不均一な分布を持つデータストリームでの決定木アルゴリズムに向けたノード構築基準値の実装と評価

Implementation and Evaluation of Node Construction Criterion for Decision Tree Algorithm in Inhomogeneous Distribution Data Stream

[1] 峰岸　達也,　　[2] 新美　礼彦
[1]Tatsuya Minegishi,　[2]Ayahiko Niimi
[1] 公立はこだて未来大学大学院システム情報科学研究科
[1]Graduate School of Systems Information Science, Future University Hakodate
[2] 公立はこだて未来大学システム情報科学部
[2]Faculty of Systems Information Science, Future University Hakodate

Abstract: In recently social world, there is a flood of various data, and it is growing a desire to discover informative information in those data and to utilize them. Moreover, those data are changing more huge and complex. In particular, it is called such data that generate intermittently and di erent interval to data stream represented by sensor network and stream mining is technology to discover informative information from data stream have been a focus of attention. In this paper, we focus on classi cation learning which is analytic method of stream mining. We are concerned with decision tree learning called VFDT to regard real data as data stream. However, there are some data whose rate of classes of data in real data is extremely di erent, and credit card transaction data is one of the those data. Therefore we propose and implement new statistical criterion used in nodes construction algorithm implemented VFDT and evaluate whether it can support in inhomogeneous distribution data stream.

1　はじめに

近年のネットワーク社会では情報処理技術の発達により大規模なデータを収集・蓄積することが容易になった．また，そのような大規模なデータの中から有益なパターンや新しい知識を発見し，有効活用できないかという要望が多く挙げられる．このような流れを受けてデータ収集から知識の発見までを行う技術としてデータマイニングが注目されてきている．しかし，インターネットの普及やセンサ技術の発展にともない，データは以前に比べてより大規模かつリアルタイム性が増してきているといった複雑なものへと姿を変えてきている．このような断続的に，異なる間隔で到着する大規模なデータが次々に消費されていく様子をデータの流れ (ストリーム) としてとらえたものをデータストリームと呼び，その中から知識発見を行う技術をストリームマイニングと呼ぶ.

ストリームマイニングには様々な分析手法があるが，なかでも分類学習が注目されている．これまでに多くの分類学習手法が提案されているが，決定木学習手法は学習が高速であることと，導出される分類機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よく利用される．データストリームに対応した決定木学習手法には VFDT[1] と呼ばれるものがあり，データの到着に従い決定木を順に成長させていき，データを分類するというものである.

しかし中には，本稿でデータストリームとして取り上げたクレジットカード取引データのように分類する際のデータのクラスの割合が極端に違うようなデータが存在し，このようなデータでは VFDT の分類精度が低下するといった問題がある.

そこで本稿では，不均一なデータ分布を持つデータストリームに適用できるような VFDT のノード構築アルゴリズムを提案し，その中でノード構築基準値の実装と評価を行う．ノード構築基準値の評価としては，重みの付け方の有効性を議論するために，重みの値を複数変更し実験を行い，傾向を観察する.

本稿の構成は次のとおりである．まず，第 2 章では VFDT の説明を行う．その後，第 3 章では提案手法として不均一なデータ分布を持つデータストリームからの VFDT 構築について述べ，第 4 章では実験から提案手法の有効性を検証する．そして，第 5 章で実験結果を示し，第 6 章で結果の考察をする．最後の第 7 章で

はまとめと今後の展望を述べる．

2 既存研究

2.1 VFDT

C4.5[2] のように初めにすべての事例を入力として受け取り，決定木を構築するものをオフライン型決定木と呼ぶ．しかし，これはすべての事例がそろわないと決定木を構築することができないということと，事例にランダムアクセスをしなければならないということからデータストリームに適用することができない．これに対して，データストリームの短い間隔で次々と新しい事例が到着し，かつ累積する事例数が大量になるという特徴に対応した決定木をオンライン型決定木といい，代表的なものに VFDT(Very Fast Decision Tree learner) が挙げられる．VFDT はすべての事例の到着を待たずに決定木を徐々に成長させていくことができるので，メインメモリに事例を蓄積しない．VFDT のアルゴリズムはメモリ消費量と処理時間を減らすために，事例そのものを決定木中に蓄積するのではなく，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蓄積する．VFDT では事例を受け取るごとにルートノードのみの決定木から枝を成長させ葉ノードを作成していくことで決定木を順に成長させていく．新規にノードを作成する際には，それまでのノードに頻度情報が蓄積し，統計基準値を満たすかどうかの判定を行い，決定木を成長させる．VFDT では Hoe ding bound と呼ばれる統計基準値が用いられる．葉ノードに蓄えられた事例は利用できる全事例の一部分でしかないので，誤差を含む可能性がある．しかし，定常分布に基づいて確率的に生成される無限に長いデータストリームを考えると，それぞれの葉ノードに到達する事例の集合はオフライン型決定木の場合の理想的なデータ集合と見なすことができる．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，Hoe ding bound は $1-\delta$ の確率で変数 r の真の平均が $\bar{r}-$ より大きくなることを保証する．ここで， は以下のように定義される．

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

ある葉ノードにおける最良の基準値と，次の基準値との差が より大きくなればその葉ノードからさらに分岐を作成する．Hoe ding boud を用いると，$\overline{\Delta G()} = \overline{G(X_a)} - \overline{G(X_b)} >$ のとき，属性 X_a でノードを分割することが $1-\delta$ の確率で正しいことがわかる．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．

2.2 データストリームからの VFDT 構築

既存研究にて、データストリームに対応した決定木学習手法である VFDT の構築を行った [3]．ここでは，クレジットカード取引データをデータストリームとみなし VFDT を構築した．

1 で挙げたようにクレジットカード取引データは分類する際のデータのクラスの割合が極端であるが，ここでは VFDT の構築アルゴリズムに変更を加えずに VFDT を構築した．ここでは，クレジットカード取引データを用いて VFDT を構築する際に，データのサンプリングによって VFDT が変化するかを検証するために不正利用率は変えずに無作為にサンプリングした 10 個のデータセットを用いて 10 本の VFDT を構築した．ここで不正利用率は 10%としたが，既存研究 [4] においてオフライン決定木の構築として C4.5 アルゴリズムで 3 通りの不正利用率で決定木を構築した際に最も良く決定木が成長したものが 10%であったためである．C4.5 アルゴリズムにおいてオフライン決定木を構築した際に設定した不正利用率は以下の通りである．

(a) 実際の不正利用率である 0.02%

(b) 提供されたデータのサンプリング比率である 0.5%

(c) この実験で設定した 10%

(a)，(b) ではルートノードにより終端ノードが 2 分されただけの決定木となってしまい，決定木の分類精度は 99%以上となったが，ほぼすべての不正利用を正常利用と分類してしまう結果となった．これは 0.02%，0.5%のどちらも不正利用率が低すぎるため，決定木自体の精度は高くなったものの，実際にはほぼすべての不正利用を分類できていないという結果になってしまった．以上のことから不正利用率 10%のデータセットを VFDT の構築に用いた．

それぞれの VFDT において 10-folds cross validation を行い VFDT の精度とサイズを算出し，10 本分の結果の平均を取った．したがって実際には VFDT は 100 本構築したことになる．VFDT の精度は 92.157%，VFDT のサイズは 91 となった．精度の分散値が 0.290 となったことから，VFDT の結果はデータのサンプリングには依存しないことがわかっている．

3 不均一な分布を持つデータストリームからの VFDT 構築

2.1 であげた VFDT のノード構築基準値である Hoe ding bound ではデータストリームのデータ分布をガウス分布と仮定している [5]．しかし，4.1 であげるクレジットカード取引データなどではデータストリームに含まれる事例のクラスがガウス分布に従わない場合がある．そのような場合であると VFDT の精度自体は高くなるが，実際には片方のクラスの分類精度がほとんど無視されてしまうような VFDT が構築されてしまう．そこで我々は，VFDT のノード構築基準値である Hoe ding bound の計算を変更することで，不均一なデータ分布を持つデータストリームに対応できるような VFDT の構築を提案する．変更点としては，VFDT において葉ノードから新たに枝を成長させる際の判定で行われる情報利得の計算 $\overline{\Delta G()} = \overline{G(X_a)} - \overline{G(X_b)} >$ に用いられる情報量 $G(X_a)$ と $G(X_b)$ にクラスごとに重み付けを行う．本稿ではクラスは 2 つとしている．ID3 や C4.5 で用いられている情報量の計算では，事例の集合 S に対して $freq(C_i, S)$ を S の中でクラス C_i に属する事例の数，集合 S に含まれる事例数を $|S|$ とすると，S からランダムに 1 つの事例を選び出し，それがクラス C_i に属しているとすると平均情報量 $\overline{G(X)}$ は

$$\overline{G(X)} = -\sum_{i=1}^{2} \frac{freq(C_i, S)}{|S|} \times \log_2 \frac{freq(C_i, S)}{|S|} \tag{2}$$

となる．ここで $\overline{G(X)}$ を求める際にそれぞれのクラスでの情報量に重み付けをして和を取る．重みは $0 \leq w \leq 1$ の範囲とし，不正利用のクラスとする．従って不正利用のクラスを C_1，正常利用のクラスを C_2 とすると，

$$\begin{aligned}\overline{G(X)} = &-w \times \frac{freq(C_1, S)}{|S|} \times \log_2 \frac{freq(C_1, S)}{|S|} \\ &-(1-w) \times \frac{freq(C_2, S)}{|S|} \times \log_2 \frac{freq(C_2, S)}{|S|}\end{aligned} \tag{3}$$

となる．また，VFDT では離散データストリームを対象としたアルゴリズムになっており，数値データストリームへの対応はしていない．しかし，本稿で VFDT を構築する際に用いたツールである VFML(Very Fast Machine Learning)[6] で公開されている VFDT のプログラムで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 Entropy-Based Discretization[5] という離散化法が導入されているが，このとき各数値属性の情報利得を最大とする属性値で 2 つの区間に離散化しているが，この情報利得の算出でも，上で挙げた同様の重み付けを行っている．

しかし，重み付けを行ったあとの情報利得と Hoe ding bound の比較である $\overline{\Delta G()} = \overline{G(X_a)} - \overline{G(X_b)} >$ において，左辺の情報利得は重み付けされたものであり，右辺の Hoe ding bound の値　とそのまま比較してしまうと重み付けされたものとされていないものを比較してしまっている．そこで，右辺の情報利得に 2 つのクラスに対しての重み w と $(1-w)$ の平均値である 0.5 をかけて，両辺の釣り合いを取ることとする．

4 実験

4.1 実験データとツール

本稿では実データとしてクレジットカード取引データをデータストリームに見立てて実験を行う. 実際のクレジットカード取引では，データは複雑に変化し，オンラインで連続して到着する．そのデータは以下のようなものである．

(i) 1 日に約 100 万件のデータが発生
(ii) 1 取引につき 1 秒未満で到着
(iii) ピーク時には 1 秒間に約 100 件
(iv) 24 時間 365 日絶え間なく到着

したがって，クレジットカード取引データはまさにデータストリームと呼ぶことができる．しかし，1 日に約 100 万件の取引データが発生するものにデータマイニングを用いたとしても，モニタリングで 1 日に対応出来るのは約 2,000 件であるのが一般的である．したがって，検出件数は多くても全体の 0.2%程度という厳しい条件のなかで，疑わしい取引データを効率的に検出しなければならない．さらに実際の不正利用率は全取引データに対して 0.02〜0.05%と極めて低い割合であり，膨大な取引データのなかから極めて低い不正利用を検出しなければならないということが課題とされる．

今回実際に扱ったデータは 1 取引ごとのデータが時系列で CSV ファイルに記述されていて，データは属性として存在する．クレジットカード取引データには 124 の属性があり，84 属性が取引データと呼ばれる生データとなっていて，このうちの 1 属性がそれぞれの取引データが不正利用かそうでないかを判別するためのものとなっている．残りの 40 属性はカード利用者の利用挙動から算出されたデータでビヘイビア属性と呼ばれる．データサイズは 1ヶ月分で 700MB ほどであり非常に大きなものである．また上で実際の不正利用率

は 0.02〜0.05%と述べたが，このデータでは 0.5%ほどにサンプリングし直している．実験に用いたデータは以下の通りである．

- データの属性数
  - 57 の取引属性と 42 のビヘイビア属性
- 不正利用のサンプリング比率
  - 正常利用：不正利用 = 9：1

通常，提供データには全部で 120 程の属性が存在するが，提供元の助言から VFDT の構築に不向きなものは除外した．また，VFDT の構築に用いたデータ数は約 5 万件である．

実験には VFDT の構築にはデータストリーム用の機械学習アルゴリズムの実装コードである VFML を使用し，VFDT を構築した．

4.2 実験方法

4.1 で挙げたデータを用いて以下の 2 つの VFDT を構築した．

(i) VFML に実装された VFDT アルゴリズムをそのまま用いた VFDT.

(ii) 葉ノードから新たに枝を成長させる際の判定で行われる情報利得の計算に用いられる情報量と，数値属性を離散化する際の情報量の計算に重み付けを行った VFDT.

重み付けは 3.2 で述べたように不正利用のクラスの重みを w とし，$0 \leq w \leq 1$ の範囲とする．今回の実験ではクレジットカード取引データの正常利用と不正利用の割合を 9 : 1 としているので，不正利用に対する重みを $w = 0.9$ と大きくした．また，重み付けの有効性を観察するために $w = 0.9$ の他に $w = 0.1, 0.5, 0.99, 0.999$ として実験を行った. ここで $w = 0.5$ は 3 で示した情報利得と Hoe ding bound の比較時の操作により，実際には既存手法である重み付けを行っていないものと一致する．どちらの実験でも VFDT の枝刈りはデフォルト値である $pruning\ confidence = 25\%$とした．

5 実験結果

クレジットカード取引データを用いて 4.2 の (i) と (ii) の VFDT の重みを変えて構築した VFDT の精度とサイズと不正ルール数を表 1 に，$w = 0.5$ と一致する既存手法の VFDT の Confusion Matrix の結果を表 2 に，提案手法の VFDT の重みを $w = 0.1, 0.9, 0.99, 0.999$ とした時の Confusion Matrix の結果を順に表 3 から表 6 に示す．表 1 の結果は VFML に実装されている cross validation を用いて 10-folds cross validation を行った結果である．VFML では VFDT のエラー率と，全ノード数がサイズとして算出される．表 1 での精度とはエラー率を 100%から引いたものである．10-folds cross validation を行っているので実際にはそれぞれの重みにおいて VFDT を 10 本構築しその平均値を算出している．

表 1: 精度とサイズと不正ルール数

	精度 (%)	サイズ	不正ルール数
$w = 0.5$	90.851	91.000	3
$w = 0.1$	71.188	27.400	3
$w = 0.9$	92.325	106.600	5
$w = 0.99$	90.881	99.400	3
$w = 0.999$	89.879	90.800	3

表 2: Confusion Matrix(既存手法 $w = 0.5$)

		実際のクラス	
		0(正常)	1(不正)
葉のクラス	0(正常)	40,825	2,174
	1(不正)	1,494	2,598

表 3: Confusion Matrix(提案手法 $w = 0.1$)

		実際のクラス	
		0(正常)	1(不正)
葉のクラス	0(正常)	32,027	1,550
	1(不正)	10,292	3,222

表 4: Confusion Matrix(提案手法 $w = 0.9$)

		実際のクラス	
		0(正常)	1(不正)
葉のクラス	0(正常)	40,174	1,982
	1(不正)	2,145	2,790

表 5: Confusion Matrix(提案手法 $w = 0.99$)

		実際のクラス	
		0(正常)	1(不正)
葉のクラス	0(正常)	41,449	3,555
	1(不正)	870	1,217

表 6: Confusion Matrix(提案手法 $w = 0.999$)

		実際のクラス	
		0(正常)	1(不正)
葉のクラス	0(正常)	42,252	4,669
	1(不正)	67	103

6 考察

表 1 の通り，VFDT 自体の精度では $w = 0.9$ のとき 92.325%で最も高くなり，$w = 0.1$ のとき 71.188%と最も低くなった．しかし，表 2 から表 6 までの Confusion Matrix から不正利用に対する再現率 (Recall) を算出した図 1 では $w = 0.1$ のときが最も多くの不正利用を検出し，VFDT 自体の精度が最も高かった $w = 0.9$ のときが 2 番目であった．しかし，$w = 0.1$ のときは正常利用を正常利用の葉ノードに分類したデータ数が他の重みの値に比べ極端に少なくなった．このため VFDT 自体の精度が低くなったと考えられる．また，$w = 0.999$ と不正利用に対する重みを 1 に近づけた場合では不正利用に対する再現率が非常に小さくなり，VFDT 自体の精度は正常利用の分類に関してがほとんどであると言える．

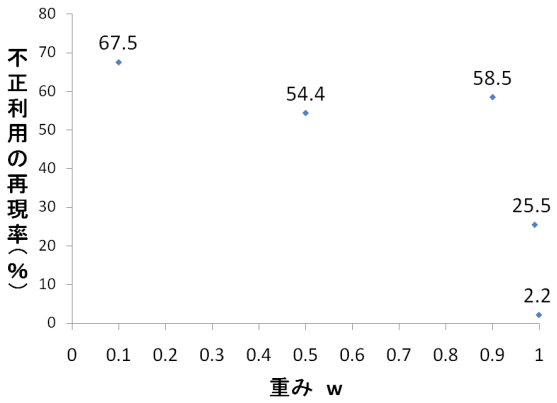

図 1: 不正利用に対する再現率

さらに，それぞれの重みの値での VFDT のサイズと不正ルール数を比較してみても $w = 0.1$ の場合のみサイズが 27.400 と小さくなった以外はほとんど安定していた．不正ルール数でも VFDT のサイズが最も大きくなった $w = 0.9$ の場合で 5 つとなった他は，どれも 3 つとなり，VFDT のサイズには依存しないと言える．

以上から，VFDT のノード構築基準値として，ノード構築時の情報利得と Hoe ding bound の比較の際の情報利得に重みをつけ付けてて学習させることにより，通常では事例数が少ないクラスが分類時にうまく分類されないような場合でも，精度を向上させることができることがわかった．

7 まとめと今後の展望

本稿では不均一な分布を持つデータストリームとしてクレジットカード取引データを取り上げ，そこから VFDT のノード構築アルゴリズムにおける新しい統計基準値の提案・実装を行い，その有効性を検証した．

ノード構築アルゴリズムにおける新しい統計基準値としては既存アルゴリズムのノード分割の際に用いられている Hoe ding bound と情報利得の計算の比較に用いられる情報量にクラスごとに重み付けを行った．今回の実験ではクレジットカード取引データの正常利用と不正利用を割合を９：１としたので，そのデータのクラス分布の逆数を重みとした．

結果としては重み付けを行った VFDT では，VFDT 自体の精度，不正利用の分類率ともに性能が向上する結果となり，今回提案した重みの付け方はデータのクラス分布が極端に異なる場合に有効であることがわかった．

今後の展開としては，重みの値を変えて実験を行うことで，重みの値によって精度がどう変化するのかを検証することで重みの値の有効性を示したい．

また，重み付けという提案手法に一般性を持たせるために VFDT 構築に用いるデータを UCI データセットなどのベンチマーク用データによる実験を行い，今回の実験と比較して考察する必要がある．

8 謝辞

株式会社インテリジェントウェイブの関係者の方々には，実験データの提供，また実験を進めていく上での様々なアドバイスなど細部にわたるご指導をいただきました．ここに感謝いたします．

参考文献

[1] P. Domingos. G. Hulten: Mining High-Speed Data Streams, Proceedings of the ACM Sixth International Conference on Knowledge Discovery and Data Mining, ACM Press, pp.71-80,2000

[2] J. Ross Quinlan: C4.5 : programs for machine learning, Morgan Kaufmann, San Mateo, Calif., 1993

[3] Tatsuya Minegishi, Masayuki Ise, Ayahiko Niimi, Osamu Konishi: Extension of Decision Tree Algorithm for Stream Data Mining Using Real

Data, IEEE, 5th International Workshop on COMPUTATIONAL INTELLIGENCE & APPLICATIONS 2009, 2009

[4] Tatsuya Minegishi, Masayuki Ise, Ayahiko Niimi, Osamu Konishi: Comparison with two attribute selection methods using actual data, stepwise procedure in logistic regression analysis and selection by decision tree, Japan Society for Fuzzy Theory and Intelligent Informatics, The 25th Fuzzy System Symposium, 1A2-02 (6 pages in CD-ROM),2009

[5] 西村　聖，寺邊　正大，橋本　和夫: 数値データストリームからの決定木導出, FIT2009, 第 8 回情報科学技術フォーラム, 2009

[6] P. Domingos. G. Hulten: VFML - a toolkit for mining high-speed time-changing data streams, http://www.cs.washington.edu/dm/vfml/, 2003

連絡先

公立はこだて未来大学大学院　峰岸達也
E-mail: g2109043@fun.ac.jp