修　士　論　文

バッギング手法による
分類システムの性能の向上

指導教官　Ho Tu Bao　教授

北陸先端科学技術大学院大学
知識科学研究科知識システム基礎学専攻

850068　西田　健一郎

審査委員：　Ho　Tu　Bao　教授（主査）
石崎　雅人　助教授
中森　義輝　教授

2000 年 2 月

目　次

図　目　次

表　目　次

第　1　章
はじめに

1.1　研究の背景と目的

機械学習において分類問題は主要な研究分野のひとつである。過去のデータを適切に分類することは未知のデータへの予想を与えることにつながる。古くは、データを扱うのは人間のみであった。扱われるデータは人間が作ったものであり、複雑さも限られていたので単純な統計手法で間に合っていた。今日高次元，混合データタイプ、巨大なデータ数の複雑なデータが存在する。これからのコンピュータ世代の人々にとって、これら複雑なデータは珍しくない。

分類システムの研究はこういった複雑なデータから有用な知識を抽出する方法の一つとして研究がなされてきた。よい分類は未知の事柄に対する適切な予想を与える。予想が正確であるかどうかは比較的確認が容易であり、興味深いものでもある。したがって分類システムの正確さを上げる研究が盛んに行われている。分類システムの性能を予想の正確さで測ると考えるとき、分類システムは予想をするシステムつまり予想機(predictor)と見ることができる。分類システムを予想機と見ることにより、予想を計算する過程を形式化することが容易になる。分類システムの性能の向上には種々の観点が考えられるが、本研究では正確さにのみ着眼している。

バッギング手法は最近発見された新しい手法である。バッギング手法は予想機の正確さの推定法を研究する過程で発見された。その後、バッギングは予想機の予想の正確さを上げる手法として提唱された。そのとき提唱者である Breiman は“お門違いの方法で予想機の正確さを向上することができる。”と、驚いたのである。予想機の正確さの向上を求めて、いくつかの予想機を複合する流れがある。バッギング手法の研究はそのような流れのひとつとも見られている。

これまでに Nearest　Neighbor 分類システムによる予想機[10] (以下 NNC) には有効でなく、C4.5[13]、CART[11],など決定木による予想機、ニューラルネットによる予想機については有効

であることが確認されている。しかしこれまでに提唱された定義では、これら以外の予想機についての有効性が判定できない。[2][6]

本研究では１.バッギング手法、２.評価法の１つである交差検定(Cross Validation 以下 CV)[10]、３．予想機、の３つを結合したシステムを３つ開発する。このシステムの役目は予想機の正確さを測ることである。

本研究の目的はバッギング手法を分類システムに応用し、バッギング手法の有効性を確認することである。そしてその概要は以下である。１.すでにバッギングの有効性が報告されている C4.5, NNC についてバッギング手法を応用し、正確さに関する実験を行う。これによりバッギングの有効性を確認する。そしてバッギングが有効であるときブートストラップ（本誌３章にて説明）[1]の回数と正確さの関係を調べる。２．有効性が確認されていない NaiveBayes による予想機（以下 NB）[10]にバッギングを応用し、正確さに関する実験を行う。

1.2 本論文の構成

本論文は７つの章よりなる。2 章は分類システムと予想機の関係について述べ、予想機に関する基本的なことの形式化を行った。さらに本研究で扱う NNC、決定木による分類システム、NB について紹介した。3 章ではバッギング手法の一般的な紹介と形式化を行った。そしてバッギングの有効範囲についてのキーワードである不安定性を紹介した。4 章では本研究で採用した CV を形式化した。5 章では本研究で開発されたシステムの概要を記した。6 章では実験で使用したデータについて記した。そして具体的な実験の進め方、実験結果、結果に対する考察を行った。７章では本研究全体についての結論を述べた。

第　2　章
分類システムとは

2.1　分類と予想

機械学習において分類システムの研究目的には、大きく分けて 2 つの目的があるといわれている。1 つは正確に予想する分類規則の生成であり、もう一つは予想の構造の解明である。本研究では前者についてのみ論じる。

分類システムは学習用サンプルから分類規則を組み立てる。クラスが未知のある状態に分類規則を当てはめることによりクラス（分類）を与える。このクラスがその状態への予想である。

分類システムを予想の正確さの観点で見ると予想機と見ることができる。予想機は学習用サンプルの情報から未知の状態に対する予想をする。

以下に Breiman[10]による予想機 (Predictor) としての分類システム、分割としての分類規則、サンプル、状態、についての形式を紹介する。

あらかじめ決まった順序である状態（a case）の測定値の集合をとる。そしてその測定値を x_1，　x_2，　...とする。ある状態で（x_1，　x_2，　...）が測定値ベクトル X として作られたとき、この X がそのある状態に相当するとする。測定値空間 $\mathbf{X}$ をとり，ここに可能なすべての X が定義されているとする。数多くの互いに異なる $\mathbf{X}$ の定義の存在が可能である。しかしいかなる測定値空間 $\mathbf{X}$ の定義がなされていても $X \in \mathbf{X}$ は $\mathbf{X}$ 空間上の 1 点となる。X は分類しようとする状態にも相当する。

今状態を J 個のクラスに分類することを考える。クラスに $1,\ 2,\ \ldots,\ C$ と番号をつける。c をクラスの集合とする。つまり　$c = \{1,\ 2,\ \ldots,\ C\}$。

すべての $X \in \mathbf{X}$ に対してシステマティックにクラスを予想する方法の 1 つはクラスを割り当

てるルールを生成することである。そのルールはクラス$1,\ \ 2,\ \ ...,\ \ C$のうち1つをすべての$X \in \mathbf{X}$に割り当てる。

定義1

分類規則は$\mathbf{X}$空間上に$d(X)$で表す。このとき$d(X)$は$c=\{1,\ \ 2,\ \ ...,\ \ C\}$の 1 つの数に等しい。

別の見方をすると$j \in c$のとき集合の列$A_j \subset \mathbf{X}$は

$$A_j = \{X \mid d(X) = j\}$$

と定義される。これらは互いに共通な要素をもたない。そして

$$\mathbf{X} = \bigcup_j A_j$$

となる。A_jは$\mathbf{X}$の分割である。これらは同値関係をなす。

定義2

分類規則は$\mathbf{X}$空間をJ個の部分集合に分ける分割である。つまり

$$\mathbf{X} = \bigcup_j A_j$$

である。このとき各々の$X \in A_j$について予想されるクラスはjである。

定義3

学習用サンプルは状態Xとクラスjの対$(X_1, j_1), ..., (X_N, j_N)$からなる。このとき状態の数はN であり、$X_n \in \mathbf{X}$、$j_n = \{1, 2, ..., C\}$、$n = 1, 2, ..., N$である。学習用サンプルは$L$で表す。つまり

$$L = \{(X_1, j_1), ..., (X_N, j_N)\}$$

測定値ベクトルはそこに表れる変数のタイプによって一般的に2つに分けることができる。

定義4

測定値が実数であるときその変数は連続値という。測定値が順序の関係ない有限集合からなるときその変数は離散値という。

以下属性値とは測定値の種類である。$X = (x_i, i = 1, ...)$において添え字iに相当するのが属性値

である。

定義 5

すべての測定値ベクトル X_n が一定の次元を持つとき、このサンプル(sample)は標準構造をもつという。

この論文で扱われるすべてのサンプルは標準構造をもつ。

分類システムは学習用サンプルから未知の状態に予想を与える。このことを予想機 φ で表す。

定義 6

学習用サンプルを $L=\{(X_n, j_n), n=1,...,N\}$ とする。今ある状態 $X \in \mathrm{X}$ を入力すると $\varphi(X, L)$ という予想を得る。関数 φ を予想機といい、$\varphi(X, L) \in \{1, 2, ..., C\}$ である。

今分類システム，予想機、分類規則，サンプル、状態、クラスの関係を図示すると次のようになる。（図２．１）

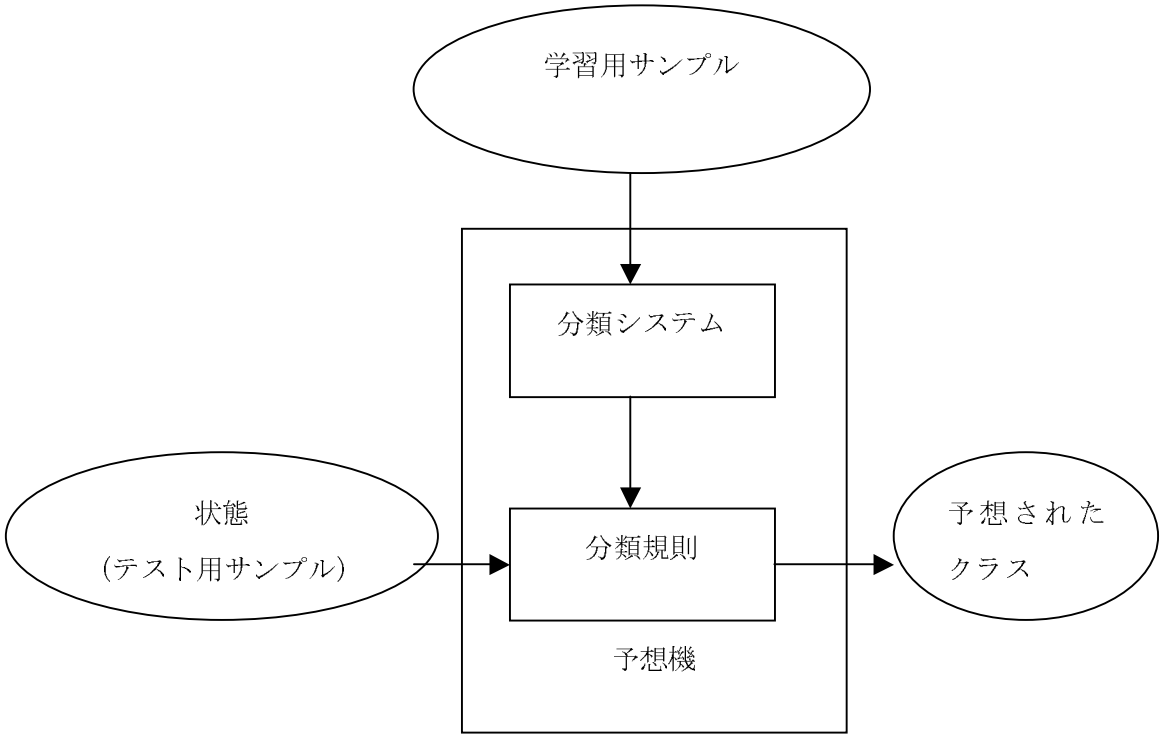


図２．１：予想機とその周辺

2.2 Nearest Neighbor による分類

いま Nearest Neighbor 分類システム（つまり NNC）に予想の必要なある状態のデータを入力する。このとき NNC は学習用サンプルから予想の必要な状態に最も近い測定値ベクトルをもつ部分集合を選び、その部分集合の持つクラスのうち最も多いクラスを予想とする。NNC は今日他の分類システムと比較されるよく知られた予想機である。

NNC による予想機は次のように 2 段階に形式化される。

１．学習用サンプルを $L=\{(X_n, j_n), n=1, ..., N\}$ とする。距離関数として $D(X_i, X_j)$ をとる。そして予想を必要とする状態を X とする。このとき X と最も近い K 個のサンプルをとる。これを近傍サンプル $L^{(N)}$ と表すと

$$L^{(N)}=\{(X_1, j_1), ..., (X_K, j_K)\}$$

と表す。

２．NNC による予想機を φ_N と表す。つまり

$$\varphi_N(X, L)=\arg\max_{j\in\{1, ..., C\}}\sum_{i=1}^{K}\delta(j, j_i)$$

となり、ここで $\delta(a,b)=\begin{cases}1 & if \quad a=b\\ 0 & if \quad a\neq b\end{cases}$

また $\arg\max_{x\in X} f(x)=\{x \mid \max imaize\ f(x)\}$

である。

2.3 Naive Bayes による分類

Naive Bayes（つまり NB） による分類システムは単純な確率論にもとづく予想機である。NB 予想機にある状態のデータを入力する、このとき学習用サンプルにおいてこの状態と各々のクラスが同時に起こる確率を計算する。 この確率が最も高いクラスを予想とする。この予想機は次のように形式化できる。

学習用サンプルを $L=\{(X_n, j_n), n=1, ..., N\}$ とする。予想を必要とする状態を

$X=(x_1, x_2, ..., x_M)$、クラスの候補を$j \in \{1, ..., C\}$とする。このとき空間Lにおいて$x_1, ..., x_M$が起こったときjの起こる条件付確率を$P(j \mid x_1, ..., x_M)$のように表す。このときNB予想機φ_{NB}は

$$\varphi_{NB}(X, L) = \arg\max_{j \in \{1, ..., C\}} P(j) \prod_{i}^{M} P(x_i \mid j)$$

と表せる。

NBは基本的には離散値の属性値を持つサンプルのために作られている。

2.4 決定木による分類（C4.5）

決定木による分類システムは、ある評価関数を用いて、学習用サンプルを分割する。これにより分類規則を生成する。分類規則はいわゆるIf－Thenルールで表すことができる。

評価関数の違いは分類システムの違いにつながる。もっともよく知られた決定木による分類システムとしてC4.5がある。C4.5において評価関数はGain - Raitioとよばれる。今これを紹介する。

Gain - RaitioはGainとSplitInfomationからなる。つまり

$$Gain - Raitio = \frac{Gain}{SplitInfomation}$$

である。Gainは情報論的エントロピーにより表される。学習用サンプル$L=\{(X_n, j_n), n=1, ..., N\}$、$j \in \{1, ..., C\}$、$X=(x_k, k=1, ..., K)$、$L$の部分集合を$S \subseteq L$とする。このときエントロピーを

$$Entropy(S) \equiv \sum_{i=1}^{C} - p_i \log_2 p_i$$

と表す。このときp_iは部分集合Sにクラスがiとなる要素の存在する確率である。そしてSのうち属性値x_kを要素として持つ部分集合を$S_{x_k} \subseteq S$とするとGainは

$$Gain(S, x_k) \equiv Entropy(S) - \sum_{k=1}^{K} \frac{|S_{x_k}|}{|S|} Entropy(S_{x_k})$$

と表すことができる。ここで$|S|$は部分集合の要素の数である。SplitInformationも同様に表せて

$$SplitInformation(S, x_k) \equiv -\sum_{k=1}^{K} \frac{|S_{x_k}|}{|S|} \log_2 \frac{|S_{x_k}|}{|S|}$$

となる。したがってGain - Raitioは

$$Gain - Raitio(S, x_k) = \frac{\sum_{k=1}^{K} \frac{|S_{x_k}|}{|S|} \sum_{i=1}^{C} p_i \log_2 p_i - \sum_{i=1}^{C} p_i \log_2 p_i}{\sum_{k=1}^{K} \frac{|S_{x_k}|}{|S|} \log_2 \frac{|S_{x_k}|}{|S|}}$$

と表すことができる。

C4.5 による予想機は学習用サンプル L を頼りに仮説空間を探索する。これにより分類規則を得る。この分類規則と予想を必要とする状態 X を比べることによりクラスの予想を得る。

C4.5 による予想機も前章と同様に $\varphi_{C4.5}(X, L)$ と表しクラス $j \in \{1,...,C\}$ をさす。

第　3　章
バッギング（Bagging）

3.1　バッギングとは

バッギングは予想機の正確さを向上させる手法である。もっとも性能のよい、つまり正確な予想機を選ぶ model selection という分野がある。バッギング手法はこの model selection における評価法の研究過程で発見された。[2]

バッギングは複数の予想機を収集する手法の 1 つである。その特徴はブートストラップという作業にあり、Bagging という名前の由来にもなっている。バッギングにより組み合わされた予想機はシステムを形成し、1 つの予想機として振舞う。そしてこの予想機はより正確な予想をすることが期待される。以下 i)収集による予想機、ii)投票（Vote）iii)ブートストラップ（Bootstrap）、iiii)バッギング予想機について述べる。

i)　収集による予想機：　収集による予想機をφ_Aと書く。このとき A は aggregate の意味である。ある予想機を$\varphi(X, L)$、予想を必要とする状態を$X = (x_k, k = 1, ..., K)$、学習用サンプルを$L = \{(X_n, j_n), n = 1, ..., N\}$とする。今 L から学習用サンプルの集合$\{L_m, m = 1, ..., M\}$を得るとき、これらを用いて収集による予想機φ_Aを次のように表す。

$$\varphi_A(X, L) = E_L \varphi(X, L)$$

このなかで$E_L \varphi(X, L)$はLとφを駆使したXに対する予想である。そしてここでは$\varphi(X, L)$は$\varphi(X, L_m)$の集まりに置き換えることが許されている。

ii)　投票：　いまφ_Aにおいて収集方法が最大得票による選出、つまり投票によるときを考える。このときクラスを　$j \in \{1, ..., C\}$と表す。そしてクラスをj、$\varphi(X, L_m)$が予想する回数をN_j

とする。つまり

$$N_j = \#\{m \mid \varphi(X, L_m) = j\}$$

するとφ_Aは

$$\varphi_A(X, L) = \arg\max_j N_j$$

と表すことができる。

iii) ブートストラップ： 学習用サンプル$L = \{(X_n, j_n), n = 1, ..., N\}$とする。このときブートストラップとは「$L$からランダムに１つ取り出し、元に戻す」という作業をN回繰り返すことにより、Lの複製$\{L_m^{(B)}, m = 1, ..., M\}$を作り出すことである。ブートストラップにより$L$のある要素$(X_n, j_n)$は、$L_m^{(B)}$のすべての要素であるかもしれないし、$L_m^{(B)}$の要素にはならないこともある。

Lの要素のうち、平均しておよそ３７％が$L_m^{(B)}$の要素として含まれないことになる。[1]

iiii) バッギング予想機： バッギング予想機は、投票による収集を行うφ_Aにおいて、学習用サンプルの集合$\{L_m, m = 1, ..., M\}$がブートストラップによる複製$\{L_m^{(B)}, m = 1, ..., M\}$の場合だといえる。今バッギング予想機を$\varphi_B$と書くと

$$\varphi_B(X, L) = av_B(X, L^{(B)})$$

$$= \arg\max_j N_j$$

ここで

$$N_j = \#\{m \mid \varphi(X, L_m^{(B)}) = j\}$$

と表すことができる。

バッギング予想機の概念を図３．１に示す。

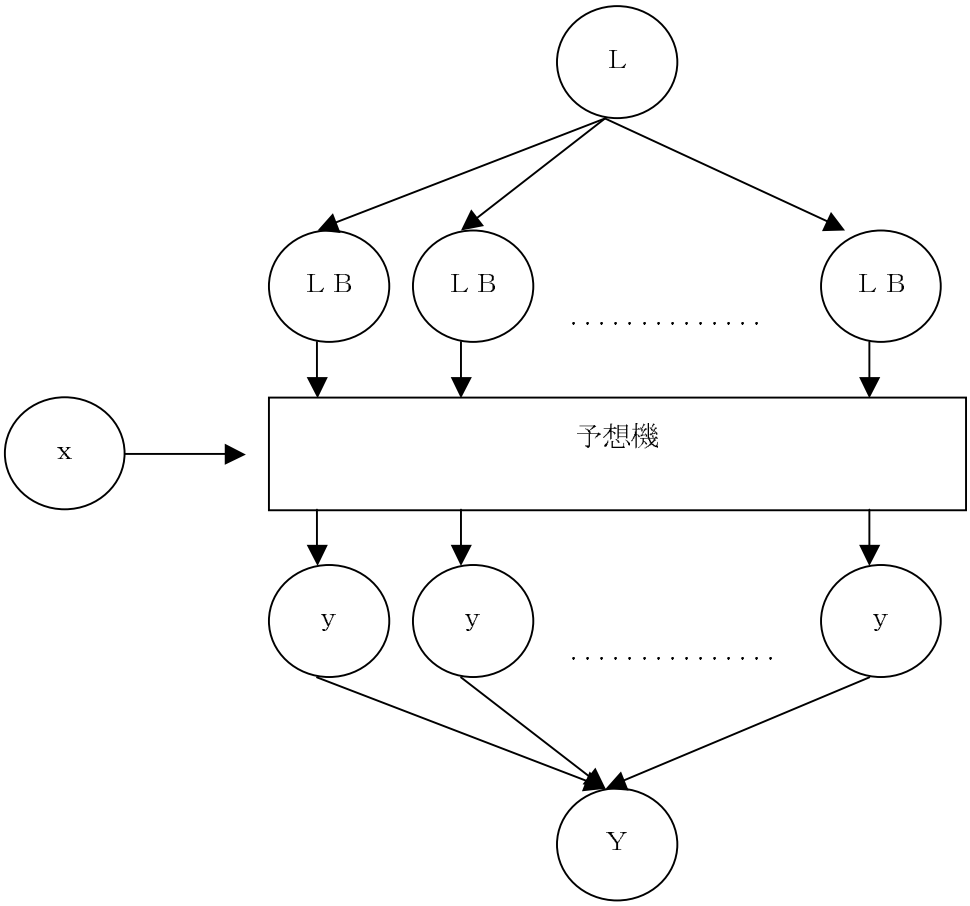


図３．１：バッギング予想機の概念図。

ここでＬは学習用サンプル、ＬＢはブートストラップによるＬの複製をあらわす。

ｙは$\varphi(X, L_m^{(B)})$による予想。Ｙは$\varphi_B(X, L)$による予想を表し、ｘは状態Xをさす。

Quinlan[9]はバッギング手法をＣ４．５ に応用し、27 個の標準的データ集合による実験を行った。このとき平均でおよそ 10%（バッギングによる C4.5 と C4.5 のエラー率の比の 100 分率）の向上を得た。そして理論的にも健全で有効であると結論している。

Breiman はバッギングの理論的提唱者であり、NNC においてはバッギングが有効でないこと。木による分類を行う CART においてバッギングが有効であること、を 7 個のデータ集合を用いて実証した。

この中でブートストラップの回数とバッギング予想機の正確さに関する実験を行った。そして２５回以上のブートストラップは労働の無駄だとしている。

３.２　予想機の不安定性

バッギング手法はどのような予想機の正確さも向上できるのだろうか。この問いに答えるために手がかりとなるのが不安定性（Unstability）という概念である。今その概念を紹介する。それと同時に本研究で扱う C4.5、NNC、NB による予想機に、バッギングが有効かどうかの可否について論文サーベイによる答えを出す。

Breiman[1]は「予想機そのものの構造において最も重要なことはそれが不安定であることである。学習用サンプルをある程度攪拌することにより、ある予想機の予想がある程度変わることがあれば、バッギングによりその予想機の正確さを向上させることができる。」
としている。この不安定さが不安定性（Unstability）である。この言葉は予想機の正確さを評価する研究において使われていた。[2]

予想機の不安定性は予想機の正確さを推定するとき障害となる。後述する交差検定（CrossValidation）は予想機の不安定性を解消するとされる。この概念は理論的明確な定義があるわけではないが、論文[2]の中で発見的定義として次のように表現されている。

発見的定義：　予想機において、学習用サンプルのわずかな違いがその予想機の作る一連の分類規則にたいして大きな違いを生じ得るとき、この予想機は不安定性である。

しかし予想機の不安定性については判断が難しく実証的に知ることしかできない。Domingos[6]は“バッギングが有効かそうでないかを予想機の特性からは判断できない”としている。

さらに同じ論文の中で決定木による分類システム、ニューラルネット、などは不安定性であり、

NNC などはそうではないとされる。つまり本研究で扱われる C4.5 は典型的な不安定性を持つ予想機であり、NNC は不安定性をもたないとされる。NB による予想機については上の定義では断言はできない。

つまり本研究で扱う予想機について

- C4.5 にはバッギング手法が有効である。
- NNC には有効でない。
- NB についてはわからない。

といえるのである。

第　４　章
評価方法

４.1　予想機の性能と真のエラー率

予想機の性能を評価する方法については多くの研究がなされてきた。特に予想の正確さは予想機の主目的であるから関心が高い。本研究で述べる分類システムの性能とは予想機の正確さのことである。この正確さは、予想の真偽のうち偽の回数の割合で測る。つまりエラー率を用いて正確さを測る。これには推定エラー率がよく使われる。真のエラー率を求めることはむづかしい。まず真のエラー率について紹介する。

予想機の真のエラー率を $R^*(\varphi)$ とすると $R^*(\varphi)$ は次のように形式化できる。確率モデルを考える。$\mathrm{X} \times \mathrm{c}$ を対 (X, j) のすべての集合とする。ここで $X \in \mathrm{X}$、$j \in \mathrm{c}$、$\mathrm{c} = \{1,2,...,C\}$ とする。（今 Borel 測度のような微妙な点は無視する。）$P(A, j)$ を $X \subset A$ とクラス j に関係のあるような確率分布と解釈する。学習用サンプルを $L = \{(X_n, j_n), n = 1, ..., N\}$ とし、$P(A, j)$ とは独立であるとする。予想機 φ を $\varphi(X, L)$ と表し、L から作られるとする。このとき $R^*(\varphi)$ は L と同じ形式の“新しいサンプル”に対する分類ミスを用いて、そのミスの確率として定義される。

定義６　対 (X, y)、$X \in \mathrm{X}$、$y \in \mathrm{c}$ を考える。いま“新しいサンプル”の確率分布を $P(A, j)$ とすると

（1）$P(X \in A, y = j) = P(A, j)$

（２）(X, y) は L と独立

このとき

$$R^*(\varphi) = P(\varphi(X, L) \neq y)$$

である。ここで$P(a)$は事象の起こる確率を表す。

4.2　交差検定

上記の真のエラー率を求めるのは困難である。本研究では$R^*(\varphi)$を推定する方法である交差検定（CV）を採用する。

本研究では限られたサンプルで予想機の真のエラー率を推定しなければならない。そして CV は次の２つの性質を持っていることがその採用の理由である。

１、　CV はサンプルのすべてを試行に使うので“けちな“手法である。つまり少ないサンプルにも有効であるとされている。

２、　現在 CV は予想機の正確さの評価手法して広く使われている。

CV はまずサンプル集合を二つに分ける。一方は学習用サンプルとして使い、もう一方はテスト用サンプルとして使う。テスト用サンプルにおいてその要素の測定値ベクトルは予想の必要な状態として使わる。また要素のクラスは予想の真偽を判断するのに使われる。

CV は分けられた部分集合の個数だけ試行を行う。そしてこれら試行の結果を集めて平均を取り、１つの推定エラー率を導く。CV は不安定性をもつ予想機に、より安定な推定を出すことも知られている。今サンプルをV個の部分集合に分ける V－foldCV について述べる。

あるサンプル$L = \{(X_n, j_n), n = 1, ..., N\}$がある。この$L$を$V$個の部分集合に分ける。このときそれぞれの部分集合は可能な限り、ほぼ同じ数の要素を持つようにする。この部分集合を$L_v \quad v = 1, ..., V$と表す。ある部分集合L_vがあるとき、このL_vに含まれない要素を含むLの部分集合を$L - L_v$と書く。ここで

$$\begin{aligned} \#L &= \#L_v + \#(L - L_v) \\ &= \sum_{v=1}^{V} \#L_v \\ &= N \end{aligned}$$

である。

予想機としてφをとる。このときL_vによるエラー率を$R^{ts}(\varphi^{(v)})$とすると

$$R^{ts}(\varphi^{(v)}) = \frac{1}{\#L_v} \sum_{(X_n, j_n) \in L_v} \delta(\varphi(X_n, L - L_v) \neq j_n)$$

となる。

ここでLすべてを使って予想機φに対するエラー率を集めたものを、真のエラー率$R^*(\varphi)$に対する推定値として採用する。これを$R^{CV}(\varphi)$と表すと

$$R^{CV}(\varphi) = \frac{1}{V} \sum_{v=1}^{V} R^{ts}(\varphi^{(v)})$$

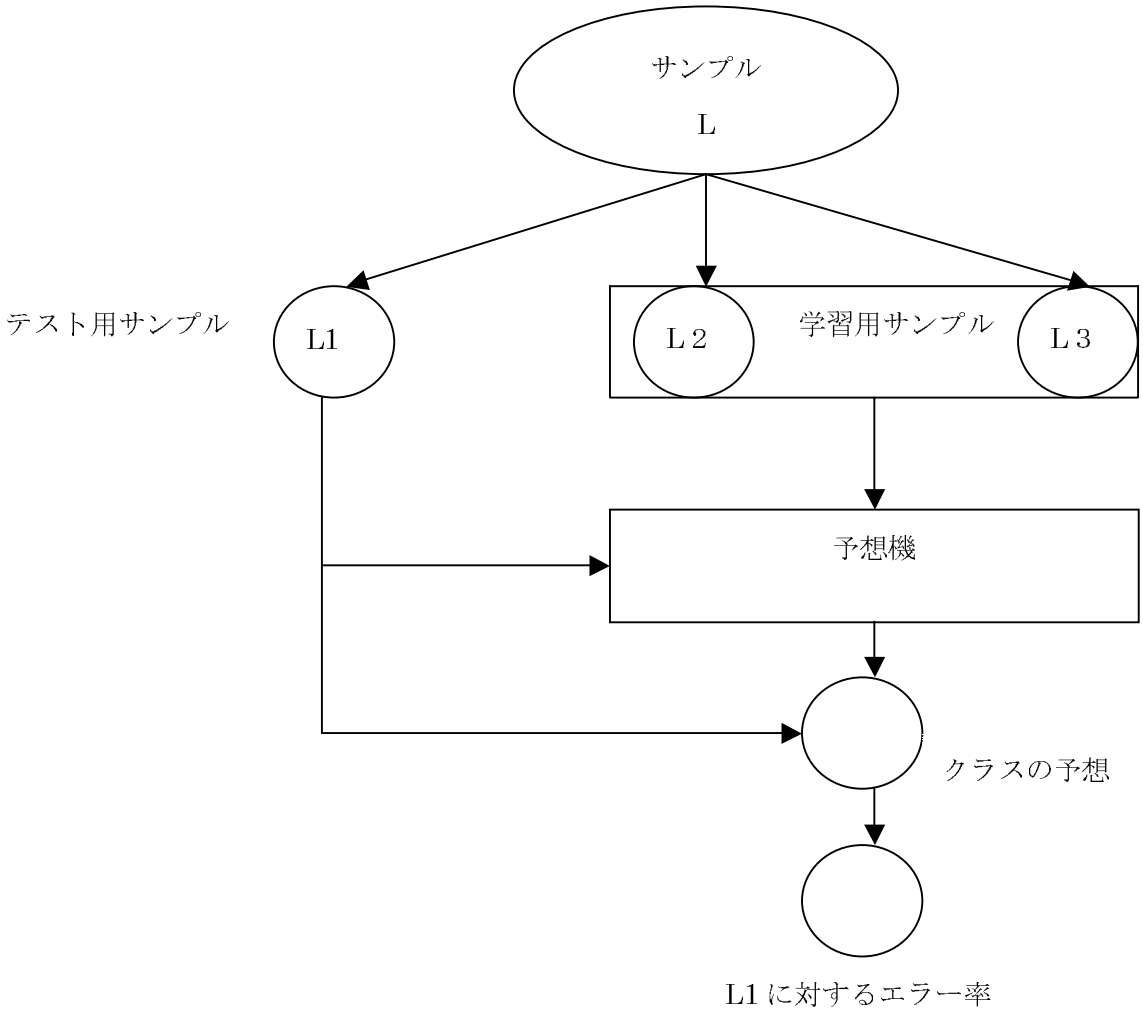
となる。この$R^{CV}(\varphi)$は$R^{ts}(\varphi^{(v)})$よりさらに$R^*(\varphi)$に近いことが期待される。

V-foldCV においてVをある程度大きい数にとると、ほとんどLの要素の数Nと同じくらいのサンプル数でφを組み立てることができる。そしてそのときの学習用サンプルの数はおよそ

$$N(1 - \frac{1}{V})$$

である。

次項にCVの概略を図示する。（図４．１）



・ここではサンプル L が L1、L2、L3 の 3 つに分けられる 3 - foldCV である。

・上の過程を L1、L2、L3 それぞれをテスト用サンプルとして 3 つのエラー率を計算する。

・それらエラー率の平均が CV による推定エラー率である。

図４．１：CV の概略図

第　５　章
システムの概要

5.1　概要

本研究ではバッギング手法を NNC、C4.5、NB のそれぞれに応用した　BNNC、BC4.5、BNB のシステムを開発した。３種類のプログラムは UNIX 上で C 言語を用いて作成した。BNNC、BC4.5、BNB はそれぞれブートストラップの回数 M を１に変更することにより、バッギングを応用していない NNC、C4.5、NB となる。３種類のプログラムにはそれぞれ CV についての記述があり、10-foldCV を行う。これら３種類のプログラムが生成するシステムを総称して BCV システム（バッギング CV システム）と呼ぶことにする。

すべての BCV システムはある１つのサンプルに対し、そのサンプルに対する１つのエラー率を標準出力に表示する。またすべてのシステムはサンプルの名前を Name としたとき Name.names、Name.sample という形式のファイルを入力とする。そして計算の過程で Name.training、Name.test、Name.data というファイルを生成する。

次項に BCV システムの概略を図示する。（図５．１）

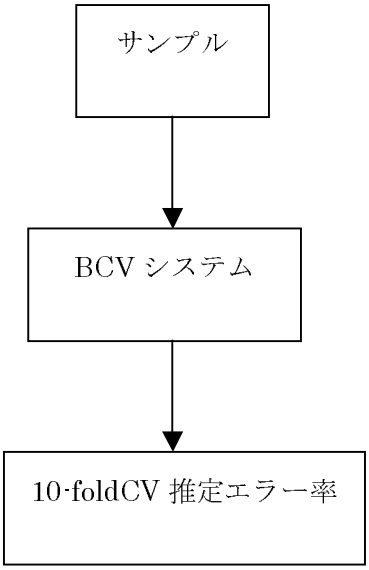


図５．１：BCV システムの概略

BCV システムにおいて１つのサンプルから１つの CV による推定エラー率を得るまでの流れを述べる。

１．あるサンプル $L^O = \{(X_n, j_n), n = 1, ..., N\}$ が予想機 φ による BCV システムに入力される。
２．L^O のうち約 9 $N/10$ を学習用サンプル L と表し、$N/10$ をテスト用サンプル $L^O - L$ と表す。
３．学習用サンプル L からブートストラップにより複製 $L^{(B)}$ を M 個作り出す。このとき M はブートストラップの回数である。
４．M 個の複製から M 個の分類機の列 $\varphi_1(L^{(B)}), ..., \varphi_M(L^{(B)})$ を作る。
５．今テスト用サンプル $L^O - L$ のある要素対を (X^{test}, j^{test}) とすると、それぞれのテスト用サンプルに対する予想機 $\varphi_1(X^{test}, L^{(B)}), ..., \varphi_M(X^{test}, L^{(B)})$ を得る。
６．予想されたクラス $\varphi_1(X^{test}, L^{(B)}), ..., \varphi_M(X^{test}, L^{(B)})$ のうちもっとも多いクラスをバッギング予想機による予想 $\varphi_B(X^{test}, L)$ とする。
７．テスト用サンプル $L^O - L$ のある要素対 (X^{test}, j^{test}) の真のクラス j^{test} と $\varphi_B(X^{test}, L)$ を比べて $L^O - L$ によるエラー率 $R^{ts}(\varphi^{(L^O - L)})$ を得る。

８．$R^{ts}(\varphi^{(L^O-L)})$を１０個集めてその平均を取る。これがサンプル$L^O$に対するCVによる推定エラー率$R^{CV}(\varphi)$である。

３種類のBCVシステムで予想機部位はそれぞれ次のように調達した。１．NNCについては自作したものを使用した。２．C4.5についてはQuilanによりC言語によるオリジナルのソースコードが出版されており、そのソースコードを使用した。３．NB は同じ研究室の学生が副テーマの研究用に作成したものを使用した。(ファイルの読み込み部分はすべてC4.5用入力ファイルと同じ形式で読み込みが可能なように作成した。)

BCVシステムの簡単なフローチャートを示す。(図5. ２)

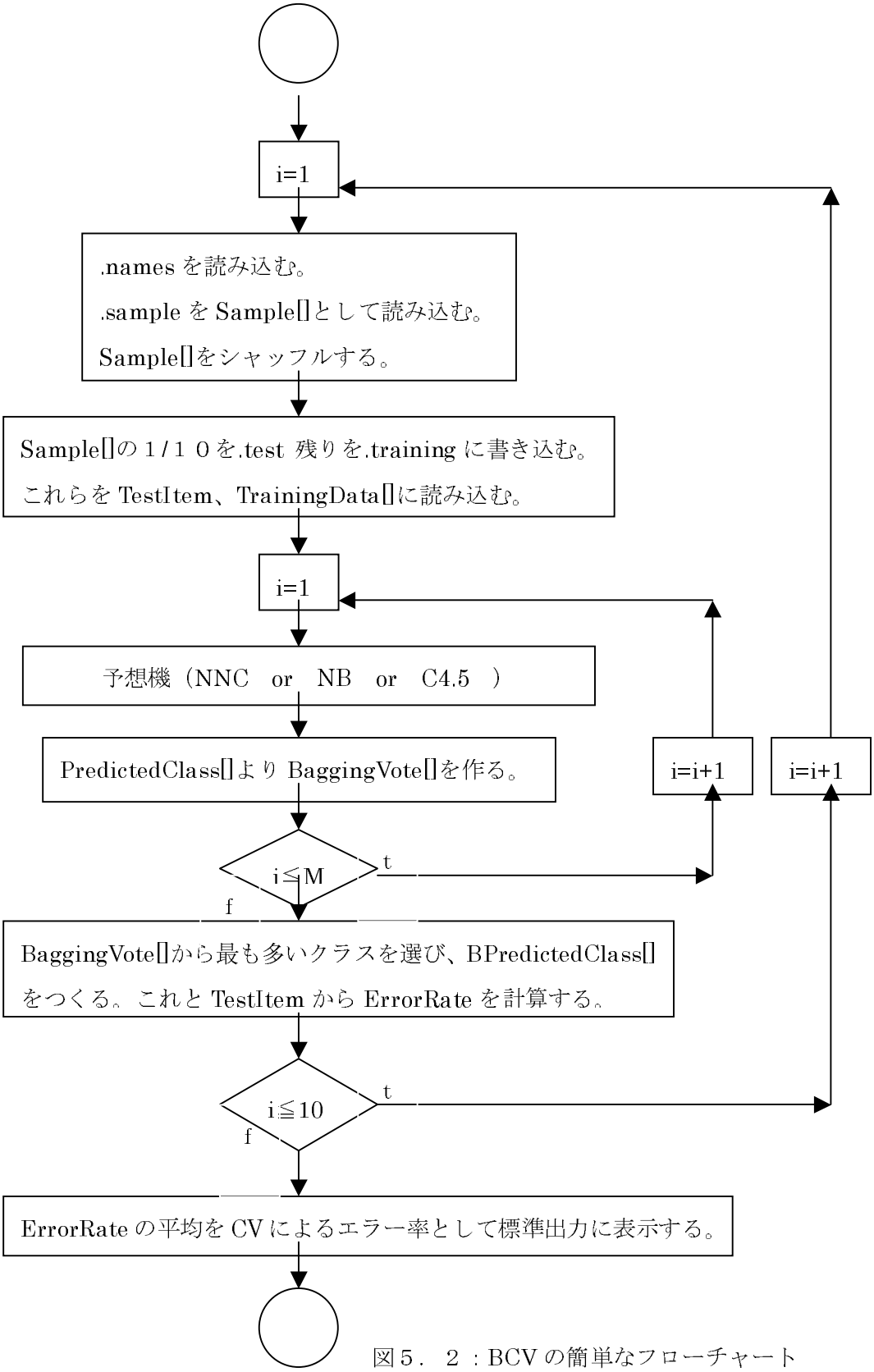


図５．２：BCV の簡単なフローチャート

5.2　NNC による BCV システム

本研究で作成した NNC 予想機は１０個の要素からなる近傍(Nearest)サンプル $L^{(N)}$ を選ぶ１０‐NN である。距離 D については状態 X の記述法により次の２種類を使用した。以下 2 種類の距離の定義を示す。

１．状態 X の要素がすべて実数値のとき。

距離関数として標準化したユークリッド距離を使用した。これを D_S と書く。いまある状態を $X=(x_k, k=1,...,K)$、学習用サンプルを $L=\{(X_n, j_n), n=1,...,N\}$、$X_n=(x_k^n, k=1,...,K)$ とする。２つの状態 X と X_n の標準化したユークリッド距離は

$$D_S(X, X_n)=\sqrt{\sum_{k=1}^{K}\left(\frac{x_k-x_k^n}{\hat{x}_k-\check{x}_k}\right)^2}$$

となる。ここで

$$\hat{x}_k=\max_{n\in\{1,...,N\}} x_k^n$$

$$\check{x}_k=\min_{n\in\{1,...,N\}} x_k^n$$

である。さらに

$$\left(\frac{x_k-x_k^n}{\hat{x}_k-\check{x}_k}\right)^2\le 1$$

といえる。

２．状態 X の要素がすべて離散値のとき。

距離関数としてハミング距離を使用した。つまり

$$D_H(X, X_n)=\sum_{k=1}^{K}\delta(x_k, x_k^n)$$

上のどちらでもないとき、つまり X に実数値 と離散値の両方が要素として存在するとき。このときは事前に実数値である要素を、離散化プログラムを使って離散値になおす。したがってすべての要素を離散値に変換してから２．のハミング距離を使うことになる。このとき実数値であ

る要素は１以上の離散値にふり分けられる。このときの離散化プログラムは研究室所有のものを使用した。この離散化プログラムはC4.5の評価関数に似た、情報論的エントロピーを利用したものである。その開発は参考文献[７]を元に行われた。

測定不能値についてはC4.5と同じのデフォルト値をあてた。

次項に簡単なフローチャートを示す。（図５．３）

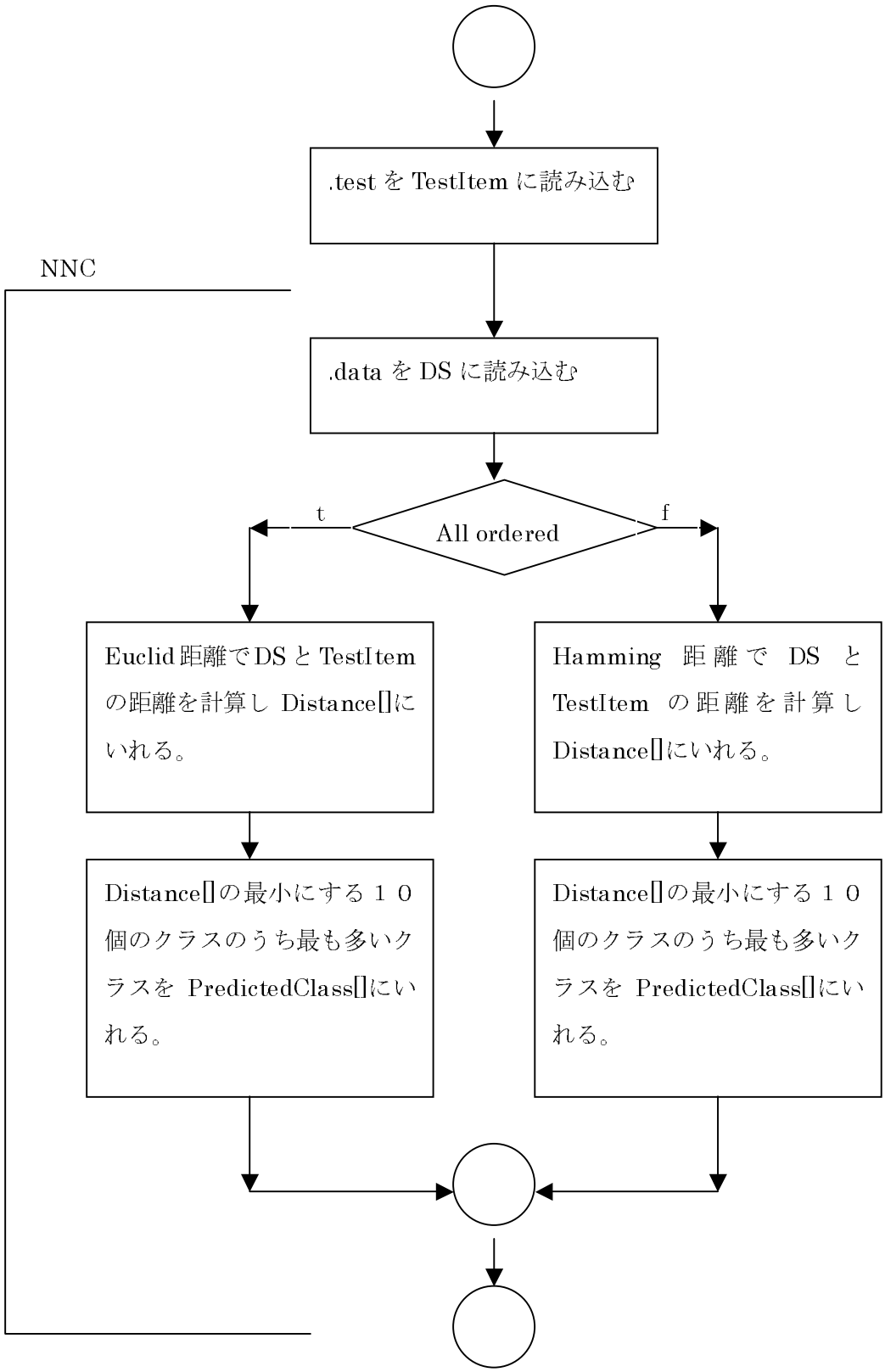


図５．３：NNC の簡単なフローチャート

BNNC システム全体について述べる。このシステムは大きく分けて 3 つの部位からなる。1 つは上で自作した NNC の部位、2 つ目は C4.5 のファイル入力部位、そしてバッギングと CV の部位である。これらの結合は直接ソースコードを操作することによりおこなった。

5.3　NB による BCV システム

本研究に採用した NB は属性値が実数値のとき、標準分布による確率を採用している。これによって実数値, 離散値どちらの属性が入っていても対応できるようになっている。しかしこれには議論の余地がある。本研究の目的はバッギングの有効性の確認にあるため、無難な選択をする。つまり予想機 φ_{NB} は純粋に 2 章で紹介した形式の計算をするのみである。したがってその入力は離散値を持つサンプルに限られる。このことは BNB による BCV システムが入力として離散値を持つサンプルしか受け付けないことを意味する。NB の入力部位は独自に開発されたものであるが C4.5 の形式に準拠している。BNB による BCV システムは作成者の違いから大きくバッギングと CV の部位、NB の部位に分けることができる。これらの部位は本研究の他のシステムと同様 C 言語のソースコードレベルで結合している。

5.4　C4.5 による BCV システム

C4.5 は木による探索を用いて分類規則を生成するが 2 種類の分類規則を生成する。一方は入力した学習用サンプルに極度に適した（過適応：over fitting）分類規則であり、もう一方は枝きり（prune）という作業により過適応を避けたものである。枝切りを行った分類規則は多少一般化された規則を生成し、こちらのほうが推定エラー率の値として低いものが得られるとされている。[１１] 本研究で作成した C4.5 による BCV システムでは、枝きりを行った後の分類規則のみを採用する。

C4.5 は入力するサンプルの要素の属性値として実数値 、離散値どちらの属性も受け付ける。

第　６　章
実験

６.1　サンプル

本研究の実験の目的は予想機の正確さを測ることである。予想機はサンプルにより違う正確さを見せる。したがってサンプルの数は多いほどよい。そして種類も豊富である必要がある。またシステムを評価する目的には奇抜なサンプルは必要ない。よく知られた有名なサンプルほど好ましい。これらの条件をすべて満たす Web 上のデータベースがある。それは UCI のホームページ上にあり（http://www.ics.uci.edu/~mlearn/MLRepository.html)、機械学習の専門家によって維持され、使用されている。

論文[8]を紹介する。ここでは３３個の予想機が３２個のサンプルにより評価されている。予想機の内訳は決定木によるものが２２、統計的手法が９、残りはニューラルネットによる予想機であった。評価の対象は予想機の正確さと計算速度であった。

本研究ではこの論文で扱われているデータを中心に、上記の UCI のデータベースから入手した。

６．１．１　サンプルの洗浄

本研究で扱う予想機 NNC と BNNC はサンプルが 2 種類の属性値をもつとき使用できない。したがって４．２で説明のとおり離散化プログラムによる変換を行った。NB と BNB は離散値サンプルしか扱えない。したがって使用するサンプルのすべての属性値が離散値でないときは離散化プログラムによる変換を行った。

本研究で扱うすべてのシステムは C4.5 の形式によるファイルしか受け付けないのですべての

サンプルはこの形式になるように変換した。

入手したサンプルの中には欠損値のあるものもあるし、ないものもある。

６．１．２　サンプルの特徴

一般的に、すべてのサンプルの属性値は連続値か離散値である（定義 5)。すべてのクラスは離散値である。クラスの数は２以上である。属性値が離散値のときその測定値は２種類以上であり、１種類のときは属性値がないことに等しい。

本実験では２０個のサンプルを使用した。以下サンプル個々の特徴を表にまとめる。

（表６．１）

サンプル名	欠損値	ケースの数	クラス数	連続値	離散値
bcw	無	683	2	9	0
house	無	506	3	12	1
voting	無	435	2	0	16
liver	無	345	2	6	0
heart	無	270	2	7	6
ta	無	151	3	1	4
crx	有	690	2	6	9
ech	有	131	2	5	1
hco	有	368	2	5	14
imp	有	205	5	13	9
att	有	1000	2	1	8
bio	有	209	2	5	0
der	有	366	6	1	33
edu	有	100	4	8	4
hur	有	209	2	6	0
inf	有	238	6	0	18
hab	有	306	2	3	0
hep	有	155	2	6	13
hyp	有	3163	2	6	9
lbw	有	189	2	2	6

表６．１：サンプルの特徴

6.2　実験の概要

本研究における実験は

1．BC4.5 と C4.5 の比較。

2．BNNC と NNC の比較。

3．BNB と NB の比較。

4．サンプル heart、BC4.5 におけるブートストラップ回数とエラー率の関係。

を求めることを目的としている。

本研究におけるすべての実験は特定の予想機，特定のサンプルに対する予想の正確さを計算する。この正確さは CV をもちいて推定エラー率により測る。

また本研究におけるすべての実験は大きく 2 つに分けることもできる。

1．　BCV システムのバッギングにおけるブートストラップの回数を一定にして，サンプルを種々のものに変える。この試行により、種々のサンプルに対する推定エラー率を求める。

2．　バッギングにおけるブートストラップの回数を変化させ，サンプルは特定のものを使用する。この試行により種々のブートストラップ回数に対する推定エラー率得る。

1.はサンプルに偏よらずに、2 つの予想機の正確さを比較することが目的である。2.はブートストラップの回数と推定エラー率の関係を示すことが目的である。

6.3　実験の手順

予想機の正確さを推定する実験の手順を以下に示す。これは本研究のすべての実験に共通する。

1．適当な BCV システムを作り上げる。このことは予想機φを選ぶことでもある。このとき予想機φの選択には次の選択肢がある。つまり C4.5、NNC、NB のうちどれかの選択、バッギング応用の可否、バッギングを応用したときブートストラップの回数、である。

2．洗浄されたサンプルの中から 1 つのサンプルを選択する。今これをLとする。

3．BCV システムを用いて選んだサンプルLに対する CV エラー率$R^{CV}(\varphi)$を計算する。

4．上の手順を同様に 10 回繰り返し、$R^{CV}(\varphi)$を 10 回集める。これらの平均値を　e　とする。この e をこの選択肢に対する推定エラー率（以下単にエラー率）として採用する。

６.４　実験の結果

６．４．１　BC4.5 と C4.5 の比較

C4.5 と BC4.5 を比較する。ブートストラップの回数は 20 である。サンプルの個数は 20 である。

サンプル名	e(C4.5)	e(BC4.5)	e(BC4.5)/e(C4.5)(%)
bcw	0.0457	0.0346	75.6
house	0.2751	0.2303	83.7
voting	0.0519	0.0470	90.7
liver	0.3691	0.3121	84.6
heart	0.2898	0.2273	78.4
ta	0.5700	0.5373	94.3
crx	0.1752	0.1265	72.2
ech	0.4347	0.3392	78
hco	0.1946	0.1604	82.4
imp	0.2713	0.2096	77.3
att	0.4361	0.4000	91.7
bio	0.1722	0.1316	76.4
der	0.0663	0.0358	54
edu	0.4833	0.4422	91.5
hur	0.2116	0.1670	79
inf	0.3331	0.2824	84.8
hab	0.3141	0.2826	90
hep	0.2191	0.1977	90.2
hyp	0.0124	0.0079	64.1
lbw	0.4121	0.3758	91.2
平均	0.2669	0.2274	81.5

表６．２：C4.5 と BC4.5 比較

６．４．２　BNNC と NNC の比較

NNC と BNNC を比較する。ここでブートストラップの回数は２０である。サンプルの個数は 20 である。

サンプル名	e(NNC)	e(BNNC)	e(BNNC)/e(NNC)(%)
bcw	0.0461	0.0461	100.0
house	0.2114	0.2114	100.0
voting	0.0728	0.0714	98.1
liver	0.4286	0.4223	98.5
heart	0.2308	0.2038	88.3
ta	0.5709	0.5381	94.3
crx	0.1509	0.1462	96.9
ech	0.3538	0.3901	110.2
hco	0.2111	0.2076	98.3
imp	0.3605	0.3484	96.6
att	0.4111	0.4042	98.3
bio	0.2810	0.2866	102.0
der	0.0389	0.0348	89.5
edu	0.4485	0.4342	96.8
hur	0.4118	0.4036	98.0
inf	0.3721	0.3665	98.4
hab	0.2936	0.2743	93.4
hep	0.1534	0.1534	100.0
hyp	0.0135	0.0134	99.2
lbw	0.3489	0.3404	97.6
平均	0.3037	0.2975	97.7

表６．３：NNC と BNNC の比較

６．４．３　BNB と NB の比較。

NB と BNB を比較する。ここでブートストラップの回数は２０である。サンプルのうち inf と hab については計算ができなかった。その理由は、サンプルの属性値をすべて離散値に変換する過程で、属性値のほとんどが意味のないものに変わってしまったことによる。したがってサンプル個数は 18 である。

サンプル名	e(NB)	e(BNB)	e(BNB)/e(NB)(%)
bcw	0.0306	0.0309	101
house	0.2724	0.2712	99.6
voting	0.0992	0.0988	99.6
liver	0.3788	0.3794	100.2
heart	0.1719	0.1769	102.9
ta	0.3536	0.3587	101.4
crx	0.1224	0.1231	100.6
ech	0.3320	0.3483	104.9
hco	0.1803	0.1795	99.6
imp	0.3295	0.3312	100.5
att	0.3707	0.3707	100
bio	0.0960	0.0965	100.5
der	0.0057	0.0080	140.4
edu	0.4217	0.4205	99.7
hur	0.1440	0.1446	100.4
hep	0.1624	0.1624	100
hyp	0.0102	0.0101	99
lbw	0.3035	0.3018	99.4
平均	0.2103	0.2118	102.8

表６．４：NB と BNB の比較

６．４．４　ブートストラップの回数とエラー率の関係

有効性が確認された BC4.5 において、ブートストラップの回数を２から 50 まで変化させた。サンプルは heart を使用した。

heart を使用した理由は以下である。１．このサンプルの減少率がサンプル中平均的な値を持つものの１つであったこと。２．属性値が連続値、離散値の両方に満遍なくあること。３．少なすぎない適当なケースの数を持つこと。

以下に図を示す。(図 6.1)

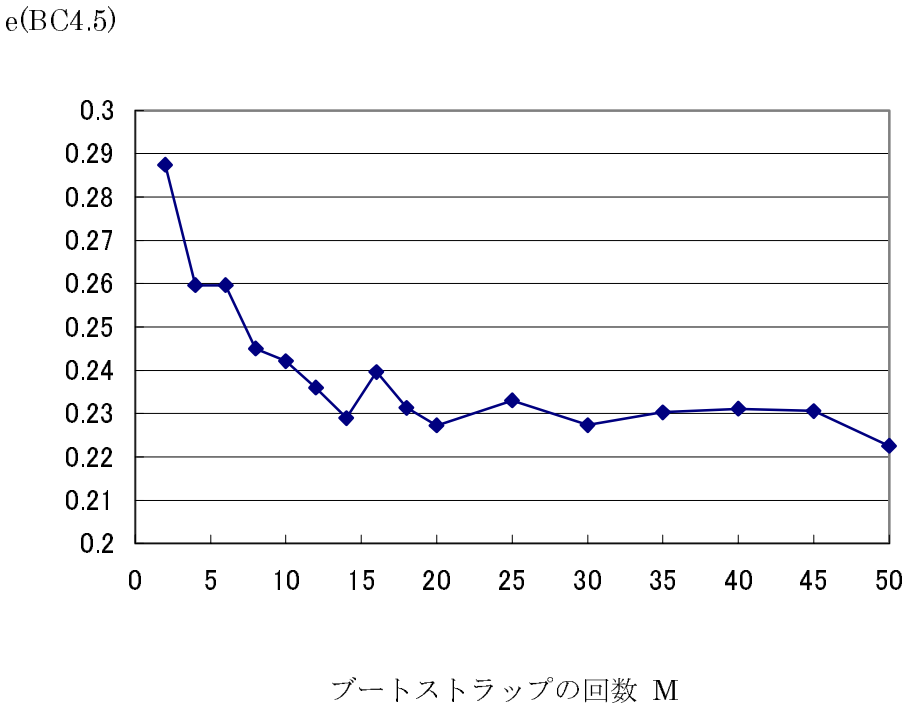


図６．１：ブートストラップの回数とエラー率の関係

6.5　考察

予想機φ_Bと予想機φのエラー率の比 e(φ_B)/e(φ)(%)はサンプルに対するバッギングの有効性を表すと考えることができる。この値が１００%からどのくらい減少しているかがバッギングの有効性を示す。今これを減少率と呼ぶ。多くのサンプルからの減少率を集めて平均値をだす。この平均値は予想機の持つバッギングの有効性を与える。以下 C4.5、NNC、NB の平均減少率を棒グラフにする。

（図 6.2）

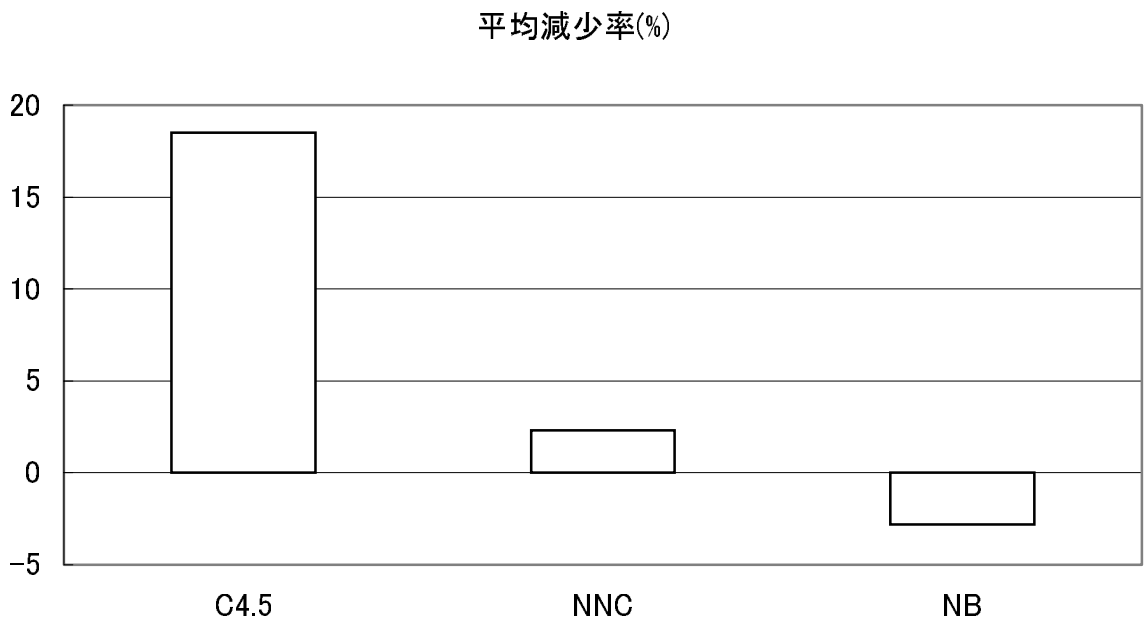


図 6．2：バッギングによる平均減少率と予想機の関係

図 6.2 において、C4.5 は 18.5%もの平均減少率を持つ。NNC はわずか 2.3%である。このことから C4.5 にはバッギングが有効であることが確認できた。そして NNC にはバッギングが有効でないことも確認できる。NB について、平均減少率は - 2.8%である。このことはバッギングを応用したことによりエラー率が全体的に増加したことを意味している。したがって NB にバッギング手法は有効でない。

BC4.5 と C4.5 の比較においてバッギングの有効なサンプルの特徴を探すことを試みた。しかしそのような特徴は発見できなかった。サンプルのうち der は BC4.5 と C4.5 の比較では減少率が 46%である。これはサンプル中最大の減少率である。また NB と BNB との比較において - 40.4%であり、エラー率がサンプル中最大に上昇している。さらに NNC と BNNC では 10.5%の減少率でありこれもサンプル中最大級の減少率である。サンプル der はどの比較においてもバッギングによりサンプル中最大の変化をもたらしている。der には次の 2 つの仮説が考えられる。1．der の e(C4.5)、e(BC4.5)はそれぞれ 0.0663、0.0358 とサンプル中最大級の小ささであり、誤差の拡大が起こっている。 2．e(C4.5)、e(NNC)、e(NB)がそれぞれ 0.0663、0.0389、0.057 と小さく、予想機による予想の効果が高いことから、バッギングによるエラー率の変化も大きい。したがって予想機の特徴を測るのに適したサンプルである。本研究では 2 つの仮説を確かめることはできないので、サンプル der が珍しい振る舞いをしたことのみ述べる。

サンプル der を除いた NB と BNB の比較では減少率の最大値と最小値がわずか 3.9%の幅の中にすべて集まっている。これに比べて NNC においては同様の幅が 21.9%であり、C4.5 では 30.2%である。このことは NB が、不安定性とは別の概念として、安定な予想機であることを示す。このことは NB が単純な確率の計算による予想機であることが原因と考えられる。

図 6.1 ではブートストラップの回数が 20 から 25 の付近からそれ以上ではエラー率が一定である。このたびの実験は Breiman の“２５回以上のブートストラップは労働の無駄だ”[1]という主張を支持する結果となった。

第　7　章
結論

7.1　結論

バッギング手法は有効である。本研究の実験の結果、バッギング手法の有効性が確認された。しかしどのような予想機についても有効であるとはいえない。このことはバッギングを応用したBNB の平均推定エラー率のほうが大きくなってしまうということが顕著に表わしている。

Breiman の発見的定義ではバッギングが有効であるかどうかは判断できない。したがって現在あるたくさんの予想機にバッギングが有効であるか否かは本研究のように実験してみないとわからない。いま１．いくつかの木による予想機、ニューラルネットによる予想機について、バッギングが有効であること、２．NNC ほかいくつかの予想機については無効であること、が実証されている。これに加えてまたひとつ NB についてはバッギングが有効でないことが実証された。

今日 NNC は予想機の比較相手としてよく引き合いに出されている。しかし Breiman はバッギングの有効な予想機とバッギングが有効でない NNC を同じように比較することは好ましくないとしている。乱立する予想機はその評価が難しい。バッギングが有効であるかどうかは予想機を特徴付ける基準となり、予想機そのものの理解を助ける。

近年の技術の発達に伴って計算機を取り巻く環境が整いつつある。バッギングにより計算時間が多少多くかかろうとも、バッギングによるエラー率の向上は無視できない。

謝辞

本研究を行うにあたり、熱心な指導と助言を賜りました Ho Tu Bao 教授に心から感謝いたします。多くの親切なご指導を受け賜りました中森義輝教授に感謝いたします。研究の指針について助言を賜りました石崎 雅人助教授に心から感謝いたします。多くのご指導を受け賜りました下嶋 篤助教授に心から感謝いたします。研究に必要な環境について多くの助言をいただいた Nguyen Ngoc Binh 助手に感謝いたします。

最後に研究に関しての相談にのって頂き、協力をしていただいた Nguyen Dung Trong さん、Ho－石崎研究室の皆様に感謝いたします。

参 考 文 献

[1] Breiman, L. 1996a. Bagging predictors. Machine Learning 24:123-140.

[2] Breiman, L. 1996b. Heuristics of instability and satabilization in model selection. The Annals of Statistics 1996, 24, pp2350-2383

[3] Breiman, L. 1998. Randomizing outputs to increse prediction accuracy. Technical Report 518, Statistics Department, University of California at Berkely, available at www.stat.berkeley.edu

[4] Breiman, L. 1999. Using adaptive bagging to debias regressions. Technical Report 518, Statistics Department, University of California at Berkely, available at www.stat.berkeley.edu

[5] Friedman, J. H. 1996. On bias, varance, 0/1 - loss, and the curse-of-dimensionality. Technical report, Department of Statistics and Stanford Linear Accelerator Centor, Stanford, CA.

[6] Domingos, P. 1997. Why Dose Bagging Work? A Bayesian Account and its Implications. Proceedings, Third International Conference on Knouledge Discovery & Data Mining, AAAI Press.

[7] Fayyad, U.M. & Irani, K.B. 1992. On the handling of continuous-valued attribute in decision tree generation. Machine Learning, 8, 87-102.

[8] Lim, T.S. & Loh, W.Y., 1999, A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms, Machine Learning.

[9] Quinlan, J.R., 1998, Bagging, boosting, and C4.5,

[10] Itchell, T.M., 1997 Machine Learning, McGraw Hill. ISBN 0-07-042807-7.

[11] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., Classification and regression tees, WADSWORTH&BROOKS/COLE ADVANCED BOOKS & SOFTWARE Pracific Grove, California.

[12] Berry, MJ.A., Linoff, G., 1997, Data Mining Techniques For Marketing, Sales, and Customer Support, WILEY COMPUTER PUBLISHING.

[13] J.R.キンラン著，古川康一監訳，AIによるデータ解析（Programs for machine learning）トッパン，１９９５。

[14] H.M.ダイテル＋P.J.ダイテル著，小嶋隆一訳，Computer Science Textbook C言語プログラミング，プレインテスホール出版，１９９８。

[15] 有本卓著，情報・確率・エントロピー，森北出版，１９８０。

[16] アブラムソン.N著，宮川洋訳，情報理論入門，昭和４４年発行。