3D2-3

CVFDT へのナイーブベイズ導入：コンセプトドリフトを含むデータストリームからの高精度な分類器導出

Application of naive-Bayes Classifiers to CVFDT :
Learning Higher Accuracy Decision Trees from Concept Drifting Data Streams

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

*1東北大学大学院 情報科学研究科
Graduate School of Information Sciences, Tohoku University

In this paper, we propose to combine the naive-Bayes approach with CVFDT, which is known as one of the major algorithms to induce a high-accuracy decision tree from time-changing data streams. The proposed improvement, called $\text{CVFDT}_{\text{NBC}}$, induces a decision tree as CVFDT does, but contains naive-Bayes classifiers in the leaf nodes of the induced decision tree. The experiment using the artificially generated time-changing data streams shows that $\text{CVFDT}_{\text{NBC}}$ can induce a decision tree with more accuracy than CVFDT does.

1. はじめに

近年，インターネットの普及やセンサー技術の発達にともない，センサーネットワークなど大量の電子化データが継続的に得られる環境が普及している．継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 [2] とよばれ，データストリームからの知識発掘が注目されている．

データストリームは，事例が継続的に得られるため全てを蓄積していると事例数が膨大になるという特徴を持つ．また，データストリームは，時間を経るにしたがいデータの特徴が変化する場合が多い．分類学習の観点からは，これは時間変化にともない学習すべき概念が変化することに相当する．このように，時間の経過にともない学習すべき概念 (コンセプト) が次第に変化することをコンセプトドリフトという．

データストリームは，(1) データが継続的に生成される，(2) データ量が膨大である，(3) コンセプトドリフトが発生する，などの通常の機械学習で扱う学習データとは異なる特徴があるため，データストリームの特徴に適した学習手法を準備する必要がある．

データストリームからの決定木学習手法の代表的なものに CVFDT[3] がある．しかし，CVFDT のアルゴリズムは，学習データが持つ分類器の学習に有用な情報の全てを利用していない．本論文では，上記の問題を解決すべく，CVFDT の葉ノードにナイーブベイズを導入した $\text{CVFDT}_{\text{NBC}}$(CVFDT with naive-Bayes Classifiers) を提案し，実験を通じて分類性能と処理速度について評価した結果を報告する．

2. CVFDT

2.1 CVFDT の特徴

コンセプトドリフトを含むデータストリームからの逐次更新型の決定木学習手法の代表的なものに CVFDT[3] がある．データストリームは，継続的にデータが得られるため，累積の事例数が膨大になるという特徴をもつ．よって，全ての事例をそのまま記憶すると，必要となる記憶容量も膨大となる．この問題に対応するため，CVFDT は事例そのものを決定木中に保存するのではなく，クラスごとの属性と属性値組み合わせの度数のみを保持する．そして，CVFDT は決定木中の各ノードに保持されている度数を用いて決定木の分岐属性を決定するために情報利得の計算を行う．

また，CVFDT ではノードの分岐属性の信頼度を考慮するために Hoeffding の不等式を用いている．Hoeffding の不等式を用いることにより，当該ノードを通過した事例を用いた分岐属性の判定の信頼性を評価でき，少数の事例数が到着した段階でノードの分岐に有効かつ信頼性が高い分岐属性の選択を行うことができ，分類精度の高い決定木の学習を可能にしている．

データストリームでは時間が経過するにしたがい，コンセプトドリフトが発生する可能性がある．コンセプトドリフトが発生した場合は分類精度を維持するため，決定木を最新の概念（コンセプト）に追従させなければならない．しかし，決定木中に保持している情報の一部はコンセプトドリフトが発生する以前の古い事例に基づくものであるため，そのまま古い情報を保持して学習に利用していると最新のコンセプトを正しく学習できない．

そこで，CVFDT は移動ウィンドウ方式を採用し，常に最新の w 個の事例の情報に基づき決定木学習を行う．これにより，最新のコンセプトに追従した情報に基づいて学習することを可能にしている．なお，w はウィンドウ幅であり，ユーザーが設定する．

また，CVFDT は定期的に決定木中の全ノードで Hoeffding の不等式を用いて最も信頼性の高い分岐属性を再判定する．そして，現在の分岐属性と新たに Hoeffding の不等式を満たす最も信頼性の高い分岐属性が一致しない場合，そのノードより下位の部分木は最新のコンセプトに適合していないと考えられるため，CVFDT は決定木中の最新のコンセプトに対応していない部分の代替木を作成する．そして，最新のコンセプトに適合した代替木が十分に成長すると，古い部分木と交換することにより，決定木を最新のコンセプトに追従させる．

2.2 CVFDT の課題

CVFDT は，コンセプトドリフトを含むデータストリームから高速に分類精度の高い決定木を学習することが可能である．しかし，CVFDT ではテスト事例の分類の際，C4.5 と同様に学習時に葉ノードへ割り当てられた学習事例のクラス頻度情報に基づき，多数クラスをその葉ノードにおける分類クラスとして判定している．一方，先にも述べたように CVFDT は葉ノードへ割り当てられた学習事例のクラス頻度以外にも，

連絡先: 西村聖，東北大学大学院情報科学研究科，〒 980-8579 宮城県仙台市青葉区荒巻字青葉 6-6-11-304，TEL: 022-795-5856，E-mail: nishimura@aiet.ecei.tohoku.ac.jp

属性と属性値の組み合わせの頻度情報も保持している．この情報を利用することにより，さらに分類精度を改善することができると考えられる．

3. $\mathrm{CVFDT_{NBC}}$

3.1 CVFDT の改善

CVFDT では C4.5 と同様に葉ノードの多数クラスを基にテスト事例の分類を行っており，葉ノードに割り当てられた学習事例の属性に関する情報はテスト事例の分類の際は利用されていない．しかし，C4.5 により学習される決定木の葉ノードにナイーブベイズ [6] を組み合わせた NBTree[5] のように，CVFDT では利用されていない学習事例の属性に関する情報も用いることによりさらなる分類精度の向上が見込まれる．

そこで，CVFDT により学習される決定木の葉ノードへナイーブベイズを導入した手法である $\mathrm{CVFDT_{NBC}}$(CVFDT with naive-Bayes Classifiers) を提案する．$\mathrm{CVFDT_{NBC}}$ では，CVFDT により学習される決定木の葉ノードへナイーブベイズを導入し，CVFDT では分類の際には利用されなかった学習時に葉ノードへ蓄積された事例の属性に関する情報をテスト事例の分類に利用する．これにより，さらなる分類精度の向上が期待される．

CVFDT により学習される決定木を用いた分類では，テスト事例に対してルートから順に各ノードに適用された属性によるテストを繰り返し，最終的にテスト事例が割り当てられた葉ノードのクラスを持ってテスト事例のクラスとしている．

一方，$\mathrm{CVFDT_{NBC}}$ により学習される決定木を用いたテスト事例の分類は，テスト事例がルートノードから葉ノードに割り当てられるまでは CVFDT と同様である．そして最終的にはテスト事例がたどり着いた葉ノードで，学習時にその葉ノードへ割り当てられた事例から生成されたナイーブベイズを用いて分類を行う．ナイーブベイズは式 (1) に基づき確率演算を行い，式 (1) 中の左辺を最大とするクラスをテスト事例のクラスとする．

$$P(C|\vec{e}) \propto P(C)\prod_{i=1}^{d} P(e_i|C). \tag{1}$$

ここで C はクラス，d は属性数，e_i は事例 $\vec{e}$ の i 番目の属性値を示している．

決定木の葉ノードにナイーブベイズを設置するということは，決定木により分割された属性空間をさらに細かく分割するということであり，属性空間をより細かく分割することにより，高い分類精度を発揮することができる．図 1 に決定木の葉ノードにナイーブベイズを設置することにより，属性空間がより細かく分割される例を示す．

図 1(a) のような分布の事例について学習した結果，図 1(b) のような決定木が学習されたとする．CVFDT は決定木のノード分岐に関する判定の信頼性評価に Hoeffding の不等式を用いているため，決定木の分岐にある程度の事例数を要し，図 1(b) の決定木はこれ以上分岐ができないとする．

通常の決定木による分類では，学習時に葉ノードへ割り当てられた事例のクラス頻度しか用いないため，図 1(b) のように一部の事例に関して正しく学習できない．しかし，図 1(c) のように決定木の葉ノードへナイーブベイズを導入することにより，学習時に葉ノードへ割り当てられた学習事例のクラス頻度のほかに属性に関する情報も利用することができ，通常の決定木では正しく学習することができなかった事例についても学習することができるようになる．

3.2 $\mathrm{CVFDT_{NBC}}$ の特徴

$\mathrm{CVFDT_{NBC}}$ はテスト事例のクラスを決定木の葉ノードに備え付けられたナイーブベイズにより予測するため，式 (1) 中の $P(C)$ と $P(e_i|C)$ の両方の情報を用いるが，これらの情報は CVFDT でも葉ノードに保持しており，決定木中に保持する情報に変化はない．

ナイーブベイズはクラスの判定に無関係な属性や外れ値やノイズに強く，クラスの判定に有効な属性がない場合にも優れた分類精度を示すという特徴を持つ．また Domingos らは，ナイーブベイズは属性間に相関がないことを前提としているが，属性間に相関がある場合でも高い分類精度を示し，学習のために準備された事例が少ない場合でも高い分類精度が得られることを報告している [1]．よって，葉ノードに割り当てられる少ない事例からでもナイーブベイズは十分な分類精度を示すことが期待される．

$\mathrm{CVFDT_{NBC}}$ では，学習の過程で決定木の葉ノードへナイーブベイズを導入するために必要な情報は各ノードで自然と維持される．よって，葉ノードにナイーブベイズを導入することにより，新たに余計な処理は発生しない．

一方，テスト事例を分類する際には葉ノードにおいてナイーブベイズを実行するため，CVFDT に比べて若干多くの処理時間を要することが予想される．CVFDT ではテスト事例の分類にかかる時間は，テスト事例を決定木の葉ノードへ割り当てる時間のみである．しかし，$\mathrm{CVFDT_{NBC}}$ では葉ノードにナイーブベイズを導入することにより，テスト事例の分類に要する時間は，テスト事例を決定木の葉ノードへ割り当てる時間と，テスト事例が割り当てられた葉ノードでナイーブベイズを実行する時間の和となる．よって，$\mathrm{CVFDT_{NBC}}$ では葉ノードにナイーブベイズを導入することにより，事例の分類時に葉ノードでナイーブベイズを実行する時間の分だけ処理時間が長くなる．しかし，ナイーブベイズによる分類は高速に実行可能であることから，テスト事例の分類に要する処理時間が長くなったとしても，多くの適用対象では問題ない程度である．

なお，$\mathrm{CVFDT_{NBC}}$ は CVFDT と同様に名義属性のデータの分類学習を対象範囲としている．

4. 実験

4.1 実験環境

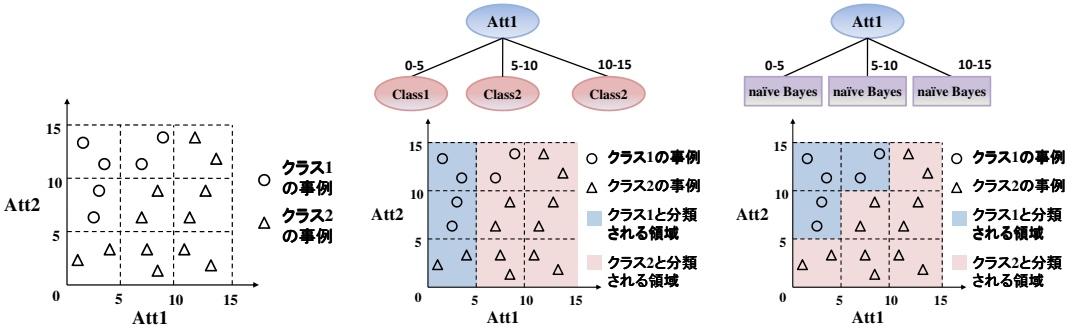
パラメータによりデータストリームの特徴を調整できるように，人工的にコンセプトドリフトを含むデータストリームを生成し，$\mathrm{CVFDT_{NBC}}$ と従来手法の性能比較を行った．

性能を比較する従来手法として，バッチ型決定木学習手法である C4.5 と逐次更新型決定木手法である CVFDT を用いた．実験には，C4.5 について C4.5 リリース 8[7]，CVFDT については VFML[4] で公開されている CVFDT のプログラムを利用した．実験に用いた計算機は，OS が Fedora Core7，CPU が Intel(R) Core(TM)2 CPU 4300 1.8GHz，メモリが 2GB というものである．

4.2 実験用データ

提案手法の性能を評価するため，Hulten らが実験に使用している人工データの作成方法 [3] を参考に，人工的にコンセプトドリフトを含むデータストリームを生成した．各事例 $\vec{e}$ のクラスは式 (2) で表わされる属性空間上の超平面にしたがい決定される．

$$\sum_{i=1}^{d} w_i e_i = w_0. \tag{2}$$



(a) 学習事例分布　(b) 決定木による分類　(c) ナイーブベイズ導入後

図 1: ナイーブベイズの導入による属性空間の細分化

ここで w_i は重みベクトル $\vec{w}$ の i 番目の値であり，i 番目の属性のクラスの判定への寄与度を示している．コンセプトドリフトは定期的に重みベクトルを変化させ，超平面を変化させることにより実現される．また，事例の各属性値はクラスの判定をした後，CVFDT でも扱えるように離散化している．詳しい説明は [3] を参照されたい．

4.3 実験方法

上記の方法により，人工的に 300 万個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 万個毎に 1 万個のテスト事例を与え分類誤差などを調べた．また，コンセプトドリフトは学習事例 5 万個毎に重みベクトルを変化させることにより発生させた．学習事例 1 万個毎にテストを行うため，合計で 300 回のテストが行われ，以下に示す結果はこの 300 回のテストでの平均値である．

なお，C4.5 はバッチ型の学習手法であるため，1 万事例毎に新規に決定木を学習させた．C4.5 のパラメータはデフォルト値を用い，CVFDT のパラメータは [3] と同じである．

4.4 実験結果

4.4.1 属性数と性能

図 2 に各学習手法の分類誤差の属性数変化を示す．図 2 中の drift level はコンセプトドリフトの前後でクラスが変化した事例の割合を百分率で表現しており，コンセプトドリフトの程度を示す．今回の実験でコンセプトドリフトの程度をほぼ一定に保ち，属性数のみを変化させたため，drift level はほぼ一定の値となった．

図 2 を見ると何れの属性数のときも $\mathrm{CVFDT_{NBC}}$ の分類誤差が小さいが，属性数が増えるにつれて $\mathrm{CVFDT_{NBC}}$ による分類誤差の改善が小さくなっていることがわかる．

今回，各属性値は連続値を CVFDT でも扱えるように離散化をしており，各属性値は離散化によりある程度情報が失われてしまっている．CVFDT や C4.5 は $\mathrm{CVFDT_{NBC}}$ と異なり全ての属性をテスト事例の分類時に用いるわけではないため，属性数が増えても離散化による影響を抑えることができ，属性数が増えて問題が簡単になった分だけ分類誤差が小さくなった．一方，$\mathrm{CVFDT_{NBC}}$ はナイーブベイズを用い式 (1) にしたがい分類を行うため，属性数が増えるほど離散化により失われる情報の総量が増え，CVFDT や C4.5 とは異なり，分類誤差を下げることができなかった．

表 1 に $\mathrm{CVFDT_{NBC}}$，CVFDT が 1 万個の事例を学習するのに要した時間と，1 万個のテスト事例を分類するのに要した時間を示す．

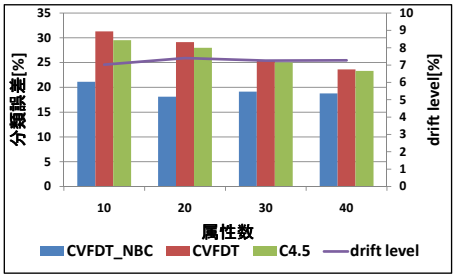


図 2: 属性数と分類誤差の関係

CVFDT により学習される決定木の葉ノードにナイーブベイズを導入するのに必要な情報は，学習の過程で自然と各葉ノードへ蓄えられるため，CVFDT と $\mathrm{CVFDT_{NBC}}$ の学習時間はほぼ等しくなる．よって，学習時間に対する要求などの面で CVFDT が適用できる問題領域であれば，$\mathrm{CVFDT_{NBC}}$ も適用可能であることがわかる．

次に，分類時間を見ると $\mathrm{CVFDT_{NBC}}$ のほうが CVFDT と比べてテスト事例の分類に要する時間がかなり長いことが分かる．しかし，ナイーブベイズ自体が高速に実行可能であるため $\mathrm{CVFDT_{NBC}}$ の分類時間の増加は微小であり，多くの適用対象で問題とならない程度である．

表 1: 学習時間と分類時間

属性数	学習時間 [s]		分類時間 [s]	
	$\mathrm{CVFDT_{NBC}}$	CVFDT	$\mathrm{CVFDT_{NBC}}$	CVFDT
10	1.40	1.74	0.0274	0.00199
20	1.91	1.79	0.0579	0.00217
30	2.18	2.05	0.08743	0.00239
40	2.84	2.91	0.153	0.00288

4.4.2 Drift level と分類誤差

表 2 に属性数を 10 と固定し，コンセプトドリフトの程度を変化させた場合の分類誤差の変化を示す．

表 2 を見ると，何れの drift level のときも $\mathrm{CVFDT_{NBC}}$ は CVFDT よりも分類誤差が小さいことがわかる．特に，drift

level が小さいときは CVFDT と比べて分類誤差が 10%近く小さくなっている．しかし，drift level が大きくなるにつれて $\text{CVFDT}_{\text{NBC}}$ の CVFDT と比べての分類精度の改善は小さくなった．

CVFDT は決定木の葉ノードに割り当てられた学習事例のクラスの出現確率 $P(C)$ が最大のクラス C をテスト事例のクラスとする．一方，$\text{CVFDT}_{\text{NBC}}$ は決定木の葉ノードにナイーブベイズを導入しているため，式 (1) に示したように，学習事例のクラスの出現確率 $P(C)$ に加えて，事後確率 $P(e_i|C)$ に基づいてテスト事例のクラスを決定している．drift level が小さいときはコンセプトドリフトの前後で推測したい本当の $P(e_i|C)$ がそれほど変化しないため，ウィンドウ中に古いコンセプトの事例を含んでいてもそれほど問題とならず，属性とクラスとの関係の情報 $P(e_i|C)$ を利用することにより，コンセプトドリフトの影響を抑制しながらクラスを正しく予測することができる．一方，drift level が増加するとコンセプトドリフトが発生するたびに推測したい本当の $P(e_i|C)$ が大きく変化し，学習事例から正しい $P(e_i|C)$ を推測することが困難となり，属性とクラスの情報を用いてもクラスを正確に予測することが難しくなる．このように，$\text{CVFDT}_{\text{NBC}}$ はコンセプトドリフトの程度が小さい場合に特に CVFDT の分類精度を改善することができる．

また，$\text{CVFDT}_{\text{NBC}}$ や CVFDT は drift level が大きくなるにつれて分類誤差が大きくなったが，C4.5 は 1 万事例毎に新たに決定木を学習するため，コンセプトドリフトの影響をほとんど受けず，drift level が大きくなっても分類誤差がほとんど変化しなかった．

今回の実験では決定木を新たにに学習する間隔が短かったため，コンセプトドリフトの影響を抑えることができた．しかし，決定木を新たに学習する間隔が長いと分類精度の劣化を招く可能性もある．よって，drift level が大きいときはバッチ型の学習手法で短い間隔で改めて分類器を学習し直すか，CVFDT のウィンドウ幅を短くし，最新の事例のみに基づいて決定木を学習しなければならない．

表 2: 分類誤差の drift level 変化

drift level[%]	分類誤差 [%]		
	$\text{CVFDT}_{\text{NBC}}$	CVFDT	C4.5
7.02	20.85	31.29	29.52
14.22	23.43	33.80	30.11
20.99	27.68	35.66	30.28
28.12	33.06	39.31	31.36
35.04	36.88	42.01	32.29

4.4.3 学習初期の分類誤差

図 3 に CVFDT と $\text{CVFDT}_{\text{NBC}}$ の分類誤差の時間変化を示す．図 3 を見ると，CVFDT は時間が経過し決定木がある程度成長するまでは属性空間を細かく分割できないため，分類誤差が大きくなってしまっている．

しかし，$\text{CVFDT}_{\text{NBC}}$ は決定木の葉ノードにナイーブベイズを導入しているため，通常の決定木よりも属性空間を細かく分割することができ，決定木があまり成長していない学習初期においても分類誤差が小さくなる．このように，$\text{CVFDT}_{\text{NBC}}$ は学習の初期にも高い分類精度が得られるという特徴を持つ．

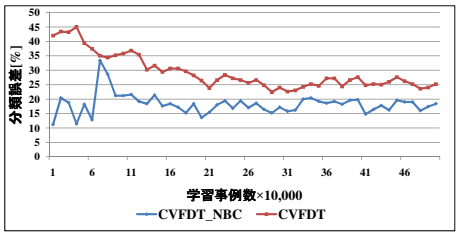


図 3: 分類誤差の時間変化

5. まとめ

本論文では，代表的なコンセプトドリフトを含むデータストリームからの決定木学習手法である CVFDT により学習される決定木の葉ノードへナイーブベイズを導入した $\text{CVFDT}_{\text{NBC}}$ を提案した．

人工的に生成したコンセプトドリフトを含むデータストリームを用いた実験により，$\text{CVFDT}_{\text{NBC}}$ が CVFDT と同等の学習時間で，CVFDT と比べて優れた分類精度の決定木を学習できることを確かめた．特に，コンセプトドリフトの程度が小さいときや，学習初期においては CVFDT よりも分類精度が大きく改善される．

$\text{CVFDT}_{\text{NBC}}$ は決定木の葉ノードへナイーブベイズを導入しているため，ナイーブベイズの実行時間だけ分類に要する時間が増加してしまう．しかし，$\text{CVFDT}_{\text{NBC}}$ の分類時間の増加は多くの適用対象では許容範囲内の増加であり，実用上は問題とならない程度の分類時間の増加である．

参考文献

[1] Domingos, P. , Pazzani, M.: On the Optimality of the Simple Bayesian Classifiers under Zero-One Loss, Machine Learning, Vol. 29, (1997) 103–130

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

[3] 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

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

[5] Kohavi, R.: Scaling Up the Accuracy of Naive- Bayes Classifiers: a Decision-Tree Hybrid, In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, (1996) 202–207

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

[7] Quinlan, J. R.: C4.5: Programs for Machine Learning, Morgan Kaufmann (1993)