論文
Technical Paper

ベイズ統計の手法を利用した決定リストのルール信頼度推定法
Estimating reliability of rules in decision lists using Bayesian learning

Summary

The decision list algorithm is one of the most successful algorithms for classification problems in natural language processing. The most important part of the decision list algorithm is the calculation of reliability for each rule, hence the estimation of probability for each contextual evidence. However, the majority of research efforts using decision lists do not think much of the estimation method. We propose an estimation method based on Bayesian learning which gives well-founded smoothing and better use of prior information on each type of contextual evidences. Experimental results obtained using Senseval-1 data set and Japanese pseudowords show that our method makes probability estimation more precise, leading to improvement of classification performance of the decision list algorithm.

概要

統計的クラス分類器としての決定リストは，近年自然言語処理における様々な分野でその有効性を示している．決定リストを構成する上で最も重要な問題の一つは，ルールの信頼度の算出法である．決定リストを用いた多くの研究では，最尤推定法と簡単なスムージングにより信頼度を算出しているが，理論的な根拠に欠け推定精度も高くないという問題がある．そこで本論文では，ベイズ学習法を利用してルールの信頼度を算出する手法を示す．さらに，証拠の種類ごとに異なる事前分布を利用することで，より正確な信頼度の推定が可能になり，決定リストの性能が向上することを示す．本手法の有効性を確かめるために，語義曖昧性解消の問題に決定リストを適用して実験を行なった．英語に関しては Senseval-1 のデータを用い，日本語に関しては疑似単語を用いた．その結果，ベイズ学習による信頼度推定手法が，ルールの確率値の推定精度を高め，決定リストの分類性能を向上させることを確認した．

1　はじめに

決定リストとは統計的なクラス分類器である．自然言語処理の多くは，クラス分類問題として捉えることが可能であり，近年，様々な自然言語処理において，決定リストによる手法の有効性が示されている (Yarowsky 1995; 新納 2000; 宇津呂他 1999; 白木, 梅村, 原田 2000)．特に，語義曖昧性解消問題に対しては，語義曖昧性解消システムの性能を競う競技会である Senseval-1 において，決定リストを階層的に拡張した手法が最も良い成績をあげている (Yarowsky 2000)．

クラス分類器としては，分類精度の点だけでいえば，最近ではサポートベクタマシン (Vapnik 1995) やアダブースト (Freund and Schapire 1999) といった手法が，その性能の高さから注目を集めている (永田 平 2001) ．しかし，それらの手法は，学習結果が人間にとってブラックボックスなのに対して，決定リストによる手法では，作成された分類器が if-then 形式のルールの並びであるために，人間が容易に理解可能であるというメリットがある．学習した決定リストに人間の手を入れることで，性能を向上させることができるとの報告もある (Li and Yamanishi 1999)．

決定リストを作成する上で最も重要な問題は，ルールの信頼度の算出法である．信頼度を計算するためには，限られた事例から，ルールに関する条件付き確率を計算する必要がある．事例の数が多ければ，確率値を最尤推定法によって頻度の比として推定することにほとんど問題はない．しかし，事例の数が少ない場合，最尤推定法による推定値の誤差は非常に大きくなってしまう．このような問題に対し，決定リストを用いた多くの研究では，事例の数が少ないルールを間引いたり，簡単なスムージングを行なうことによって対処している．しかし，ルールを間引く手法では重要なルールを取りこぼしてしまう危険があり，計算式に適当な数値を足してスムージングを行なう手法では加算する値の設定の理論的な指針がないという問題がある．

他方，決定リスト手法の改良として，特徴の種類ごとに異なった信頼度の重み付けを与える手法が提案され，日本語の同音異義語解消の実験によってその有効性が示されている (新納 1998)．このことは，特徴の種類によって，ルールの信頼度に最尤推定法では考慮することのできない違いが存在することを示唆している．

そこで本論文では，ルールの確率値の推定にベイズ統計の手法を利用する．ベイズ統計では，確率変数に関する推定を行なう際に，学習者の持っている事前知識を活用することができる．そのため，適切な事前知識を利用することができれば，最尤推定よりも正確な推定を行なうことができる．また，上記の，証拠の種類による信頼度の違いも，事前分布の違いとして自然に導入することができる．

本論文では，語義曖昧性解消の問題を例にとり，ベイズ学習による信頼度の算出が，決定リストの性能を向上させることを示す．本論文の構成は以下の通りである．2 章で決定リストによるクラス分類の手法を説明する．3 章で，ベイズ学習による確率値の算出法を示す．4 章で，他のルールの確率値を利用して事前分布を構成する方法を示す．5 章で，決定リストを語義曖

表 1　決定リストの例
表 1　An example of a decision list

信頼度	証拠	語義
8.10	*plant* **life**	A
7.58	**manufacturing** *plant*	B
7.39	**life** (within ±2–10words)	A
7.20	**manufacturing** (within ±2–10words)	B
6.27	**animal** (within ±2–10words)	A
4.70	**equipment** (within ±2–10words)	B
4.39	**employee** (within ±2–10words)	B
:	:	:

昧性解消問題に適用した実験結果を示す．6 章で，まとめを行なう．

2　決定リスト

決定リストとは，クラス分類のためのルールを，その信頼度の高い順に並べたものである．それぞれのルールは,「もし（証拠 E_i）ならば，クラスは C_j である」という形式をしている．証拠というのは，判定の手がかりとなる事例の特徴である．

例として，英語の多義語 *plant*（A: 植物, B: 工場）に関して Yarowsky が行なった実験での決定リストを表 1に示す (Yarowsky 1994)．最上位のルールは,「右隣に life という単語があったら，語義は A」という意味，4 番目のルールは,「距離 2〜10 単語以内に manufacturing という単語があったら，語義は B」という意味である．

実際にクラスの分類を行なう際には，その事例に対して適用可能なルールのうち，最も上位のルールを用いて分類が行なわれる．例えば入力文が,

... divide life into *plant* and animal kingdom ...

であるとすると，適用可能なルールのうち最上位なのは 3 番目のルールであるから，*plant* の語義は A だと判定されることになる．

このように，決定リストによる手法では，他の多くの機械学習手法と異なり，特徴を単独で利用する．単独でしか利用しないのは一見不利なようであるが，語義曖昧性解消などの，文脈の語彙的な特徴を利用する問題に関しては，単独の証拠が分類の決定的な証拠になることが多いため，決定リストによる手法が有効であるといわれている．

本論文で提案する決定リストのルール信頼度の推定手法は，特定の自然言語処理に特化したものではないが，本論文では，決定リストの適用例として，上記のような語義曖昧性解消の問題を取り上げる．

ここで，本論文で用いる文脈上の特徴を以下に示す．

- Window
  ターゲットから，距離 10 単語以内に出現する単語
- Adjacent
  ターゲットの左隣に出現する単語
  ターゲットの右隣に出現する単語
- Pair
  ターゲットの左隣にある単語対
  ターゲットを挟む単語対
  ターゲットの右隣にある単語対

すなわち，文脈情報としての詳細さが異なる三つのタイプの特徴を利用する．文脈情報として最も詳細なのは Pair であり，最も粗いのが Window である．

2.1　ルールの信頼度

決定リストは，事例とその正解ラベルを含む訓練コーパスから作成される．決定リストの作成において最も重要な問題は，それぞれのルールの信頼度の計算法である．文献 (Yarowsky 1994) では，次の式に従って信頼度を計算している．

$$（信頼度）= \log\Big(\frac{P(C_A|E_i)}{P(C_B|E_i)}\Big) \tag{1}$$

すなわち，証拠 E_i のもとでクラス（語義）が A である確率と，同じ証拠 E_i のもとでクラスが B である確率との比の対数をとったものである．

従来の決定リストを用いた自然言語処理の研究では，ルールの信頼度の算出法として，式（1），あるいは，式（1）をクラスが 3 つ以上の場合でも適用できるように変形した次の式,

$$（信頼度）= \log\Big(\frac{P(C_A|E_i)}{1-P(C_A|E_i)}\Big) \tag{2}$$

が用いられることが多い (Yarowsky 1994, 2000; 新納 2000)．また，対数をとらずに,

$$（信頼度）= P(C_A|E_i) \tag{3}$$

とする場合もある (白木他 2000).

ここで，式（3）と式（2）を見比べてみると，式（2）は，式（3）に関して単調増加であり，決定リストでは信頼度の大小関係しか問題にならないのだから，後述するスムージングの問題を考慮しなければ，式（2）を用いた場合と，式（3）を用いた場合では，結果的に作成される決定リストは等価になる．一般にクラス分類器の目標は，分類の正解率を最大にすることであるから，ルールの信頼度としては，そのルールが正解する確率である式（3）を用いるのが自然

である．また，クラス分類器が，自然言語処理システムの一部を構成している場合，分類の信頼度は確率として出力された方が扱いやすいことが多い．そこで本論文では，ルールの信頼度として式（3）を用いることにする．

式（3）の値は，訓練事例が多ければ，ベルヌーイ試行における最尤推定により，次のように計算することができる．

$$(\text{信頼度}) = P(C_A|E_i) = \frac{f(C_A, E_i)}{f(E_i)} \tag{4}$$

ただし，$f(C_A, E_i)$ は，クラス A に属するターゲットと証拠 E_i が同時に出現した回数．$f(E_i)$ は，証拠 E_i の出現回数である．ところが，通常は出現回数が少ない証拠も多い．例えば，

$$f(C_A, E_i) = 1, f(E_i) = 1 \tag{5}$$

の場合，信頼度は $1/1 = 1$ と計算されるが，たった一つの事例しかないのに，その信頼度は 100%，すなわち最も信頼度の高いルールだとみなされてしまう．このように，出現回数の少ない事例において，そのままでは統計的に信頼性のある確率値が算出できないことをスパースネスの問題という．

そこで本論文では，ベイズ学習の手法を用いてこの問題の解決を試みる．

3　ベイズ学習によるルール確率値の推定

いま，求めたいルールの確率値を θ とする．最尤推定の枠組では，確率モデルの尤度が最大となるように θ を決定するが，ベイズ学習の枠組では，θ を確率変数と考えて，その確率分布を求める問題と考える．本論文では，得られた確率分布を決定リストのルールの信頼度として利用したいのだから，その確率分布から θ の期待値を計算して利用すればよい．

訓練コーパスにおいて，確率を求めたいルールに関する事例が n 個あり，そのうちの k 個において，そのルールが正しいというデータがあるとする．このデータを y とすると，データ y を観測した後の θ の事後密度は，

$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{p(y)} \tag{6}$$

$$= \frac{p(\theta)p(y|\theta)}{\int_0^1 p(\theta)p(y|\theta)d\theta} \tag{7}$$

で与えられる．ここで，事象 y はベルヌーイ試行と考えられるから，その確率は二項分布により次のように与えられる．

$$p(y|\theta) = {}_nC_k\theta^k(1-\theta)^{n-k} \tag{8}$$

これを式（7）に代入して，

$$p(\theta|y) = \frac{p(\theta)_n C_k \theta^k (1-\theta)^{n-k}}{\int_0^1 p(\theta)_n C_k \theta^k (1-\theta)^{n-k} d\theta} \tag{9}$$

$$= \frac{p(\theta)\theta^k (1-\theta)^{n-k}}{\int_0^1 p(\theta)\theta^k (1-\theta)^{n-k} d\theta} \tag{10}$$

を得る．ここで，事前分布 $p(\theta)$ をどのように設定するのか，という問題が浮上する．事前分布は，θ について学習者が持っている事前知識を表す．ベイズ学習における事前分布の設定方法に関しては，大きく分けて 2 つのアプローチがある．一つは，できるだけ公平で無知の状態を表すように事前分布を設定する方法である．そのような事前分布としては，一様分布や Jeffreys の無情報事前分布などが提案されている (繁桝 1985)．もう一つは，学習者が事前に持っている知識を積極的に表現するような事前分布を設定する方法である．

3.1　一様分布

まず，無知の状態を表す事前分布として，一様分布を用いた場合について説明する．いま，あるルールの確率に関して，事前知識が全くないものと考えると，すべての確率値の事前確率について同じ値とするのが自然である．θ は [0,1] を定義域とする連続の確率変数であり，$p(\theta)$ は密度関数であるから，

$$p(\theta) = 1 \tag{11}$$

とすればよい．そうすると，事後分布は次のようになる．

$$p(\theta|y) = \frac{\theta^k (1-\theta)^{n-k}}{\int_0^1 \theta^k (1-\theta)^{n-k} d\theta} \tag{12}$$

$$= \frac{\theta^{(k+1)-1}(1-\theta)^{(n+2)-(k+1)-1}}{\int_0^1 \theta^{(k+1)-1}(1-\theta)^{(n+2)-(k+1)-1} d\theta} \tag{13}$$

この確率分布は，ベータ分布と呼ばれ，期待値は次式で与えられる (鈴木 国友 1989)．

$$E[\theta] = \frac{k+1}{n+2} \tag{14}$$

いま，k と n は，それぞれ，$f(C_A, E_i)$ と $f(E_i)$ に対応しているのだから，

$$\text{（信頼度）} = P(C_A|E_i) = \frac{f(C_A, E_i)+1}{f(E_i)+2} \tag{15}$$

となる．結論は非常にシンプルである．すなわち，頻度 $f(C_A, E_i)$ と $f(E_i)$ をそのまま用いる代わりに，$f(C_A, E_i)+1$ と $f(E_i)+2$ を用いればよい，ということである．

4　事前分布の利用による確率値の正確な推定

前章では，ベイズ学習において事前情報が全くないものとし，事前分布を一様分布として事後分布の導出を行なった．しかし，5章で述べる実験結果から明らかなように，実際の正解率と，ベイズ学習による確率から計算された期待正解率との間には開きがある．これは，推定された確率が真の確率からずれていることを示している．この原因には，以下の 3 つが考えられる．

- トレーニングデータ vs. テストデータ
  もし，学習のためのトレーニングデータと，テストデータの性質が異なっている場合，実際の正解率は低下する．これは，コーパスベースの手法の本質的な問題である．
- Global vs. history-conditional
  決定リストにおいて，あるルールが適用されるということは，そのルールより上位のルールが，その文脈に適用できなかったことを示している．したがって，確率値はその条件を反映したものでなければならない．ところが，式（15）では，そのような条件を考慮せず，単に事例全体の中での確率しか考慮していない．そのような条件を考慮した確率値を算出するためには，決定木を構成するように，決定リストにルールを追加するたびに，それに適合する事例を削除していく，というようなことをする必要がある．しかし，そのようにすると，下位にいくにしたがって事例の数が少なくなっていくため，確率値の推定誤差が大きくなってしまうことや，計算量が事例数の 2 乗に比例するようになってしまうという問題がある．文献 (Yarowsky 2000) では，ルールの確率値を，上記の 2 つの確率，すなわち事例全体の中での確率と，上位のルールにマッチしなかったという条件付き確率との重み付き平均をとることによって計算している．
- 事前分布
  前章では，事前分布を一様分布と仮定した．しかし，例えば，分類すべきクラスの数が 5 個あり，学習者が全く情報を持たないとすれば，特定のクラスを出力するルールが正解する確率の事前分布としては 0.2 を期待値とするような分布であるべきであろう．しかし，一様分布の期待値は 0.5 である．この例からもわかるように，一様分布はどんな場合でも適切な事前分布というわけではない．

上記の三つの問題に対して，最初の二つの問題については本論文では扱わない．本章では，他のルールの確率値を利用して適切な事前分布を設定する手法を提案する．

事前分布とは，θ に関するデータがない段階で仮定される，θ がとる値の確率分布である．いま，θ は，あるルールの確率を表しているが，ここで θ を単独で考えるのではなく，θ は，同じ証拠タイプ内の他の多くのルールの確率値と同じような性質を持っていると考えることにする．つまり，あるルールの事前分布を，他のルールの確率値を利用して構成する[1]．

1 このような考え方は，経験ベイズと呼ばれることもある (Gelman, Carlin, Stern, and Rubin 1995)．また，ベイズ統計の枠組を用いてはいないが，単語の出現確率の代表的なディスカウンティング手法であるグッド・チューリング推定

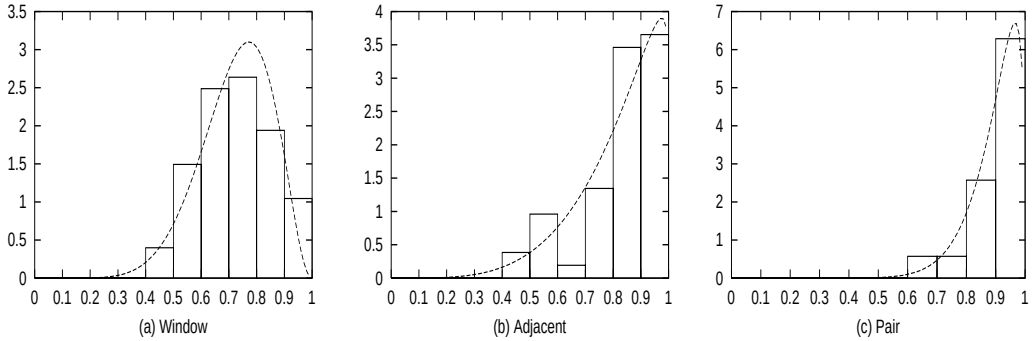

図 1　ルールの確率値の分布
図 1　Distribution of rule probabilities

まず，ルールの確率値の分布がどのような性格を持っているのかを見るために，実際のルールの確率値の分布の例を図 1に示す．これは，5章の実験で用いられた多義語 accident において，それぞれの証拠のタイプに属するルールの確率値の，正規化されたヒストグラムを示したものである（グラフ中の曲線については後述する）．ただし，各々のルールの確率値は，事前分布を一様分布としたベイズ学習により算出し，出現回数が 10 回未満のルールは除いている．

ここで，ルールの確率値の統計的性質は，そのルールの証拠の事例数に依存しないと仮定すれば，図 1 に示したような，事例の数が多いルールの実際の確率値の分布を利用して，事前分布を構成することができる．事前分布の確率分布としては，ベータ分布を採用する．ベータ分布は，ベルヌーイ試行において自然共役事前分布と呼ばれる確率分布であり，事後分布の導出が解析的に可能であることが知られている (繁桝 1985)．ベータ分布は，2 つのパラメータによって決定されるが，本論文では最も簡単なパラメータ推定法の一つであるモーメント法によってパラメータを決定する．モーメント法とは，母集団 j 次モーメント

$$E_{(a,b)}[\theta^j] = \int \theta^j p(\theta) d\theta \tag{16}$$

と，標本 j 次モーメント

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} (\frac{k_i + 1}{n_i + 2})^j \tag{17}$$

がそれぞれ等しいと置いた連立方程式を得くことでパラメータ a, b を計算する方法である．図 1のグラフ中の曲線は，ヒストグラムで示した確率値のデータから，モーメント法によって得たベータ分布を表している．

以下に事前分布をベータ分布とした場合の，事後分布の導出の過程を示す．まず，ベータ分布は次の式で与えられる．

法の考え方ともよく似ている (北 1999)

$$p(\theta) = \frac{1}{B(a,b)}\theta^{(a-1)}(1-\theta)^{(b-1)} \tag{18}$$

ただし，$B(a,b)$ はベータ関数

$$B(a,b) = \int_0^1 \theta^{(a-1)}(1-\theta)^{(b-1)}d\theta \tag{19}$$

である．

ベータ分布の 1 次モーメントは，

$$\frac{a}{a+b} \tag{20}$$

2 次モーメントは，

$$\frac{a+1}{a+b+1}\cdot\frac{a}{a+b} \tag{21}$$

で与えられるから，同じタイプの証拠に属し，出現頻度が閾値（本論文では 10 とした）以上のルールの確率値の，1 次モーメント，2 次モーメントをそれぞれ μ_1, μ_2 とすれば，ベータ分布の 2 つのパラメータは，

$$a = \frac{\mu_1(\mu_1-\mu_2)}{\mu_2-\mu_1^2} \tag{22}$$

$$b = \frac{(\mu_1-\mu_2)(1-\mu_1)}{\mu_2-\mu_1^2} \tag{23}$$

と指定すればよい．

この事前分布を式（10）に代入することにより，事後分布は次のようになる．

$$p(\theta|A) = \frac{1}{B(a+k,b+n-k)}\theta^{(a+k-1)}(1-\theta)^{(b+n-k-1)} \tag{24}$$

事後分布の期待値，すなわちルールの信頼度は次のように得られる．

$$E[\theta] = \frac{a+k}{a+b+n} \tag{25}$$

このように，信頼度は最終的に加算スムージングのような形式で得られることから，信頼度の計算自体は非常に簡単に行なうことができる．

5　実験

提案手法の有効性を確かめるため，決定リストを用いて，英語の語義曖昧性解消と，日本語の疑似単語判定問題に関して実験を行なった．語義曖昧性解消とは，多義語の語義を文脈から判定する問題で，自然言語処理における典型的なクラス分類問題である．また，疑似単語判定とは，複数の単語をシステムの側からは単一の単語にしか見えないようにしておき，どの単語であるのかを文脈から判定させるという問題である．この問題は，人工的な語義曖昧性解消問

題ということができる．

実験によって評価すべき点は二つある．一つはもちろん，クラス分類の正解率である．従来手法と比べて，正解率が向上するかどうかを評価する．もう一つは，出力する確率値の正確さである．つまり，ルールの確率値が，どの程度正確に推定できているかということである．それを評価するために,「期待正解率」というものを考える．これは，それぞれの分類に用いられたルールの確率値を平均したものである．もし，確率値の推定が理想的に行なわれたとすれば，実際の正解率と，期待正解率はほぼ等しくなるはずである．すなわち，実際の正解率と期待正解率のずれは，確率値の推定の「悪さ」を表すことになる．

比較対象とする従来手法は以下の二つである．

- 間引き
  出現回数が閾値未満の証拠のルールは使用しないようにする手法．ルールの確率値は式（3），すなわち最尤推定により算出する．確率値が等しい場合は，出現回数の多いルールを優先する．
- 対数尤度比
  式（2）を用いる手法．文献 (Yarowsky 1994)(新納 1998) などで用いられている．この場合，式（2）の分母が 0 になってしまう可能性があるため，頻度の比の式の分母と分子に小さな値 α を足す．このようにすることで，分母が 0 になってしまう問題を防げる．また，同じ確率であれば，頻度の高い証拠のルールが優先されるようになる．

5.1 Senseval-1 データセットによる実験

英語の語義曖昧性解消については，語義曖昧性解消の競技会である Senseval-1 のデータセットが公開されているので，それを利用して実験を行なった[2]．Senseval-1 データセットには，訓練データが利用可能な多義語が 36 個含まれている．表 2に，それぞれの多義語の語義数，訓練事例数，テスト事例数を示す．

本実験では，品詞タグ付けなどの前処理は行わず，生のテキストデータを利用して決定リストの学習と評価を行なった．

従来手法に関しては，最も良い場合と比較するため，間引きの閾値を変化させて，最も正解率が高くなる値を採用した．本データセットに関しては，最も良い閾値は 2 であった．また，対数尤度比でのスムージングのパラメータ α に関しても 0.1 きざみで変化させ，最も正解率が高くなる値を採用した．本データセットに関しては，最も良い α は 0.9 であった．

表 3に結果を示す．表中の数字は正解率を表している．正解率の右側にある括弧内の数字は，先に述べた「期待正解率」との差の絶対値を表している．この値が小さいほど，確率値の推定が正確であることを示している．

2 http://www.itri.brighton.ac.uk/events/senseval/

表 2　Senseval-1 データセット
表 2　Senseval-1 data set

多義語	品詞	語義数	訓練事例数	テスト事例数	多義語	品詞	語義数	訓練事例数	テスト事例数
accident	n	8	1234	267	giant	a	5	315	97
amaze	v	1	133	70	giant	n	7	342	118
band	p	32	1326	302	invade	v	6	45	207
behaviour	n	3	994	279	knee	n	22	417	251
bet	n	15	106	273	modest	a	9	374	270
bet	v	9	59	116	onion	n	4	26	214
bitter	p	17	144	372	promise	n	8	586	113
bother	v	8	282	209	promise	v	6	1160	224
brilliant	a	10	440	229	sack	n	11	97	82
bury	v	14	272	201	sack	v	4	185	178
calculate	v	5	218	218	sanction	p	10	96	431
consume	v	6	56	183	scrap	n	14	27	156
derive	v	6	255	217	scrap	v	3	30	186
excess	n	8	178	186	seize	v	11	287	259
float	n	12	61	75	shake	p	39	963	356
float	v	16	182	229	shirt	n	8	531	184
floating	a	5	39	47	slight	a	6	380	218
generous	a	6	307	227	wooden	a	4	361	196

（品詞が p とは品詞情報が判定システムに与えられないことを示す）

まず，正解率に関して見ると，間引きの正解率が最も低い．これは，間引きによって重要なルールを捨ててしまっていることが原因だと考えられる．対数尤度比による手法と，事前分布を一様分布としてベイズ学習による手法が，ほぼ同じ正解率である．ただし，ここで注意するべきなのは，対数尤度比による手法では，スムージングのパラメータに関して，正解率が最もよくなるようにチューニングがなされたうえでの結果だということである．事前分布を一様分布としたベイズ学習による手法は，そのようなチューニングを全く必要としないにもかかわらず，それとほぼ同じ正解率を達成している．また，期待正解率と実際の正解率とのずれに関しても，最尤推定（間引き）に比べてかなり小さく，ベイズ学習による推定の有効性を示している．

最も正解率が高いのは，他のルールの確率値を利用してベータ分布によって事前分布を構成する手法である[3]．これは，適切な事前分布によって，ルールの確率値の推定が正確になり，本

3 ただし，本手法で得られた正解率（72.7%）は，Senseval-1 参加システムでの最高正解率（78.9%）(Yarowsky 2000) よりも低い．本論文では正解率を追求することが目的ではないため，stemming や品詞タグ付けなどの前処理を行なっ

表 3 Senseval-1 データセットによる評価
表 3 Evaluation by Senseval-1 data set

単語	品詞	間引き		対数尤度比	ベイズ 一様分布		ベイズ ベータ分布	
accident	n	85.0%	(13.6)	85.0%	83.9%	(7.8)	89.9%	(3.2)
amaze	v	100.0%	(0.1)	100.0%	100.0%	(1.0)	100.0%	(0.2)
band	p	86.8%	(11.4)	84.4%	84.1%	(5.9)	87.4%	(2.6)
behaviour	n	95.3%	(4.6)	94.6%	94.6%	(2.7)	94.6%	(3.1)
bet	n	48.4%	(30.7)	44.3%	45.1%	(27.8)	50.5%	(9.5)
bet	v	66.4%	(26.2)	69.0%	70.7%	(10.1)	76.7%	(2.3)
bitter	p	44.9%	(31.9)	49.2%	49.2%	(22.2)	51.1%	(4.0)
bother	v	76.1%	(18.8)	78.5%	78.5%	(5.5)	78.0%	(5.1)
brilliant	a	48.5%	(33.6)	48.5%	47.6%	(24.9)	49.3%	(11.7)
bury	v	44.3%	(34.7)	49.8%	49.3%	(22.4)	49.3%	(10.0)
calculate	v	88.5%	(11.0)	86.7%	86.7%	(2.6)	86.7%	(2.8)
consume	v	43.7%	(37.1)	42.1%	42.6%	(29.1)	42.1%	(18.3)
derive	v	55.3%	(33.1)	53.9%	54.4%	(20.8)	64.1%	(3.5)
excess	n	79.6%	(13.8)	81.7%	81.7%	(8.9)	81.2%	(9.7)
float	n	50.7%	(38.4)	53.3%	53.3%	(21.8)	56.0%	(5.3)
float	v	41.5%	(38.0)	45.4%	45.0%	(27.7)	42.8%	(16.5)
floating	a	63.8%	(20.5)	59.6%	59.6%	(13.0)	61.7%	(7.7)
generous	a	44.1%	(36.9)	48.9%	49.3%	(23.5)	46.7%	(9.8)
giant	a	97.9%	(1.5)	96.9%	96.9%	(1.9)	96.9%	(0.9)
giant	n	74.6%	(20.1)	78.8%	79.7%	(5.9)	79.7%	(4.2)
invade	v	44.4%	(35.0)	46.4%	46.9%	(24.4)	50.2%	(15.1)
knee	n	71.3%	(18.8)	70.5%	70.9%	(8.3)	72.9%	(0.2)
modest	a	66.7%	(24.0)	67.0%	66.3%	(10.8)	66.3%	(2.5)
onion	n	84.6%	(15.1)	84.6%	84.6%	(6.5)	84.6%	(9.8)
promise	n	74.3%	(19.8)	74.3%	74.3%	(7.4)	74.3%	(5.4)
promise	v	87.5%	(12.0)	88.4%	88.4%	(5.4)	92.9%	(1.7)
sack	n	82.9%	(10.1)	85.4%	81.7%	(3.3)	85.4%	(2.6)
sack	v	97.8%	(2.3)	97.8%	97.8%	(0.6)	97.8%	(1.5)
sanction	p	72.6%	(21.9)	74.5%	74.5%	(7.6)	77.7%	(2.3)
scrap	n	42.3%	(47.6)	41.7%	41.7%	(38.7)	45.5%	(31.8)
scrap	v	87.6%	(11.7)	87.6%	87.6%	(1.8)	87.6%	(4.5)
seize	v	59.5%	(26.9)	60.6%	60.2%	(17.1)	64.5%	(4.3)
shake	p	60.1%	(25.0)	62.1%	61.2%	(19.7)	61.2%	(17.6)
shirt	n	82.1%	(13.0)	83.7%	83.7%	(3.7)	82.6%	(2.6)
slight	a	90.8%	(8.3)	94.0%	94.0%	(0.3)	94.0%	(0.8)
wooden	a	93.9%	(6.1)	93.9%	93.9%	(3.1)	93.9%	(3.9)
平均		70.4%	(20.9)	71.2%	71.1%	(12.3)	72.7%	(6.6)

（括弧内の数字は正解率と期待正解率との差）

当に信頼できるルールが上位に位置するようになったからだと考えられる．そのことを裏付けるように，実際の正解率と期待正解率のずれが，一様分布の場合と比較して半減している．つまり，確率値の推定がそれだけ正確になったということを示している．

5.2　日本語の疑似単語判定の実験

日本語の語義曖昧性解消に関しては，Senseval-1 のようなデータセットが公開されていないことから，疑似単語を用いて実験を行なった．疑似単語とは，複数の異なる単語を判定システムの側からは同一の単語にしか見えないようにし，文脈からどの単語であるのかを判定させる手法である．例えば,「銀行」という単語と,「土手」という単語を用いて疑似単語を作ったとすると，判定システムからは，入力文は例えば，

... お金をおろしに＊＊へ行く途中 ...

のように見える．＊＊の部分が疑似単語である．そして，文脈から,「銀行」であるのか「土手」であるのかを判定させるというわけである．これは，文脈から多義語の語義の判定を行う多義性解消の問題とかなり似た問題になる．

実験に用いる疑似単語に関しては，ベースラインとしての正解率（単純に最も出現頻度の高い単語を選ぶ方法の正解率）が高くならないように，一つの疑似単語を構成する各々の単語の出現頻度がほぼ等しくなるようにして構成した．コーパスとしては,「CD-毎日新聞 97 年版」を JUMAN version 3.6 (京都大学大学院情報学研究科 1998) で形態素解析したものを用いた．事例の数に関しては，各々の疑似単語について，1024 の訓練事例，1000 のテスト事例を重なりがないようにコーパスからランダムに抽出して，トレーニングとテストを行なった．

従来手法のパラメータに関しては，英語の多義語での実験と同様に，正解率が最も高くなる値を採用した．間引きの閾値に関しては 3，対数尤度比のスムージングパラメータ α に関しては 0.4 とした．

表 4に結果を示す．傾向は，表 3に示した英語の多義語での結果とほとんど同じである．最も正解率が悪いのは，間引きによる手法である．一様分布のベイズ学習は，対数尤度比とほぼ同じ正解率を達成している．最も正解率が高いのは，他のルールの確率値を利用してベータ分布によって事前分布を構成する手法である．

ここで，事前分布をベータ分布とした場合の，証拠のタイプによる事前分布の違いを見るために，表 5に事前分布の期待値を示す．この表からわかるように，事前分布の期待値の傾向は，$Window < Adjacent < Pair$ となっている．すなわち，あるルールに関して何もデータがなければ，そのルールが Window であるよりも Adjacent である方が，さらに，Adjacent であるよりも Pair である方が信頼できるということである．これは，より詳細な文脈情報を用いた方が

ていない．そのような前処理や，(Yarowsky 2000) のような言語学的知識を利用した決定リストの階層化などを行なえば，正解率を上昇させることは可能だと考えられる．

表 4　日本語疑似単語による評価
表 4　Evaluation by Japanese pseudo words

疑似単語	間引き	対数尤度比	ベイズ 一様分布	ベイズ ベータ分布
政策／テレビ	90.4% (7.1)	92.4%	92.0% (1.5)	92.6% (2.5)
大統領／首相	84.9% (9.9)	90.5%	88.6% (0.0)	89.4% (1.2)
仕事／言葉／資金／文化	66.2% (18.8)	72.8%	71.6% (9.6)	75.1% (6.3)
持つ／含む	83.7% (13.1)	87.2%	86.6% (2.3)	90.3% (0.9)
考える／見る／目指す	67.8% (17.8)	67.6%	70.5% (8.4)	73.6% (2.1)
入る／示す／開く／進める	73.1% (15.2)	77.2%	76.3% (6.8)	79.3% (2.8)
近い／難しい	84.1% (12.3)	88.5%	88.6% (2.0)	90.5% (0.2)
新しい／高い／強い	65.7% (18.2)	68.9%	68.7% (10.1)	72.6% (2.6)
若い／厳しい／大きい／よい	67.8% (16.3)	72.8%	72.0% (7.8)	77.4% (1.4)
平均	76.0% (14.3)	79.8%	79.4% (5.4)	82.3% (2.2)

（括弧内の数字は正解率と期待正解率との差）

正確な判断ができるという我々の直感とも一致する．また，これらの事前分布に影響によって，最終的な信頼度も全体として，Pair や Adjacent のルールが上位に位置することになる．

6　おわりに

本論文では，統計的クラス分類器である決定リストに対して，二つの改善方法を示した．

- ベイズ学習によるルール確率値の推定
  決定リストを作成するにあたって最も重要なことは，ルールの信頼度をどのようにして計算するかということである．本論文では，ベイズ学習の手法を用いることにより，理論的な裏付けのあるスムージングによる計算が可能なことを示した．
- 証拠の種類ごとに事前分布を設定することによる精度向上
  証拠の種類ごとに，他のルールの確率値を利用して事前分布を構成することによって，より正確な確率値の推定ができることを示した．また，その結果，決定リストにおいて，より信頼性の高いルールが上位に位置するようになり，決定リストの分類性能が向上することを示した．

本論文では，ベイズ学習の枠組で証拠の種類ごとに異なった確率値を算出することで，決定リストの性能を向上させることができることを示した．このように，証拠のタイプごとに信頼

表 5　事前分布の期待値
表 5　Expectation value of prior distribution

疑似単語	Window	Adjacent	Pair
政策／テレビ	0.74	0.77	0.82
大統領／首相	0.67	0.75	0.80
仕事／言葉／資金／文化	0.54	0.64	0.70
持つ／含む	0.70	0.82	0.89
考える／見る／目指す	0.57	0.65	0.68
入る／示す／開く／進める	0.51	0.54	0.72
近い／難しい	0.68	0.74	0.85
新しい／高い／強い	0.54	0.64	0.64
若い／厳しい／大きい／よい	0.48	0.64	0.71

度の値を変えることで決定リストの性能向上を図った研究として，(新納 1998) がある．この研究では，決定リストによる同音異義語判別において，複合語からの証拠に重みを付けることで，分類精度の向上を図っている．そこでは，決定リストの信頼度として，式（2）を用い，複合語からの証拠には，信頼度に重み付けのための係数を掛けることで，複合語からの証拠を用いたルールを優先させている．本論文では，証拠の種類ごとに対する異なった重みづけをベイズ推定の枠組における事前分布を使用して行なったと考えることができる．本手法の利点は，どの種類の証拠にどの程度重み付けをするのかを，言語学的な直観に頼ることなく，実際のルールの確率値の分布から事前分布を構成することによって自動的に決定できるという点であるといえる．

また，本論文で提案した手法は，決定リストの分類性能を向上させるだけでなく，出力する確率値の推定精度も向上させる．クラス分類器は，大きな自然言語処理システムの構成要素として用いられることも多い．その場合，各構成要素であるクラス分類器の出力は，その後の処理に利用されることになるが，出力された信頼度自体が不正確では，それらを利用する後の処理の性能を低下させる恐れがある．したがって，分類性能だけでなく，分類器の出力確率の精度も向上させる本手法は，そのような場合にさらに有効になる可能性があるだろう．

本論文では，似たような性質を持つ他の多くの確率値を利用することで，少ない事例から計算される確率値の精度を高められることを示した．この手法は同様の性質をもつ他の統計的手法に適用できると考えられる．例えば，最大エントロピー法では素性の確率を求める必要がある．最大エントロピー法を利用した多くの研究では，素性の確率値を最尤推定法によって求めているが，その場合，本論文で指摘したような確率値の推定誤差の問題がある．実際には，事例の数が少ない素性を使用しないようにすることが多いためにその問題が顕在化することは少

ないが，本論文で示した手法によって，事例の数が少ない素性も利用することで最終的な性能向上につながる可能性もあり，興味深い課題といえる．

参考文献

Freund, Y. and Schapire, R. (1999). “A short introduction to boosting.” *J. Japan. Soc. for Artif. Intel. 14(5)*, 771–780.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995). *Bayesian Data Analysis*. Chapman & Hall.

北研二 (1999). 確率的言語モデル. 東京大学出版会.

京都大学大学院情報学研究科 (1998). 日本語形態素解析システム JUMAN version 3.6.

Li, H. and Yamanishi, K. (1999). “Text Classification Using ESC-based Stochastic Deicision Lists.” *Proc. of ACM-CIKM*, 122–130.

永田昌明 平博順 (2001). “テキスト分類 -学習理論の「見本市」-.” 情報処理, **42** (1), 32–37.

繁桝算男 (1985). ベイズ統計入門. 東京大学出版会.

新納浩幸 (1998). “複合語からの証拠に重みをつけた決定リストによる同音異義語判別.” 情報処理学会論文誌, **39** (12), 3200–3206.

新納浩幸 (2000). “日本語形態素解析のクラス分類問題への変換とその解法.” 情報処理学会研究報告（2000-NL-135）, **2000** (11), 149–156.

白木伸征, 梅村祥之, 原田義久 (2000). “複数決定リストの順次適用による文節まとめあげ.” 自然言語処理, **7** (4).

鈴木雪夫 国友直人 (1989). ベイズ統計学とその応用. 東京大学出版会.

宇津呂武仁 他 (1999). “コーパスからの日本語従属節係り受け選好情報の抽出およびその評価.” 自然言語処理, **6** (7), 29–60.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. New York.

Yarowsky, D. (1994). “Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French.” *Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics*, 88–95.

Yarowsky, D. (1995). “Unsupervised Word Sense Disambiguation Rivaling Supervised Methods.” *Proc. of the 33rd Annual Meeting of the Association for Computational Linguistics*, 189–196.

Yarowsky, D. (2000). “Hierarchical Decision Lists for Word Sense Disambiguation.” *Computers and Humanities*, 34(2):179–186.

(0000 年 5 月 6 日 受付)

(0000 年 7 月 8 日 再受付)

(0000 年 9 月 10 日 採録)