1 値を持つが，図 2 に示すように $|\Theta_k| = 10^4$ ときは人名のケースで最高の F1 値を持つ．これは $\chi^2$ がスコア関数として有用な可能性を残すものの，$|\Theta_k|$ 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，CET と GSS は $|\Theta_k|$ の大小によらず安定した性能を保ち，$|\Theta_k| = 10^3$ 以上では全語彙を用いた All used ($\lambda^*$) 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，尤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 ($\lambda^*$) 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 に対して尤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，およ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 $|\Theta_k|$ 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 $|\Theta_k|$ 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 $|\Theta_k|$ の点を矢印で指示している．なお，実行時間とメモリ使用量はプログラム実行毎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 ($\lambda^*$) とほぼ一致する．そして図 3(b) と図 3(c) の対応点を見る

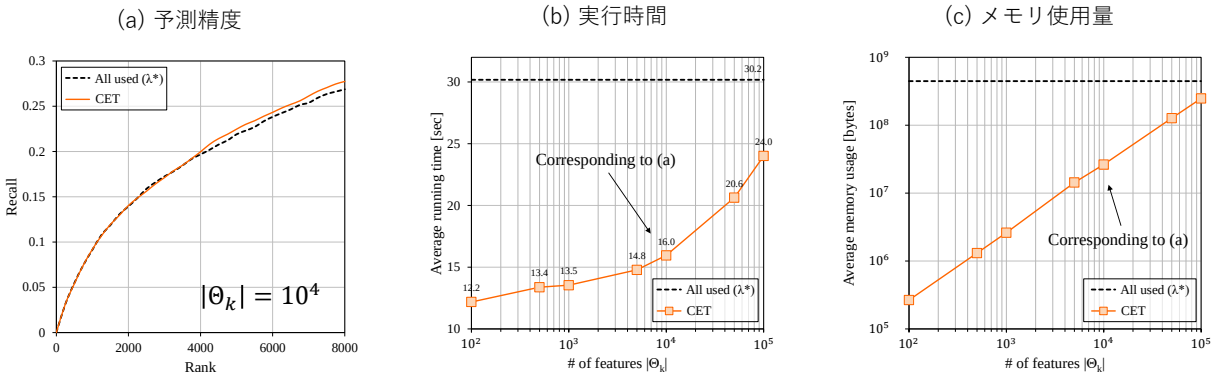


**図 3　特徴選択の有無に対する予測精度，実行時間，メモリ使用量の比較（地名）．**

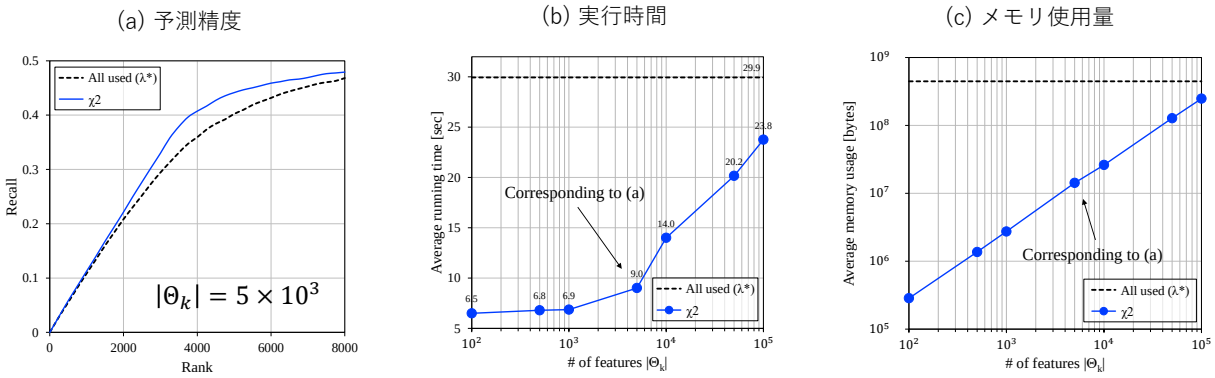


**図 4　特徴選択の有無に対する予測精度，実行時間，メモリ使用量の比較（人名）．**

と，CET の実行時間は All used ($\lambda^*$) の 1/2，メモリ使用量は 1/10 程度に削減できている．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，$\chi^2$ の曲線は All used ($\lambda^*$) と比較して上位 4,000 位程度まで傾きが大きい．これは上位において $\chi^2$ が All used ($\lambda^*$) よりも高い適合率を持つことを意味する．さらに 4,000 位以降でも $\chi^2$ は高い再現率を維持した．よって $\chi^2$ 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，$\chi^2$ の実行時間は All used ($\lambda^*$) の 1/3，メモリ使用量は 1/10 程度に削減できている．以上から提案手法の有効性を確認した．

## 6　考察

本節では，正則化パラメータ $\lambda$ 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，$\lambda$ 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 ($\lambda^*$) は正則化パラメータに最適値 $\lambda^*$ を設定した手法であり，評価実験で用いた手法 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尤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 ($\lambda^*$) に着目すると，F1 値は常に高く，急

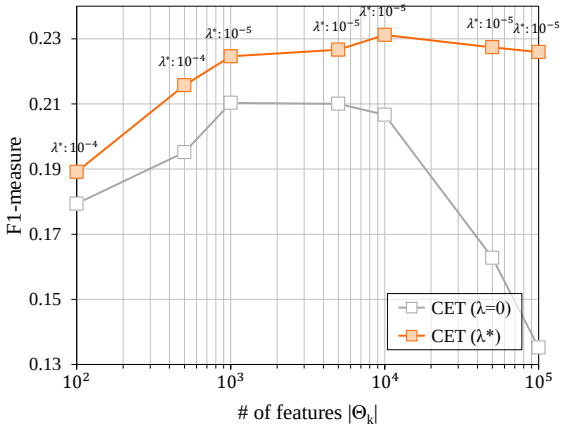


**図 5　$\lambda$ の有無に対する F1 値の比較（地名）．**

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 $\lambda^*$ 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 $\lambda$ がロバストであることを意味する．この性質は $\lambda^*$ 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．

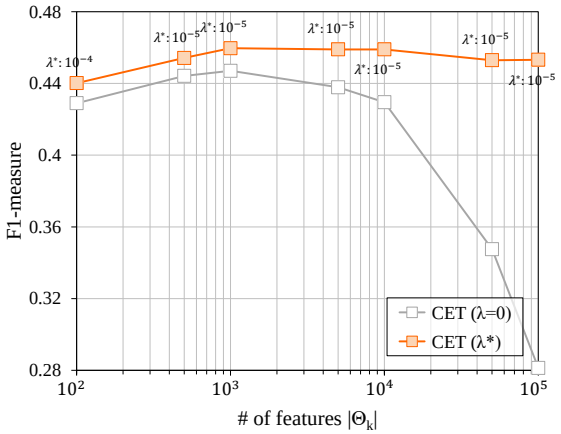


**図 6　$\lambda$ の有無に対する F1 値の比較（人名）.**

## 7　おわりに

本稿では，低頻度およびゼロ頻度 N-gram の両方に対処できる尤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 $t_k$ を個別に扱い，それらの尤度比の積を取ることである．また，$t_k$ の尤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 $t_k$ を個別に扱うには，実際にはあまり成立しない統計的な独立性を $t_k$ 間に仮定する必要がある．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 を尤度比で予測した．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，かつ効率よく尤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，$\chi^2$，および GSS を用い，それぞれのふるまいを比較した．結果として，$\chi^2$ は低頻度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 を尤度比の推定対象とした．今後は，推定対象が同様の性質を持つ実用的なタスクを探し，尤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 19K12266 の助成を受けたものです．

### 参考文献

[1] Christopher D. Manning and Hinrich 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. 電子情報通信学会論文誌 D, Vol. J102-D, No. 4, pp. 289–301, 2019.
[3] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, Vol. 13, No. 4, pp. 359–394, 1999.
[4] Wolfgang Härdle, Marlene Müller, Stefan Sperlich, and Axel Werwatz. *Nonparametric and semiparametric models*. Springer Science & Business Media, 2012.
[5] Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, pp. 601–608, 2007.
[6] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *ICML*, pp. 81–88, 2007.
[7] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, pp. 1433–1440, 2008.
[8] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, Vol. 10, pp. 1391–1445, July 2009.
[9] Xuelian Deng, Yuqing Li, Jian Weng, and Jilian Zhang. Feature selection for text classification: A review. *Multimedia Tools and Applications*, Vol. 78, No. 3, pp. 3797–3816, 2019.
[10] Dunja Mladenić and Marko Grobelnik. Feature selection for unbalanced class distribution and naive bayes. In *ICML*, pp. 258–267, 1999.
[11] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, pp. 412–420, 1997.
[12] Luigi Galavotti, Fabrizio Sebastiani, and Maria Simi. Experiments on the use of feature selection and negative evidence in automated text categorization. In *TPDL*, pp. 59–68, 2000.
[13] Jennifer G. Dy and Carla E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, Vol. 5, pp. 845–889, Aug. 2004.
[14] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, Vol. 15, No. 11, pp. 1119–1125, 1994.
[15] Luis Carlos Molina, Lluís Belanche, and Àngela Nebot. Feature selection algorithms: A survey and experimental evaluation. In *ICDM*, pp. 306–313, 2002.
[16] Sanmay Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *ICML*, pp. 74–81, 2001.
[17] Eric P. Xing, Michael I. Jordan, and Richard M. Karp. Feature selection for high-dimensional genomic microarray data. In *ICML*, pp. 601–608, 2001.
[18] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL*, pp. 363–370, 2005.