RF-005

数値データストリームからの決定木導出
Decision Tree Induction from Numeric Data Stream

西村 聖† 寺邊 正大‡ 橋本 和夫†
Satoru Nishimura Masahiro Terabe Kazuo Hashimoto

1 はじめに

近年，インターネットの普及やセンサ技術の発達にともない，センサネットワークなど大量の電子化データが断続的に得られる環境が普及している．このような環境から得られる，断続的に，異なる間隔で到着するデータ系列はデータストリーム [1] とよばれており，データストリームからの分類学習が注目されている．

データストリームでは，時間を経るにしたがい同じ属性値を持つ事例の属するクラスが変化することがある．時間経過にともなう学習対象 (コンセプト) と，対象から得られるデータ傾向の変化をコンセプトドリフト [2] とよぶ．コンセプトドリフトが発生すると，コンセプトドリフト以前の古い事例で学習された分類器の精度は一般に劣化する．したがって，コンセプトドリフトを含むデータストリームから精度の高い分類器を学習するためには，分類器を最新のコンセプトへと追従させなければならない．

これまでに多くの分類器学習手法が提案されている．そのうち決定木学習手法は，学習が高速であり，かつ導出される分類器の表現が人間にとって理解しやすいものであることから，データマイニングの実務においてよく利用される手法の 1 つである．

データストリームからの逐次更新型の決定木導出手法として，Hoeffding Tree アルゴリズム [5, 7, 8, 9, 11] が提案されている．このうち，特にコンセプトドリフトを含むデータストリームに対応するものとして，CVFDT[9]，$\text{CVFDT}_{\text{NBC}}$[11]，UFFT[8] がある．

これまでに提案されている Hoeffding Tree アルゴリズムの多くは名義データストリームを前提としており，数値データストリームの取り扱いに関する研究は少ない．決定木導出手法で数値属性を取り扱う主な方法として，(1) 全ての数値属性値を異なる属性値として扱う方法，(2) 離散化する方法，(3) 数値属性値の分布を仮定する方法，がある．これら数値属性の取り扱い方法を，数値属性のモデル化法とよぶこととする．

Pfahringer らは，コンセプトドリフトがなく記憶容量に制限のある条件では，数値属性値の分布をガウス分布と仮定するモデル化法が Hoeffding Tree アルゴリズムに適していることを示している [4]．しかし，コンセプトドリフトを含む場合に適した数値属性のモデル化法については，これまでに検討されていない．

コンセプトドリフトを含む数値データストリームから分類器を学習する場合，コンセプトドリフトにより最適な数値属性のモデルが変化する場合がある．ここで数値属性のモデルとは，離散化によるモデル化法であれば離散化する際の閾値，分布を仮定するモデル化法では分布を表すパラメータのことである．したがって，数値データストリームから高精度の決定木を導出するためには，(1) コンセプトドリフトを検出し，(2) 検出されたコンセプトドリフトに対応し，数値属性のモデルを変更する必要がある．特に，対応へのきっかけとなるコンセプトドリフトの検出が重要となる．

CVFDT などの Hoeffding Tree アルゴリズムでは，Hoeffding の不等式を用いてコンセプトドリフトの検出が行われる．しかし，CVFDT は名義データストリームを前提としているため，Hoeffding の不等式にもとづいたコンセプトドリフト検出手法の，数値データストリームにおける有効性については検証の必要がある．また，その他のコンセプトドリフトの検出手法として，UFFT で採用されている分類精度の時間変化にもとづいた検出手法がある．

本論文では，コンセプトドリフトを含む数値データストリームから Hoeffding Tree アルゴリズムを用いて決定木を導出する際の，(a) 数値属性のモデル化法と，

†東北大学大学院　情報科学研究科，Graduate School of Information Sciencies, Tohoku University
‡株式会社　三菱総合研究所，Mitsubishi Resarch Institute, Inc.

(b) コンセプトドリフト検出手法について，各手法の性能を実験を通じて比較し，最適な手法を明らかにする．

実験に用いる Hoeffding Tree アルゴリズムには，代表的な手法である CVFDT とその改良である $\text{CVFDT}_{\text{NBC}}$ を用いる．$\text{CVFDT}_{\text{NBC}}$ の葉ノードにはナイーブベイズが導入されているため，CVFDT とは最適なモデル化法が異なる可能性がある．

本論文の構成は，次のとおりである．まず，第 2 章では Hoeffding Tree アルゴリズムについて説明を行う．その後，第 3 章で数値属性のモデル化に関する説明を行い，第 4 章でコンセプトドリフト検出手法について述べる．そして，第 5 章で実験を通じて Hoeffding Tree アルゴリズムに最も適した数値属性のモデル化法とコンセプトドリフト検出手法について分析する．最後に，第 6 章で結論を述べる．

2 データストリームからの決定木導出

本章では，Hoeffding Tree アルゴリズムに関する説明と，主な Hoeffding Tree アルゴリズムの特徴を比較する．

2.1 Hoeffding Tree アルゴリズム

データストリームから決定木を導出する場合，データストリームの短い間隔で新しい事例が到着し，かつ累積する事例数が大量となるという特徴に対応しなければならない．Hoeffding Tree アルゴリズムはメモリ消費量と処理時間を減らすため，事例そのものを決定木中に保持するのではなく，事例のクラスと属性値の同時出現頻度のみを各ノードで保持する．

Hoeffding Tree アルゴリズムは最初はルートノードのみの決定木を，新規事例の到着に合わせて更新する．ノードに頻度情報が蓄積し，後述する評価基準が充たされた場合にノード分岐は行われる．一般的には，ノード分岐の判断に用いる事例数が多いほど，信頼性が高い分岐属性の選択を行うことができる．しかし，十分な信頼性が得られるまで事例が蓄積するのを待つとノード分岐ができず，決定木が成長しにくいという問題も生じる．よって，十分に成長した，分類精度の高い決定木を得るためには，必要最低限の事例数で信頼性の高いノード分岐を行うことが期待される．

Hoeffding Tree アルゴリズムは Hoeffding の不等式を用いて．ノードにおける分岐属性判定の信頼度を評価する．あらかじめ分岐属性判定の信頼度基準を与えておけば，その基準を充たす最小限の事例が蓄積された時点で分岐属性が選択され，分岐属性の信頼性を担保しながら，決定木を成長させることができる．

以下では Hoeffding の不等式を用いたノード分岐について説明する．値域が R の数値変数 r を n 回独立に観測し，その平均が $\bar{r}$ のとき，Hoeffding の不等式は $1-\delta$ の確率で変数 r の真の平均が $\bar{r}-\epsilon$ より大きくなることを保証してくれる．ここで，ε は以下のように定義される．

$$\epsilon = \sqrt{\frac{R^2 \ln\left(\frac{1}{\delta}\right)}{2n}} \tag{1}$$

Hoeffding の不等式を用いると，$\overline{\Delta G()} = \overline{G(X_a)} - \overline{G(X_b)} > \epsilon$ のとき，属性 X_a でノードを分岐することが $1-\delta$ の確率で正しいことが分かる．ここで $G()$ は情報利得関数，X_a は情報利得を最も大きくする属性，X_b は情報利得を 2 番目に大きくする属性である．

コンセプトドリフトが発生すると，ノードの最適な分岐属性が変化することがある．そのため，Hoeffding Tree アルゴリズムは，コンセプトドリフト検出手法を用いてコンセプトドリフトを検出し，決定木中の最新のコンセプトへ追従できていない部分を修正し，分類器全体を常に最新のコンセプトへと追従させる．

2.2 主な Hoeffding Tree アルゴリズムの特徴

主要な Hoeffding Tree アルゴリズムである，VFDT[5]，CVFDT[9]，$\text{CVFDT}_{\text{NBC}}$[11]，UFFT[8] について．各手法の特徴を表 1 に示す．

VFDT，CVFDT，$\text{CVFDT}_{\text{NBC}}$ は単一の決定木からなるが，UFFT は複数の決定木によるアンサンブルを構成する．VFDT，CVFDT では事例が割り当てられた葉ノードの多数クラスにもとづき事例の分類は行われる．一方，$\text{CVFDT}_{\text{NBC}}$，UFFT では各葉ノードに導入されたナイーブベイズにもとづき分類が行われる．

VFDT と CVFDT は，論文中では名義データストリームを対象としたアルゴリズムとなっており，数値データストリームへの対応については発表されていない．しかし，VFML[10] で公開されている VFDT と CVFDT のプログラムには，数値属性を取り扱うための改良が加えられている．具体的には，VFDT はユー

表 1: 主な Hoeffding Tree アルゴリズムの特徴.

特徴	分類方法	決定木のタイプ	数値属性のモデル	コンセプトドリフト検出
VFDT	多数クラス	単一の決定木	離散化	(検出不可)
CVFDT			網羅的	Hoeffding の不等式
$CVFDT_{NBC}$	ナイーブベイズ		(取扱不可)	
UFFT		アンサンブル	ガウス分布仮定	分類精度の時間変化

ザが定義する事例数にしたがって数値属性を離散化する方法をとっている．一方，CVFDT は決定木中の各ノードで数値属性値を網羅的に全て保持する．

$CVFDT_{NBC}$ は，CVFDT の葉ノードにナイーブベイズを導入した手法である．ナイーブベイズを実行するためには，ある属性値が発生する確率を計算しなければならない．しかし，CVFDT のように全ての属性値をただ保持するだけでは，ナイーブベイズ実行のための確率は計算できない．したがって，$CVFDT_{NBC}$ は現状では数値属性へは対応していない．

一方，$CVFDT_{NBC}$ と同じくナイーブベイズが導入された UFFT は，数値属性がガウス分布にしたがうことを仮定することにより，数値属性のモデル化を行っている．ガウス分布を仮定することにより，ナイーブベイズ実行のための確率計算は容易となる．

CVFDT，$CVFDT_{NBC}$ は定期的に決定木の各ノードで Hoeffding の不等式を用いて最適な分岐属性の再判定を行い，コンセプトドリフトの検出を行う．一方，UFFT は決定木の各ノードに導入されたナイーブベイズの分類精度に着目しコンセプトドリフトの検出を行う．

本論文では，コンセプトドリフトを含むデータストリームからの代表的な決定木導出手法である CVFDT と，CVFDT の葉ノードへナイーブベイズを導入した $CVFDT_{NBC}$ を用い，最適な数値属性のモデル化法とコンセプトドリフト検出手法を実験により明らかにする．

3 数値属性のモデル化法

数値属性を含むデータストリームから決定木を導出する際には，処理時間とメモリ消費量の観点から，数値属性をモデル化し，決定木中に保持する情報量を低減する工夫が必要である．

Hoeffding Tree アルゴリズムにおける数値属性のモデル化法として，次の 3 つの方法が考えられる．第 1 の方法は，CVFDT[10] や $VFDT_C$[7] のように，決定木中の各ノードで数値属性値を完全に保存する方法 (網羅法)，第 2 の方法は，VFDT[5] のように数値属性を離散化し，離散化区間ごとに事例の出現頻度を数える方法 (離散化法)，第 3 の方法は，数値属性値の分布を仮定する方法である．一般的には，ガウス分布を仮定する (ガウス分布法)．以下では，3 通りの数値属性のモデル化法について説明する．

3.1 網羅法

CVFDT に実装されている数値属性のモデル化法は，各ノードで数値属性値をそのまま保持する．ノード分岐の際は，情報利得が最大となる値で属性値を 2 つに分割，離散化する．各ノードで数値属性値をそのまま保持するため，正確なノード分岐が可能となるが，決定木の各ノードに保持される情報量が多く，メモリ消費や処理時間の面で問題が発生する．一方，数値属性値をそのまま保持するため，コンセプトドリフトが発生した際も，これに追従し，適切なノード分岐を行うことができる．

$CVFDT_{NBC}$ のように，決定木の葉ノードへナイーブベイズを導入した Hoeffding Tree アルゴリズムの場合，事例の分類には葉ノードに導入されたナイーブベイズが用いられる．ナイーブベイズ実行のためには，ある属性値が取りうる確率を事例から推定しなければならない．しかし，CVFDT のように数値属性値をそのまま保持した場合，ナイーブベイズ実行前に離散化などを行わなければナイーブベイズ実行のための確率を計算できないという問題がある．

3.2 離散化法

VFDT のように，ノードへ割り当てられた事例から数値属性を離散化するモデル化法がある．離散化後は数値属性値を保持する必要はなく，名義属性と同様の扱

いができるため決定木中に保持する情報量を減らすことができる．

これまでに多くの離散化手法が提案されているが，Entropy-Based Discretization は導出される決定木の精度を高めることができるということが知られている [6]．したがって，本論文でも離散化手法として Entropy-Based Discretization を採用する．数値属性を離散化する際は，離散化の粒度が学習される決定木や，葉ノードに導入されたナイーブベイズの分類精度に影響する．離散化の粒度が細かいほうが複雑なクラス境界を学習できるが，粒度が細かすぎると過学習の問題を起こす．一般には，最小記述長原理にもとづき離散化の粒度を求める方法が良いとされているが，本論文では簡単のために数値属性を 2 分割することとする．

逐次更新型の手法では，新規事例が到着するたびに決定木の状態が変化していく．離散化を行うタイミングにより離散化の結果が大きく異なる場合もあるため，適切な離散値を求めるのに必要な最小限の事例数に対する議論が必要であるが，本論文ではユーザが定義する k 個の事例にしたがって数値属性を離散化することとし，$k = 300$ とする．

つまり，数値属性の離散化は，(1) 決定木の葉ノードに k 個の事例の情報を蓄える，(2) 各数値属性を情報利得を最大とする属性値で 2 つの区間に離散化する，(3) 離散化後はクラスごとに各属性で閾値との大小の度数のみを保持する，という手順で行われる．

コンセプトドリフトにより適切な離散化が変化する可能性があるが，離散化後は離散化の閾値との大小の度数しか保持しないため，適切な離散値を求め直すことができない．したがって，コンセプトドリフトを検出手法により検出し，再びゼロから離散化を行う必要がある．

3.3 ガウス分布法

UFFT[8] のように数値属性がガウス分布にしたがうことを仮定する数値属性のモデル化法がある．各ノードでガウス分布の形状を特定するのに必要な，クラスごとの属性値の平均と分散のみを事例から推測するため，処理時間が少なく，メモリ消費量も少ない．

ある数値属性で，ノード分岐に最も適した属性値は以下のようにして推測される．$P(A)$ をクラス A の事例の出現確率，x_A をクラス A に属する事例 x の属性値，$\overline{x}_A$ を x_A の平均，$\sigma^2(x_A)$ を x_A の分散，$\phi(x_A, \sigma(x_A))$ を平均 $\overline{x}_A$，分散 $\sigma^2(x_A)$ のガウス関数．式 (2) を満たす解がノード分岐に最も適した属性値となる．

$$P(A)\phi(\overline{x}_A, \sigma(x_A)) = P(B)\phi(\overline{x}_B, \sigma(x_B)) \quad (2)$$

式 (2) を満たす解 x_1, x_2 のうち，式 (3) を用い $\overline{x}_A$，$\overline{x}_B$ に近いほうの解を用いる．

$$Min(|x_1 - \overline{x}_A| + |x_1 - \overline{x}_B|, |x_2 - \overline{x}_A| + |x_2 - \overline{x}_B|) \quad (3)$$

数値属性ごとに式 (3) を満たす属性値で分岐した場合の情報利得を計算し，情報利得が最大となる属性を分岐属性とし，2 つの子ノードに分岐する．実際の分布がガウス分布とは異なる場合には不適切なノード分岐をしてしまう可能性があるが，一般的にはガウス分布を仮定しても問題ないことが知られている [4]．

コンセプトドリフトが発生した際も，ガウス分布のパラメータを修正し，モデルを追従させることができるため，正確なノード分岐が可能である．また，ガウス分布の組み合わせにより複雑なクラス境界を学習できるため，分類精度の高いナイーブベイズが学習可能である．

4 コンセプトドリフト検出手法

コンセプトドリフトが発生した場合，分類精度を維持するために，数値属性のモデルを最新のコンセプトへと追従させなければならない．モデル自身にコンセプトドリフトへの追従性がなかったり，既存のモデルを修正するよりも再モデル化したほうが精度や処理速度の面で優れる場合には，コンセプトドリフトを検出し，再度数値属性のモデル化を行わなければならない

以下では Hoeffding Tree アルゴリズムで用いられている 2 つのコンセプトドリフト検出手法の説明を行う．

4.1 Hoeffding の不等式にもとづく手法

コンセプトドリフトが発生した場合，分類精度を維持するために決定木を最新のコンセプトへと追従させなければならない．CVFDT[9] や $\text{CVFDT}_{\text{NBC}}$[11] は，定期的に Hoeffding の不等式を満たす属性の再判定を行う．現在の分岐属性と Hoeffding の不等式を満たす属性が異なる場合，コンセプトドリフトが発生したとみなし，最適な分岐属性にもとづいた代替木を生成する．

代替木が十分に成長したら古い部分木との入れ替えを行い，決定木を最新のコンセプトへと追従させる．

Hoeffding の不等式にもとづくコンセプトドリフト検出手法は，現在の数値属性のモデルにしたがった場合，どの属性が最もノード分岐に適するかを調べる．したがって，コンセプトドリフトにより数値属性の最適なモデルが変化した場合，コンセプトドリフト以前の古いモデルで分岐に最適な属性を調べていることになり，コンセプトドリフトの検出性能に疑問が残る．

4.2 分類精度の時間変化にもとづく手法

UFFT[8] で用いられているコンセプトドリフト検出手法について説明する．Gama らは，分類精度の時間変化を用いたコンセプトドリフト検出方法 [12] を提案している．一般的には，学習に用いる事例数が増えるほど分類精度は向上する．したがって，学習事例数が増えたのに分類精度が低下した場合，コンセプトドリフトが発生したと予想される．

Hoeffding Tree アルゴリズムは新規事例の到着のたびに新規事例を決定木で分類し，事例が通過した各ノードで頻度情報の更新を行う．Gama らの手法は，学習時に事例が通過した各ノードで頻度情報を更新すると共に事例の分類を行い，ノードごとに分類精度の平均，標準偏差の更新を行う．そして，分類精度が統計的に確かに低下した場合，コンセプトドリフトが発生したとみなす．

現在の数値属性のモデルがコンセプトドリフトに追従できていない場合，数値属性のモデル化法に関わらず分類精度は低下する．したがって，Hoeffding の不等式にもとづく手法とは異なり，分類精度の時間変化にもとづく手法は数値データストリームからでも正確にコンセプトドリフトを検出できる．しかし，学習時に事例が通過した各ノードで分類を行うため，Hoeffding の不等式にもとづく手法の簡易な処理と比べて，学習時間が増加することが予想される．

あるノードでコンセプトドリフトを検出した場合，UFFT では CVFDT のように代替木を生成せず，当該ノードより下位の部分木の枝刈りを行う．UFFT は複数の決定木によりアンサンブルを構成するため，急激に決定木を枝刈りしてもアンサンブル分類器全体としての影響は少ない．しかし，CVFDT は単一の決定木であるため，急激に決定木の枝刈りすると決定木のサイズが極端に小さくなる可能性もある．そこで，本論文ではあるノードでコンセプトドリフトを検出した場合，従来の CVFDT と同じく代替木を作成する．

5 実験

5.1 実験方法

CVFDT，$\text{CVFDT}_{\text{NBC}}$ に対して，(1) 数値属性のモデル化法，(2) コンセプトドリフトの検出手法，の 2 点を変更した場合の分類精度，学習時間，メモリ消費の変化を分析した．

実験データは Hulten らのデータストリーム作成方法 [9] を参考に作成した．人工的に 300 万の学習事例からなるデータストリームを作成し，学習事例 1 万毎にテスト事例 1 万を与え分類精度などを調べた．学習事例 1 万毎にテストを行うため，合計で 300 回のテストが行われ，次節に示す結果はこの 300 回のテストでの平均値である．なお，コンセプトドリフトは学習事例 5 万毎に発生させてある．

5.2 実験データ

作成するデータストリームの各事例 $\vec{e}$ は，d 個の属性とクラスからなる．各属性は，[0,1] の間でいずれかの値をとり，クラスは positive，または negative の 2 クラスである．なお，属性数は $d = 10$ とする．

事例空間として d 次元の属性によりはられる $[0,1]^d$ の空間を考える．実験データとして生成する N 個の事例は，この空間上に一様に分布させる．この事例空間を，以下の式 (4) により部分空間に分割し，ラベル付けを行う．

$$0.2jw_0 \leq \sum_{i=1}^{d} w_i e_i \leq 0.2(j+1)w_0. \qquad (4)$$

ここで，w_i は事例 $\vec{e}$ の i 番目の属性に与えられた重み，w_0 は定数，j は自然数である．各事例 $\vec{e}$ について，式 (4) を満たす j が偶数ならば positive にラベル付けされ，奇数ならば negative にラベル付けされる．また，ラベルを付与したのち，ノイズとして 5%の事例のクラスを反転させる．

全ての重み w_i の初期値は $w_i = 0.2$ とする．ただし，w_0 については $w_0 = 0.25d$ とする．ここで，重みベクトル $\vec{w}$ を変化させることにより，クラス境界をゆるや

かに変化させることができる．その結果，事例空間中の同じ点のクラスが変化する．すなわち，コンセプトドリフトを発生させることが出来る．

今回のデータでは，学習用事例を 5 万生成するごとに，ドリフトさせる属性の重み w_i に $0.007d\sigma$ を加えてコンセプトドリフトを発生させる．ここで，σ_i は重み w_i の変化の方向を意味しており，初期値は $\sigma_i = 1$ とする．コンセプトドリフトを発生させるたびに，5%の確率で σ_i の符号を反転させ，w_i の増減を反転させる．また，w_i の定義域は $0 \leq w_i \leq 0.2d$ とし，この範囲から外れる場合には σ の符号を反転させ，$0 \leq w_i \leq 0.2d$ の範囲に収まるようにする．

図 1 に，コンセプトドリフトにより，クラス境界がゆるやかに変化する様子を示す．重みベクトル $\vec{w}$ はクラス境界の形状に関連するパラメータである．一方，w_0 はクラス境界間の距離に関連するパラメータである．$\vec{w}$ と w_0 により，学習対象であるコンセプトの複雑さが決定する．

重みベクトル $\vec{w}$ を変化させると，クラス境界が変化する．図 1 の右図の点線は古いクラス境界を表しており，実線は新しいクラス境界を表している．古いクラス境界と，新しいクラス境界に囲まれた領域がコンセプトドリフトによりクラスが変化した領域であり，全属性空間に対する，クラスが変化した領域の占める割合を drift level とよぶ．

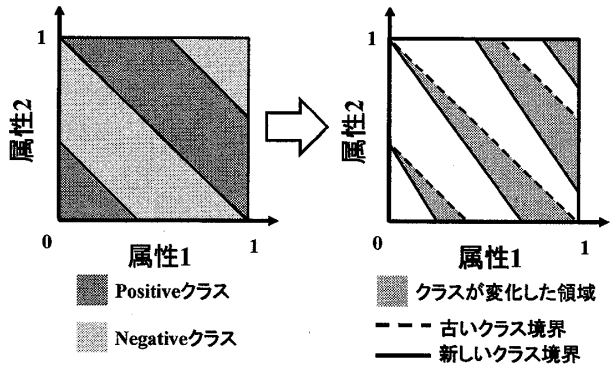


図 1: コンセプトドリフトによるクラス境界の変化．

5.3 実験結果

(1) 数値属性のモデル化法の影響

CVFDT，$\text{CVFDT}_{\text{NBC}}$ により導出される決定木の分類精度，学習時間，メモリ消費量の数値属性のモデル化法による変化を調べるために実験を行った．生成したデータストリームでは，学習事例 50,000 事例ごとにコンセプトドリフトとしておよそ 7% の事例のクラスを変化させてある．

表 2 に CVFDT，$\text{CVFDT}_{\text{NBC}}$ の分類精度，10,000 個の事例の学習に要した時間，メモリ消費量を示す．表中の太字の数字は，他手法の結果と比べて有意水準 5%で有意差が見られた部分である．なお，$\text{CVFDT}_{\text{NBC}}$ で網羅法を用いる際は，ナイーブベイズ実行のために分類前に数値属性の離散化などが必要となるため，今回は用いていない．

表 2 を見ると，CVFDT では網羅法の分類誤差が 0.15%ほど高いが，どの方法を用いても分類精度に大きな差はないことが分かる．一方，学習時間を見ると，網羅法が他の 2 つの方法と比べて約 9 倍大きい．また，メモリ消費量についても他の 2 つの方法と比べて約 8 倍も大きい．したがって CVFDT には，学習時間とメモリ消費量の観点から，離散化法かガウス分布法が好ましい．

一方，$\text{CVFDT}_{\text{NBC}}$ ではガウス分布法が離散化法よりも 6%ほど分類精度が高い．また，学習時間，メモリ消費量の両方の観点からもガウス分布法のほうが優れている．ガウス分布法も離散化法も，CVFDT では分類精度はほとんど変わらないため，$\text{CVFDT}_{\text{NBC}}$ での分類精度の違いは，葉ノードに導入されたナイーブベイズの影響であるということが分かる．

今回の実験では，離散化法は，離散化の粒度が非常に粗いため複雑なクラス境界を学習できず，分類精度が低くなった．一方，ガウス分布法は，ガウス分布の組み合わせにより複雑なクラス境界を学習できるため，離散化法を用いた場合よりも高い分類精度を示した．

よって，分類精度，学習時間，メモリ消費の観点から CVFDT ではガウス分布法，または離散化法が適しており，$\text{CVFDT}_{\text{NBC}}$ ではナイーブベイズの精度を高めることができるガウス分布法が適していることが分かった．ただし，$\text{CVFDT}_{\text{NBC}}$ での離散化法も離散化の粒度を細かくすれば分類精度の改善が見込まれるため，離散化法に関しては更なる検討の余地がある．

(2) コンセプトドリフト検出手法の影響

実験 5.3(1) より，数値属性のモデル化法としてガウス分布法が CVFDT と $\text{CVFDT}_{\text{NBC}}$ に適していることが

表 2: 数値属性のモデル化法の性能比較.

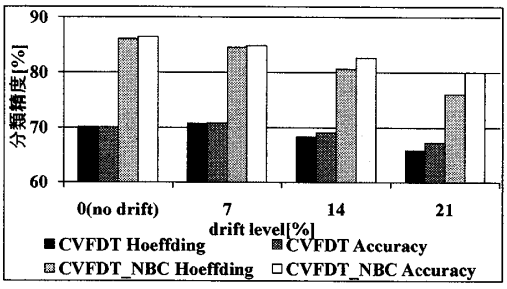
	CVFDT			$CVFDT_{NBC}$		
	分類精度 [%]	学習時間 [s]	メモリ [MByte]	分類精度 [%]	学習時間 [s]	メモリ [MByte]
網羅法	70.86	8.88	339.12	———		
離散化法	70.62	**0.98**	41.68	78.05	1.12	41.71
ガウス分布法	70.72	1.07	**40.99**	**84.56**	**1.06**	**40.99**

分かった．そこで，ガウス分布を仮定した CVFDT と $CVFDT_{NBC}$ に，分類精度の時間変化を用いたコンセプトドリフト検出手法 [12] を導入した場合の分類精度と学習時間への影響を，コンセプトドリフトの程度を変化させた複数のデータストリームを用いて調べた．

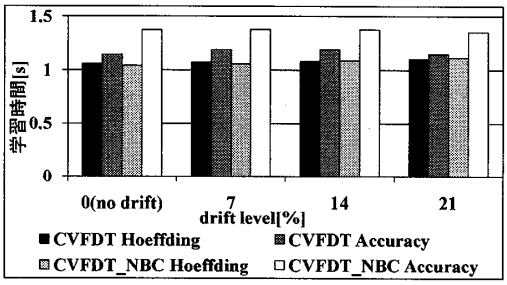
Hoeffding の不等式にもとづくコンセプトドリフト検出手法を用いた場合と，分類精度の時間変化にもとづく手法を用いた場合の各手法の分類精度を図 2(a) に，10,000 個の事例を学習するのに要した時間を図 2(b) に示す．図中の Hoeffding は，Hoeffding の不等式にもとづくコンセプトドリフト検出手法を用いたことを示し，Accuracy は分類精度の時間変化にもとづくコンセプトドリフト検出手法を用いたことを示している．図中の drift level はコンセプトドリフトの程度を表し，drift level が 0%の場合はコンセプトドリフトが全く発生していない．また，Hulten らの論文 [9] を参考にコンセプトドリフトを発生させた場合，drift level が 7%となったため，drift level を 7%単位で変化させている．

図 2(a) を見ると，drift level が 0%から 7%と低いときにはコンセプトドリフトの程度が緩やかであるため，Hoeffding の不等式にもとづく手法でも十分対応でき，分類精度の時間変化にもとづく手法の導入による分類精度の改善はほとんどない．しかし，drift level が大きくなるにつれて Hoeffding の不等式にもとづく手法ではコンセプトドリフトに追従できなくなり，分類精度の時間変化にもとづく手法の導入による分類精度の改善は大きくなった．特に，$CVFDT_{NBC}$ で分類精度の改善が大きく，$CVFDT_{NBC}$ への分類精度の時間変化にもとづく手法導入の効果が大きいことが分かる．一方，図 2(b) を見ると，CVFDT では学習時間の増加は 10%以下と小さいが，$CVFDT_{NBC}$ では最大 36%も学習時間が増加している．

drift level が 7%以下と小さい場合は分類精度の時間変化にもとづく法の導入による分類精度の改善はほとんどないが，drift level が大きくなるにつれて分類精度の時間変化にもとづく手法による分類精度の改善は大きくなり，学習時間の増加を考慮しても分類精度の時間変化にもとづく手法を導入すべきであることが分かった．



(a) 各組み合わせの分類精度.



(b) 各組み合わせの学習時間.

図 2: コンセプトドリフト検出手法による性能比較.

6 結論

本論文では CVFDT や $CVFDT_{NBC}$ で実際に 3 通りの数値属性のモデル化法を分類精度，学習時間，メモリ消費量の観点から比較し，Hoeffding Tree アルゴリズムに最も適した数値属性のモデル化法を明らかにし

た．また，CVFDT に備わっている Hoeffding の不等式にもとづくコンセプトドリフト検出手法には課題があったため，Gama ら [12] が用いた分類精度の時間変化にもとづく手法を CVFDT などの単一の決定木用に修正，導入し，分類精度の時間変化にもとづく手法を導入した場合の分類精度や学習時間の面から評価した．

人工的に作成したデータストリームを用いた実験により，以下の知見を得ることができた．

- 数値属性のモデル化法については，CVFDT ではガウス分布法，または離散化法の 2 つが適している．一方，ナイーブベイズが導入されている $\text{CVFDT}_{\text{NBC}}$ では，ガウス分布法が最適である．
- コンセプトドリフトの検出手法は，ドリフトレベルが高い場合に，分類精度の時間変化にもとづく手法が，Hoefdfing の不等式にもとづく手法よりも優れる．

参考文献

[1] Han, J., Kamber, M.:Data Mining : Concepts and Techiniques (Second Edition), Morgan Kaufmann (2006)

[2] Widmer, G., Kubat, M.:Learning in the Presence of Concept Drift and Hidden Contexts, Machine Learning Vol.23, (1996) 69–101

[3] Langley, P., Iba, W., Thompson, K.:An Analysis of Bayesian Classifiers, In Proceedings of the Tenth National Conference on Artificial Intelligence, (1992) 223–228

[4] Pfahringer, B., Holmes, G., Kirkby, R.:Handling Numeric Attributes in Hoeffding Trees, In Proceedings of the Twelfth Pacific Asia Conference on Knowledge Discovery and Data Mining, (2008) 296–307

[5] Domingos, P., Hulten, G.:Mining High-Speed Data Streams, In Proceeding of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2000) 71–80

[6] Dougherty, J., Kohavi, R., Sahami, M.:Supervised and Unsupervised Discretization of Continuous Features, In Proceedings of the Twelfth International Conference on Machine Learning, (1995) 194–202

[7] Gama, J., Rocha, R., Medas, P.:Accurate Decision Trees for Mining High-speed Data Streams, In Proceedings of the Nineth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2003) 523–528

[8] Gama, J., Medas, P., Rodrigues, P.:Learning Decision Trees from Dynamic Data Streams, In Proceedings of the 2005 ACM Symposium on Applied computing, (2005) 573–577

[9] Hulten, G., Spencer, L., Domingos, P.:Mining Time-changing Data Stream, In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data mining, (2001) 97–106

[10] Hulten, G., Domingos, P.:VFML–A Toolkit for Mining High-speed Time-changing Data Streams, http://www.cs.washington.edu/dm/vfml/ (2003)

[11] Nishimura, S., Terabe, M., Hashimoto, K., Mihara, K.: Learning Higher Accuracy Decision Trees from Concept Drifting Data Streams, In Proceedings of the Twenty First International Conference on Industrial,Engineering & Other Applications of Applied Intelligent Systems, (2008) 179–188

[12] Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection, Lecture Notes in Computer Science, Vol. 3171, (2004) 286–295