決定リストによる期待損失を用いた能動学習

紺野憲一
茨城大学工学部
システム工学科

概要

本論文では Roy らが提案した期待損失を利用した能動学習の手法と、決定リストを組み合わせた手法を提案する。期待損失を用いた能動学習を用いて、低機能な決定リストを用いた識別での精度向上を目指す。比較対象としては現在もっとも一般的な能動学習の手法である QBC を用いる。それぞれの手法を同音異義語の識別問題に適用し、QBC によりも良い結果を出す事を目標とする。

能動学習は解答をつける前の例題から学習に対して効果の高い物を選別して学習させる手法であり、訓練データの構築のコストを低減できる。学習において訓練データの作成がもっとも労力を必要とする事から能動学習は有効である。

能動学習の手法としては QBC が一般的である。これは訓練データからランダムにデータを選出した新しい訓練データ複数個作り、例題を判別した結果が分かれた物を追加するデータの候補とする手法である。つまり、作成した訓練データ間の差異によって結果に違いが出ると言う事であり、元の決定リストのその事例に対する識別の曖昧生を表していると考える事ができる。QBC はこの曖昧性の高い例題を訓練データの追加候補とする事で訓練データを補強する手法であり、識別規則の精度向上に高い効果が期待できる。

期待損失は判別した例題がもつ損失率（誤りの程度）を計算する事によって、損失率をより低くする例題を選択する手法である。損失率が高い場所とは決定規則において曖昧性の高い部分と考えられる。

つまり損失率が大きく向上する例題とはこの曖昧な部分を補う例題と考える事ができ、これによって学習に効果的な例題を選択する事ができる。これは QBC の選択基準と類似しており、同傾向の結果が得られる事になる。実際 Roys らはこの期待損失を利用した手法を試み、大きな成果をあげている。

ただし、Roy らの能動学習では問題の対象が文書分類であり、ベースとなる機械学習手法は Naive Bayes である。しかも Naive Bayes に bagging の手法も組み入れている。文書分類問題に対しては Naive Bayes がよい結果を出すことが知られており、彼らの手法がうまくいったのは、ベースとなる機械学習手法のパフォーマンスが高かったためとも考えられる。

しかし現実の問題ではどのような機械学習手法を利用してもパフォーマンスの低い規則しか学習できないケースもある。このためパフォーマンスあまり高くない学習手法を用いる場合でも期待損失を利用した能動学習の手法が利用可能か調べておく事は重要である。

本論文ではこの期待損失を用いた決定リストによる機械学習手法を実装し、現在使われている手法である QBC との性能比較する。

実験は同音異義語の識別問題を扱う。同音異義語とはある単語において読みが同じでも意味が違う単語の事で、例えば「橘」と「端」は同音異義語である。同音異義語問題とは文書中の単語が同音異義語である場合に、どの意味で用いられているかを識別する問題である。

期待損失を利用した能動学習に決定リストを利用する場合、二つの実装上の工夫が必要である。

一つ目は逐次的な学習を実現することである。期待損失を利用した能動学習の手法は、単純に実現すると計算コストが高いため、逐次的な学習が可能であるような学習手法を利用する必要がある。

二つ目は期待損失の計算の為に決定リストの共起頻度計算に確率を用いる事である。

これらの工夫を施して決定リストによる期待損失を利用した学習手法を実装させ、QBC の手法と精度比較を行なった。

9 つの同音異義語に対して実験を行なった結果、精度は QBC に劣り、さらに正解率が大きく変動し安定しないも事例もあった。これは高頻度の証拠がテストデータ中に大量に含まれるため、損失率がほとんど無いはずのこれらの証拠が最終的な値に影響を与え過ぎている事が問題であると考える。また、期待損失と決定リストの default 値との相性の悪さも原因であると考える。決定リストは証拠の含まれない事例に対しては、Default のラベルのクラスを解答とする。この Default も出現頻度が高いと期待損失の値に影響を与え過ぎてしまう。

これらの問題を踏まえさらにパラメータの調整と証拠としての確率が高い例題を切り捨てるなどの改良を施して再度実験を行なった。

結果、改良前よりは明らかによい結果が得られた。しかし、QBC と比較すると同程度の性能を表すものの、依然として正解率の不安定さが残っているなどの問題が見受けられた。

これは Ddefault のラベルが変化する事による問題である。QBC では最も大規模なクラスの事例が選ばれやすい傾向があるが、期待損失ではその傾向が低い事がわかった。それにより Default のクラスの入れ替わりが激しくその影響で正解率の変動がおこる。しかし、これは希少クラスの事例もバランスよく抽出できると言う利点と考える事ができる。また、学習の繰り返し回数が増えるほど Default のラベルも安定し、その影響力も減少する事が期待できる。

これらの事から、繰り返し回数が多ければ期待損失を用いた手法の方が有効と考える事ができる。しかし、繰り返し回数を増やすと計算コストが増えるため問題となる。よって限り少ない繰り返し回数で高い成果を得る方法を模索するのが今後の課題である。

abstract

In this thesis , I describe the technique of active learning use of the expected loss that Roy proposed that uses the decision lists. It aims at the accuracy improvement in the active learning that used expected loss and decision lists. QBC is used as a comparison,that the technique of the most general active learning now . These technique is applied to the identification problem of the homonym , and the target is that the result technique that uses the expected loss is better than QBC.

Active learning is a technique that selects the data with high effect from the unlabeled training data , and it can reduce the cost of the construction of the labelled training data. Active learning is effective because labelling to the training data is the highest in the cost of the entire learning .

QBC is general as the technique of active learning. This method is devised by preparing some training data as data that adds the one that the result divided by each identification. In a word, the effective degree of the training data without the label is measured according to the separation condition of the identification result of more decision lists. QBC is a technique for reinforcing the training data by adding high vague exercise, and a high effect can be expected of the accuracy improvement of the identification rule.

The expected loss is a technique for selecting the exercise that makes the wastage rate lower by calculating the wastage rate of the loss ratio (level of the mistake). It is thought that the place where the wastage rate is high is a part that is vague in the decision rule and high. In a word, the exercise to which the wastage rate greatly decreases can select an exercise that supplements this vague part and an effective exercise can be selected. It resembles the selection criterion of QBC, and the result of same tendency will be obtained. Roys actually try the technique using this expected loss, and have given the big result.

However, the object in the question is a document classification in active studies of Roy, and the machine study technique for becoming basic is Naive Bayes. Moreover, the technique of bagging is put in to Naive Bayes. To the document classification problem it is known to put out the result in which Naive Bayes is good. Therefore, it is thought that their techniques succeeded because the ability of the basis mechanical learning technique was high.

However, there is a problem where only the rule that the function is low can be used in the case. Therefore ,it is important to examine whether the technique of active learning using the expected loss can be used even when the learning technique is used.

In this thesis,the mechanical study technique by the decision list that uses this achieved loss is achieved, and compares it with QBC.

The experiment treats the identification problem with the homonym. In the homonym, even if reading is the same in a certain word, the meaning is different word . For example ,Japanese "Bridge" and "Edge" are homonyms. The homonym problem is an identification problem with the homonym in the document.

It is necessary to devise it in two points to use the decision list for active learning using the expected loss.

The first is to achieve consecutive learning. The technique of active learning using the expected loss should use a consecutive learning technique as the calculation cost is high.

The second is a thing to use the probability for the co-occurrence frequency calculation of the decision list to calculate the expected loss.

These devices were given, the learning technique by the decision list using the expected loss was achieved, and accuracy was compared with the technique of QBC.

It experimented to nine homonyms, and as a result accuracy was inferior to QBC. In addition, there was a case where the correct answer rate increases and decreases greatly, too. This problem happens actually by the thing that the loss is given to evidence without the wastage rate. This influences harmfully in the evidence of high frequency. Moreover, it is thought that poor compatibility with the value of default of the expected loss and the decision list is a cause. As for the decision list, the class of the label of Default is answered for the case where evidence is not included. When the appearance frequency is high, Default also influences the value of the expected loss too much.

To solve these problems, it has improved it as follows. The computational method of the expected loss was improved to these problems and it experimented again.

The result was improving than before. However, it was more unsuccessful than QBC, and the correct answer rate was still unstable.

This is a problem that the label of Ddefault changes. The case with the largest-scale class in QBC tends to be chosen easily and the tendency is low in the expected loss. It has been understood that the correct answer rate changes greatly because the class of Default changes. However ,it can be thought that it is an advantage that it is not influenced by the size of the class when the exercise is selected. Moreover, if the repetition frequency of learning increases, the label of Default is fixed , and the influence power of Default decreases.

In a word, it can be thought that the expected loss is more effective if the repetition frequency of study increases. However, the calculation cost's increasing becomes a problem when the frequency of learning is increased. Therefore, the performance improvement by a little study frequency is a next problem.

目 次

図 目 次

表 目 次

第1章　序論

1.1　背景と目的

本論文では Roy らが提案した期待損失を利用した能動学習の手法 [1] を同音異義語問題に応用する。能動学習のベースとなる学習手法としては決定リストを用いる。決定リストでは低頻度の証拠の正確な分布が得られないと言う問題が起こるため、スムージングによって補正を行なう。しかし、スムージングを行なうと高頻度の証拠の損失率が高くなると言う問題が発生する。能動学習は学習に有効な訓練データの選択を目的とする。スムージングによって付与された損失の値は、この訓練データを選択する際の選択基準として用いる期待損失の値に大きな影響を与えるため、適切な選択がなされない事となる。そこで、本論文では高頻度の証拠のスムージングによる影響を無くす事によって適切な訓練データの選択を目指す。

自然言語処理では個々の問題を分類問題として定式化し、帰納学習の手法を利用して、その問題を解決するアプローチが大きな成功を納めている。しかし、このアプローチには帰納学習で必要とされる大量の訓練データを用意しなければならないという大きな問題がある。そこで小量の訓練データから得られる識別規則の精度を、大量のラベルなし訓練データによって高めていく手法が提案されている。

ラベルとはその訓練データのクラスの事である。学習において必要となるのはラベル付きの訓練データである。ラベル無しの訓練データは機械的に作成する事ができ、作成は容易である。しかし、訓練データにラベルを付ける作業は、自動化することができず作成に時間と手間がかかる。このため、ラベルを付ける作業を減らす事ができれば人的労力の削減を行なうことができる。そこで用いられるのが能動学習手法である。能動学習とは、学習者が学習に用いる訓練データを選択する学習手法である。

ラベル無しの訓練データの中から学習に有効な訓練データを選択する事ができれば、学習に有効な訓練データのみにラベルを付けて学習に利用する事により、ラ

ベルを付ける作業の削減を行なえる。

能動学習の手法としては Query By Committee が一般的だが、Roy らは期待損失を利用した手法を試み大きな成果をあげている。ただし、Roy らの能動学習では問題の対象が文書分類であり、ベースとなる機械学習手法は Naive Bayes である。しかも Naive Bayes に bagging の手法も組み入れている。文書分類問題に対しては Naive Bayes がよい結果を出すことが知られており、彼らの手法がうまくいったのは、ベースとなる機械学習手法のパフォーマンスが高かったためとも考えられる。

しかし現実の問題ではどのような機械学習手法を利用してもパフォーマンスの低い規則しか学習できないケースもある。このためパフォーマンスがあまり高くない学習手法を用いる場合でも、期待損失を利用した能動学習の手法が QBC よりもよい結果が得られるのかどうかは調べておく必要がある。

本論文では問題として同音異義語の分類問題を扱う。またベースとなる機械学習手法として決定リストを用いる。期待損失を利用した能動学習に決定リストを利用する場合には、二つの実装上の工夫が必要となる。

一つ目は逐次的な処理の導入である。期待損失を利用した能動学習の手法ではラベルなし訓練データの数だけ決定リストを作成し、さらにそれぞれの決定リストを用いて全てのラベル無し訓練データの識別を行なうという作業が必要となる。これを単純に計算すると計算コストが非常に高い。しかし作成された決定リストや識別結果の差異は極小量であるので、その差分のみを逐次的に計算させる事によって計算コストの低下を計る。

二つ目は決定リストの予測力の算出に確率を利用する事である。訓練データの期待損失の計算はその訓練データから作られた決定リストの予測力を用いる事で行なう。期待損失は確率を用いて計算される値であるので、予測力を確率を与える事で期待損失の計算に利用できるようにする。

これらの工夫を施し、期待損失を用いた手法を実装しても QBC よりも性能は良くならない。スムージングを行なう事により高頻度の確実性の高い証拠に小量の損失の値が生じてしまう事が問題と考えられる。ここで生じる損失の値は極小量ではあるが、決定リスト中の高頻度の証拠はラベルなし訓練データ中での頻度も高くなる傾向があるため、一つ一つの期待損失への影響力が小さいにも関わらず、訓練データ全体としての影響力が必要以上に大きくなる事が原因として考えられる。そこで、本論文では高頻度の証拠の損失率を固定することによって期待

損失への影響力を無くし、期待損失を用いた能動学習の精度向上を目指す。

1.2 本論文の構成

第2章では決定リストについて説明する。第3章では能動学習手法の QBC について説明する。第4章では期待損失を用いた能動学習の手法とその改良点について述べる。第5章では能動学習手法を同音異義語の分類問題に適用した実験およびその結果について説明する。第6章では本研究の考察を述べる。第7章でまとめを行なう。

第2章　同音異義語問題

同音異義語の例として「ハシ」がある。「ハシ」と発音する単語には「橋」、「端」、「箸」などがある。このように同じ読み方をする異なった語義を持つ単語の事を同音異義語と言う。

人間同士の会話では音声を媒体とするため、同じ音の単語は混同する可能性がある。文字情報においても同じ音をもつ単語は変換を間違えたり、似た意味をもつ単語を誤って利用する場合がある。

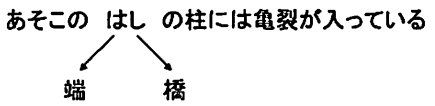


図 2.1: 同音異義語の判別

図2.1の文では「橋」と「端」のどちらとして用いられているのか判断し難い。

そこで、「はし」という単語が含まれる文が与えられた時、その文中の「はし」がどの「はし」であるか分類するという問題を考える事ができる。このようにある問題が与えられた時に複数のクラスの内、どれに分類されるかを考えるのが分類問題である。

このような同音異義語の分類問題を同音異義語問題と言う。

この同音異義語問題の中でも、似た意味をもつ単語は間違った使われ方をする可能性が高い。例えば「かいほう（解放、開放）や「せいさん(清算、精算)」や「ろてん(露天、露店)」などがある。

「かいほう」の同音異義語一覧を表2.1に載せる。

表2.1の中でも「解放」と「開放」はどちらも対象を自由にするといったニュアンスが含まれており混同しやすい。また「会報」と「回報」においても情報を複数に伝えるといったニュアンスが両方に含まれており混同しやすい。逆に、「介抱」

	意味	用例
開放	開け放つ、出入り自由にする	窓を開放する
解放	体や心の制限を取り除き、自由にする	奴隷を解放する
介抱	病人の世話をする	お年寄りを介抱した
快方	病気が良くなる事	病気が快方に向かった
回報	返事	戦争に関する回報が届いた
会報	会員に対する知らせ	クラブの会報を配った
解法	問題を解く方法	数学の解法を暗記する

表 2.1: 同音異義語：かいほう

と「会報」などは意味的に混同する可能性は少ない。

本論文ではこのような混同しやすい単語 [6] に対して実験を行なう事とする。

実験対象は表 2.2 の 9 単語とする。

同音異義語	クラス 1	クラス 2
さいけん	「債券」	「債権」
かいほう	「解放」	「開放」
きょうちょう	「協調」	「強調」
じしん	「自身」	「自信」
たいがい	「体外」	「対外」
うんこう	「運航」	「運行」
かてい	「過程」	「課程」
しょくりょう	「食糧」	「食料」
しょうがい	「障害」	「傷害」

表 2.2: 実験対象単語

それぞれの同音異義語の定義と用例を表 2.3 に載せる。

単語	意味	用例
「債券」	国などが発行する有価証券	債券の購入
「債権」	給付を請求しうる権利	債権者の訴訟問題
「解放」	体や心の制限を取り除き、自由にする	奴隷を解放する
「開放」	開け放つ、出入り自由にする	窓を開放する
「協調」	力を併せて物事を成すこと	協調性を大事にする
「強調」	物事を強めに言う、または主張すること	強調したい文字
「自身」	自らの事	自分自身を大事にする
「自信」	自分を信じる事	自信をもって答える
「体外」	肉体の外	体外に摘出された腫瘍
「対外」	外部または外国に対する事	対外政策
「運航」	船や飛行機が進路を進む事	船の運航
「運行」	バス・電車や星などの定まった行動	星の運行
「過程」	物事が変化していく道筋	今までの過程
「課程」	学種などで習得すべき事項	博士課程
「食糧」	食べ物、とくに主食の米や麦	食糧難
「食料」	食べ物 またはその材料	食料品の買い出し
「障害」	物事を妨げるもの、または身体の機能不全	障害となる法律
「傷害」	傷つける事	傷害事件

表 2.3: 単語の意味と用例

第3章　決定リスト

決定リスト [7] では識別対象の中に含まれる全ての事象の中から識別に利用する素性を選択し、そこから識別に用いる証拠を作成する。その証拠の有効度の順に並べてリスト化したものが決定リストである。また、決定リストでは Default と言う証拠を作成し、証拠の現れない事例の識別に利用する。

低機能な手法であるが、実装が容易で人間がその動作を確認できるため、改良しやすいという利点もある。

3.1　決定リストの作成アルゴリズム

決定リストでは識別に有効な素性を選択し、そこから識別に用いる証拠を作成する。素性とは識別においてその判断材料となるであろう事例の特徴の事である。

例えば、動物の分類に「哺乳類」、「両生類」、「爬虫類」などがある。これをそれぞれ振り分けるクラスと考え、ある動物がこれらのどの分類（クラス）に当てはまるかを考えるとする。この時、素性としては「体毛があるかないか」や「足の本数」などが考えられる。

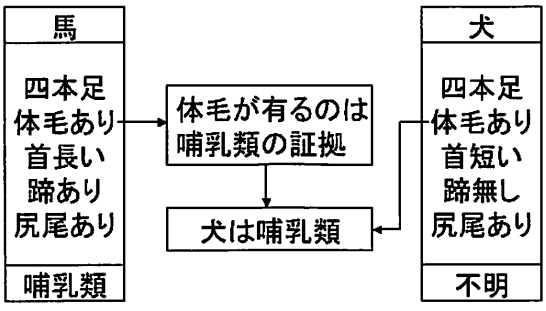


図 3.1: 動物の素性

図3.2のように学習に用いる事例として「馬」が与えられているとする。ここで、その素性として、「体毛があるかどうか」、「足の本数」を考えると「体毛＝有り」、「足の本数＝4」と言う証拠ができる。証拠とはそれぞれの事例の素性と実際の事例中に含まれるその素性の値である。

「馬」は哺乳類であるのでこの事例の分類されるクラスは「哺乳類」とする。この事例のクラスをラベルと呼ぶ。識別規則の学習にはラベル付きの事例が用いられる。

決定リストにおいて、事例から作成された証拠はその事例のラベルに対する証拠として考える。つまり、事例「馬」から作られる「体毛＝有り」と言う証拠はクラス「哺乳類」である証拠だと考える。よって「哺乳類」を「体毛＝有り」のラベルとする。

ここで、動物「犬」の識別を考える。「犬」の特徴と証拠とを照らし併せて一致するものがあれば、その証拠のクラスが「犬」の識別クラスとなる。先ほど作成した証拠「体毛＝有り」は「犬」の特徴と一致するので、「犬」は「哺乳類」と識別される。

ここで、「体毛があるかないか」という特徴が素性として適当か考える。体毛があると言うのは哺乳類独特の特徴であり、突き詰めれば例外もあるが哺乳類以外にはこの特徴は当てはまらないので、「体毛＝有り」は哺乳類である証拠として適切だと考える事ができる。では「足の数＝4本」と言う証拠はどうであろうか。確かに「犬」や「馬」などは「哺乳類」でかつ4本足である。しかし、同じ「哺乳類」である「猿」は2本足である。また、「爬虫類」の「トカゲ」や両生類の「蛙」など「4本足」だが「哺乳類」でない動物も多い。よって証拠「4本足」を「哺乳類」の証拠として考えるのは適切とは言い難い。

つまり、設定した素性から得られる証拠が、識別における証拠として適切かどうかを考える必要がある。また、「4本足」のようにほとんどの証拠は複数のクラスの証拠となる可能性をもつため、その証拠をどのラベルの証拠として扱うかと言う判断も必要となる。

決定リストでは、共起頻度と予測力によってこれを決定している。

共起頻度とは全ての訓練データ中である証拠が同じクラスをもって現れる頻度である。

例えば素性として「足の本数」を設定する。学習に用いる事例の集合を訓練データと呼ぶ。

図 3.2: 訓練データ：動物

図3.2のように訓練データとして「馬」、「猫」、「ニワトリ」、「猿」、「トカゲ」があったとするとこの中で証拠「足の本数＝4」を含むものは「馬」、「猫」、「トカゲ」である。それぞれのクラスは「哺乳類」、「哺乳類」、「爬虫類」であり、「足の本数＝4」と「哺乳類」が共起する頻度は2、「4本足」と「爬虫類」が共起する頻度は1となる。決定リストでは共起頻度が一番高いクラスがその証拠のラベルとなる。この場合、「足の本数＝4」のラベルは「哺乳類」である。

しかし、実際には「足の本数＝4」であったからと言って「哺乳類」とは断定できない。識別で実際に用いる証拠は識別対象の特徴と一致する証拠の中でも確実性の高いものを用いるのが望ましい。「犬」のように「足の本数＝4」で「体毛＝有り」動物の場合、「4本足」で識別を行なうと学習に用いた事例によっては「4本足」のラベルは「爬虫類」や「両生類」などになる可能性があるため、「体毛＝有り」で識別した方が確実である。

そこで証拠の確実性を計る目安として予測力という値を用いる。予測力の詳しい計算方法は後に述べる。予測力が高ければ確実性の高い証拠と考える。よって比較対象中の特徴と一致する証拠の中で一番予測力の高いものを識別に用いる事で確実性の高い識別を行なう事ができる。先ほどの例で証拠「体毛＝有り」について考えた場合、その証拠を含む訓練データは「馬」、「猫」、「猿」である。これらは全て「哺乳類」と言うラベルを持ちその予測力は「足の本数＝4」よりも高くなる。よって識別対象がどちらの証拠も持つ場合、「足の本数＝4」と言う証拠よりも「体毛＝有り」と言う証拠を用いて識別を行なった方が確実性が高いと考え

る事ができる。

この予測力の順に証拠をリストかしたものが決定リストである。決定リストを用いた識別はこのリストの上の証拠からと識別対象の特徴とを照らし合わせ、証拠と特徴が一致した時その証拠のラベルを識別結果とする。

証拠	予測力	クラス
M	0.99234	a
N	0.98272	b
S	0.97457	a
T	0.93453	c
...	...	...
...	...	...
Default	0.3331	a

表 3.1: 決定リスト 1

表 3.1 は決定リストの例である。リストの一番最後には Default と言う証拠が入る。決定リストはリスト上に現れる証拠を含むものしか識別できない。そこで、リスト上の証拠が一切含まれない識別対象の識別を行なうのがこの Default である。

Default のラベルには訓練データ中に一番多く現れるクラスが設定される。また、その予測力は訓練データのラベルの割合から計算される。これは、クラスが不明の事例がそれぞれのクラスである可能性は、事象全体のクラスの割合と同じであると考える事ができるからである。

例えば、事例 x をクラス (A,B) のどちらかに分類する。この事例が含まれる事象の全ての事例のクラスがわかっている時、事象全体に現れるクラスの割合を見るとクラス A である事例が 9 割でクラス B である事例が 1 割であった。この場合識別を行なわなくてもこの事例 x は 9 割の確率でクラス A であると考える事ができる。よって、Default のラベルとして A を設定しておけばこれを用いた識別の 9 割は正解となる。

このように一番割合が多いクラスを Default のラベルとして設定する事によって、決定リスト上の証拠で識別ができない時に完全にランダムな判定を行なう事を防ぐ。

また、Default の予測力は適当にクラスを選択した時の確からしさとして考える事ができるので、決定リストでは Default より小さい予測力を持つ証拠は識別の証拠にはならないとして、切り捨てる。

実際には事象全体のクラスの割合は未知である。訓練データは事象全体からランダムで抽出したものと考えられるので、訓練データのクラスの割合は事象全体のクラスの割合と類似していると考える。

よって実際の Default の計算には訓練データのクラスの割合を用いる。

決定リストの作成は以下の 4 つの手順で行われる。

step1:証拠の作成

step2:共起頻度の計算

step3:予測力の計算

step4:リストの作成

ある事例 x がクラス $(c_1, \cdots, c_n)$ のどのクラスに含まれるかを判定する事とする。x を識別する規則の学習には訓練データの集合 D を用いる。決定リストの作成手順について以下に述べる。

step 1 : 証拠 evd の作成。

訓練データ集合 D の素性の値より証拠 evd を作成する。同音異義語の識別においては訓練データは識別対象の同音異義語を含む文となる。よって素性はその文中の単語に対して設定する。

例えば同音異義語問題「はし」に対して、「墨田川の橋の交通量を減らしたい」という文を訓練データとする。この時、素性を「文中の名詞」とすると証拠として「墨田川」や「交通量」などが作成される。「はし」を含む文に「墨田川」や「交通量」という単語が含まれていれば、その文中の「はし」が「橋」である証拠と考える。

ここで、素性をどのように設定すると良いかを考える。単純に素性を「訓練データに含まれる全ての単語」として設定すると、識別に役立たない予測力の小さな

証拠が大量にできてしまう危険性があり、計算効率が悪くなるなどの問題がある。

例えば、助詞の「の」はどのような文にも多く出現するであろうから証拠として役に立たない。

そこで、通常は単語の中からよりクラスの識別に強く関わっているだろう素性を選択し利用する。しかし、素性の設定をあまりに限定したものにすると識別に用いるデータが足りなくなり、識別の精度が低くなる。つまり証拠の設定は決定リストの性能を決定する重要な問題である。

本論文では文中での単語の位置関係や品詞情報を利用して素性の設定を行なう。

品詞情報を考えると名詞が一番その文の特徴を表しやすいと考える事ができる。また、単語の位置関係では問題の同音異義語に近い物ほどその同音異義語との関連性が高い事が期待できる。

例えば「利根川の橋」という文では「利根川」と「橋」の関連性は高いと考えられるが、「利根川の上流に位置する山々の北に位置する村の端に実家の家がある」という文では「利根川」と「端」の関連性は低いと考えられる。

このような考え方から本論文では素性として表 3.2 の e1〜e5 を設定する。

e1=直前の単語
e2=直後の単語
e3=直前の 2 単語
e4=直後の 2 単語
e5=文書中に現れる名詞

表 3.2: 素性の設定

これは e1 と e2 で文書中での単語の接続のされ方、e3 と e4 で文書の流れ、e5 で文書の内容を読みとる事を目的として設定した。

「このページは利根川の橋や写真を紹介しています」を学習に用いる事例として考えた場合、証拠は図 3.3 でラインをひいた部分の単語となる。

実際にはまず chasen を用いて形態素解析を行ない、以下のように単語に分割する。

これを形態素解析したものが表 3.3 である。

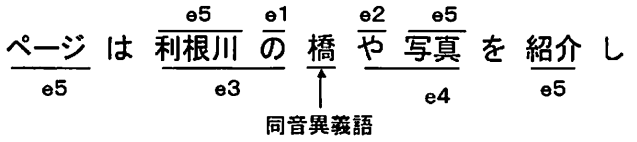


図 3.3: 証拠の設定

この	この	連体系
ページ	ページ	名詞-般
は	は	助詞-係助詞
利根川	利根川	名詞-固有名詞-人名-姓
の	の	助詞-連体化
橋	橋	名詞-一般
や	や	助詞-並立助詞
写真	写真	名詞-一般
を	を	助詞-並立助詞
紹介	紹介	名詞-サ変接続
し	する	動詞-自立
て	て	助詞-接続助詞
い	いる	動詞-非自立
ます	ます	助動詞

表 3.3: 形態素解析

これより、素性として設定した部分を抽出し、表3.4のような証拠を作成する。

実際に決定リストの作成に必要な事例中に含まれる情報はこの証拠とラベルである。よって、本論文では決定リストの作成に用いる訓練データは事例中の証拠とラベルのみを抽出した形を用いる。

次の3つは同音異義語問題「さいけん」(債券、債権) の実験で実際に利用した、訓練データの一部である。

e1=偽造 e2=が e3=の-偽造 e4=が-埼玉 e5=偽造 e5=ワリコー e5=債 e5=埼玉 e5=県内 e5=東京 債券

素性	証拠	ラベル
e1	e1=の	橋
e2	e2=や	橋
e3	e3=利根川-の	橋
e4	e4=や-写真	橋
e5	e5=ページ	橋
e5	e5=利根川	橋
e5	e5=写真	橋
e5	e5=紹介	橋

表 3.4: 証拠

e1=（ e2=担保 e3=ロンバードレート-（ e4=担保-貸付 e5=ロンバードレート e5=% e5=５ e5=担保 e5=貸付 e5=金利 債券

e1=に e2=相場 e3=背景-に e4=相場-の e5=背景 e5=回復 e5=景気 e5=相場 e5=急騰 e5=受ける 債券

訓練データ中の全ての事例から m 個の証拠 $(evd_1, \cdots, evd_m)$ が作成される。

step 2 : 共起頻度 frq の計算

共起頻度 $frq(evd_j, c_i)$ は事例中で evd_j と c_i が共起した回数である。訓練データ全体でそれぞれの証拠の共起頻度を計算する。

例えば証拠「墨田川」が設定されていた場合、訓練データから「墨田川」が出現する事例だけを抽出しそのラベルが「端」の場合と「橋」の場合の回数をカウントする。それぞれの場合の訓練データの個数が共起頻度である。

表 3.5 は先の事例中の証拠とそれぞれのクラス「端」と「橋」との共起頻度の例である。

step 3 : 予測力 est の計算

証拠がそのクラスの証拠である確からしさが予測力 est である。予測力 $est(c_i, evd_j)$

証拠	共起頻度	
	「橋」	「端」
e1=の	10	5
e2=や	14	12
e3=利根川-の	30	0
e4=や-写真	28	1
e5=ページ	2	20
e5=利根川	30	3
e5=写真	29	2
e5=紹介	12	13

表 3.5: 共起頻度

は共起頻度 $frq(c_i, evd_j)$ から計算し、一つの証拠の中で予測力がもっとも高くなるクラスをその証拠が指し示すクラスとみなせる。

予測力 $est(c_i, evd_j)$ には通常以下の式が用いられる。

$$est(c_i, evd_j) = \log \frac{frq(c_i, evd_j)}{sum - frq(c_i, evd_j)} \tag{3.1}$$

$$sum = \sum_i frq(c_i, evd_j) \tag{3.2}$$

ここでは期待損失の計算に確率を用いる必要があるため、予測力は以下の式で計算する。

$$est(c_i, evd_i) = \frac{frq(c_i, evd_j)}{sum} \tag{3.3}$$

例えば「e2=渡る」と言う証拠に対し、テストデータからクラス「橋」が 5 回、クラス「端」が 2 回、クラス「箸」が 1 回出現した場合を考える。この時クラスを (橋、端、箸、梯) と定義するならばその共起頻度は $(5, 2, 1, 0)$ となる。この時証拠「e2=渡る」の指し示すクラスは共起頻度が最も大きい「橋」であり、その予測

力は以下のようになる。

$$\frac{5}{5+2+1+0} = 0.625$$

また、決定リストではDefault値と言う値が用いられる。Default値は識別の際、識別対象に決定リスト中の証拠が含まれなかった時に識別の基準として用いる値である。全訓練データのラベルの割合を計算し、最もその割合が高いクラスをこのDefaultのラベルとする。Defaultの予測力は以下の値を用いる。

$$est(c_i, Default) = \frac{max(|c_i|)}{|D|} \tag{3.4}$$

例えば訓練データが100個あった場合、「橘」をラベルに持つ訓練データが74個、「端」をラベルに持つ訓練データが26個であったとする。この時のDefaultのラベルは「橘」となりその予測力は以下のようになる。

$$\frac{74}{100} = 0.74$$

step 4 : 決定リストの作成

それぞれの証拠をを予測率の大きい順に並べたものが決定リストとなる。Defaultもこのリストに加え、Defaultよりも予測力が低いものは切り捨てる。これはDefaultは識別不能な場合の最低限の予測力しか持たないと考えるため、Defaultよりも予測力の小さい証拠は信頼性に欠けると考えるためである。

識別では、このリストの上から識別対象中の素性とリスト中の証拠とを比較する。この素性と決定リスト上の証拠が一致した時、その証拠が指し示すクラスを解答として返す。この時、決定リスト上に識別対象中の要素が現れない可能性がある。この時はDefaultの指し示すクラスを解答として返す事にする。ここでDefaultよりも予測力が低い証拠のみが問題中に含まれている場合もDefaultの値が選ばれる事になる。

3.2 低頻度証拠に対する確率の調整

予測力の計算をする上で証拠の出現頻度が問題となる。

例えば、証拠evd_jがクラスA, B, Cのどれのラベルを持つとして、その証拠が現れる大量の事例からそれぞれのクラスと共に現れる割合を調べるとAが50%、

証拠	予測力	クラス
e1=が	0.94894	橋
e3=形-の	0.94272	端
e2=に	0.93457	端
e3=日本-の	0.93453	橋
...	...	...
e2=渡る	0.625	橋
...	...	...
Default	0.4331	端

表 3.6: 決定リスト 2

B が 30%、C が 20% だったとする。これを evd_j のクラスの正しい分布の割合と考える。

この時ある訓練データにおいてこの証拠が 100 回現れ、A のラベルを持つ証拠が 50 個、B のラベルを持つ証拠が 30 個、C のラベルを持つ証拠が 20 個であったならクラスの正確な分布を得る事ができる。証拠の出現頻度が多いならば分布の誤差は小量であると考えられる。

しかし、図 3.4 で表すように証拠の出現頻度が少なくなるほどこの誤差は大きくなる。例えば、証拠が一度しか現れずそのラベルが A であった場合、その時現れた証拠のラベルとなるクラスの分布は A が 100% となってしまう。

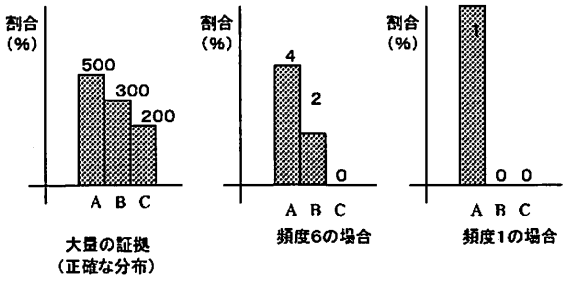


図 3.4: 頻度と分布

そこでスムージング [2] と言う処理を用いて低頻度の分布に修正を加える。図 3.5 で表すようにスムージングとは特出したクラスの割合を他のクラスに分散させる事によって、データに無いクラスの割合を予測するものである。

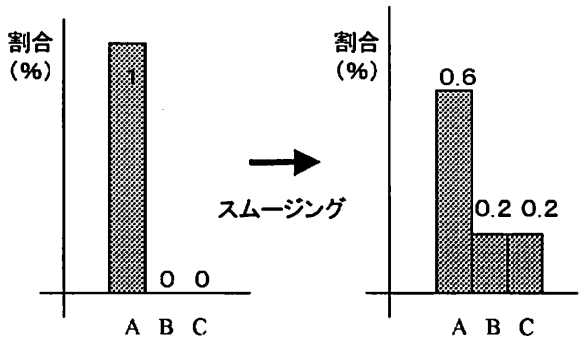


図 3.5: スムージング

今回は式 3.5、3.6 のように予測力の分母にパラメータ α を付け足す事によってスムージングを実現した。

$$est(c_i, evd_i) = \frac{frq(c_i, evd_j)}{sum + \alpha} \tag{3.5}$$

$$Default = \frac{max(|c_i|)}{|D| + \alpha} \tag{3.6}$$

第4章　能動学習

能動学習とは学習者が能動的に学習データを選択する機械学習手法である。

訓練データが学習にどの程度効果があるのかを学習前に予測し有効な訓練データのみを学習に利用する事により、訓練データの作成コストと学習に掛かる時間的コストを低下させる事ができる。

実際問題では学習に用いる訓練データの作成で一番手間が掛かるのが事例に対するラベルの作成である。ある問題xに対するラベルyは全て未知である。よって学習にはラベル付きのデータが必要となる。ほとんどの場合これは人間の手で与える必要がある。つまり、どうしてもコンピュータによる自動化ができないのがこの訓練データの作成である。

能動学習はこのラベルをつける作業を必要最低限に押える事ができる。これが能動学習の最大の利点となる。

能動学習で現在最も一般的な手法として QBC [3] がある。また、今回利用する期待損失を用いたものも能動学習手法の一つである。

QBC は既に確立されている手法であり QBC と期待損失を用いた手法とで学習した識別規則がどの程度の正解率を持つかを比較する事によって、期待損失を用いた手法の有効性を考える事ができる。

また、QBC と期待損失を用いた手法はどちらも決定規則の曖昧性解決を行なう手法であり、選択されるデータの傾向もある程度の類似性を持つと考えられる。よって QBC の手法と期待損失を用いた手法とで選択されたデータの傾向を比較する事で、期待損失を用いた手法で選択された事例の有効性を考える。

今回実際に利用するのはこの Query By Bagging [4] である。ここでは Query By Bagging も QBC の1種として位置付けて、QBC と呼ぶ。

4.1 QBC

4.1.1 QBCによる曖昧性の解消

QBC の手法は証拠の曖昧性の解消を目的とする。証拠は全てのクラスに対しての証拠となる可能性を持ち、場合によってはどのクラスの証拠となるかはっきりしない事がある。これが、証拠の曖昧性である。

証拠の曖昧性は訓練データの偏りや証拠を含む事例の不足によって起こる問題である。例えば、ある事例のラベルの出現頻度の分布がクラス A が 30% でクラス B が 70% だったとする。しかし、偶然に訓練データとして用いた事例ではクラス A のラベルを持つものの方が多い場合がある。この時、この証拠を含む事例を追加する事を考えると、追加する事例のラベルはクラス A より クラス B の可能性が高い。それは事象全体でクラス B の事例が多いだけではなく、既にクラス A の事例は既にいくつか用いられているため、それ以外の事例がクラス B の事例である確率は高くなるからである。

よってこの証拠の事例を追加していくとクラス A の証拠からクラス B の証拠に変わる事となる。

つまり事例を追加していく事でラベルが変化する証拠を曖昧性の高いクラスとする。これは、訓練データ中の証拠の数が少ないほど顕著な問題になる。頻度 1 の証拠はどのクラスになるかはその証拠が現れた事例次第であり、その証拠の確からしさは低く、曖昧性が高いと考えられる。

曖昧性が高い証拠は、証拠を追加していけば曖昧性を解消し、正しいラベルを持つ証拠になる事が期待できる。

しかし、訓練データに追加するものは証拠ではなく証拠を含む事例である。よって曖昧性の高い証拠を含む事例を曖昧性の高い事例とし、事例の曖昧性の高さを考える事とする。

QBC ではこの曖昧性を複数の訓練データを用いた識別によって計測する。

複数の訓練データでそれぞれの識別規則を学習し、ある事例の識別を行なったとする。この時、全ての識別規則が同じクラスであると判定するならこの事例の曖昧性は低いと考える。しかし、識別規則ごとに識別結果に違いが現れた場合、それは識別規則ごとで識別に用いた証拠が違う、または証拠のラベルが違うという事であり、これを曖昧性の高い事例と考える。

実際の識別問題では訓練データは一つである。そこでその訓練データから複数の訓練データを作成する。単純に訓練データを複数に分割して用いた場合、一つの訓練データのデータ数が少なくなる事によりその訓練データから作られる識別規則は分割前の訓練データから作られるものより性能が著しく低下してしまう。例えば、QBC を利用するために 10 個の訓練データを作成する場合を考える。学習に用いる訓練データを 10 分割してしまえば当然識別精度は落ちる。精度を下げずに QBC を利用する為には与えられた訓練データの 10 倍近い訓練データの作成を行なう必要があり能動学習のコスト低下の目的に反してしまう。そこで最初に与えられている訓練データより、できる限り訓練データの事例数を減らさずに複数の訓練データを作成する為の工夫が必要となる。

QBC ではランダムサンプリングと復元抽出を用いてこの問題を解決している。ランダムサンプリングと復元抽出は次の項で解説する。

4.1.2 QBC アルゴリズム

QBC アルゴリズムの手法は以下のとおりである。ある事例 x がクラス $c_1, \cdots, c_n$ の内どれかに含まれるかを識別する。図 4.1 は QBC の動作の流れ図である。

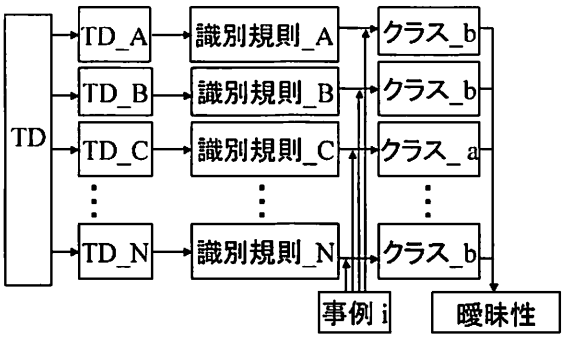


図 4.1: QBC の流れ

まず識別規則を学習する為の訓練データ D が与えられているとする。そこからランダムサンプリングを用いて訓練データ D の 8 割程度の訓練データ D_1 を作成

する。ここで訓練データ D は D_1 と残りの2割 D_1' に分割されているが、D_1 と D_1' を再結合し、そこからまたサンプリングを行なう事を復元抽出という。この復元抽出を用いてトレーニングデータ D から10個前後のトレーニングデータ $D_1, \cdots, D_n$ を作成する。

この $D_1, \cdots, D_n$ より、識別規則 $R_1, \cdots, R_n$ を作成する。ある事例 x をそれぞれの識別規則で識別し、クラス $c_1, \cdots, c_n$ に識別された回数をそれぞれ $a_1, \cdots, a_n$ とし、その事例の曖昧性を表す値 y を以下の式4.2で計算する。

$$y = \prod_{i=1}^{n} \frac{a_i}{n} \tag{4.1}$$

(4.2)

この値が大きいものほど識別が難しい事例と考え、最も大きい y^* を訓練データに追加する事例とする。

$$\forall(x) \quad y^* < y \tag{4.3}$$

4.2 期待損失

期待損失による能動学習は識別規則の曖昧な部分を見つけ出し、その部分を補強するデータを追加する事により曖昧性の解消を行なう手法であり、その考え方は QBC の手法と類似している。

訓練データには事例の偏りが存在する。訓練データは識別対象が現れる事例の一部から作られるため、この訓練データ集合が事例全体の特徴をうまく表しているかは分からない。この時事例全体の特徴と訓練データの特徴とのずれが訓練データの偏りである。よって、訓練データ中に含まれる証拠がそれぞれのクラスを指し示す予測力の分布にも偏りが生じる。この分布の偏りはその証拠を含むデータを訓練データに追加する事によって少しずつ解消されると考えられる。

ある証拠に対して、その証拠が現れる事例のほぼ全てを訓練データ中に追加する事ができるのなら、その時の証拠の分布をその証拠の真の分布と言い替える事ができる。

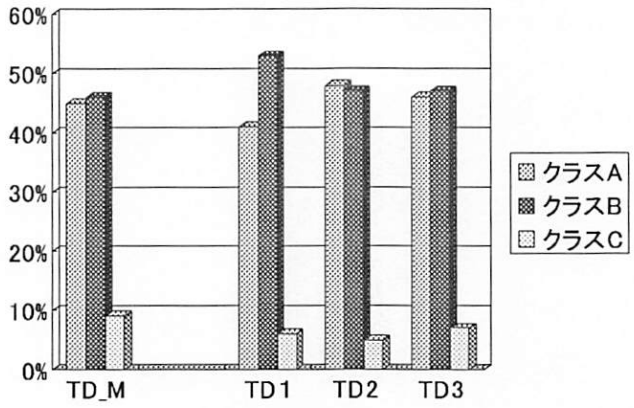


図 4.2: 期待損失

図 4.2 は全ての事例を含む訓練データ TD_M と通常の訓練データ TD1,TD2,TD3 があった時、それぞれの訓練データからある証拠の予測力を計算した時識別されるクラス A, B, C の予測力の分布が訓練データによって違いが生じている例である。

ここでこの真の分布とある訓練データからえられる分布との差を損失と考える事ができる。

ここで訓練データに含まれる証拠の全体の損失を訓練データのもつの損失と考える。この訓練データの期待損失は計算できるとする。この時、一つのデータを追加した時の訓練データの期待損失は追加した事例の分だけ減少する。よってこの期待損失をより低くする事ができるデータを選定する事で、識別規則の曖昧性の解消を試みるのが期待損失の手法である。

4.2.1 識別規則の期待損失

本実験で利用する識別規則は決定リストである。理想的な決定リストは全ての証拠の予測力が 100% となる。しかし、実際には 100% の証拠だけで確実に識別を行なえるという事は考えられず、複数のクラスの可能性をもつ不確定な証拠を利用しなくてはならない。よって、識別を行なった結果が外れる可能性を含んでいる。この識別が不正解になる可能性を損失とする。決定リストよりそれぞれの

証拠の持つ損失は計算できる。

ある証拠に対し事例を追加した時その証拠力の変化によって期待損失の値も変化する。そこで、この時の期待損失の減少量を計る事によって決定リストに対するその事例の貢献度を計るのが期待損失の手法である。

4.2.2 能動学習への利用

期待損失の能動学習への適用はこの識別規則における期待損失を利用する。図4.3は訓練データから期待損失導出までの流れ図である。

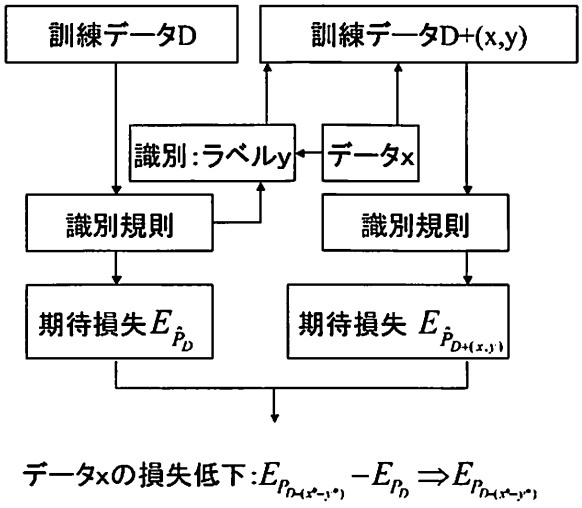


図 4.3: 期待損失の導出の流れ

まず、期待損失を計算する訓練データ D を用意する。ここで、この D から学習した識別規則に対する期待損失は計算可能とする。ある事例をこの訓練データに追加しその識別規則を学習したとすると、もちろんその期待損失の値は追加以前より微妙に低下する。この期待損失の微妙な下降値を追加した事例の学習に対する有効度として利用する。事例追加前の期待損失の値は一定であるので、実際にはデータ追加前と追加後との差を計算する事はせずデータ追加後の期待損失の値自体を学習に対する有効度として利用する。

全ての追加候補のデータに対してこの処理を行ない、その中から最も期待損失

の値が小さいものを選出する。

4.2.3 損失率の計算

ここに $P(x|y)$ の分布が未知な入力 x がある。この x が識別されるであろうクラスは $y \in (y_1, \cdots, y_n)$ であり、適当な入力の分布 $P(x)$ があると考える。学習者にはラベル（正解）付きの訓練データ D が与えられているものとする。訓練データ D より識別規則を学習し、x が入力された時出力分布の予測 $\hat{P}_D(y|x)$ が導き出せる。

ここで、期待損失 $E_{\hat{P}_D}$ は以下の式で表せる。

$$E_{\hat{P}_D} = \int_x L(P(y|x), \hat{P}_D(y|x))P(x) \tag{4.4}$$

ここで L は損失の程度を表す関数である。損失関数の基本的なものには以下の2種類がある。

logloss:

$$L = \sum_{y \in \mathcal{Y}} P(y|x) log(\hat{P}_D(y|x)) \tag{4.5}$$

0/1loss:

$$L = \sum_{y \in \mathcal{Y}} P(y|x)(1 - \delta(y, argmax_{y' \in \mathcal{Y}} \hat{P}_D(y'|x)) \tag{4.6}$$

実際の手順としてはある入力 x^* と そのラベル y^* をトレーニングセット D に加え新しいトレーニングセット $(D + (x^*, y^*))$ を作成する。ここで期待損失を計算し他の x との比較より以下の条件を満たす x^* を見付ける。

$$\forall(x, y) \quad E_{\hat{P}_{D+(x^*,y^*)}} < E_{\hat{P}_{D+(x,y)}} \tag{4.7}$$

しかしここでいくつかの問題が発生する。

まず一つは訓練データに追加する x のラベル y は未知だと言う事である。これは訓練データ D によって得られる識別規則から識別結果の値をラベルとする事で補う。

また期待損失 $E_{\hat{P}_D}$ の計算において真の分布 $P(y|x)$ は未知である。したがってここでは現在利用可能なものを用いてそれを期待損失を評価する事になる。例えばlogloss ではエントロピーによって損失を評価する事と言い替える事ができる。

先ほどの $(D+(x^*,y^*))$ を D^* と表す事にして期待損失の式は以下のようになる。エントロピーについては次の項で解説をする。

logloss:

$$\tilde{E}_{\hat{P}_{D^*}} = \frac{1}{\mid \mathcal{P} \mid} \sum_{x \in \mathcal{P}} \sum_{y \in \mathcal{Y}} \hat{P}_{D^*}(y|x) log(\hat{P}_{D^*}(y|x)) \tag{4.8}$$

0/1loss:

$$\tilde{E}_{\hat{P}_{D^*}} = \frac{1}{\mid \mathcal{P} \mid} \sum_{x \in \mathcal{P}} (1 - max_{y \in \mathcal{Y}} \hat{P}_{D^*}(y|x)) \tag{4.9}$$

4.2.4 エントロピー

エントロピーとは「不確定性、乱雑さ、無秩序の度合い」などの意味を表す単語である。もともとは物理学の概念で、物質やエネルギーの局在（偏り）を表す。

物理的なエントロピーの例としては水に垂らしたインクがあげられる。最初はインクの分子は水の中のある部分、インクを垂らした周辺にもやのように固まっている。これがエントロピーの低い状態である。しかし、時間の経過と共にインクは水全体に広がっていき、やがて水全体を濁らせる事となる。この状態がエントロピーの高い状態となる。自然界ではエントロピーは系全体として減少する事はなく、時間と共に増加をつづける。これは物理学の「熱力学第2法則」である。

このエントロピーの概念を、情報量の定義指標として情報理論に導入したものが情報エントロピーである。情報科学分野では、物理的なエントロピーとの混同しない為にエントロピーの事を情報エントロピーと呼び使い分けている。

情報エントロピーとは事象の不確かさとして考え、ある情報による不確かさの減少分が、その情報の情報量であると考える。情報を受け取る前後の不確かさの相対値が「情報エントロピー」となる。

例えばさいころを振った時、結果を見る前はどの目が出るかは全くわからない。この時の不確かさ、情報エントロピーは最大となる。実際に振ってみて奇数の目が出たという情報を受け取ると情報エントロピーは減少する。さらに1の目がでた事つまりは結果、を知った時の情報エントロピーは最小となる。

情報エントロピーとは情報科学における損失と考える事もできる。これを利用して期待損失の値を導出する事を考える。

情報源 D からある事象 x を観測した時、クラス y についての曖昧さが減少する事が期待される。まず $y \in (y_1, \cdots, y_n)$ に対して定義される情報エントロピーは

$$H_0 = -\sum_j p_D(y_j) \log p_D(y_j) \tag{4.10}$$

である。ある x を観測した後の情報エントロピーは

$$H_x = -\sum_j p_D(y_j|x) \log p_D(y_j|x) \tag{4.11}$$

となる。x が確率 $p_D(x)$ で生じる場合の条件つき情報エントロピーは

$$\begin{aligned} H_{p_D(x)} &= -\sum_i p_D(x_i) H[p_D(y|x_i)] \\ &= -\sum_i \sum_j p_D(x, y_j) \log p_D(y_j|x_i) \end{aligned} \tag{4.12}$$

となる。これより情報源 D の情報量はエントロピーの減少量として表せる。

$$\begin{aligned} I(D, x) &= H_0 - H_{p_D(x)} \\ &= -\sum_j p_D(y_j) \log p_D(y_j) + \sum_i \sum_j p_D(x, y_j) \log p_D(y_j|x_i) \end{aligned} \tag{4.13}$$

情報源 D に追加情報 (x^*, y^*) を追加したものを D^* とするならば D と D^* とのエントロピーの差より x^* のエントロピーの減少量を推量する事ができると期待される。

$$\begin{aligned} I(D, x) - I(D^*, x) &= [H_0 - H_{p_D(x)}] - [H_0 - H_{p_D(x^*)}] \\ &= [-\sum_j p_D(y_j) \log p_D(y_j) + \sum_i \sum_j p_D(x, y_j) \log p_D(y_j|x_i)] \\ &\quad - [-\sum_j p_D^*(y_j) \log p_D^*(y_j) + \sum_i \sum_j p_D^*(x, y_j) \log p_D^*(y_j|x_i)] \end{aligned} \tag{4.14}$$

この時 D と D^* の観測前確率 H_0 は同値であると考えられるので以下のように変換できる。

$$I(D,x) - I(D^*,x) = [+\sum_i \sum_j p_D(x,y_j) \log p_D(y_j|x_i)] - [\sum_i \sum_j p^*_D(x,y_j) \log p^*_D(y_j|x_i)] \tag{4.15}$$

この時のエントロピーの減少量はつまりは期待損失の減少量とも考えられる。これを期待損失の値として利用する事が可能である。

$$\begin{aligned} \tilde{E}_{\hat{P}_{D^*}} &= -\frac{1}{|\mathcal{P}|}[I(D,x) - I(D^*,x)] \\ &= -\frac{1}{|\mathcal{P}|}[\sum_i \sum_j p_D(x,y_j) \log p_D(y_j|x_i)] \\ &\quad + \frac{1}{|\mathcal{P}|}[\sum_i \sum_j p^*_D(x,y_j) \log p^*_D(y_j|x_i)] \end{aligned} \tag{4.16}$$

ここで必要となるのは事例 (x^*, y^*) 追加後の損失の低下量、つまり情報エントロピーの減少量であり追加前の情報エントロピーは一定である為省略する事ができる。

$$\tilde{E}_{\hat{P}_{D^*}} = \frac{1}{|\mathcal{P}|}[\sum_i \sum_j p^*_D(x,y_j) \log p^*_D(y_j|x_i)] \tag{4.17}$$

これが実際に利用する期待損失となる。

4.2.5 逐次的な学習

期待損失のアルゴリズムは追加候補である x の回数だけ識別リストの学習が必要となる。単純に計算を行うと計算コストが高くなるため、逐次的な処理を行い計算コスト低下を図る。

図 4.4 からわかるように期待損失の導出では訓練データ D と、それに事例を追加した訓練データ $(D + (x、y))$ に対した同じ処理を行なっている。

訓練データ D に対する識別規則学習は既にされており、訓練データ D と訓練データ $(D + (x、y))$ との差異は小量である。

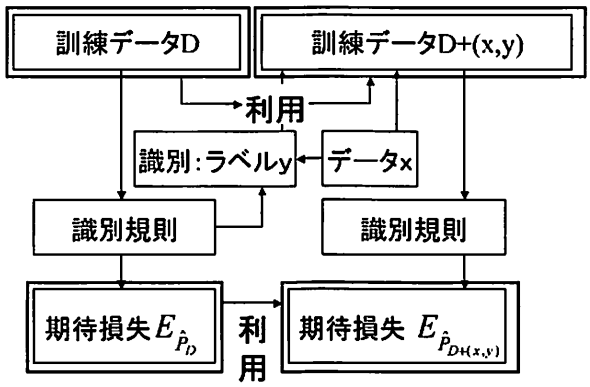


図 4.4: 逐次的な学習

よって訓練データ D の結果を用い、それに事例 (x, y) と関わる部分のみを修正する事により訓練データ $(D + (x、y))$ の識別規則を学習する。

また、期待損失の計算においても同様の処理ができる。期待損失の計算にはテストデータを識別した時の識別の証拠が持つ予測力の値を利用する。ここで訓練データ D によるテストデータの識別結果は既知であり、$D + (x, y)$ からの識別規則による結果と比較すると事例 x に含まれる証拠に対してのみ変化する。よって、その証拠に対する識別のみを行なえば識別と予測力の計算をする事ができる。

これによって大幅な計算量の低下を見込む事ができる。

4.3 期待損失手法の改良

期待損失の手法を単純に実装すると QBC よりも精度が落ちる [5]。この原因を探る為、実験結果をトレースしてみた結果、QBC と期待損失とで追加候補の選出に違いが出ている事が分かった。

以下は同音異義語「さいけん」について能動学習を行なった結果の上位 5 である。

QBC :

1:e1=偽造 e2=が e3=の-偽造 e4=が-埼玉 e5=偽造 e5=ワリコー e5=債 e5=埼玉 e5=県内 e5=東京 債券

2:e1=（ e2=担保 e3=ロンバードレート-（ e4=担保-貸付 e5=ロンバードレート

e5=% e5=5 e5=担保 e5=貸付 e5=金利 債券

3:e1=に e2=相場 e3=背景-に e4=相場-の e5=背景 e5=回復 e5=景気 e5=相場 e5=急騰 e5=受ける 債券

4:e1=は e2=の e3=根拠-は e4=の-動き e5=根拠 e5=動き 債券

5:e1=の e2=は e3=米国-の e4=は-昨年 e5=米国 e5=昨年 e5=十月 e5=天井 債券

期待損失:

1:e1=、 e2=、 e3=為替-、 e4=、-株 e5=為替 e5=こと e5=奇妙 e5=株 e5=金融 e5=市場 債券

2:e1=、 e2=の e3=株式-、 e4=の-トリプル e5=株式 e5=ドル e5=裏 e5=トリプル e5=安 e5=市場 債券

3:e1=、 e2=の e3=株式-、 e4=の-トリプル e5=株式 e5=ドル e5=決裂 e5=トリプル e5=暴落 e5=市場 債券

4:e1=不良 e2=が e3=いったい-不良 e4=が-いくら e5=不良 e5=ある e5=の e5=不良 債権

5:e1=共同 e2=買取 e3=た-共同 e4=買取-機構 e5=共同 e5=する e5=設立 e5=買取 e5=機構 e5=不良 債権

それぞれの証拠を素の訓練データから作成した決定リストと照らし合わせその単語の予測力を調べると表 4.1 のようであった。

QBC	期待損失
1:	1:
e5=東京 債券 0.857143 6 債券 6	e5=市場 債券 0.941176 16 債券 16
e2=が 債券 0.8 4 債券 4	e5=金融 債権 0.875 7 債権 7
e5=埼玉 債券 0.5 1 債券 1	e5=こと 債券 0.8 4 債券 4
e5=債 債券 0.5 1 債券 1	e1=、 債券 0.727273 10 債券 8 債権 2
e5=県内 債券 0.5 1 債券 1	e5=市 債権 0.5 1 債権 1
e5=偽造 債券 0.5 1 債券 1	e5=為替 債券 0.5 1 債券 1
e5=ワリコー 債券 0.5 1 債券 1	e2=、 債券 0.5 1 債券 1
e4=が-埼玉 債券 0.5 1 債券 1	
e3=の-偽造 債券 0.5 1 債券 1	
e1=偽造 債券 0.5 1 債券 1	
2:	2:
e5=金利 債券 0.9 9 債券 9	e5=市場 債券 0.941176 16 債券 16
e5=貸付 債券 0.666667 2 債券 2	e1=、 債券 0.727273 10 債券 8 債権 2
e5=ロンバードレート 債券 0.666667 2 債券 2	e5=市 債権 0.5 1 債権 1
e4=担保-貸付 債券 0.666667 2 債券 2	e5=ドル 債券 0.5 1 債券 1
e3=ロンバードレート-（ 債券 0.666667 2 債券 2	e2=の 債券 0.478261 22 債券 11 債権 11
e2=担保 債券 0.666667 2 債券 2	
e1=（ 債券 0.6 4 債券 3 債権 1	
e5=担保 債券 0.5 3 債券 2 債権 1	
e5=５ 債券 0.5 1 債券 1	
e5=% 債券 0.5 1 債券 1	

QBC	期待損失
3:	3:
e5=相場 債券 0.875 7 債券 7	e5=市場 債券 0.941176 16 債券 16
e2=相場 債券 0.875 7 債券 7	e1=、 債券 0.727273 10 債券 8 債権 2
e4=相場-の 債券 0.833333 5 債券 5	e5=暴落 債券 0.5 1 債券 1
e5=急騰 債券 0.666667 2 債券 2	e5=市 債権 0.5 1 債権 1
e5=背景 債券 0.5 1 債券 1	e5=ドル 債券 0.5 1 債券 1
e5=受ける 債券 0.5 1 債券 1	e2=の 債券 0.478261 22 債券 11 債権 11
e5=回復 債券 0.5 1 債券 1	
e3=背景-に 債券 0.5 1 債券 1	
e1=に 債券 0.5 1 債券 1	
e5=景気 債券 0.333333 2 債券 1 債権 1	
4:	4:
e5=動き 債券 0.5 1 債券 1	e5=不良 債権 0.970588 33 債権 33
e5=根拠 債券 0.5 1 債券 1	e1=不良 債権 0.970588 33 債権 33
e4=の-動き 債券 0.5 1 債券 1	e2=が 債券 0.8 4 債券 4
e3=根拠-は 債券 0.5 1 債券 1	e5=の 債券 0.5 1 債券 1
e1=は 債券 0.5 3 債券 2 債権 1	
e2=の 債券 0.478261 22 債券 11 債権 11	
5:	5:
e5=米国 債券 0.8 4 債券 4	e5=不良 債権 0.970588 33 債権 33
e3=米国-の 債券 0.75 3 債券 3	e5=買取 債権 0.666667 2 債権 2
e5=十月 債券 0.666667 2 債券 2	e5=共同 債権 0.666667 2 債権 2
e1=の 債券 0.666667 11 債券 8 債権 3	e5=設立 債権 0.5 1 債権 1
e5=天井 債券 0.5 1 債券 1	e5=機構 債権 0.5 1 債権 1
e5=昨年 債券 0.5 1 債券 1	e4=買取-機構 債権 0.5 1 債権 1
e4=は-昨年 債券 0.5 1 債券 1	e2=買取 債権 0.5 1 債権 1
e2=は 債権 0.5 3 債券 1 債権 2	e1=共同 債権 0.5 1 債権 1
e5=十 債券 0.428571 6 債券 3 債権 3	e5=する 債券 0.454545 10 債券 5 債権 5

表 4.1: 予測力

QBC ではそれぞれのクラスに対する予測力が同程度の証拠や低頻度の証拠が含まれている事例が選ばれているが、期待損失の物は片方の予測力が極端に高いものが含まれる事例が選ばれている。このような結果となる原因は、訓練データ中で頻度が多い証拠はテストデータ中にも多く含まれる可能性が高い為、識別する時のヒット率が高くなる事により、一度の損失率の低下が小量でも最終的な期待損失に与える影響力が高くなる事と考えられる。

これは低頻度の証拠への修正パラメータ α によって高頻度の証拠でも予測力が100% にならない事が根本の原因と考える。

それを確かめるため訓練データ中の出現回数が多い単語5個と出現回数が一度のみの物からランダムに5個を抽出してテストデータ中の出現回数をカウントし表 4.2 に表す。

高頻度	回数	低頻度	回数
e5=不良	3250	e5=物件	10
e1=不良	3204	e4=国-側	0
e5=市場	163	e3=社長-	0
e2=市場	113	e2=引き受け	2
e3=の-不良	1133	e1=国内	0

表 4.2: 予測力

パラメータ α を 1 として。それぞれの証拠が一つ追加された時の期待損失の減少量は

「e5=不良」と「e1=不良」が 0.000840336
「e5=市場」0.00326797 、「e2=市場]0.00367647

これらのトレーニングデータ全体を通しての期待損失減少量は

「e5=不良」0.000840336×3250=2.73109
「e1=不良」0.000840336×3204=2.69244
「e5=市場」0.00326797×163=0.53268
「e2=市場] 0.00367647×113=0.415441

頻度が 1 の証拠の期待損失減少量が 0.166667 となるのでこれらは比較的大きな値という事である。

しかし、高頻度証拠に事例の追加を行なっても曖昧性の解消が行なわれているとは考え難い。例えば「e1=不良」はテストデータ中に 33 回現れておりその全てがクラス「債権」である事を指し示しているとする。現実的に考えてこの証拠が現れた時にクラスが「債権」である事は確実であり、これを補強して 34 個目の証拠を追加したとしても実際に識別を行なう上では何の影響も与えず、曖昧性が減少したとは考えられない。

つまりこの既に十分な確からしさを持つ証拠が期待損失低下に有効だと判別される問題が発生する。

この問題を回避するには確からしさの高いの証拠に対しては期待損失計算を行なわなければ良い。

つまり一定以上の出現頻度を持ち、それが指し示すクラスが全て同一のものであった場合、その証拠に事例を追加したとしても損失の減少量は 0 とする。

また、Default もこのような証拠と同様の傾向をもつと考えられるため、Default の損失の減少量も 0 と考える。

このような改良を施した結果を QBC と比較し、その有効性を判断する。

第5章　実験

本手法の有効性を確認するために、同音異義語の識別問題への適用を試みた。有効度の目安としてQBCでの実装も同時に行ないその結果を比較する。

同音異義語	クラス1	クラス2
さいけん	「債券」	「債権」
かいほう	「解放」	「開放」
きょうちょう	「協調」	「強調」
じしん	「自身」	「自信」
たいがい	「体外」	「対外」
うんこう	「運航」	「運行」
かてい	「過程」	「課程」
しょくりょう	「食糧」	「食料」
しょうがい	「障害」	「傷害」

表 5.1: 同音異義語のクラス定義

実験に用いる同音異義語は表5.1の9単語とする。

今回はそれぞれのクラスの事例50事例にラベルを付けした計100事例をトレーニングデータとした。

テストデータはそれぞれの事例が現れる文を95年度の毎日新聞の記事データから抽出して用いる。それぞれのクラスのテストデータの事例数は表5.2のようになる。

それぞれの手法で曖昧性解消に有効であると判断された事例の上位5事例にラベルをつけ、テストデータに追加する動作を16回まで繰り返した。決定リストのパラメータαは0.5とする。表5.3、5.4、5.5に学習後のトレーニングデータを用いてテストデータを識別した時の正解率をのせる。回数が0の時は能動学習を行なう前のトレーニングデータで識別を行なった時の正解率である。

同音異義語	クラス１	クラス２	総数
さいけん	727	5228	5955
かいほう	3725	2764	6489
きょうちょう	1952	9182	11134
じしん	5441	3476	8917
たいがい	372	889	1261
うんこう	623	675	1298
かてい	2285	406	2691
しょくりょう	2387	1903	4290
しょうがい	6661	760	7241

表 5.2: テストデータ

表 5.3、5.4、5.5 をグラフで表したのが図 5.1、5.2、5.3 である。

	さいけん		かいほう		きょうちょう	
回数	期待損失	QBC	期待損失	QBC	期待損失	QBC
0	0.684688	0.684688	0.767951	0.767951	0.925191	0.925191
1	0.793989	0.794157	0.768567	0.767951	0.912528	0.912528
2	0.794157	0.790967	0.678891	0.782897	0.891154	0.913875
3	0.855776	0.709872	0.680431	0.782435	0.912528	0.913875
4	0.861484	0.791471	0.682435	0.77812	0.922586	0.913875
5	0.871894	0.846709	0.76718	0.781048	0.894746	0.913875
6	0.884486	0.846709	0.681972	0.780894	0.907409	0.913875
7	0.887508	0.846709	0.77057	0.782435	0.898518	0.913875
8	0.885158	0.842176	0.681818	0.783051	0.899506	0.913875
9	0.887508	0.853425	0.675655	0.790139	0.900763	0.913875
10	0.888012	0.861652	0.77057	0.800616	0.922586	0.913875
11	0.833781	0.858966	0.675809	0.801233	0.918186	0.913875
12	0.88818	0.858966	0.675809	0.790909	0.900763	0.913875
13	0.770651	0.86266	0.769954	0.80755	0.918186	0.917288
14	0.888516	0.86266	0.676425	0.803698	0.919713	0.917288
15	0.888684	0.863163	0.676271	0.803698	0.900943	0.915941
16	0.889691	0.861988	0.7698	0.811248	0.919713	0.915941

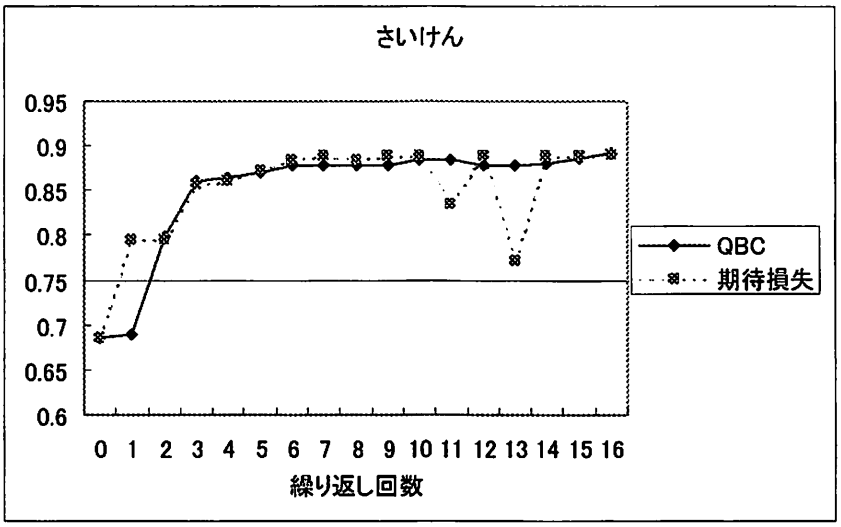
表 5.3: 実験 1-1

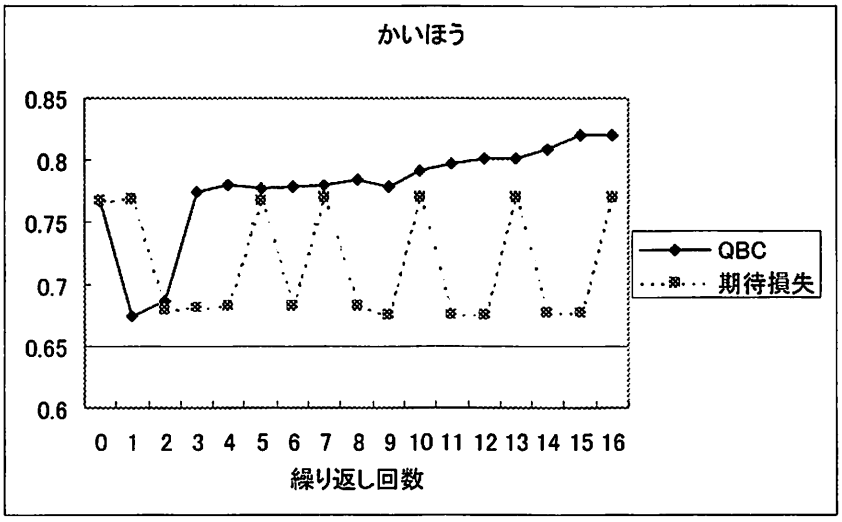
	じしん		たいがい		うんこう	
回数	期待損失	QBC	期待損失	QBC	期待損失	QBC
0	0.647791	0.647791	0.473851	0.473851	0.528099	0.528099
1	0.836959	0.833931	0.683835	0.683835	0.515012	0.515012
2	0.837632	0.834044	0.744057	0.683835	0.541186	0.515012
3	0.676048	0.835277	0.751189	0.704437	0.543495	0.550423
4	0.842229	0.835277	0.751189	0.729794	0.545804	0.552733
5	0.67717	0.843911	0.751189	0.729002	0.545804	0.552733
6	0.842229	0.843911	0.749604	0.729002	0.545804	0.553503
7	0.672909	0.857031	0.752773	0.735341	0.548114	0.555812
8	0.673133	0.866674	0.750396	0.747227	0.543495	0.568899
9	0.847612	0.867011	0.749604	0.747227	0.550423	0.568899
10	0.673133	0.853218	0.752773	0.747227	0.550423	0.568899
11	0.673133	0.848957	0.752773	0.754358	0.555042	0.569669
12	0.676273	0.849518	0.725832	0.754358	0.555812	0.588145
13	0.852097	0.813075	0.755151	0.754358	0.573518	0.585835
14	0.677394	0.820363	0.727417	0.754358	0.571978	0.584296
15	0.849069	0.816102	0.756735	0.754358	0.574288	0.585835
16	0.846715	0.816102	0.756735	0.754358	0.554273	0.585835

表 5.4: 実験 1-2

	かてい		しょくりょう		しょうがい	
回数	期待損失	QBC	期待損失	QBC	期待損失	QBC
0	0.973626	0.973626	0.485435	0.485435	0.529688	0.529688
1	0.973626	0.973626	0.540667	0.48893	0.831262	0.845347
2	0.973626	0.973626	0.551853	0.489396	0.846313	0.845485
3	0.593239	0.973626	0.556048	0.487998	0.919635	0.845485
4	0.973626	0.973997	0.496854	0.487532	0.917564	0.845485
5	0.593239	0.973997	0.485202	0.487532	0.926263	0.845485
6	0.97474	0.973997	0.484969	0.487532	0.928335	0.845485
7	0.593239	0.975854	0.496621	0.486134	0.928611	0.845485
8	0.97474	0.975854	0.485668	0.486134	0.928611	0.845485
9	0.975483	0.975854	0.485668	0.486134	0.930682	0.845485
10	0.618128	0.976226	0.485901	0.486833	0.928611	0.845485
11	0.975483	0.976226	0.487066	0.510371	0.928611	0.928335
12	0.618128	0.978455	0.490096	0.509671	0.928611	0.928335
13	0.975483	0.978455	0.490329	0.510371	0.928749	0.928335
14	0.618128	0.978455	0.488464	0.509671	0.588788	0.928335
15	0.975854	0.978455	0.48893	0.51107	0.66211	0.928335
16	0.618128	0.978455	0.490795	0.509904	0.93082	0.928335

表 5.5: 実験 1-3


さいけん
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
繰り返し回数
QBC
期待損失


かいほう
0.85
0.8
0.75
0.7
0.65
0.6
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
繰り返し回数
QBC
期待損失

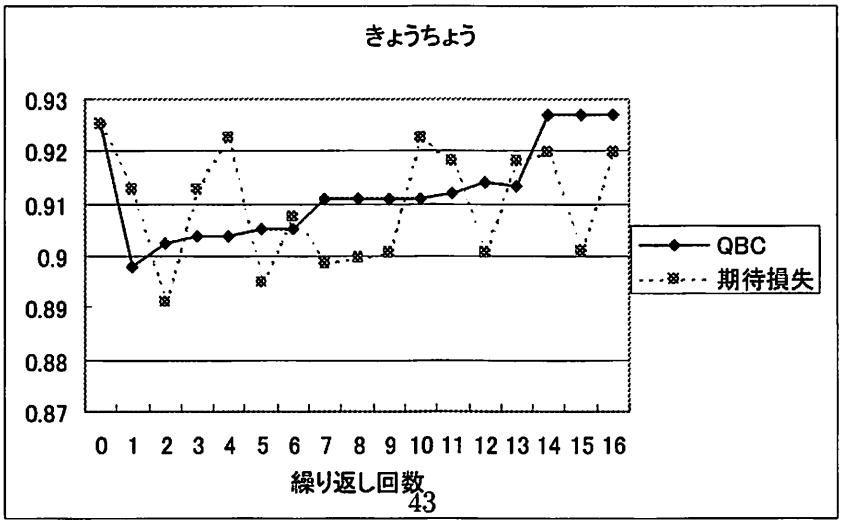

きょうちょう
0.93
0.92
0.91
0.9
0.89
0.88
0.87
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
繰り返し回数
QBC
期待損失

図 5.1: 実験 1-1

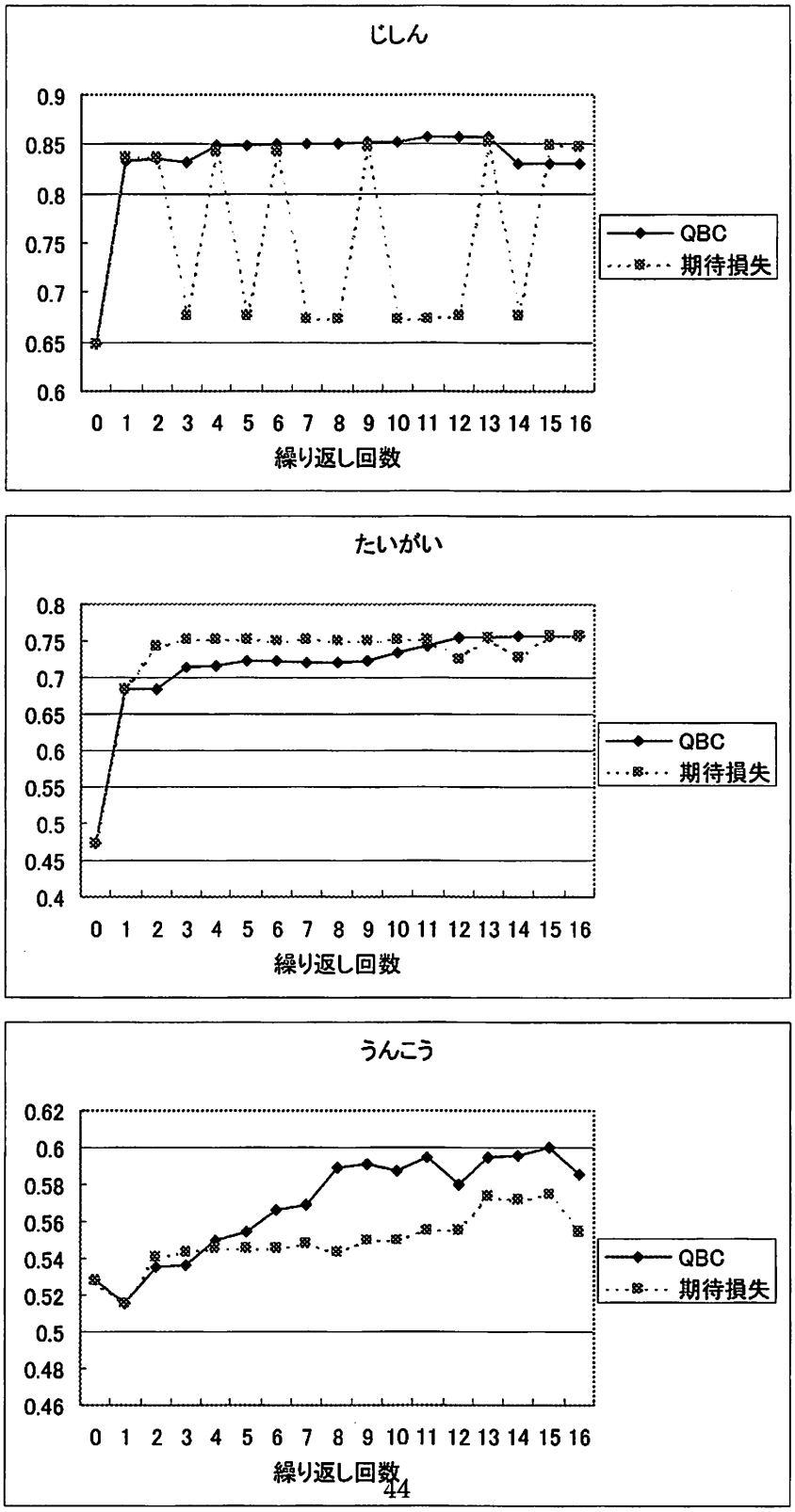
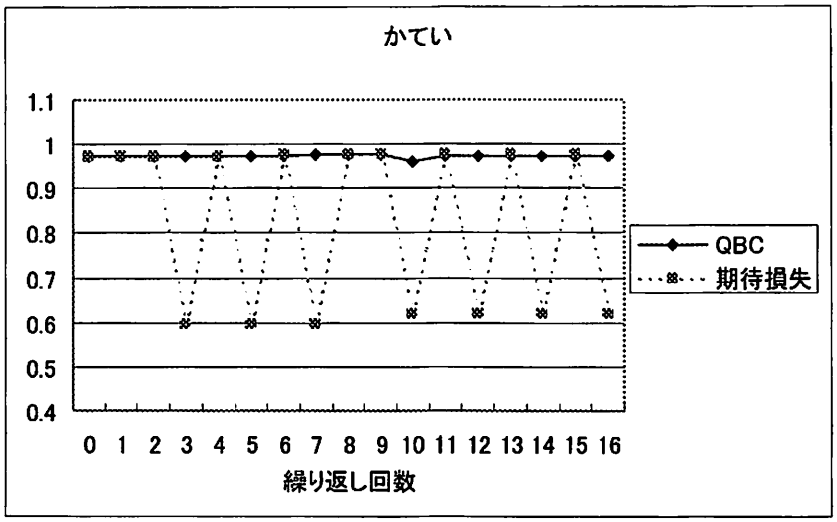
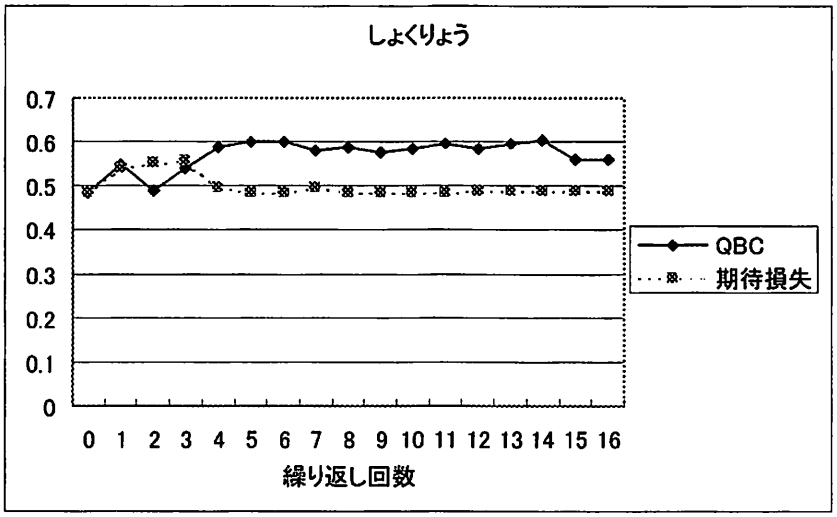

じしん
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
繰り返し回数
QBC
期待損失
たいがい
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
繰り返し回数
QBC
期待損失
うんこう
0.62
0.6
0.58
0.56
0.54
0.52
0.5
0.48
0.46
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
繰り返し回数
QBC
期待損失

図 5.2: 実験 1-2


かてい
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
繰り返し回数
QBC
期待損失


しょくりょう
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
繰り返し回数
QBC
期待損失

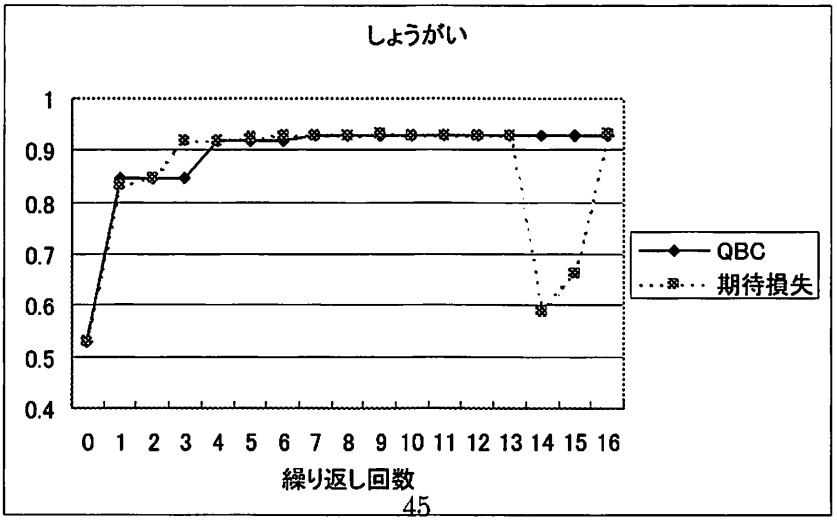

しょうがい
1
0.9
0.8
0.7
0.6
0.5
0.4
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
繰り返し回数
QBC
期待損失

図 5.3: 実験 1-3

次に、期待損失を用いた手法に改良を施し、同じ実験を行なう。改良を施した期待損失を用いた手法とQBCとの実験の正解率が表5.6、5.7、5.8である。

図5.4、5.5、5.6はQBCと、改良前と改良後の期待損失を用いた手法の結果のグラフである。

	さいけん		かいほう		きょうちょう	
回数	期待損失	QBC	期待損失	QBC	期待損失	QBC
0	0.684688	0.684688	0.767951	0.767951	0.925191	0.925191
1	0.856111	0.794157	0.679815	0.767951	0.926807	0.912528
2	0.861484	0.790967	0.777966	0.782897	0.909564	0.913875
3	0.857623	0.709872	0.672419	0.782435	0.926807	0.913875
4	0.862324	0.791471	0.691371	0.77812	0.926807	0.913875
5	0.864674	0.846709	0.689214	0.781048	0.92555	0.913875
6	0.869543	0.846709	0.778582	0.780894	0.92555	0.913875
7	0.877099	0.846709	0.798613	0.782435	0.925819	0.913875
8	0.890363	0.842176	0.800616	0.783051	0.926179	0.913875
9	0.885829	0.853425	0.801541	0.790139	0.926807	0.913875
10	0.886165	0.861652	0.818336	0.800616	0.927975	0.913875
11	0.89137	0.858966	0.818182	0.801233	0.926807	0.913875
12	0.888348	0.858966	0.724037	0.790909	0.926807	0.913875
13	0.88734	0.86266	0.724191	0.80755	0.926987	0.917288
14	0.886837	0.86266	0.725732	0.803698	0.931118	0.917288
15	0.891034	0.863163	0.741911	0.803698	0.934531	0.915941
16	0.891538	0.861988	0.748536	0.811248	0.934531	0.915941

表 5.6: 実験 2-1

	じしん		たいがい		うんこう	
回数	期待損失	QBC	期待損失	QBC	期待損失	QBC
0	0.647791	0.647791	0.473851	0.473851	0.528099	0.528099
1	0.834044	0.833931	0.710777	0.683835	0.541955	0.515012
2	0.658444	0.834044	0.57607	0.683835	0.561201	0.515012
3	0.662593	0.835277	0.723455	0.704437	0.533487	0.550423
4	0.663714	0.835277	0.715531	0.729794	0.539646	0.552733
5	0.693317	0.843911	0.715531	0.729002	0.541955	0.552733
6	0.69556	0.843911	0.751189	0.729002	0.555042	0.553503
7	0.700269	0.857031	0.726624	0.735341	0.579677	0.555812
8	0.70139	0.866674	0.555468	0.747227	0.582756	0.568899
9	0.701951	0.867011	0.667195	0.747227	0.56428	0.568899
10	0.701951	0.853218	0.776545	0.747227	0.590454	0.568899
11	0.705876	0.848957	0.778922	0.754358	0.589684	0.569669
12	0.713949	0.849518	0.720285	0.754358	0.588915	0.588145
13	0.722584	0.813075	0.711569	0.754358	0.60662	0.585835
14	0.740637	0.820363	0.723455	0.754358	0.60893	0.584296
15	0.744898	0.816102	0.79477	0.754358	0.598152	0.585835
16	0.75028	0.816102	0.724247	0.754358	0.61047	0.585835

表 5.7: 実験 2-2

	かてい		しょくりょう		しょうがい	
回数	期待損失	QBC	期待損失	QBC	期待損失	QBC
0	0.973626	0.973626	0.485435	0.485435	0.529688	0.529688
1	0.599554	0.973626	0.559543	0.48893	0.853908	0.845347
2	0.973626	0.973626	0.573992	0.489396	0.853908	0.845485
3	0.973254	0.973626	0.574691	0.487998	0.851698	0.845485
4	0.973254	0.973997	0.576556	0.487532	0.851975	0.845485
5	0.973626	0.973997	0.538103	0.487532	0.851975	0.845485
6	0.973626	0.973997	0.539268	0.487532	0.852803	0.845485
7	0.978083	0.975854	0.563738	0.486134	0.834438	0.845485
8	0.978083	0.975854	0.574458	0.486134	0.835957	0.845485
9	0.980312	0.975854	0.570729	0.486134	0.846727	0.845485
10	0.981426	0.976226	0.570729	0.486833	0.846727	0.845485
11	0.981798	0.976226	0.564437	0.510371	0.920602	0.928335
12	0.981798	0.978455	0.563738	0.509671	0.919912	0.928335
13	0.983284	0.978455	0.547192	0.510371	0.915217	0.928335
14	0.983284	0.978455	0.547658	0.509671	0.919912	0.928335
15	0.983284	0.978455	0.551853	0.51107	0.848246	0.928335
16	0.983284	0.978455	0.609415	0.509904	0.922673	0.928335

表 5.8: 実験 2-3

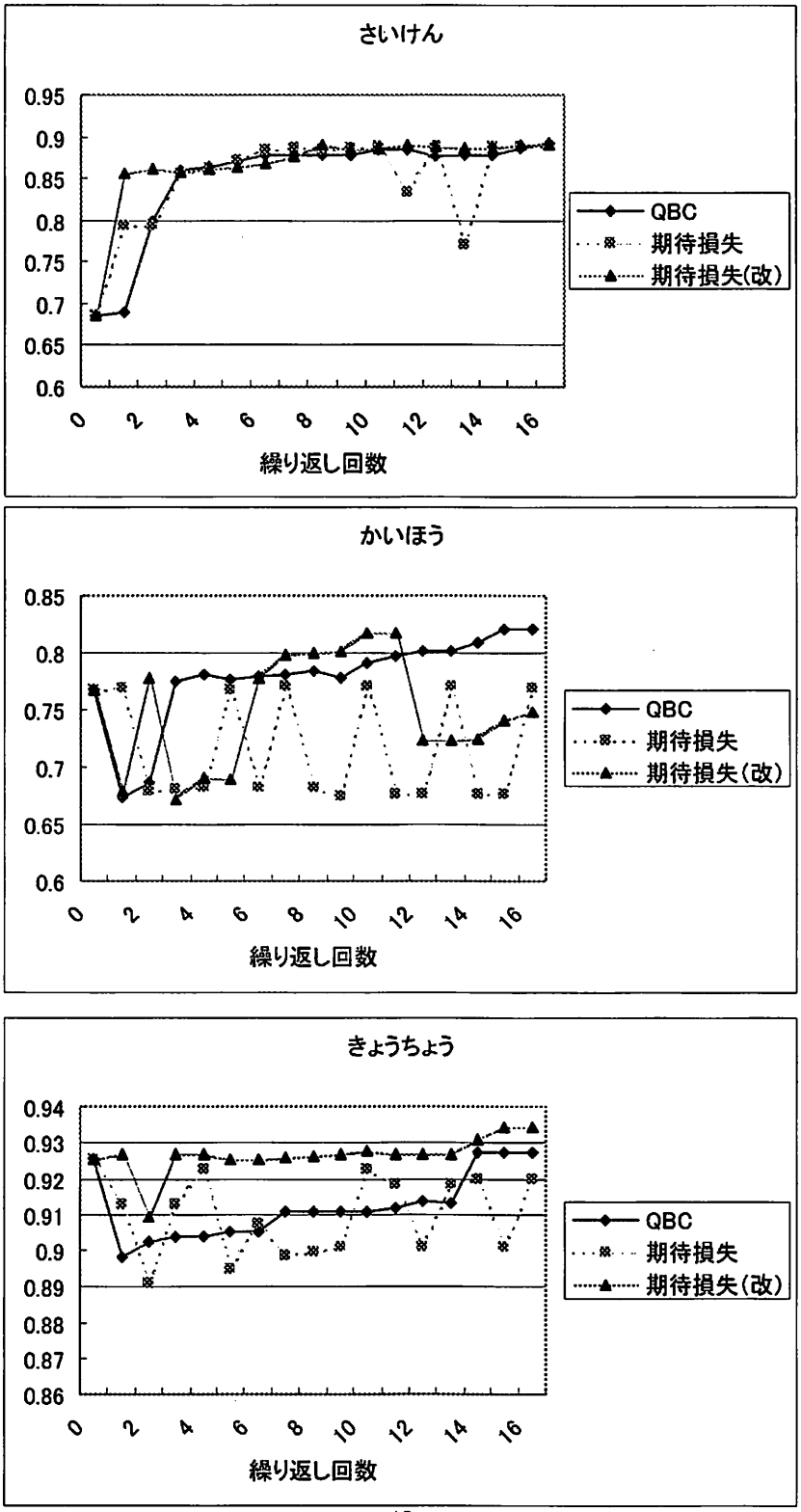
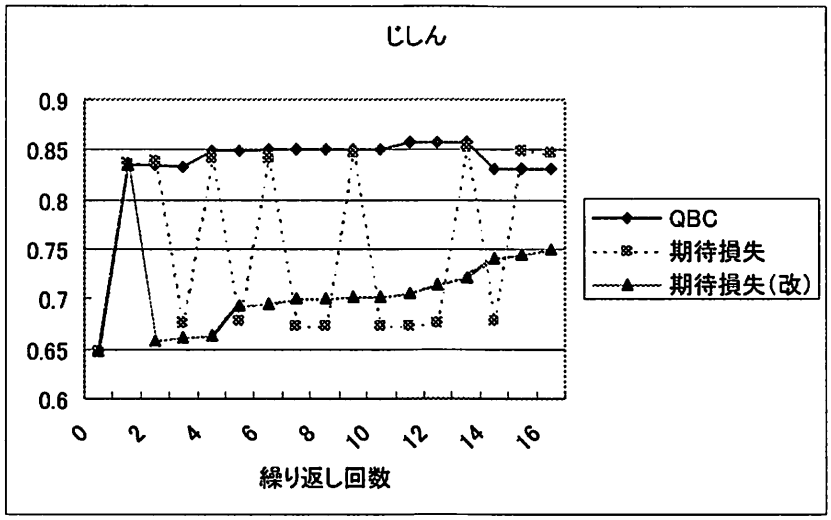
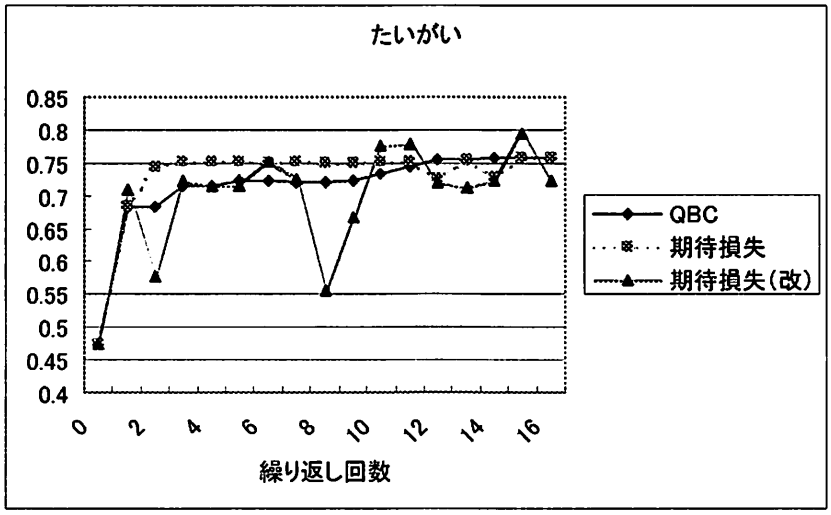

さいけん
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失
期待損失(改)
かいほう
0.85
0.8
0.75
0.7
0.65
0.6
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失
期待損失(改)
きょうちょう
0.94
0.93
0.92
0.91
0.9
0.89
0.88
0.87
0.86
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失
期待損失(改)

図 5.4: 実験 2-1


じしん
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失
期待損失(改)


たいがい
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失
期待損失(改)

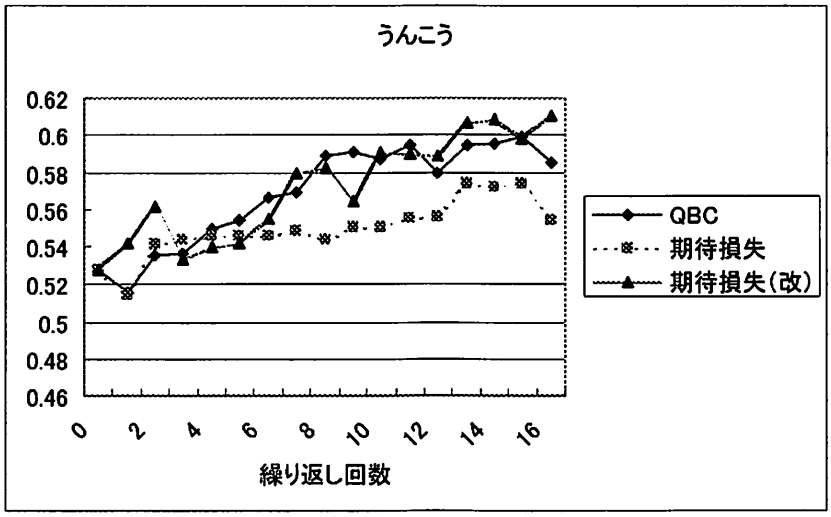

うんこう
0.62
0.6
0.58
0.56
0.54
0.52
0.5
0.48
0.46
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失
期待損失(改)

図 5.5: 実験 2-2

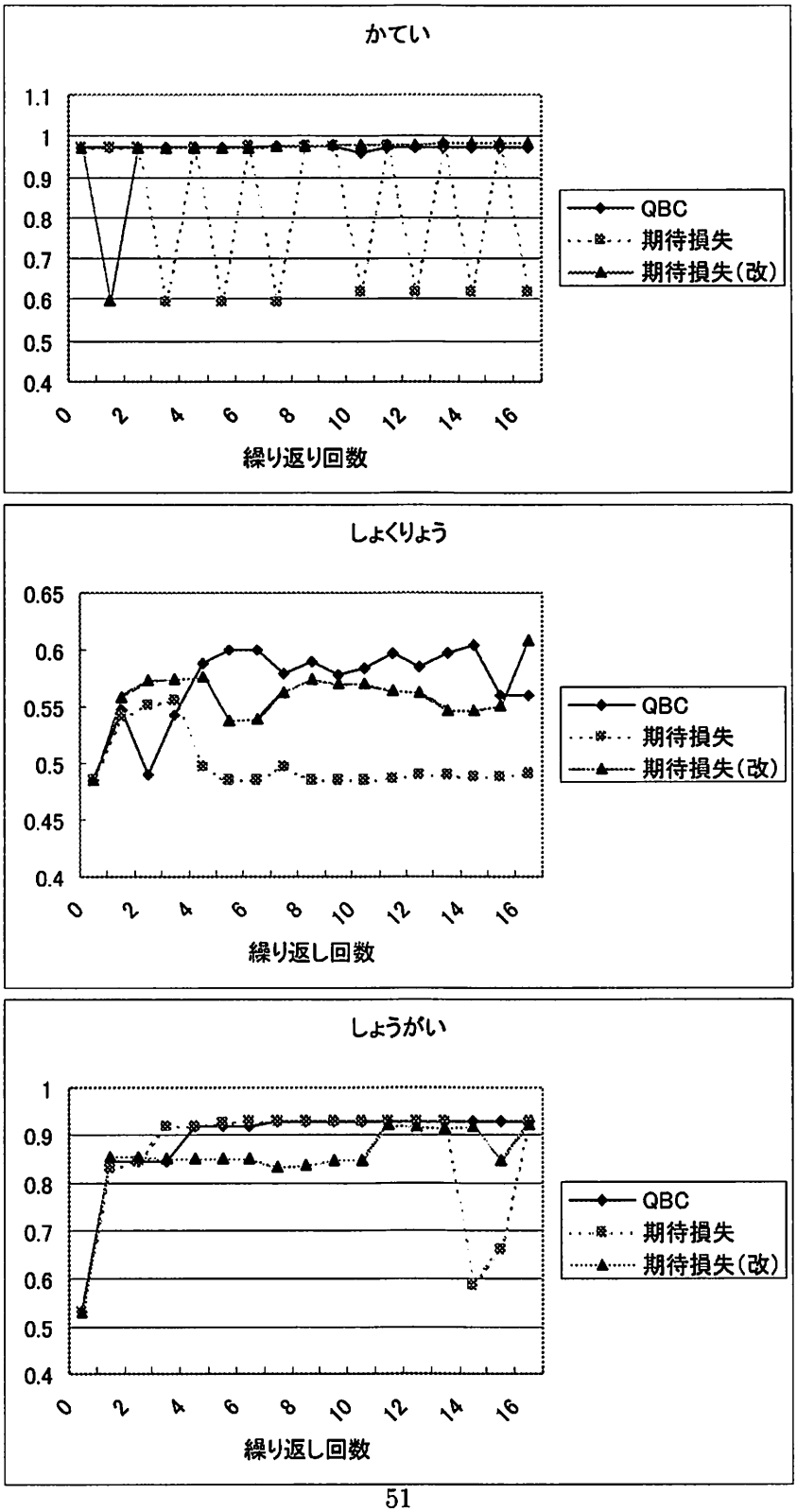

かてい
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0 2 4 6 8 10 12 14 16
繰り返り回数
QBC
期待損失
期待損失(改)
しょくりょう
0.65
0.6
0.55
0.5
0.45
0.4
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失
期待損失(改)
しょうがい
1
0.9
0.8
0.7
0.6
0.5
0.4
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失
期待損失(改)

図 5.6: 実験 2-3

第6章　考察

実験の結果「さいけん」、「きょうちょう」、「かてい」では期待損失を用いた手法の方がQBCより良い結果となった。逆に「じしん」、「しょくりょう」、「しょうがい」ではQBCの方がよい結果となった。これら以外の同音異義語ではどちらがよいとも言えない結果となった。

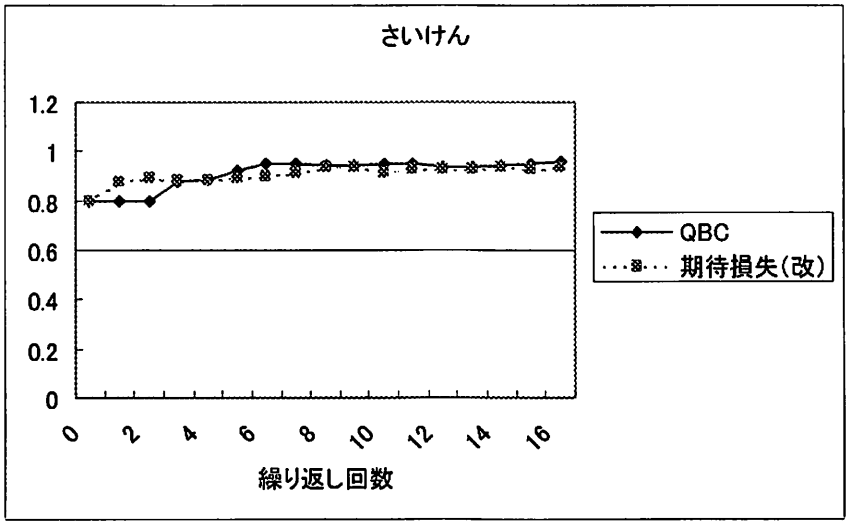
正解率の表をみると正解率が急激に変化する減少が所々にみられる。例えば「かてい」の改良前の期待損失を用いた手法での実験結果などはその現象がはっきりと見てとれる。これは、Defaultのラベルが変わる事によって引き起こされていると考えられる。実験に用いた同音異義語の中で一番この現象が顕著に現れているのが改良前の期待損失を用いた手法を用いた「かてい」の結果である。そこで、「かてい」に対して改良前の期待損失を用いた手法による能動学習の過程で作成されたそれぞれの決定リストのDefaultのラベルを調べた。結果、決定リストの正解率が9割強の決定リストではDefaultのラベルが「過程」であり、6割前後の決定リストではラベルが「課程」であった。さらにこれらの決定リストを用いた時、テストデータの判定においてDefaultで判定される割合を調べる。

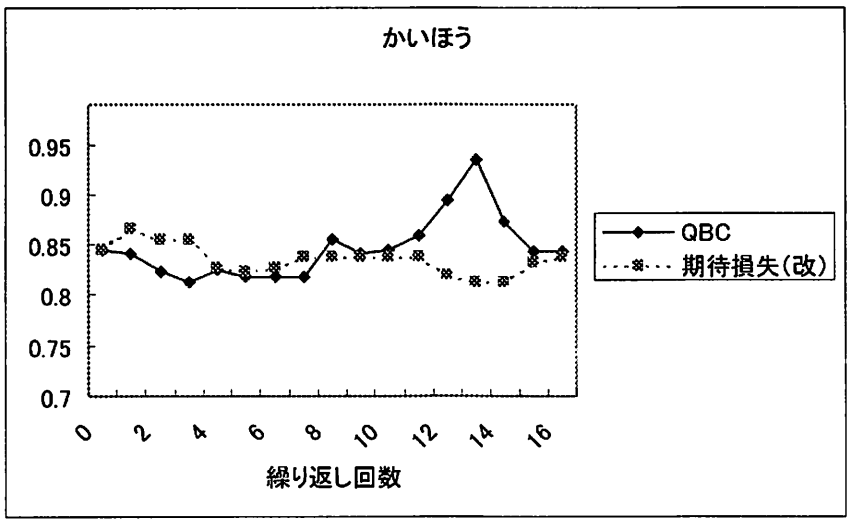
内容	個数	全体に占める%
テストデータの総数	2691	100
Default 以外で判定された個数	1551	58
Default 以外で判定されかつ正解	1534	57
Default 以外で判定されかつ不正解	17	1
Default で判定された個数	1140	42
Default で判定されかつ正解	1087	40
Default で判定されかつ不正解	53	2

表 6.1: 実験結果「かてい」

表 6.1 は同音異義語「かてい」の実験で用いたテストデータを素の訓練データで識別した時の判定結果の内分けである。つまり、「かてい」の識別においては、識別の半分近くは Default によって行なわれている事がわかる。ここで Default のラベルは「過程」である。よって正解率は 57 + 40 の 97% である。しかし、このラベルが「課程」となると Default の正解と不正解が逆転するので正解率は 57 + 2 の 58% まで下がる。つまり実験での正解率の急激な変化はこの Default のラベルの入れ替わりが原因である。

図 6.1、6.2、6.3 は Default 以外の証拠で識別されたテストデータの正解率である。


さいけん
1.2
1
0.8
0.6
0.4
0.2
0
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失(改)


かいほう
0.95
0.9
0.85
0.8
0.75
0.7
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失(改)

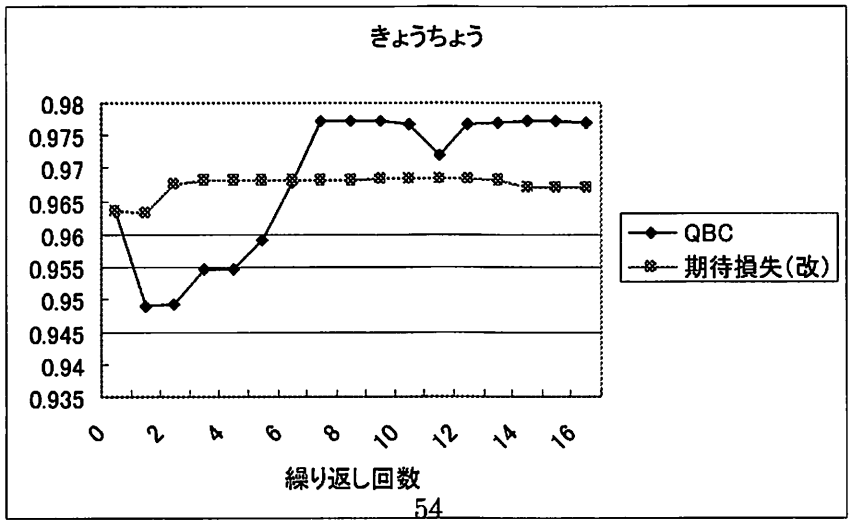
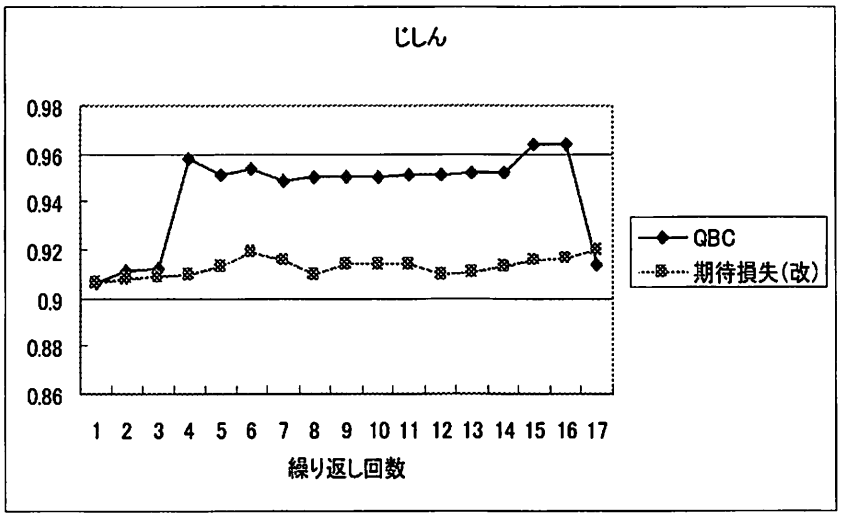
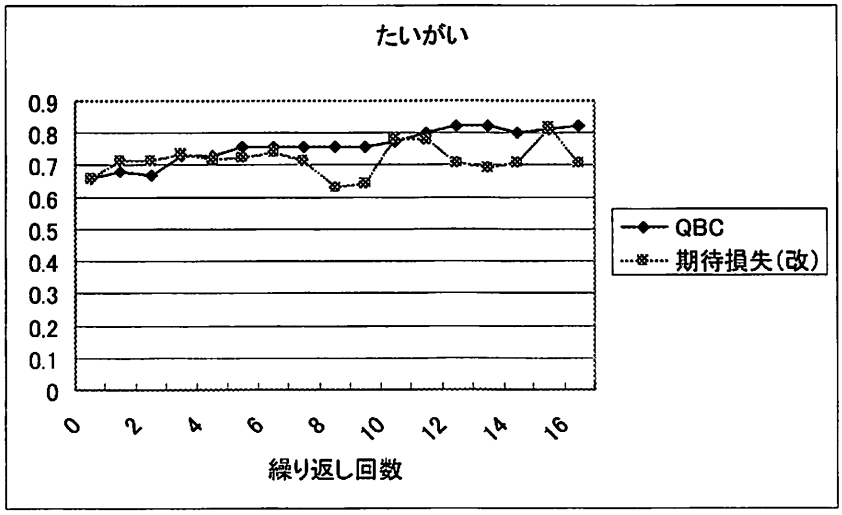

きょうちょう
0.98
0.975
0.97
0.965
0.96
0.955
0.95
0.945
0.94
0.935
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失(改)

図 6.1: 実験 3-1


じしん
0.98
0.96
0.94
0.92
0.9
0.88
0.86
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
繰り返し回数
QBC
期待損失(改)


たいがい
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失(改)

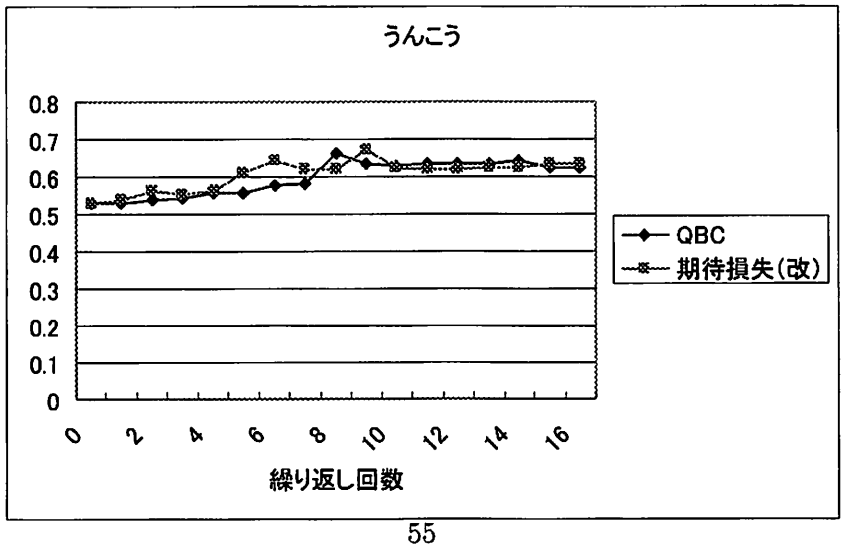
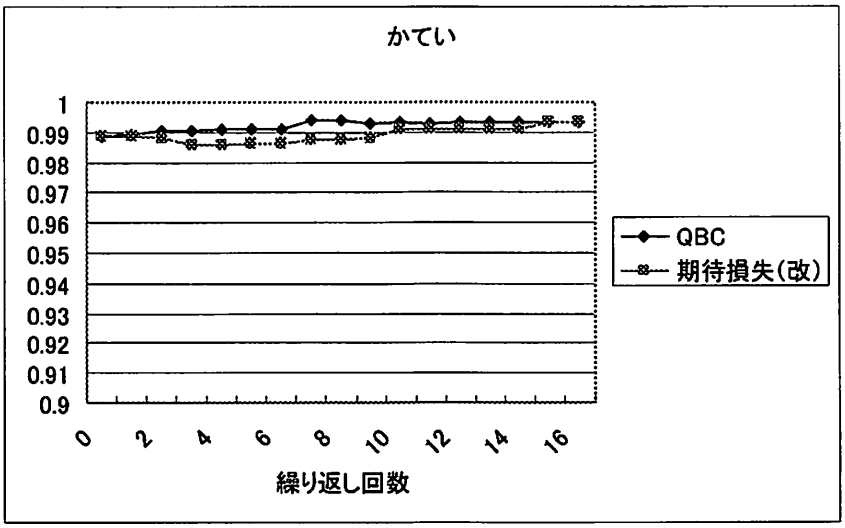
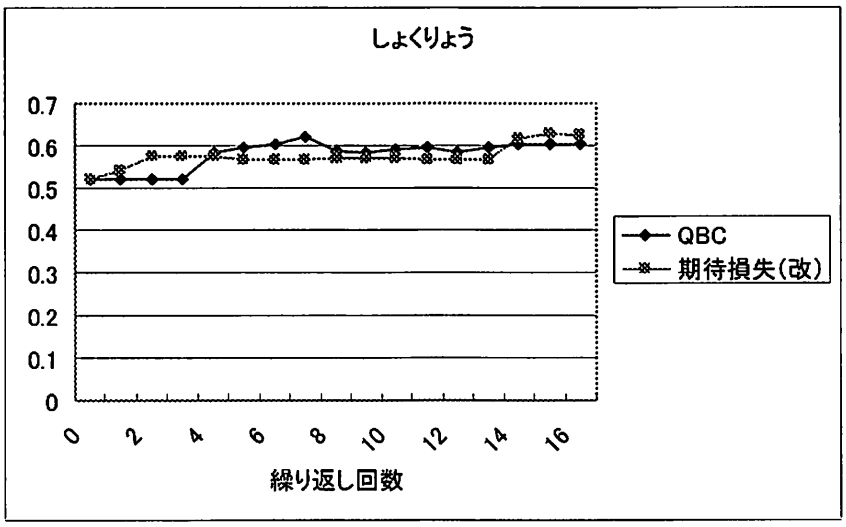

うんこう
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失(改)

図 6.2: 実験 3-2


かてい
1
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0.92
0.91
0.9
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失(改)


しょくりょう
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失(改)

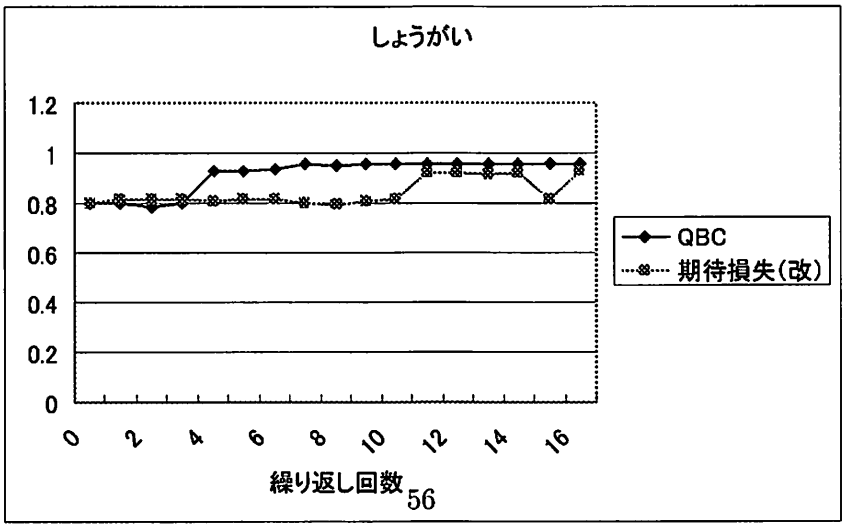

しょうがい
1.2
1
0.8
0.6
0.4
0.2
0
0 2 4 6 8 10 12 14 16
繰り返し回数
QBC
期待損失(改)

図 6.3: 実験 3-3

Default のラベルも能動学習によって最適なクラスが選択されることが期待できる。最も事例が多いクラスが Default の最適なラベルであり、事例数が多いと言う事は能動学習によって選ばれる追加事例のラベルもそのクラスのラベルである割合は他のクラスより多くなる。よって、実際に「さいけん」や「しょうがい」などでは一度目の学習で正解率が急上昇しそのまま正解率が維持される事から、学習によって Default のラベルが最適なクラスに固定されている事がわかる。しかし、一番割合の高いクラスと2番目以降のクラスとでその割合が近いほど Default のラベルが間違う確率が高くなる。実際、訓練データ中の事例のそれぞれのクラスの事例数が近い「うんこう」では Default のラベルの入れ替わりが激しい。「かいほう」や「じしん」などの結果より期待損失を用いた手法では QBC よりこの傾向が強いと考えられる。

表 6.2 は学習を 16 回繰り返した後の訓練データから作成された決定リスト中の事例のラベルの個数である。QBC では追加されるクラス偏りが見られるが、期待損失手法では偏りが少ない事が見てとれる。

事例の選択にクラスの大きさの影響を受けにくいと言う事は希少クラスの事例に対してきちんと学習できるという事であり、必ずしも悪い特性とはいえない。

しかし、実際に能動学習を行なった時に Default の判定によって正解率が落ちてしまうのは問題である。実際に能動学習を利用する場合、常に Default の影響が少なくなるまで学習を繰り返すとは限らないので Default のラベルは常に最適なクラスが選択される事が望まれる。

この Default のラベルについての問題は今後の課題とする。

さいけん	期待損失	債券:110	債権:150
	QBC	債券:108	債権:152
かいほう	期待損失	解放:125	開放:135
	QBC	解放:135	開放:125
きょうちょう	期待損失	協調:133	強調:127
	QBC	協調:90	強調:170
じしん	期待損失	自信:136	自身:124
	QBC	自信:83	自身:177
たいがい	期待損失	体外:125	対外:135
	QBC	対外:163	体外:97
うんこう	期待損失	運航:131	運行:129
	QBC	運航:121	運行:139
かてい	期待損失	過程:152	課程:108
	QBC	過程:181	課程:79
しょくりょう	期待損失	食料:129	食糧:131
	QBC	食料:130	食糧:130
しょうがい	期待損失	傷害:112	障害:148
	QBC	傷害:68	障害:192

表 6.2: Default のクラス分布

第7章　結論

本論文では期待損失を用いた能動学習の手法を決定リストに対して実装し、さらにその精度向上のため証拠の期待損失に対して改良を加えた。同音異義語9単語に対してこの手法と現在利用されている QBC の手法で実験を行なった。実験の結果、改良後の期待損失を用いた手法では QBC に近い精度の学習を行なう事ができた。しかし、現在の手法では Default のラベルとなるクラスが変わりやすく識別結果の正解率に不安が残る結果となった。どこで能動学習を修了させても Default のラベルが常に最適になるような手法の考案が今後の課題である。

謝辞

本研究の遂行および論文作成において多大な御助言及び御指導を賜りました 新納 浩幸 教官（茨城大学工学部システム工学科）に深い感謝の意を表します。また、その他様々な助言を頂きました 岩崎 唯史 教官（茨城大学工学部システム工学科）システム工学科計算機応用学科講座の教官の方々、同研究室の 藤井 丈明 氏（修士）正木 祐一 氏（修士）谷津 哲平 氏（修士）大北 高広 氏（学部）木元 俊（学部） 藤村 元彦（学部）茂木 啓悟（学部）に深く感謝致します。

参考文献

[1] Nicholas Roy　Andrew Mc Callum :''Toward Optimal Active Learning through Sampling Estimation of Error Reduction''(2001).

[2] David Yarowsky: ''Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French'',32th Annual Meeting of the Association for Computational Linguistics,pp.88-95,(1994).

[3] Seung,H.S.,Opper,M.,&Sompolinsky,H:Qiery by committee. Priceedings of the Fifth Annual ACM Workshop on Computational Learning Teeory,pp.287-294(1992).

[4] Breiman,L:Bagging predictors.Machine Learning,24,123-149(1996).

[5] 紺野憲一, 新納浩幸, 佐々木稔:決定リストと期待損失を用いた同音異義語識別規則の能動学習, 言語処理学会第 10 回年次大会,pp.757-760(2004).

[6] 新納浩幸,:誤りやすい同音異義語の抽出,
言語処理学会第自然言語処理研究会,126-1,pp.1-8(1998).

[7] 新納浩幸,:決定リストとアダブースを利用した訓練データ中の誤り検出,
言語処理学会第 7 回年次大会,pp.26-29(2001).

付録：プログラムリスト

付録A：決定リスト作成プログラム

メインプログラム：list.sh(bash)

訓練データからパラメータがαの決定リストを作成する。
入力:訓練データ パラメータα
出力:訓練データ+list

```
if [ -s "$1" ]
then
  li1 $1 $2
  sort "$1""l" >"$1""li"
  li2 "$1""li" $2
  sort +2 -r -n "$1""liz" >"$1""list"
  rm -f "$1""li" "$1""l" "$1""liz"
fi
```

サブプログラム1：li1(c++)

訓練データから証拠とラベルのペアを作る。

入力:訓練データ パラメータ α

出力:訓練データ名＋li

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
void main(int argc,char *argv[])
{
  char A[500],B[30][30],D[50],DEF[5][50],DEF_MAX[50];
  int i,j,k,v,def[5],def_all=0;
  double in;
  char IN[30];
  double def_max=0,def_count=0,def_ans,def_a=0;

  strcpy(IN,argv[2]);
  in=atof(IN);
  strcpy(D,argv[1]);
  strcat(D,"l");
  ofstream of_file(D);
  ifstream in_file(argv[1]);
  while(! in_file.eof()){
    in_file.getline(&A[0],sizeof(A));
    if(strlen(A)<=4)continue;
    i=0; j=0; k=0;
    if(A[0]==' ')i++;
```

```
    while(1){
      B[k][j]=A[i];
      if(A[i]=='\0')break;
      if(A[i]==' '){
        if(A[i-1]==' '){
  i++;
  continue;
}
B[k][j]='\0';
k++;
j=-1;
      }
      i++;
      j++;
    }
    if(def_all==0){
      strcpy(DEF[0],B[k]);
      def_all=1;
      def[0]=1;
    }
    else{
      for(v=0;v<=def_all;v++){
        if(strcmp(DEF[v],B[k])==0){
          def[v]++;
          break;
}
else{
          if(v==def_all){
            strcpy(DEF[v],B[k]);
            def[v]=1;
            def_all++;
            break;
```

```
  }
}
      }
    }
    for(v=0;v<k;v++){
      of_file<<B[v]<<" "<<B[k]<<endl;
    }
  }
  for(v=0;v<def_all;v++){
    def_a=def_a+def[v];
    if(def_max<def[v]){
      def_max=def[v];
      strcpy(DEF_MAX,DEF[v]);
    }
  }
  def_ans=def_max/(def_a+in);
  of_file<<"default "<<DEF_MAX<<" "<<def_ans<<" "<<def_a;
  for(v=0;v<def_all;v++){
     of_file<<" "<<DEF[v]<<" "<<def[v];
  }
    of_file<<endl;

}
```

サブプログラム2：li2(c++)

証拠とラベルのペアから証拠の共起頻度と予測力を計算する
入力:訓練データ名＋li
出力:訓練データ＋liz

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
void main(int argc,char *argv[])
{
  /*ANS は解答行列,ANS_MAX は解答の選択,DEF はデフォルト計算用*/
  char FILE[15],IN_A[100],IN_B[1][30],ANS[5][30],ANS_MAX[30],DEF[5][30],
check_A[100],check_B[1][30];
  /*c_ans が解答の種類数*/
  int i,j,k,ans[5],def[5],c,c_ans=1,c_def=1;
  double anser,ans_max,ans_all;

  char IN[30];
  double in;

  strcpy(IN,argv[2]);
  in=atof(IN);

  strcpy(FILE,argv[1]);
  strcat(FILE,"z");
  ofstream of_file(FILE);
  ifstream in_file(argv[1]);
```

```
in_file.getline(&IN_A[0],sizeof(IN_A));
of_file<<IN_A<<endl;
i=0;j=0;k=0;
while(1){
  IN_B[k][i]=IN_A[j];
  if(IN_A[j]=='\0')break;
  if(IN_A[j]==' '){
    IN_B[k][i]='\0';
    k++;
    i=-1;
  }
  i++;
  j++;
}
strcpy(ANS[0],IN_B[1]);
strcpy(DEF[0],IN_B[1]);
def[0]=0;
ans[0]=1;
ans_all=1;

/*after of line2 */

while(! in_file.eof()){
  in_file.getline(&IN_A[0],sizeof(IN_A));
  if(strlen(IN_A)<=2)continue;
  strcpy(check_A,IN_A);
  i=0;j=0;k=0;
  while(1){
    check_B[k][i]=check_A[j];
    if(check_A[j]=='\0')break;
    if(isspace(check_A[j])!=0){
```

```
      check_B[k][i]='\0';
      k++;
      i=-1;
    }
    i++;
    j++;
  }

  /* 証拠が同一の場合*/
  if(strcmp(IN_B[0],check_B[0])==0){
    for(c=0;c<=c_ans;c++){

      /*既知の解答があった場合*/

      if(c==c_ans){
i=0;j=0;k=0;
        while(1){
          IN_B[k][i]=IN_A[j];
          if(IN_A[j]=='\0')break;
          if(isspace(IN_A[j])!=0){
            IN_B[k][i]='\0';
            k++;
            i=-1;
          }
          i++;
          j++;
        }

strcpy(ANS[c],IN_B[1]);
        ans[c]=1;
        c_ans++;
break;
```

```
}

        /*既知の解答がなかった場合*/

if(strstr(IN_A,ANS[c])!=0){
  ans[c]++;
  break;
}
      }
    }

    /*証拠が変わった場合*/

    else{

      /*前証拠出力*/
      if(strcmp(IN_B[0],"default")!=0){
      ans_all=0;
      ans_max=0;
      of_file<<IN_B[0];
      for(c=0;c<c_ans;c++){
       ans_all=ans_all+ans[c];
       if(ans_max<ans[c]){
  ans_max=ans[c];
  strcpy(ANS_MAX,ANS[c]);
}
      }
      anser=ans_max/(ans_all+in);
      of_file<<" "<<ANS_MAX<<" "<<anser<<" "<<ans_all;
      for(c=0;c<c_ans;c++){
        of_file<<" "<<ANS[c]<<" "<<ans[c];
      }
```

```
        of_file<<endl;

        /*デフォルト計算*/
        for(c=0;c<=c_def;c++){
if(c==c_def){
  strcpy(DEF[c],ANS_MAX);
  def[c]=1;
  c_def++;
  break;
}
if(strcmp(DEF[c],ANS_MAX)==0){
  def[c]++;
  break;
}
        }

        }

        /*新証拠入力*/
        c_ans=1;
        i=0;j=0;k=0;
        while(1){
          IN_B[k][i]=IN_A[j];
          if(IN_A[j]=='\0')break;
          if(isspace(IN_A[j])!=0){
            IN_B[k][i]='\0';

            k++;
            i=-1;
          }
          i++;
```

```
            j++;
          }
          strcpy(ANS[0],IN_B[1]);
          ans[0]=1;
        }
      }
    }
```

付録B：QBCプログラム

メインプログラム：QBC.sh(bash)

QBCのプログラム出力はテストデータから訓練データに１０事例移動したものになる。出力ファイル名には後ろに繰り返し回数がつく。

入力:訓練データ テストデータ パラメータ α

出力:訓練データ＋1 テストデータ＋1

```
rond $1 "a"
sleep 5
rond $1 "b"
sleep 1
rond $1 "c"
sleep 8
rond $1 "d"
sleep 2
rond $1 "e"
sleep 7
rond $1 "f"
sleep 4
rond $1 "g"
sleep 7
rond $1 "h"
sleep 3
rond $1 "i"
sleep 7
rond $1 "j"
sleep 8
list.sh "a" $3
list.sh "b" $3
list.sh "c"  $3
```

```
list.sh "d"  $3
list.sh "e"  $3
list.sh "f"  $3
list.sh "g"  $3
list.sh "h"  $3
list.sh "i"  $3
list.sh "j"   $3
Qhan "alist" "$2"
Qhan "blist" "$2"
Qhan "clist" "$2"
Qhan "dlist" "$2"
Qhan "elist" "$2"
Qhan "flist" "$2"
Qhan "glist" "$2"
Qhan "hlist" "$2"
Qhan "ilist" "$2"
Qhan "jlist" "$2"
ent "$2" "all.lost"
sort +1 -r all.lost > all2.lost
n-file "$1" "$2"
rm -f a b c d e f g h i j **list TEST**.h all.lost all2.lost
```

サブプログラム：rond(c++)

訓練データからランダムに８割程度のデータを抜き出した訓練データを作成する。
入力:訓練データ 出力ファイル名
出力:任意

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
#include <time.h>
void main(int argc,char *argv[])
{
  char A[500];
  int i,j,k;
  ifstream in_file(argv[1]);
  ofstream of_file(argv[2]);
  srand((unsigned)time(NULL));
  while(! in_file.eof()){
    in_file.getline(&A[0],sizeof(A));
    k=rand()%10;
    if(k==1 || k==0)continue;
    of_file<<A<<endl;
  }
}
```

サブプログラム：Qhanbetu(c++)

決定リストでテストデータの識別を行なう
入力:決定リスト テストデータ
出力:テストデータ名＋-＋決定リスト名＋.h

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
void main(int argc,char *argv[])
{
  char A[600],B[30],C[100],D[15][30],E[50];
  int i,j,k,t=0,len;
  strcpy(E,argv[2]);
  strcat(E,"-");
  strcat(E,argv[1]);
  strcat(E,".h");
  ofstream of_file(E);
  ifstream in_file(argv[2]);
  while(! in_file.eof()){
    in_file.getline(&A[0],sizeof(A));
    if(strlen(&A[0])<=3)continue;
    i=0;
    while(1){
      B[i]=A[i];
```

```
      if(A[i]==' '){
        B[i]='\0';
break;
      }
      i++;
    }
    ifstream ch_file(argv[1]);
    while(! ch_file.eof()){
      i=0;j=0;k=0;
      ch_file.getline(&C[0],sizeof(C));
      while(1){
        D[k][i]=C[j];
        if(C[j]=='\0')break;
        if(C[j]==' '){
          D[k][i]='\0';
  k++;
  i=-1;
        }
        i++;
        j++;
      }
      if(strstr(&A[0],&D[0][0])!=0){

        of_file<<D[1]<<endl;
        ch_file.close();
        break;
      }
      if(strstr(&C[0],"default")!=0){
of_file<<D[1]<<endl;
        ch_file.close();
        break;
      }
```

サブプログラム：ent(c++)

複数の決定リスト (alist〜jlist) の識別結果から事例の曖昧性を計る
入力:テストデータ
出力:all.lost

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
#include <time.h>
void main(int argc,char *argv[])
{
  char FILE[20];
  char A[10][20],ANS[10][20];
  int i,j,ans[10],count,flag,line=0;
  double p;

  strcpy(FILE,argv[1]);
  strcat(FILE,"-alist.h");
  ifstream a_file(FILE);
  strcpy(FILE,argv[1]);
  strcat(FILE,"-blist.h");
  ifstream b_file(FILE);
  strcpy(FILE,argv[1]);
  strcat(FILE,"-clist.h");
  ifstream c_file(FILE);
  strcpy(FILE,argv[1]);
  strcat(FILE,"-dlist.h");
```

```
ifstream d_file(FILE);
strcpy(FILE,argv[1]);
strcat(FILE,"-elist.h");
ifstream e_file(FILE);
strcpy(FILE,argv[1]);
strcat(FILE,"-flist.h");
ifstream f_file(FILE);
strcpy(FILE,argv[1]);
strcat(FILE,"-glist.h");
ifstream g_file(FILE);
strcpy(FILE,argv[1]);
strcat(FILE,"-hlist.h");
ifstream h_file(FILE);
strcpy(FILE,argv[1]);
strcat(FILE,"-ilist.h");
ifstream i_file(FILE);
strcpy(FILE,argv[1]);
strcat(FILE,"-jlist.h");
ifstream j_file(FILE);
ofstream of_file(argv[2]);
while(! a_file.eof()){
  line++;
  of_file<<line<<" ";
  a_file.getline(&A[0][0],sizeof(A));
  b_file.getline(&A[1][0],sizeof(A));
  c_file.getline(&A[2][0],sizeof(A));
  d_file.getline(&A[3][0],sizeof(A));
  e_file.getline(&A[4][0],sizeof(A));
  f_file.getline(&A[5][0],sizeof(A));
  g_file.getline(&A[6][0],sizeof(A));
  h_file.getline(&A[7][0],sizeof(A));
  i_file.getline(&A[8][0],sizeof(A));
```

```
    j_file.getline(&A[9][0],sizeof(A));
    count=1;flag=0;
    strcpy(ANS[0],A[0]);
    ans[0]=1;
    for(i=1;i<10;i++){
      for(j=0;j<count;j++){
if(strcmp(A[i],ANS[j])==0){
  flag=1;
  ans[j]++;
  break;
}
      }
      if(flag==0){
strcpy(ANS[count],A[i]);
        ans[count]=1;
count++;
      }
      else{
flag=0;
      }
    }
    p=1;
    for(i=0;i<count;i++){
      p=p*ans[i];
      //    of_file<<ANS[i]<<" "<<ans[i]<<" ";
    }
    p=p/100;
    of_file<<p<<endl;
  }
}
```

サブプログラム：n-file(c++)

曖昧性の値のリスト（all2.lost）から訓練データとテストデータを更新する。
入力:訓練データ テストデータ
出力:訓練データ+1 テストデータ+1

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
void main(int argc,char *argv[])
{
  char A[500],B[30][50],C[30][50],CH[10];
  char FILEA[30],FILEB[30],NUMBER[11]={"0123456789"};
  int i,j,k,l,v,line=1,ch[10],count=0,check=0;
  strcpy(FILEA,argv[1]);
  strcpy(FILEB,argv[2]);
  if(isdigit(FILEA[strlen(FILEA)-2])==0){
  strcat(FILEA,"01");
  strcat(FILEB,"01");
  }
  else{
    if(FILEA[strlen(FILEA)-1]=='9'){
      FILEA[strlen(FILEA)-1]='0';
      FILEB[strlen(FILEB)-1]='0';
      for(i=8;i>=0;i--){
        if(FILEA[strlen(FILEA)-2]==NUMBER[i]){
  FILEA[strlen(FILEA)-2]=NUMBER[i+1];
```

```
            FILEB[strlen(FILEB)-2]=NUMBER[i+1];
          }
        }
      }
      else{
        for(i=8;i>=0;i--){
          if(FILEA[strlen(FILEA)-1]==NUMBER[i]){
    FILEA[strlen(FILEA)-1]=NUMBER[i+1];
            FILEB[strlen(FILEB)-1]=NUMBER[i+1];
    break;
          }
        }
      }
    }
    ofstream of_file(FILEA);
    ofstream re_file(FILEB);
    ifstream ch_file("all2.lost");
    while(! ch_file.eof()){
      ch_file.getline(&A[0],sizeof(A));
      if(strlen(A)<=2)continue;
      for(i=0;i<=10;i++){
        CH[i]=A[i];
        if(A[i]=='.'){
CH[i]='\0';
ch[count]=atoi(CH);
count++;
break;
        }
      }
      if(count==10)break;
    }
    ifstream in_file(argv[2]);
```

```
  while(! in_file.eof()){
    in_file.getline(&A[0],sizeof(A));
    for(l=0;l<=9;l++){
      if(line==ch[l]){
        check=1;
      }
    }
    if(check!=1){
      re_file<<A<<endl;
    }
    else{
      check=0;
      of_file<<A<<endl;
    }
    line++;
  }
  ifstream in2_file(argv[1]);
  while(! in2_file.eof()){
    in2_file.getline(&A[0],sizeof(A));
      of_file<<A<<endl;
  }
}
```

付録C：期待損失プログラム

メインプログラム：lost.sh(bash)

期待損失のプログラム。入出力はQBCプログラムと同一
QBCと同一共通のプログラムもQBCの項参照
入力:訓練データ テストデータ パラメータ α
出力:訓練データ＋1 テストデータ＋1

```
list.sh $1 $3
han "$1""list" "$2"
new $1 $2
wc -l "$2"  >length.txt
for name1 in "0" "1" "2"
do
  for name2 in "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
  do
    for name3 in "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
    do
      for name4  in "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
      do
        for name5  in "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
        do
          if [ -s "$1""$name1""$name2""$name3""$name4""$name5"".data" ]
          then
            reli1 "$1""$name1""$name2""$name3""$name4""$name5"".data"
            rm -f "$1""$name1""$name2""$name3""$name4""$name5"".data"

            reli2_n "$1" "$1""$name1""$name2""$name3""$name4""$name5""
.datali"  $3
            rm -f "$1""$name1""$name2""$name3""$name4""$name5"".datali"
```

```
            han2_n "$1" "$2" "$name1""$name2""$name3""$name4""$name5"
            rm -f "$1""$name1""$name2""$name3""$name4""$name5"".datalist"

            son "$name1""$name2""$name3""$name4""$name5"".h" "length.txt"
            rm -f "$name1""$name2""$name3""$name4""$name5"".h"

            echo "$name1""$name2""$name3""$name4""$name5"".h" end
          fi
        done
      done
    done
  done
done
rm -f length.txt
sort +1  all.lost > all2.lost
n-file $1 $2

rm -f all.lost all2.lost
rm -f "$2""-"$1""list.h" "$1""list"
```

サブプログラム1：han(c++)

決定リストからテストデータの識別を行なう
入力:決定リスト テストデータ
出力:テストデータ名＋-＋決定リストデータ名＋.h

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
void main(int argc,char *argv[])
{
  char A[600],B[30],C[100],D[15][30],E[50];
  int i,j,k,t=0,len;

  /*出力*/

  strcpy(E,argv[2]);
  strcat(E,"-");
  strcat(E,argv[1]);
  strcat(E,".h");
  ofstream of_file(E);

  /*テストファイル呼出*/

  ifstream in_file(argv[2]);
  while(! in_file.eof()){
    t++;
    in_file.getline(&A[0],sizeof(A));
```

```
    if(strlen(&A[0])<=3)continue;
    i=0;
    while(1){
      B[i]=A[i];
      if(A[i]==' '){
        B[i]='\0';
break;
      }
      i++;
    }

    /*判別リスト呼出*/

    ifstream ch_file(argv[1]);
    while(! ch_file.eof()){
      i=0;j=0;k=0;
      ch_file.getline(&C[0],sizeof(C));
      while(1){
        D[k][i]=C[j];
        if(C[j]=='\0')break;
        if(C[j]==' '){
          D[k][i]='\0';
  k++;
  i=-1;
        }
        i++;
        j++;
      }
      if(strcmp(&B[0],&D[0][0])==0){
//cout<<B<<endl<<endl;
        of_file<<t<<" "<<D[1]<<" "<<D[2];
        if(strstr(A,D[1])==0)of_file<<" "<<"F"<<endl;
```

```
        else {
  of_file<<" "<<"T"<<endl;
}
//        of_file<<D[0]<<"  "<<A<<endl;
        ch_file.close();
        break;
      }
      if(strcmp(&D[0][0],"default")==0){
of_file<<t<<" "<<D[1]<<" "<<D[2];
        if(strstr(A,D[1])==0)of_file<<" "<<"F"<<endl;
        else{
  of_file<<" "<<"T"<<endl;
}

//     of_file<<D[0]<<"   "<<A<<endl;
        ch_file.close();
        break;
      }
    }
  }
}
```

サブプログラム2：reli1(c++)

追加候補事例中の証拠を分解してラベルをつける
入力:訓練データ名＋事例ナンバー＋.data
出力:訓練データ名＋事例ナンバー＋.datali

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
void main(int argc,char *argv[])
{
  char A[500],B[30][50],D[50];
  int i,j,k,v;
  strcpy(D,argv[1]);
  strcat(D,"li");
  ofstream of_file(D);
  ifstream in_file(argv[1]);
  in_file.getline(&A[0],sizeof(A));
  i=0; j=0; k=0;
  if(A[0]==' ')i++;
  while(1){
    B[k][j]=A[i];
    if(A[i]=='\0')break;
    if(A[i]==' '){
      if(A[i-1]==' '){
i++;
continue;
      }
```

```
            B[k][j]='\0';
            k++;
            j=-1;
          }
          i++;
          j++;
        }
        for(v=0;v<k;v++){
          of_file<<B[v]<<" "<<B[k]<<endl;
        }
        of_file<<"default "<<B[k]<<" "<<k<<endl;
      }
```

サブプログラム3：reli2(c++)

追加候補に含まれる証拠を決定リストと照合し、更新する（改良前）
入力:訓練データ名＋事例ナンバー＋.datali
出力:訓練データ名＋事例ナンバー＋.datalist

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
void main(int argc,char *argv[])
{
  char FILE[30],OFILE[30],A[600],B[600],C[20][70],G[3][70];
  double max,e,in,lost;
  int i,j,k,t,u,c[70],ma,sum,check,def,all;
  char IN[30];

  strcpy(IN,argv[3]);
  in=atof(IN);

  ifstream in_file(argv[2]);
  strcpy(OFILE,argv[2]);
  strcat(OFILE,"st");
  ofstream of_file(OFILE);
  strcpy(FILE,argv[1]);
  strcat(FILE,"list");
  while(! in_file.eof()){
    in_file.getline(&B[0],sizeof(B));
    if(strlen(B)<=2)continue;
```

```
    def=0;
    check=0;
    i=0;j=0;k=0;
    while(1){
      G[k][i]=B[j];
      if(B[j]=='\0')break;
      if(B[j]==' '){
        G[k][i]='\0';
k++;
i=-1;
      }
      i++;
      j++;
    }

    ifstream ch_file(FILE);
    while(! ch_file.eof()){
      ch_file.getline(&A[0],sizeof(A));
      if(strlen(A)<=2)continue;
      i=0;j=0;k=0;
      while(1){
        C[k][i]=A[j];
        if(A[j]=='\0')break;
        if(A[j]==' '){
          C[k][i]='\0';
          k++;
  i=-1;
      }
      i++;
      j++;
    }
```

```
      if(strcmp(&C[0][0],&G[0][0])==0){
        check=1;
        all=atoi(C[3]);
        max=0;
        ma=0;
        for(t=4;t<=k;t=t+2){
          c[t]=atoi(C[t+1]);
  if(strcmp(&G[1][0],&C[t][0])==0){
            if(strcmp(&G[0][0],"default")==0){
              c[t]=c[t]+atoi(G[2])-1;
              all=all+atoi(G[2])-1;
    }
    c[t]++;
  }
          if(c[t]>max){
    max=c[t];
    ma=t;

  }
}
e=max/(all+1+in);
        lost=e-atof(C[2]);
        sum=0;
        of_file<<C[0]<<" "<<C[ma]<<" "<<e<<" "<<all+1<<" "<<lost;
        for(t=4;t<=k;t=t+2){
  of_file<<" "<<C[t]<<" "<<c[t];
          sum=sum+c[t];
}
        if(sum==atoi(C[3]))of_file<<" "<<G[1]<<" 1";
of_file<<endl;
break;
      }
```

```
        }
        ch_file.close();
      }
    }
```

サブプログラム3': reli2n(c++)

追加候補に含まれる証拠を決定リストと照合し、更新する（改良後）
入力:訓練データ名＋事例ナンバー＋.datali
出力:訓練データ名＋事例ナンバー＋.datalist

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
void main(int argc,char *argv[])
{
  char FILE[30],OFILE[30],A[600],B[600],C[20][70],G[3][70];
  double max,e,in,lost;
  int i,j,k,t,u,c[70],ma,sum,check,def,all;
  char IN[30];

  strcpy(IN,argv[3]);
  in=atof(IN);

  ifstream in_file(argv[2]);
  strcpy(OFILE,argv[2]);
  strcat(OFILE,"st");
  ofstream of_file(OFILE);
  strcpy(FILE,argv[1]);
  strcat(FILE,"list");
  while(! in_file.eof()){
    in_file.getline(&B[0],sizeof(B));
    if(strlen(B)<=2)continue;
```

```
    def=0;
    check=0;
    i=0;j=0;k=0;
    while(1){
      G[k][i]=B[j];
      if(B[j]=='\0')break;
      if(B[j]==' '){
        G[k][i]='\0';
k++;
i=-1;
      }
      i++;
      j++;
    }

    ifstream ch_file(FILE);
    while(! ch_file.eof()){
      ch_file.getline(&A[0],sizeof(A));
      if(strlen(A)<=2)continue;
      i=0;j=0;k=0;
      while(1){
        C[k][i]=A[j];
        if(A[j]=='\0')break;
        if(A[j]==' '){
          C[k][i]='\0';
          k++;
  i=-1;
      }
      i++;
      j++;
    }
```

```
      if(strcmp(&C[0][0],&G[0][0])==0){
        check=1;
/*追加部分*/
if(atof(C[2])>0.8){
  continue;
}
/*ここまで*/
        all=atoi(C[3]);
        max=0;
        ma=0;
        for(t=4;t<=k;t=t+2){
          c[t]=atoi(C[t+1]);
  if(strcmp(&G[1][0],&C[t][0])==0){
            if(strcmp(&G[0][0],"default")==0){
              c[t]=c[t]+atoi(G[2])-1;
              all=all+atoi(G[2])-1;
    }
    c[t]++;
  }
          if(c[t]>max){
    max=c[t];
    ma=t;

  }
}
e=max/(all+1+in);
        lost=e-atof(C[2]);
        sum=0;
        of_file<<C[0]<<" "<<C[ma]<<" "<<e<<" "<<all+1<<" "<<lost;
        for(t=4;t<=k;t=t+2){
  of_file<<" "<<C[t]<<" "<<c[t];
          sum=sum+c[t];
```

```
}
        if(sum==atoi(C[3]))of_file<<" "<<G[1]<<" 1";
of_file<<endl;
break;
      }
    }
    ch_file.close();
  }
}
```

サブプログラム4：han2(c++)

追加候補事例によって識別が変わる部分を抽出しその時の予測力の差を計る (改良前)

入力:訓練データ名＋事例ナンバー＋.datalist

出力:事例ナンバー＋.h

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
void main(int argc,char *argv[])
{
  char FILE[100],CH[20][20][50],A[300],B[200],C[3][50];
  int i,j,k,t,len=0;

  strcpy(FILE,argv[1]);
  strcat(FILE,argv[3]);
  strcat(FILE,".datalist");
  ifstream ch_file(FILE);
  while(! ch_file.eof()){
    ch_file.getline(&A[0],sizeof(A));
    if(strlen(A)<3)continue;
    i=0;j=0;k=0;
    while(1){
      CH[len][k][i]=A[j];
      if(A[j]=='\0')break;
      if(A[j]==' '){
        CH[len][k][i]='\0';
```

```
k++;
i=-1;
      }
      i++;
      j++;
    }
    len++;
  }
  ch_file.close();

  ifstream in1_file(argv[2]);
  strcpy(FILE,argv[2]);
  strcat(FILE,"-");
  strcat(FILE,argv[1]);
  strcat(FILE,"list.h");
  ifstream in2_file(FILE);
  strcpy(FILE,argv[3]);
  strcat(FILE,".h");
  ofstream of_file(FILE);
  while(! in1_file.eof()){
    in1_file.getline(&A[0],sizeof(A));
    in2_file.getline(&B[0],sizeof(B));
    if(strlen(A)<3)continue;
    i=0;j=0;k=0;
    while(1){
      C[k][i]=B[j];
      if(B[j]=='\0')break;
      if(B[j]==' '){
        C[k][i]='\0';
k++;
i=-1;
      }
```

```
        i++;
        j++;
      }
        if(atof(C[2])<atof(CH[len-1][2])){
of_file<<C[0]<<" "<<CH[len-1][1]<<" "<<atof(CH[len-1][2])-atof(C[2])<<endl;
        }
        else{
          for(t=0;t<len-1;t++){
            if(strstr(&A[0],&CH[t][0][0])!=0 && atof(C[2])<atof(CH[t][2])){
              of_file<<C[0]<<" "<<CH[t][1]<<" "<<CH[t][4]<<endl;
              break;
            }
          }
        }
    }
}
```

サブプログラム4': han2n(c++)

追加候補事例によって識別が変わる部分を抽出しその時の予測力の差を計る (改良後)

入力:訓練データ名＋事例ナンバー＋.datalist

出力:事例ナンバー＋.h

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
void main(int argc,char *argv[])
{
  char FILE[100],CH[20][20][50],A[300],B[200],C[3][50];
  int i,j,k,t,len=0,check;

  strcpy(FILE,argv[1]);
  strcat(FILE,argv[3]);
  strcat(FILE,".datalist");
  ifstream ch_file(FILE);
  while(! ch_file.eof()){
    ch_file.getline(&A[0],sizeof(A));
    if(strlen(A)<3)continue;
    i=0;j=0;k=0;
    while(1){
      CH[len][k][i]=A[j];
      if(A[j]=='\0')break;
      if(A[j]==' '){
        CH[len][k][i]='\0';
```

```
k++;
i=-1;
      }
      i++;
      j++;
    }
    len++;
  }
  ch_file.close();

  ifstream in1_file(argv[2]);
  strcpy(FILE,argv[2]);
  strcat(FILE,"-");
  strcat(FILE,argv[1]);
  strcat(FILE,"list.h");
  ifstream in2_file(FILE);
  strcpy(FILE,argv[3]);
  strcat(FILE,".h");
  ofstream of_file(FILE);
  while(! in1_file.eof()){
    in1_file.getline(&A[0],sizeof(A));
    in2_file.getline(&B[0],sizeof(B));
    if(strlen(A)<3)continue;
    i=0;j=0;k=0;
    while(1){
      C[k][i]=B[j];
      if(B[j]=='\0')break;
      if(B[j]==' '){
        C[k][i]='\0';
k++;
i=-1;
      }
```

```
            i++;
            j++;
          }
          check=0;
            for(t=0;t<len-1;t++){
              if(strstr(&A[0],&CH[t][0][0])!=0 && atof(C[2])<atof(CH[t][2])){
                check=1;
                of_file<<C[0]<<" "<<CH[t][1]<<" "<<CH[t][2]<<endl;
                break;
              }
            }
        }
    }
```

サブプログラム5：son(c++)

期待損失の計算
入力:事例ナンバー＋.h
出力:all.lost

```
#include <iostream.h>
#include <fstream.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#include <math.h>
void main(int argc,char *argv[])
{
  char A[500],B[20][30],C[30],FILE[50];
  int i,j,k,length;
  double ZOLOSS=0;
  strcpy(FILE,"all.lost");
  ofstream of_file(FILE,ios::app);
  ifstream in_file(argv[2]);
    in_file.getline(&A[0],sizeof(A));
    for(i=0;i<100;i++){
      C[i]=A[i+3];
      if(A[i+3]==' '){
        C[i]='\0';
break;
      }
    }
    in_file.close();
```

```
    length=atoi(C);
  in_file.close();
  ifstream in_file2(argv[1]);
  while(! in_file2.eof()){
    in_file2.getline(&A[0],sizeof(A));
    if(strlen(A)<=2)continue;
    i=0;j=0;k=0;
    while(1){
      B[k][i]=A[j];
      if(A[j]=='\0')break;
      if(A[j]==' '){
        B[k][i]='\0';
k++;
i=-1;
      }
      i++;
      j++;
    }
    ZOLOSS =ZOLOSS +atof(B[2]);
  }
  of_file<<argv[1]<<" "<<"ZOLOSS="<<ZOLOSS<<endl;
}
```