# probabilistic Polynomial Semantic Indexingの提案とテキストジャンル推定への適用

○箕浦健太郎，田村哲嗣，速水悟（岐阜大）

## 1. はじめに

機械学習において，クラス分類は音声処理，音楽情報処理，画像処理，テキスト情報処理など幅広い分野において利用されている重要なタスクである．

本稿では，検索システムにおいて与えられたクエリに対して適切なWebページを表示するためのページランク付け手法として提案されたPolynomial Semantic Indexing(PSI;多項意味索引)[Bai 2009]をクラス分類へと応用する手法について提案し，これを用いてテキストジャンル推定を行う．

PSIはパラメータであるテンソルを行列の積として近似することで次元数の大きいデータに対しても圧縮などの処理を行うことなく適応することが可能である．この近似は潜在的概念空間への写像とその空間上での内積及びその拡張(以下拡張内積と呼ぶ)によって定義される．この特徴により次元数の大きいデータに対しても次元圧縮(次元低減)，特徴抽出などの処理を行うことなく扱うことができる．本稿では，このPSIについて，指数関数，事後確率を用いて確率的表現probabilistic PSI(pPSI;確率的多項意味索引)を提案する．これにより，出力，及び精度の安定化，モデルに基づく訓練，半教師あり学習の効率化，データ数に応じた推定手法の利用が可能になるなどのメリットが挙げられる．これをクラス分類へ適用することでテキストジャンル推定を行う．

先行研究[箕浦 2011]では，分類器としてpPSIを利用した場合，データによってSVMよりも優れた分類精度となることを示したが，本稿ではテキスト情報処理における分類精度を，毎日新聞記事コーパスを用いて示し，(multiclass-)SVMの分類精度と比較する．

## 2. 関連研究

### 2.1 潜在的概念空間

PSIは訓練によって$N$次元潜在的概念空間を形成する．同様に$N$次元概念空間を形成する手法に後述するLatent Semantic Indexing(LSI;潜在的意味索引)[Deerwester 1990], probabilistic LSI(pLSI; 確率的潜在的意味索引)[Hofmann 1999]などがある．

これらはいずれもテキスト情報処理のための手法であるが，ラベル付けされていないデータに基づく教師なしモデルであり，パラメータが学習データのみに依存する(このためデータの解析に向いているといえる)．したがって限られたデータに基づいた結果となる．またデータの解析のみで処理後のデータの扱いには関与しないため，目的とするタスクに直接的に作用しないという欠点がある．

#### (1) Latent Semantic Indexing(LSI)

LSIは，処理の対象とするデータの特異値分解(SVD)によって得た特異値の大小関係を元にして優先的に使用する潜在的概念空間軸を定めることによって概念空間を生成し，その概念空間への写像によって元のデータを$N$次元へと次元圧縮する手法である．多くはテキスト情報処理に利用されている．LSIは最小二乗法によって計算される．

#### (2) probabilistic LSI(pLSI)

pLSIは，処理の対象とするデータに内在するトピックを潜在パラメータとして用いることによって，データの生成モデルを表現したものである．pLSIは最尤法に基づいてEMアルゴリズムなどを利用して計算される．この場合，パラメータがSVDと同様の形式で表現されるため，それを用いて$N$次元概念空間を導き出すことができ，これによりデータに内包されたトピックを抽出することが可能である．

### 2.2 クラス分類

本稿の目的であるクラス分類では，一般的にSupport Vector Machine(SVM), Naïve Bayes Classifier(単純ベイズ分類器)などの分類器が用いられている．

#### (1) Support Vector Machine(SVM)

SVMは2クラス分類に用いられ，学習データをクラス毎に各データ点との距離が最大となる平面(分離平面)によって判別する手法である．SVMの特徴は，この分離平面に最も近いデータとの距離(マージン)を最大にするように分離平面を決定する点にある．これにより高精度な分類を実現している．また多クラスに対する適応についても様々な手法が編み出されている．

#### (2) Naïve Bayes Classifier

Naïve Bayes Classifierは，入力ベクトルの各要素に対して強い独立性を仮定したモデルであり，特定のクラスに該当する確率は入力ベクトルの要素毎の条件付き確率(入力ベクトルが既知のとき，対象のクラスに該当する確率)の積で表現される．このモデルは学習に多くのデータを必要とせず学習が高速である点でメリットがある．

## 3. Polynomial Semantic Indexing(PSI)

### 3.1 拡張内積

PSIの定義を行うのに先立ち，定義において利用する拡張内積について定義する．$H$個の$I$次元のベクトル$\boldsymbol{x_h}(h = 1 \ldots H)$があったとき，拡張内積を(1)のように定義する．

$$<_{h=1}^{H} \boldsymbol{x_h} >= \sum_{i=1}^{I} \prod_{h=1}^{H} (\boldsymbol{x_h})_i \qquad (1)$$

ただし添字$i$は$i$番目の要素を示す．この拡張内積は，$H = 2$のときに内積となる．

### 3.2 定義

$K$次のPolynomial Semantic Indexing(PSI; 多項意味索引)の近似は次のような関数として定義される(元の定義は[Bai 2009]に詳しい)．個数$M(\geqq 2)$の入力変数ベクトル$(\boldsymbol{x_1}, \ldots, \boldsymbol{x_M})$があるとき，入力変数ベクトルから考慮したい2個以上最大$K$個の重複組み合わせについて考え，その重複組み合わせの集合を$V$とし，組み合わせの個数$|V|$を$G$とする．$V$のうち，$g(\leqq G)$番目の組み合わせに含まれる入力変数ベクトル$(\boldsymbol{x_{g1}}, \ldots, \boldsymbol{x_{gk_g}}, \ldots, \boldsymbol{x_{gK_g}})$(ただし$K_g$は変数ベクトルの個数)をそれぞれ行列$(\boldsymbol{X_{g1}}, \ldots, \boldsymbol{X_{gk_g}}, \ldots, \boldsymbol{X_{gK_g}})$によって$N$次元潜在的概

念空間上へと写像し，$g$番目の組み合わせについて拡張内積をとる．$G$個全てについて同様に拡張内積を計算し，その和が PSI の関数の値である．これを式で表すと(2)となる．

$$f^{[K]}(\boldsymbol{x_1},\dots,\boldsymbol{x_M}) = \sum_{g=1}^{G} < \prod_{k_g=1}^{K_g} \boldsymbol{X_{gk_g}} \boldsymbol{x_{gk_g}} > \quad (2)$$

ここで，重複組み合わせの最大個数$G_{max}$: $\sum_{k=2}^{K} {}_M H_k$となる．また$\boldsymbol{X_{gk_g}}$はベクトル$\boldsymbol{x_{gk_g}}$の次元数を$Dim.$とすると，$\boldsymbol{X_{gk_g}} \in \boldsymbol{R}^{N\times Dim.}$となり，写像:$\boldsymbol{R}^{Dim.} \rightarrow \boldsymbol{R}^{N}$となる．

### 3.3 例

ここでは，$K=3$の場合について具体的に述べる．なお，入力変数を$(\boldsymbol{x},\boldsymbol{y})$とし$(M=2)$，$\boldsymbol{x}$の要素同士の組み合わせと$\boldsymbol{y}$の要素の関係性を考慮して，$V=\{(\boldsymbol{x},\boldsymbol{x},\boldsymbol{y}),(\boldsymbol{x},\boldsymbol{y})\}$とおく$(G=2)$．この$V$について関数 PSI を式で表すと(3)となる．

$$f^{[3]}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{N} (\boldsymbol{A}\boldsymbol{x})_i (\boldsymbol{B}\boldsymbol{x})_i (\boldsymbol{C}\boldsymbol{y})_i + \sum_{i=1}^{N} (\boldsymbol{D}\boldsymbol{x})_i (\boldsymbol{E}\boldsymbol{y})_i \quad (3)$$

これは，(2)において$M=2$，$G=2$，$K_1=3$，$\boldsymbol{X_{11}}=\boldsymbol{A}$，$\boldsymbol{x_{11}}=\boldsymbol{x}$，$\boldsymbol{X_{12}}=\boldsymbol{B}$，$\boldsymbol{x_{12}}=\boldsymbol{x}$，$\boldsymbol{X_{13}}=\boldsymbol{C}$，$\boldsymbol{x_{13}}=\boldsymbol{y}$，$K_2=2$，$\boldsymbol{X_{21}}=\boldsymbol{D}$，$\boldsymbol{x_{21}}=\boldsymbol{x}$，$\boldsymbol{X_{22}}=\boldsymbol{E}$，$\boldsymbol{x_{22}}=\boldsymbol{y}$と対応する．

文章分類において，$\boldsymbol{x}$を文章ベクトル，$\boldsymbol{y}$をクラス識別のためのベクトルとすると，(3)の第一項はクラスにおける単語の共起情報を，第二項は生起情報を表すこととなる．

### 3.4 訓練

このモデルの訓練は，損失関数を margin rank loss[Herbrich 2000]によって定め，これを最小化することによって行う．入力変数を$(\boldsymbol{x},\boldsymbol{y})$としたとき，ある$\boldsymbol{x}$について，$\boldsymbol{y}^-$より$\boldsymbol{y}^+$の方が高く評価されるべきであるとする．このとき，損失関数$E(\theta)$は(4)のようになる．

$$E(\theta) = \sum_S \max\left(0, 1 - f^{[K]}(\boldsymbol{x},\boldsymbol{y}^+;\theta) + f^{[K]}(\boldsymbol{x},\boldsymbol{y}^-;\theta)\right) \quad (4)$$

ここで$\theta$はパラメータ(潜在的概念空間上へ写像する行列)，$S$はデータ集合である．この関数の最小化は確率的勾配法を用いてデータのペア$s=(\boldsymbol{x},\boldsymbol{y}^+,\boldsymbol{y}^-)\in S$について(5)のように逐次的に行う．

$$if\left(1 - f^{[K]}(\boldsymbol{x},\boldsymbol{y}^+;\theta^{(t)}) + f^{[K]}(\boldsymbol{x},\boldsymbol{y}^-;\theta^{(t)}) > 0\right)$$

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \lambda^{(t)} \frac{\partial}{\partial\theta} E_{s\in S}(\theta^{(t)}) \quad (5)$$

ここで$t$はステップ数，$0<\lambda^{(t)}\leqq 1$は学習率パラメータである．

## 4. probabilistic PSI(pPSI)

### 4.1 定義

(3)について，PSI を確率の枠組みで構成するため，その確率的表現として probabilistic PSI(pPSI; 確率的多項意味索引)を，指数関数，及び事後確率を用いて(6)のように定義する．

$$\Pr^{[K]}(\boldsymbol{y}|\boldsymbol{x};\theta) = \frac{\exp\left(f^{[K]}(\boldsymbol{x},\boldsymbol{y};\theta)\right)}{\sum_{y\in Y}\exp\left(f^{[K]}(\boldsymbol{x},\boldsymbol{y};\theta)\right)} \quad (6)$$

ここで$Y$は有限集合であり，$\theta$はパラメータ(潜在的概念空間上へ写像する行列)である．これは$\boldsymbol{x}$が既知のときの$\boldsymbol{y}$の事後確率である．

### 4.2 訓練

このモデルの訓練は最尤推定法を用いて行う．ある$\boldsymbol{x}$について，$\boldsymbol{y}$が$\boldsymbol{y}'\in Y'(=Y-\{\boldsymbol{y}\})$より高く評価されるべきであるとき，更新は(7)によって逐次的に行う．

$$if\left(Pr^{[K]}(\boldsymbol{y}|\boldsymbol{x};\theta) < e * \max_{\boldsymbol{y}'\in Y'} Pr^{[K]}(\boldsymbol{y}'|\boldsymbol{x};\theta)\right)$$

$$\theta^{(t+1)} \leftarrow \theta^{(t)} + \lambda^{(t)} \frac{\partial}{\partial\theta} l(\theta^{(t)}) \quad (7)$$

更新条件$if(\cdot)$は，(6)を$f^{[k]}(\boldsymbol{x},\boldsymbol{y})$について解いたものを，$\boldsymbol{y}^+\rightarrow\boldsymbol{y}$，$\boldsymbol{y}^-\rightarrow\boldsymbol{y}'$と置換した(5)に代入したものである．また(7)中の対数尤度の微分値$\frac{\partial}{\partial\theta} l(\theta)$は(8)のようになる．

$$\frac{\partial}{\partial\theta} l(\theta) = \frac{\partial}{\partial\theta} f^{[K]}(\boldsymbol{x},\boldsymbol{y};\theta) - \sum_{\boldsymbol{y}\in Y} Pr\,(\boldsymbol{y}|\boldsymbol{x};\theta) \frac{\partial}{\partial\theta} f^{[K]}(\boldsymbol{x},\boldsymbol{y};\theta) \quad (8)$$

ここで$t$はステップ数，$0<\lambda^{(t)}\leqq 1$は学習率パラメータである．

ただし，このモデルは訓練中に入力ベクトルのスケールに依存してパラメータがオーバーフローを起こすことがある．この問題を解決するために，訓練で入力変数ベクトルのスケールを動的に変更している．

### 4.3 確率モデルであるメリット

PSI を確率モデルとしたことで，次の利点が考えられる．

- 出力が[0,1]と正規化され安定する．
- 異なる入力変数$\boldsymbol{x}$についてどちらがより$\boldsymbol{y}$とのスコアが高いか，他の$\boldsymbol{y}$を考慮した上で比較することができる．
- パラメータの更新式に確率が存在することで，更新が現在のパラメータの分布に則ったものとなる．これは訓練中のパラメータの微調整や外れ値による過学習の対策に有効であると考えられる．またこの特性は半教師あり学習のときにより強く働く．これにより，更新時 PSI と pPSI を併用することによって計算時間が短縮される(これを予備実験に示す)．
- 半教師あり学習を行う際に，求めた数値をそのまま利用できるため効率が良い．
- 最尤推定の他，MAP 推定などデータ数に応じた確率に基づく推定法を用いることができる．また他の確率モデルとの統合を図ることも可能である．

### 4.4 CRFs との関連

Conditional Random Fields(CRFs; 条件付き確率場)は，入力変数$\boldsymbol{x}$について素性$\boldsymbol{y}$の成立する箇所数をベクトル$\boldsymbol{\Phi}(\boldsymbol{x},\boldsymbol{y})$で表現したものと，各素性の重要度を示す重み$\boldsymbol{w}$の内積を条件付き確率の形で表現したものである．これは(9)のように表される．

$$f(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp<\boldsymbol{w},\Phi(\boldsymbol{x},\boldsymbol{y})>}{\sum_{y\in Y}\exp<\boldsymbol{w},\Phi(\boldsymbol{x},\boldsymbol{y})>} \quad (9)$$

CRFs は条件付き確率を用いて表現されている点，内積(及び拡張内積)を用いるという点で pPSI と類似の表現であるといえる．CRFs はテキスト情報処理に特化したモデルであるが，pPSI はどのような情報に対しても用いることができる．

## 5. pPSI のクラス分類への利用

### 5.1 目的変数パラメータ

pPSI をクラス分類に適用するに先立ち，(10)のように目的変数パラメータ$\boldsymbol{t}^{(u)} \in R^U$を定める．ここで$t_i^{(u)}$は$\boldsymbol{t}^{(u)}$の$i$次元目の要素を示す．

$$t_i^{(u)} = \begin{cases} \text{if } i = u & 1 \\ \text{otherwise} & 0 \end{cases} \ (i = 1 \dots U) \qquad (10)$$

ただし，$U$はクラス数，$u$はクラス番号で，$u$は分類するクラスに対して一意に与えられるものとする．また全ての$\boldsymbol{t}^{(u)}$の集合を$T$とおく．

### 5.2 クラス分類への適用

あるクラス$C_j$において，そのクラスに属する$i$番目のデータベクトルを$\boldsymbol{d}_i^{(\mathrm{j})}$，対応する目的変数パラメータを$\boldsymbol{t}^{(j)}$とする．ここですべてのクラスについて$(\boldsymbol{d}_i^{(u)}, \boldsymbol{t}^{(u)})(u = 1 \dots U)$のペアを考える．この二変数ベクトルのペアについて，データ$\boldsymbol{d}_i^{(u)}$が既知のときに正しい組み合わせの目的変数パラメータ$\boldsymbol{t}^{(u)}$の事後確率が最も高くなるとして pPSI を用いてクラス分類を行う．このとき，あるデータ$\boldsymbol{d}$のクラス$C_u$に属する確率$p(\boldsymbol{d}, C_u)$は，(3)において$\boldsymbol{x} \to \boldsymbol{d}$，$\boldsymbol{y} \to \boldsymbol{t}^{(u)}$とすることで(11)と表される．

$$p(\boldsymbol{d}, C_u) = \Pr^{[3]}\left(\boldsymbol{t}^{(u)} \middle| \boldsymbol{d}\right) = \frac{\exp\left(f^{[3]}(\boldsymbol{d}, \boldsymbol{t}^{(u)}; \theta)\right)}{\sum_{t \in T} \exp\left(f^{[3]}(\boldsymbol{d}, \boldsymbol{t}; \theta)\right)}$$

$$f^{[3]}\left(\boldsymbol{d}, \boldsymbol{t}^{(u)}\right) = \sum_{i=1}^{N} (\boldsymbol{A}\boldsymbol{d})_i (\boldsymbol{B}\boldsymbol{d})_i \boldsymbol{C}_{ui} + \sum_{i=1}^{N} (\boldsymbol{D}\boldsymbol{d})_i \boldsymbol{E}_{ui} \qquad (11)$$

ただし，行列に対する添字 $ui$ は$u$行$i$列の要素を示すものとする．

### 5.3 クラスの推定

あるデータベクトル$\boldsymbol{d}$について，その属するクラスのクラス番号を$u$とすると，その推定値$\hat{u}$は(12)のようになる．

$$\hat{u} = argmax_u \, Pr^{[3]}(\boldsymbol{t}^{(u)} | \boldsymbol{d}) = argmax_u \, f^{[3]}(\boldsymbol{d}, \boldsymbol{t}^{(u)}; \theta) \ (12)$$

また，推定されたクラスは$C_{\hat{u}}$である．

### 5.4 分類器としての pPSI のメリット

pPSI を分類器として利用することは，他の分類器に対して次のような利点が考えられる．

- 入力変数の分布を仮定しない．したがってデータ及び目的変数パラメータがそれぞれどのような分布を示すものであっても訓練によってそれぞれに適合することとなる．これにより入力変数の分布を考慮する必要がない．
- 予め入力変数全体の素性を知る必要がない．これは前述の分布に加え，平均，分散，共分散などが未知であったとしても，モデルの形成に影響しないということである．したがって逐次更新を行うことが可能であり，更新時にすべての学習データを同時にメモリ上に配置する必要がない．
- (11)に示されるように，あるデータに対して式中のデータに関わる積$\boldsymbol{A}\boldsymbol{x}$，$\boldsymbol{B}\boldsymbol{x}$，$\boldsymbol{D}\boldsymbol{x}$を一度計算しておくと，クラスの推定についてはただ(拡張)内積と和の計算のみを行えばよい．これは訓練時にも推定時にも寄与し，計算コストが軽減されることとなる．特にデータの次元数が大きいとき，また対象とするクラスが多いときに，この特徴が強く活かされる．
- 高次元のデータであっても，近似されたパラメータを用いるため，一更新あたりに必要なメモリ，計算時間ともに次元数に比例して増加するのみでそれ以上を必要としない．したがって，文書分類のようなタスクを次元圧縮などの手法を用いることなく行うことができる．

## 6. 実験

### 6.1 概要

pPSI の性能評価のため，コーパスを用いて pPSI と m-SVM†(Linear Kernel)でクラス分類を試行する．また訓練に要した時間を併記する．使用コーパスは以下の通りである．

コーパス　：毎日新聞記事コーパス'94~'07
ジャンル　：8 ジャンル
(国際，経済，家庭，文化，読書，芸能，スポーツ，社会)
訓練記事数：{1000,2000,4000}記事/ジャンル
テスト記事数：1000 記事/ジャンル
単語数　：{10000,20000,40000}単語

### 6.2 条件

訓練は，次の手順で行う．(i) 訓練データを 3 分割し，初期値を正規乱数によって決定したパラメータについて，過去$h = 100000$回の更新判定において更新しなかった割合(以下，非更新割合)γが 50%になるまで訓練を行う．(ii) 3 パラメータのうち学習に使用しなかったデータでの尤度が最大であったパラメータを採用する．(iii) 採用したパラメータについて，全ての訓練データで以下の条件に基づいて訓練を行う．

更新式は，pPSI は(7)を用い，$N = 10$である．更新の終了は，以下のいずれかの条件を満たすこととする．

- γ = 0.99となる場合：目標への達成とする
- 開始からの更新判定回数$cnt$が$h$以上で，$cnt$が最良のγからの更新判定回数の 2 倍以上の場合(そのとき最良のγでのパラメータを採用する)：最良のパラメータが更新される見込みがないとする

また，m-SVM はオプション c(trade-off between training error and margin)を除いて全てデフォルトを用いた．

### 6.3 結果

pPSI によるクラス分類のデータ数/ジャンル，単語数別訓練時間と終了時点での非更新割合を表 1 に示す．

**表 1　pPSI による訓練時間と終了時の非更新割合**

| データ数<br>単語数 | 1000 | 2000 | 4000 |
|---|---|---|---|
| 10000 | 52.78min.<br>99.00% | 101.21min.<br>99.00% | 339.78min.<br>69.74% |
| 20000 | 112.08min.<br>99.00% | 251.14min.<br>99.00% | 506.13min.<br>69.71% |
| 40000 | 249.26min.<br>99.00% | 594.51min.<br>99.00% | 3581.92min.<br>96.78% |

† m-SVM / SVM-Light Support Vector Machine
http://svmlight.joachims.org/svm_multiclass.html

また，クラスの推定精度を表 2 に示す．なおモデルの特性上，初期値及び訓練に一意性がないため，精度に確率的要素を含むことを特記する．またクラスの推定に要する時間は 1 データあたり 3.37e-4～4.10e-4sec.であった．

**表 2　pPSI によるクラス推定精度**

| データ数<br>単語数 | 1000 | 2000 | 4000 |
|---|---|---|---|
| 10000 | 73.64% | 77.16% | 77.90%. |
| 20000 | 74.40%. | 75.75%. | 79.56% |
| 40000 | 74.14% | 74.32% | 79.83% |

比較実験として，SVM(c=1000000)によるクラス分類のデータ数/ジャンル， 単語数別精度とその訓練時間を表 3 に示す．

**表 3　SVM によるクラス推定精度と訓練時間**

| データ数<br>単語数 | 1000 | 2000 | 4000 |
|---|---|---|---|
| 10000 | 79.34%<br>31.86sec. | 81.42%<br>38.02sec. | 81.92%<br>53.49sec. |
| 20000 | 79.21%<br>60.83sec. | 82.12%<br>58.82sec. | 82.46%<br>81.41sec. |
| 40000 | 79.59%<br>74.87sec. | 81.68%<br>81.17sec. | 82.04%<br>106.65sec. |

## 7. 考察

pPSI は，局所最適性があるため一概には言えないが，単語数の増加に伴って精度が向上するとは言えないものの，データ数の増加に伴って精度が向上する傾向はある．訓練時間は推定する数値が多いためいずれも長い．ただし分類の際に要する時間は短く，利用の際にネックとなる可能性は低い．これは文章ベクトルが疎ベクトルであるためである．

一方，SVM ではデータ数及び単語数の増加に伴って精度が向上する傾向にある．計算時間についてはデータ数よりも単語数に依存することが多く，サポートベクターを選択するという SVM の特徴が表れている．

両者を比較すると，いずれのデータ数･単語数の場合にも pPSI の精度を SVM の精度が上回っている．また訓練時間についても SVM の方が遥かに早い．

## 8. 結論

PSI の確率的表現 pPSI とそれを用いたテキストのクラス分類精度を示した．SVM でのクラス分類精度には及ばないものの，言語処理において多用される生起情報・共起情報の近似確率モデルとして，クラスタリングや概念形成，キーワード抽出などに応用できるのではないかと考えられる．

## 付記　潜在概念空間次元数と分類精度

潜在概念次元数によって，クラス分類推定精度がどのように推移するかを表 4 に付記する．

なお，用いたデータは前述の毎日新聞記事コーパス 1000 記事/カテゴリ，10000 単語であり，$N = 10$の結果は前述の実験から流用した．

**表 4　潜在概念次元数とクラス推定精度及び訓練時間**

| N | 5 | 10 | 20 | 50 | 100 |
|---|---|---|---|---|---|
| | 70.34% | 73.64% | 72.37% | 73.26% | 74.40% |
| | 26.36m. | 52.78m. | 106.66m. | 247.33m. | 442.43m. |

表 4 に示すように，精度に確率的要素を含むものの，潜在概念次元数 N が大きくなるにつれ精度が改善する傾向にある．しかしながらその傾向は緩やかであるため，SVM との比較において$N = 10$で十分であると判断した．

## 参考文献


[Bai 2009] B. Bai, J. Westion, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, C. Cortes, and M. Mohri.: Polynomial Semantic Indexing, Neural Information Processing Systems (NIPS) Conferences 2009.

[Deerwester 1990] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman.: Indexing by latent semantic analysis, JASIS, 41 (6) pp. 129-136, 1990.

[Hofmann 1999] T. Hafmann.: Probabilistic latent semantic indexing, SIGIR, pp. 50-57, 1999.

[Herbrich 2000] R. Herbrich, T. Graepel, and K. Obermayer.: Advances in Large Margin Classifiers, chapter Large margin rank boundaries for ordinal regression, MIT Press, Cambridge, MA, 2000.

[箕浦 2011] 箕浦健太郎，田村哲嗣，速水悟.: PSI の確率的表現とクラス分類への応用，人工知能学会年次会, 2011.