社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－ＳＥ－160 (1)
2008－EMB－ 9 (1)
2008／6／19

決定木の比較を利用したコンセプトドリフトの解析

久 保 晴 信[†1]

多くの分野において，時系列に変化するストリーミングデータの分析は重要な研究テーマとなってる．例えば，購買履歴動向の分析において購買動向のトレンドと，その変化を捕らえることは，企業にとって死活問題となっている．ストリーミングデータマイニングの分野では分別器を用いて時系列データの変化の検出が行われている．そのような変化のことをコンセプトドリフトと呼んでいる．本研究では，分別器として決定木を用いる，なぜならば決定木には豊かな説明能力があるからである．つまり決定木の変化はコンセプトドリフトの発生を意味していると考えられる．一般的には決定木の比較は難しい問題である．そこで我々は決定木の比較方法を提案し，コンセプトドリフトの発生を決定木の比較により検出する．

Comparison Method for Decision Trees to Analyze Concept Drift

Harunobu Kubo[†1]

In various applications, it is an important research theme to get to know the feature of streaming data which are changing in time. For example, in analysis of purchase history information, it is very important to catch the continuity of a purchase trend and its change, and it forces a life-and-death problem for the company. In stream data mining, a change of the feature of time series data is detected by using classifiers. Such change is called a concept drift. The change of decision trees are considered as the concept drift. We use decision tree as classifiers because it has rich explanation ability of the streaming data. In general to compare decision trees is difficult problem. We introduce the method to compare two decision trees and detect a concept drift based on the comparison method of decision trees.

1. はじめに

多くの分野で，時系列データの分析が研究されている．そのような研究分野では，トレンドの変化の検出が重要なテーマとなっており，いろいろな方法で研究が行われてる．データマイニングの分野では，そのような変化のことをコンセプトドリフトと呼んでいる．そのような変化を検出する技法として多くのものが提案されている[1)-9)]．動的に変化する決定木を用いた手法として VFDT[5)] が知られている．アンサンブル分別器を用いたコンセプトドリフトを扱う方法も提案されている[1),10)]．また，マルコフモデルを利用した過学習を防ぐ方法が研究されている[11)]．

コンセプト間の類似性の比較方法を与えている研究がある[12)]．このような方法では、いくつかのパラメータを導入し，そのパラメータの値を変えることでコンセプトドリフトが発生する頻度を変えることが出来る．コンセプトドリフトがあったか無かったについての判定を行うことが主流となっている．つまりその変化の詳細にまでは言及していない．我々の研究では，より細かいレベルでコンセプトドリフトを扱えるような手法を提案する．

この論文は，次のセクションで問題点について議論をし，セクション 3 では問題を扱うためのアルゴリズムとその手法を議論する．セクション 4 では人工データを用いた実験結果を報告し．セクション 5 では結論をまとめる．

†1 日本アイ・ビー・エム株式会社
System Development Laboratory
kuboh@jp.ibm.com

2. 問題の定義

このセクションでは，決定木を用いた方法についてその問題点を明らかにする．まず始めに決定木をモデルと見て，その変化からコンセプトドリフトを見つけ出すことを考える．つまり、決定木の変化をコンセプトドリフトとみなす．

2.1 ストリーミングデータとコンセプト

まず、我々は，時系列データをチャンクに分解することから始める．

$$data\ \ chunks : \mathrm{D(1)}, \mathrm{D(2)}, ..., \mathrm{D(N)} \qquad (1)$$

さらに，それぞれのチャンクデータから決定木を用いてモデルを以下のように作成する．

$$data\ \ models : \mathrm{M(1)}, \mathrm{M(2)}...\mathrm{M(N)} \qquad (2)$$

各データチャンクにモデルが対応している。つまり，モデル M(1) はチャンクデータ D(1) より決定木を作成することで得られる．次に、ストリーミングデータの一つ一つのデータ（インスタンス）の定義を与える．

$$X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\mathrm{n}, y\} \qquad (3)$$

全部で n 個の説明変数があり，ひとつの目的変数 y がある．目的変数は，以下の二つの値をとるとする．

$$y = \ yes\ \ or\ \ no. \qquad (4)$$

この目的変数は，決定木を機械学習により作成するときに使いる．より一般には，目的変数は複数の値をとることが可能である．そのような場合への拡張は容易なので，ここでは二つの値をとる場合について考える．

決定木を使ってモデルを作成すると，データチャンクの各データは，それぞれリーフノードに分類される．つまりデータ D(i) はルートノードから始まって，中間ノードを経て各リーフノードに分類され，そのデータの分布の様子がモデルを表していると考える．この決定木に対して，コンセプトドリフトの概念を導入する．つまり時系列データから決定木を作成し，その決定木からコンセプトドリフトが発生したかどうかを検出する．決定木の説明能力の高さから，単にコンセプトドリフトが起きたかどうかだけでなく、より詳細な変化の様子が得られると期待される．

まず始めに，決定木の形が時間に依存しない場合について考えてみる．ただし決定木の形は変化していないが，チャンクデータの各インスタンスが，わずかに変化しているとする．つまり決定木の構造のレベルでは変化は無いが、個々のインスタンスレベルでは変化があったとする．あるノードに注目したときに，インスタンスレベルでの分布の変化があったとする．この場合には，二つのノード間でコンセプトドリフトが起きたと考えることが出来る．これは最も簡単なコンセプトドリフトの例となっていると言える．しかし一般的には，決定木の形はインスタンスの違いにより簡単にその形を変えてしまうので、このようなわかりやすい例ばかりとは限らない．つまり決定木の構造が変化した場合に二つの決定木を比較する方法が必要だと言える．そこで，異なった構造を持つ二つの決定木を比較する方法を導入する必要性がある．

2.2 二つの決定木の比較

まず始めにそれぞれの決定木においてどのようにインスタンスが分類されるかを調べることから始める．決定木では，インスタンスはルートノードから始まり，中間ノードを経由して最終的にリーフノードにたどり着く．もし二つの決定木が同じ構造を持っているのならば，インスタンスの流れは、まったく同じものとなるであろう．一般的にはインスタンスの流れの一部は共通だが，ある一部分は異なっていたりする．このように考えると，二つの決定木の違いは，インスタンスがたどる道筋の違いとして捉えることが可能であると分かる．それでは，次に頻度の概念を決定木の全てのノードに対して導入する．

定義 1 (インスタンスの頻度) D をインスタンスの集合とする．各インスタンスはルートノードからリーフノードへと流れていく．もしあるインスタンスが，ノード i を経由したならば，そのノードに割り当てられている頻度 $b(i)$ を+1 増加させる．

この定義により，決定木の比較方法の基礎が与えられる．その方法について後ほど詳しく説明を行う．次にノードのラベルを定義する．

定義 2 (ノードのラベル) インスタンスのもっとも頻度の多かった目的変数をノード i のラベル $l(i)$ とする．

ここで注意しておきたいのは，この二つの定義 1， 2 は，目的変数の種類の数にはよらずに定義できている点である．したがって，目的変数が多数ある場合にも，同様にこれらの定義を使うことが出来る．

そこで，興味深い点は，インスタンスの流れとして，決定木を作成したときに用いたインスタンス以外のインスタンスを使って頻度を計算したらどうなるかという点である．時系列に並んでいるデータチャンクから作成した決定木について考えてみる．時系列の変化を捉えるという観点から，時刻 T と時刻 T+1 の決定木を比較することが一番自然である．この場合に，時刻 T を作成するために使ったインスタンスの集合を，時刻 T+1 の決定木に入力として与えた場合について考えてみる．例えばこれら二つの時刻の決定木が全く同じであったとすると，時刻 T と時刻 T+1 のインスタンスを与えたときに得られる頻度は全く同じになるはずである．次に，二つの決定木に異なっている部分があったとする．この場合に時刻 T のインスタンスを時刻 T+1 の決定木に与えて分類をさせてみる．そのときのインスタンスがルートノードからリーフノードへと流れる様子を追いかけて，各ノードでの頻度を記録するとする．この場合に，もし二つの決定木が類似

している部分があるなら，その部分でのインスタンスの頻度は，かなり似た値となることが予想される．また逆に，決定木の形が異なっている部分では，そのインスタンスの頻度もかなり異なるであろう．例えば、全く異なっているとすると，時刻 T のインスタンスに対して，時刻 T+1 の決定木は何の説明能力も無いはずなので，その目的属性の yes と no の頻度は共に 1/2 程度となるであろう．

我々は，これから時刻 T と時刻 T+1 の時間的に連続した決定木の間の変化に注目する．そして二つの決定木を比較するために，定義 1 を利用し，決定木を直接比較するのではなく，決定木を流れるインスタンスの様子を比べることで，比較を行うこととする．

3. コンセプトドリフトの解析

このセクションでは，二つの決定木をどのように比較するかを説明する．この方法によりコンセプトの違いがわかるようになる．この方法は定義 1 を基本として構成されている．

3.1 決定木と流れの同一視

全インスタンスの集合は，ルートノードで必ずカウントされていることが，定義 1 でわかる．このことは全てのインスタンスがルートノードからたどって，各リーフノードへたどり着くことより明らかである．他の中間ノードやリーフノードの頻度は，インスタンスの一部のみがたどるので、ルートノードの頻度よりも少ないことがわかる．我々は，インスタンスの流れが，決定木のコンセプトを表していると同一視する．

時間の経過によらずに，あるコンセプトが存在したとしてみる．いくつかのノードが時刻 T と T+1 で変化せずに存在し続けたとする．そのようなノードを見つけるために，時刻 T のあるノードのインスタンスを時刻 T+1 の決定木に入力として与える．このことは，時刻 T と T+1 の決定木が同じ構造を持っている場合に，同じ頻度が得られることより自然な定義であることがわかる．さらに，各ノードにラベルを定義 2 にしたがって与える．もし持続的なコンセプトが存在した場合には，多くのインスタンスはそのコンセプトを表しているノードをたどるであろう．また，他のノードをたどる頻度は少ないものとなるであろう．

それでは，決定木とインスタンスの流れを同一視することを定義しよう．

定義 3 (同一視) 各インスタンスはルートノードから始まり中間ノードを経由して，最終的にリーフノードに分類される．このインスタンスの流れを F と定義する．決定木の分類能力を C とする．そこで，$F = C$ と同一視する．

アルゴリズム: 二つの決定木を利用した頻度の計算
入力:
・時刻 T と T+1 の二つの決定木
1. 時刻 T の決定木の全てのノードについて番号 n をふる．全ノード数を N とする．
全ての n について do
2. ノード n でカウントされているインスタンスの D(n) とする．
3. $D(n)$ を，時刻 T+1 の決定木を用いて分類を行う．
4. 時刻 T+1 の決定木より得られた頻度を、時刻 T+1 のノード i について $a(n,i)$ とする．
5. $a(n,i)$ のなかでもっとも頻度の多かった目的属性を，ノードのラベル $m(n,i)$ とする．
end

図 1 時刻 T のノードのインスタンスを時刻 T+1 の決定木の入力として与えて頻度を集計する．

この同一視は，まずインスタンスの集合があれば決定木を再現できること、また決定木があれば，そのリーフノードに分類されているインスタンスに注目するとで，ノードをたどる様子が再現できることより，妥当なものであると考えられる．我々のこのレベルでの内容の一致をもとに，同一視する。二つの決定木を直接比較することは非常に難しい問題である．しかし，この定義により，インスタンスを用いて二つの決定木を比較することが出来るようになる．

3.2 正規化された重み

我々は，時刻 T の決定木のどのノードのインスタンスも，時刻 T+1 の入力データのインスタンスとして考えることが出来る．つまり，時刻 T の決定木のあるノードを選択し，そのノードでカウントされたインスタンスを，時刻 T+1 の決定木への入力とすることが出来る．したがって決定木により与えられる全てのコンセプトについて，時系列の変化を見ることが出来る．時刻 T の決定木に含まれる全ノード数を N とする．時刻 T+1 の決定木の頻度を定義 1 に基づいて計算をする．次のステップとして，時刻 T+1 の決定木に与えられたインスタンスがルートノードからたどる様子を観察し，各ノードにおける頻度を集計する．時刻 T+1 の決定木を作成するときに用いたインスタンスの流れを時刻 T+1 の決定木を用いて集計することでインスタンスの頻度が得られる．このインスタンスのことを基本頻度と呼び b(i) と表す．

また，アルゴリズ 1 により時刻 T のノード n にカウントされているインスタンスを，時刻 T+1 の決定木に与えて頻度を得る．つまり時刻 T のインスタンスを時刻 T+1 の決定木に適用して得られる頻度であ

るので，適用頻度と呼び $a(n,i)$ と表す．ただし分類させるときに何処にも分類されなかったインスタンスも存在し，その場合にはインスタンスを無視することとする．これまでに導入された二つの頻度を比較するために，頻度を正規化して重みを定義する．

定義 4 (基本重み) ノード i において，もっとも目的属性の数が多かったインスタンスについてその頻度を s とする．残りのインスタンスの頻度を t とする．全インスタンス数は，$s+t$ である．基本重み $x(i)$ は $s/(s+t)$ と定義される．

この定義 4 を用いることで定義 1 で与えられる頻度から基本重みを定義することが出来る．つぎに以下のようにして適用重みを定義する．

定義 5 (適用重み) ノード i において，基本重みの目的属性のラベルをもつ適用頻度のインスタンス数を s とする．残りのインスタンスを t とする．全インスタンス数は $s+t$ である．適用重み $y(n,i)$ は $s/(s+t)$ と定義される．

定義 5 で適用重みを定義した．定義をするときに，ラベルを定義 4 と同じものと定義した．頻度を比較するときに，同じ目的属性で比較しないと意味を成さないので，このように定義した．

$$0.5 \leq x(i) \leq 1. \tag{5}$$

これは定義 4 により明がである．一方，適用重みの値の範囲は以下のように定義される．

$$0 \leq y(n,i) \leq 1. \tag{6}$$

値をとる範囲がこのようになるのは適用重みのラベルを基本重みのラベルと一致させたのが理由である．

ここで，我々は二つの重みを定義することが出来た，一つは基本重み $x(i)$ であり，もう一つは適用重み $y(n,i)$ である．基本重みは，決定木を作成するときに用いたインスタンスを，その決定木で分類させたときに各ノードでカウントされる頻度より計算される．適用重みは時刻Tの決定木を作成するときに用いたインスタンスを，時刻 T+1 の決定木に与えて得られた頻度より計算される．

3.3 二次元表示

$x(i)$ と $y(n,i)$ を二次元上で表示する．水平軸を X として垂直軸を Y として，以下のように定義をする．

$$(X,Y) = (x(i), y(n,i)). \tag{7}$$

全ての点は $0.5 \leq x(i) \leq 1$ and $0 \leq y(n,i) \leq 1$ の範囲に表示される．どのようにコンセプトドリフトが起きているかをこの二次元上の点の分布から読み取る方法を説明する．つまり二次元上の点の分布がコンセプトドリフトそのものを表していると考える．

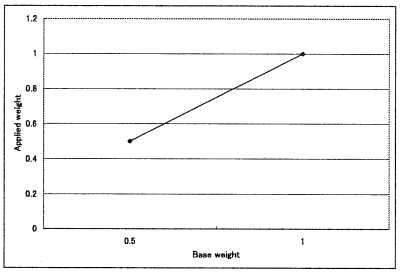

図 2 線分 $Y = X$ 上の点は，コンセプトが持続していることを表している．X 軸は基本重み (base weight)，Y軸は適用重み (applied weight)．

3.4 持続するコンセプト

もし、時刻 T と T+1 に全く同じノードが存在していた場合には，その重みは全く同じものになるであろう．何故ならば，インスタンスがノードをたどる様子が全く同じになるからである．これらのノードは同じ重みを持ち，

$$x(i) = y(n,i). \tag{8}$$

の条件を満たすであろう．つまり，$Y = X$ 上にマップされ図 2 で与えられるような直線上に表示されることになり式 8 を満たすであろう．これらの点については，コンセプトドリフトは起きていないと言うことが出来るだろう．$x(i)$ に含まれる全てのインスタンスが，$y(n,i)$ でカウントされていることになる．このような条件がどのようにして満たされているかを見てみよう．全てのインスタンスはルートノードをスタートとして決定木をたどることになる．もし時刻 T と T+1 のあるノードについてのインスタンスが全く同じであるならば，それらのインスタンスは当然同じノードをたどることになる．つまり二つのインスタンスの集合は同じコンセプトを表している．また一方で，コンセプトドリフトが起こっていることは，点が $Y = X$ にマップされない事から見つけることが出来る．

3.5 コンセプトの消滅

次に，コンセプトが継続しない例を見てみよう．時刻Tであるノードに分類されていたインスタンスを時刻 T+1 の決定木に与えたときに，インスタンスが決定木全体の散らばる場合がある．この場合には，時刻 T で存在していたコンセプトが時刻 T+1 では失われてしまい，特定のコンセプトとして同定できないと考えられる．つまり時刻 T+1 の決定木には，このインスタンスを説明する能力が失われており，約 1/2 の割合で，各ノードに散らばっていくと考えられる．つま

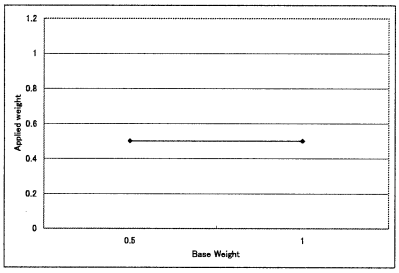

図 3 線分 $Y = 1/2$ 上の点はコンセプトが消滅していることを表している．X 軸は基本重み (base weight)，Y軸は適用重み (applied weight)．

り，適用重みは $Y = 1/2$ の線分上にマップされることになる図 3.

3.6 コンセプトの絞込み

時刻 T のインスタンスの一部が全て $Y = 1$ にマップされた場合を考えてみよう．この場合には，インスタンスの一部が絞込みを受けて $Y = 1$，の点にマップされていると考えることが出来る．したがってこの場合については，コンセプトの絞込みが起きていると考えられる．

3.7 新しいコンセプト

時刻 T+1 の決定木のノードの中で，時刻 T 全インスタンスに対して、ほとんどたどられることの無いノードがある．このような場合には，時刻 T のインスタンスでは表現されていなかった新しいノードが時刻 T+1 で発生していると考えられる．つまり，定義 6 $y(n,i) = 0$ の線分上にマップされる点が存在することになる．もちろん，時刻 T の特定のノードを選んで，時刻 T+1 の決定木に与えた場合にも，ほとんどたどられることの無いノードがある．これは新しいコンセプトの発生というよりは，時刻 T のノードのもつコンセプトに制限したのが理由であると考えられる．したがって新しいコンセプトが出来ているかどうかは，時刻 T のインスタンスを全て入力として与えて判定することになる．

定義 6 (新しいコンセプト) 新しいコンセプトの出現条件: $y(n,i) = 0$

3.8 インスタンス数が少ない場合

次にインスタンスの数が少ない場合について考えてみよう．このようなインスタンスは誤差であると見ることも出来る．しかし他の観点からみると，コンセプトドリフトがピンポイントで発生していると見ることも出来る．そこで、以下のような簡単な例を考えてみよう．二つのインスタンスが一つのノードに分類されているとする．しかしそのインスタンスが持つラベルは yes と no であったとする．この場合には，このノード完全にノイズであり，説明能力を持っていないといえる．次にこのノードのインスタンスを時刻 T+1 の決定木にインプットとして与える．そのときに，一つのインスタンスのみを含むノードにマップされたとする．つまり $Y = 1$ の条件を満たす点にマップされたとする．この場合には明らかに説明能力があり，その予測は 100%yes のコンセプトを表していると考えることも出来る．したがって，インスタンス数が少ない場合には，誤差なのか，それともピンポイントで発生しているとコンセプトドリフトなのかを見極める必要がある．

4. 実　　験

この章では時系列な人工データを用いてどのようにコンセプトドリフトが検出されるかを説明する．決定木としては， 決定木 C4.5[13] を実装している Weka J48[14] を用いる．

4.1 デ ー タ

表 1 文字を値に取る人工データ

data0	data1	data2	data3	objective
A2	B1	C5	D3	Yes
A4	B3	C1	D4	No
A1	B5	C4	D2	Yes

表 2 時刻 T の決定木

Tree	objective	count	id of node
data0 = A1:	No	(16.0/2.0)	1
data0 = A2:	No	(27.0/2.0)	2
data0 = A3			3
— data2 = C1:	No	(5.0)	4
— data2 = C2:	No	(5.0)	5
— data2 = C3:	No	(2.0)	6
— data2 = C4:	Yes	(1.0)	7
— data2 = C5:	Yes	(6.0/1.0)	8
data0 = A4:	Yes	(17.0/5.0)	9
data0 = A5:	Yes	(21.0/2.0)	10

我々は，今まで述べてきたテクニックを moving hyper plane[1)11)15)] を用いた実験に適用する．データは d 次元上に一様分布しているものとする．

$$\sum_{i=1}^{d} a_i x_i = a_0 \tag{9}$$

目的変数の値を $\sum_{i=1}^{d} a_i x_i \leq a_0$ の場合を yes とし他の場合を no とする．この hyper plane がデータを yes

表 3 時刻 T+1 の決定木

Tree	objective	count	id of node
data2 = C1:	No	(24.0)	1
data2 = C2			2
— data0 = A1:	Yes	(6.0)	3
— data0 = A2:	No	(4.0)	4
— data0 = A3:	No	(5.0)	5
— data0 = A4:	No	(5.0)	6
— data0 = A5:	No	(2.0)	7
data2 = C3			8
— data0 = A1:	Yes	(2.0)	9
— data0 = A2:	Yes	(6.0)	10
— data0 = A3:	Yes	(2.0/1.0)	11
— data0 = A4:	No	(1.0)	12
— data0 = A5:	No	(7.0)	13
data2 = C4:	Yes	(20.0/1.0)	14
data2 = C5:	Yes	(16.0/2.0)	15

と no に分類しその値は，hyper plane が動くことにより変化する．したがってこの hyper plane が動くことでコンセプトドリフトが発生するとする．データの要素を d 次元 $[0,1]^d$ の一様分布したランダムな値とする．重み $a_i i = 1,\ldots,d$ は $[0,1]^d$ のランダムな値とする．a_0 は $a_0 = (1/2)\sum_i^d a_i$ と定義し，およそ半数が yes で残りの半数が no となるようにする．我々は $d = 4$ の四次元空間上で実験を行い，各要素は *data0*, *data1*, *data2*, *data3*, *objective* の値をとる．この時系列データを 100 件単位でチャンクに分解する．時系列データは時間と共に変化し，途中でコンセプトドリフトが発生しているとする．四次元空間の各軸を五つの領域に分割しする．これは議論をわかりやすくするためのものである．*data0* は 'A1', 'A2', 'A3', 'A4', 'A5' を含み，*data1* は 'B1', 'B2', 'B3', 'B4', 'B5' を含み，*data2* は、'C1', 'C2', 'C3', 'C4', 'C5' を含み，*data3* は，'D1', 'D2', 'D3', 'D4', 'D5' を含むとする．説明変数の値に応じて 5 つの区間 $[0,0.2]$, $[0.2,0.4]$, $[0.4,0.6]$, $[0.6,0.8]$, $[0.8,1.0]$ に分割する．そして文字 'A1' を区間 $[0,0.2]$ に、'A2' を $[0.2,0.4]$ に，'A3' を $[0.4,0.6]$ に 'A4' を $[0.6,0.8]$ に 'A5' を $[0.8,1.0]$ に対応させる．同様な定義を 'B*', 'C*', 'D*'(*は 1,2,3,4,5 を表す) にも導入する．説明変数と目的変数は時間と共に変化し，それがコンセプトドリフトを引き起こすことになる．データ全体に対して 5%のの誤差を入れた．実際のデータはテーブル 1 のように与えられる．

4.2 リーフノードを用いたコンセプトドリフトの解析

決定木の例として、テーブル. 2 と 3 を例にとって説明する．水平軸を X と呼び垂直軸を Y と呼ぶことにする．ここで改めて X 軸の範囲が $[0.5,1]$ に限られることを説明する．これはノードのインスタンスをその目的属性が一番大きかったものを選ぶことにしたためである。一方，Y 軸については値は 0 から 1 までの範囲を取る．例を用いて説明をすると，時刻 T のインスタンスが目的変数が yes でありその頻度をあらわす重みが $X = 0.7$ であったとする．このときに時刻 T+1 の注目しているノードの目的属性が no でありその重みが $Y = 0.8$ であったとする．このとき目的属性の値が異なっているために直接重みを比較することが出来ない．そこで先に説明をした通りに，時刻 T のインスタンスの目的属性を no に入れ替えてその値を $Y = 0.3$ にする．その結果，値のとる範囲が広くなることがわかる．

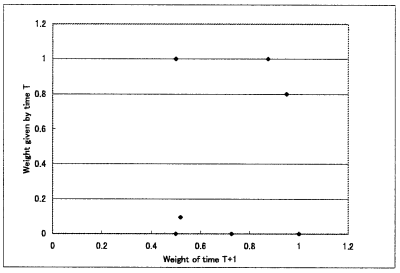

図 4 ほとんどコンセプトが継続していない様子が分かる．X 軸は基本重み，Y軸は適用重み．

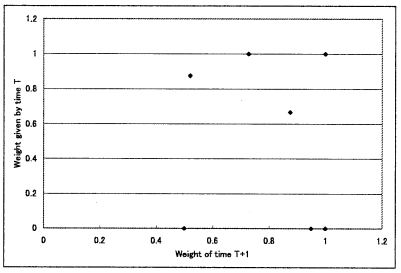

図 5 $(X,Y) = (0.5,1),(0.8,1),(1,1)$ はコンセプトドリフトが発生したことを表している.X 軸は基本重み，Y軸は適用重み.

決定木のテーブル 2 と 3 について見るとそれぞれ 10 と 15 のノードがあることが分かる．つまり，二つの決定木の形は異なっており直接比較するのは難しい問題となる．そこで我々が提案した手法を使って二つの決定木を比較してみる．まず始めに決定木 2 の id=1 のノードと時刻 T+1 の決定木 3 を比較してみる．その結果が図 4 である.$X = Y$ の線分に乗っている点はほとんど無い，つまりコンセプトが継続していなかったことを表している．また $Y = 0$ の上に乗っている点があるが，これは新しいコンセプトの出現を意味しているとは限らず，単に時刻 T の中から選んだノードのインスタンス数が少なかったために，インスタン

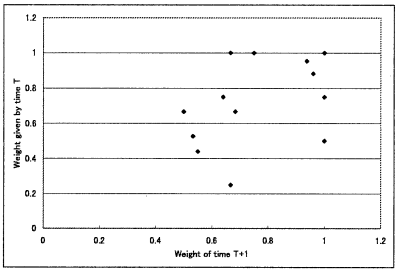

図 6 時刻 T=1（チャンク 1）.$Y = X$, $Y = 1/2$ 上に点が集まっている．コンセプトドリフトは起きていない．X 軸は基本重み，Y軸は適用重み．

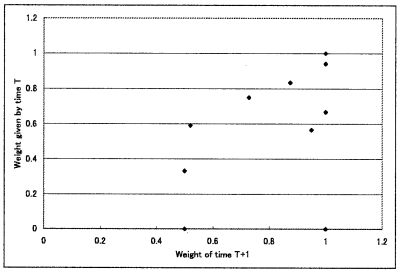

図 8 時刻 T=3（チャンク 3）．点は散らばっており $Y = 0$ 上にも点が存在する．コンセプトドリフトが起きている．X 軸は基本重み，Y軸は適用重み．

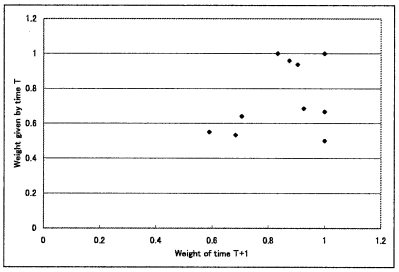

図 7 時刻 T=2（チャンク 2）.$Y = X$, $Y = 1/2$ 上に点が集まっている．コンセプトドリフトは起きていない．X 軸は基本重み，Y軸は適用重み．

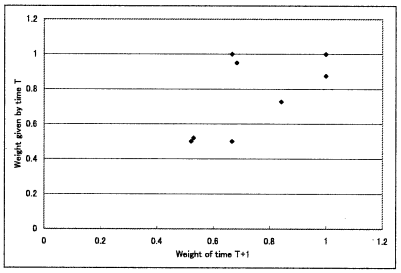

図 9 時刻 T=4（チャンク 4）.$Y = X$, $Y = 1/2$ 上に点が集まっている．コンセプトドリフトは起きていない．X 軸は基本重み，Y軸は適用重み．

スが分類されていないだけと考えられる．

図 4 を詳細に見ていこう．点は大まかに言って 3 つに分類される．点 $(X, Y) = (0.5, 1)$ は，時刻 T+1 の決定木の基本重みが $X = 1/2$ であることから分類能力を失っていると考えられる．しかし $Y = 1$ であることを考慮に入れると，インスタンスの絞込みが行われたと考えられ，コンセプトが絞り込まれたと言える．このようにして時刻 T と時刻 T+1 の決定木を比較することが出来る．次に，時刻 T の決定木の id=10 のノードについて考えてみよう，時刻 T+1 の決定木にインスタンスを与えた結果は図 5 である．この場合には $(X, Y) = (1.0, 1.0)$ の点がある．これはコンセプトが引き続き存在していることを表していると考えられる．

4.3 ルートノードを使ったコンセプトドリフトの解析

次に全てのインスタンスを使ってコンセプトドリフトを解析する．我々の定義では，全てのインスタンスはルートノードから決定木をたどるので，ルートノードで頻度を数えられたインスタンスを時刻 T+1 の決定木に入力として与えると言うことも出来る．図 6，7，8，9，これらの図は時系列に並んだものである．このデータでは時刻 $T = 3$（データチャンク 3）でコンセプトドリフトを発生させている．二つのタイプのグラフがあることが分かる．一つは，図 8 であり，コンセプトドリフトが起きている図である．$Y = 0$ 上に乗っている点が複数観察される．特に $(X, Y) = (1, 0)$ の点には複数の点が重なって表示されており，この点がコンセプトドリフトの発生を表している．もう一つは，図 6，7，9 であり $Y = X$ or $Y = 1/2$ の近くに点が集まっている様子が分かる．このような図中の点の偏りを分散を使ってあらわすことも出来る．平均は $Y - X$ に対して定義する．注意が必要なのは，直線 $Y = X$ と点との距離を測っているのではないという点である。$Y - X$ は，二つの重みの差分を表しており，まさにその差分がコンセプトドリフトの大きさを表しているからである．分散は図 10 に与えた．我々が導入したコンセプトドリフトの発生位置で，丁度分散が大きくなっていることが分かる．

4.4 三次元表示

時系列データを考慮して，三次元上に点を表示させることとする．各時刻での重みの分布を相互に比較できるように，インスタンスとしてはルートノードでカウントされたもの，つまり各チャンクデータを用いて

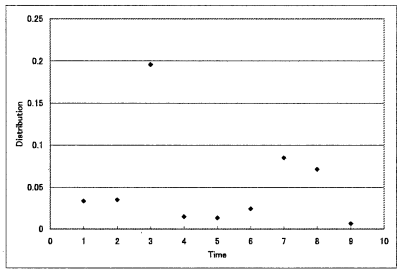

図 10　時刻 1 から 9 の $Y - X$ の分散.

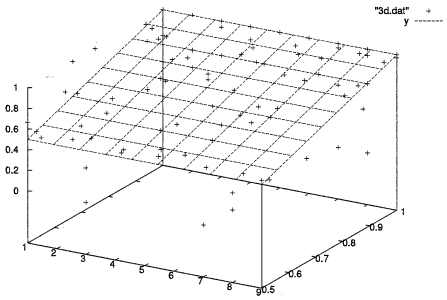

図 11　(X, Y, Z) 三次元表示. 時間の範囲は [1,9]. X 軸の範囲は $[0.5, 1]$ Y 軸の範囲は $[0, 1]$.

いる. 多くの点が $Y = X$ の平面上に集まっている. またコンセプトドリフトが起きている点では, $Y = X$ から離れた場所に点が分布している. この三次元表示の各時刻で見た二次元マップのいくつかは, 図 6 - 9 で与えられている.

5. おわりに

時系列データから決定木を作成し, その相互比較をする方法を提案した. 直接比較することが難しい決定木に対してインスタンスの流れを比較することで, 決定木の比較を可能とした. その結果コンセプトドリフトの起きている様子を明らかにすることが出来た。

参考文献

1) P. Y. H. Wang, W. Fan and J. Han, "Mining concept-drifting data stream using ensemble classifiers." in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003, pp. 226–235.

2) M. Scholz and R. Klinkenberg, "An ensemble classifier for drifing concepts." in *Proceedings of the 2th International Workshop on Knowledge Discovery and Data Stream at ECML/PKDD*, 2005, pp. 53–64.

3) M. R. B.Habcock, S.Babu and J.Widom., "Models and issues in data stream system." in *POSD,*, 2002.

4) J.B. Y.Chen, G.Dong and J.Wang., "Multi-dimensional regresshin analysis of time-seeries data streams." in *VLDB,*, 2002.

5) P.Domingos and G.Hulten., "Mining high=speed data streams." in *SIGKDD,*, 2000, pp. 71–80.

6) R.C.SHafer and M.Mehta., "Sprint: A scalable parallel chassifier for data mining." in *VLDB,*, 1996.

7) R.J.Gehrke and V.Ganti, "Rainforest: A framwork for faster decision tree construction of large datasets." in *VLDB,*, 1999.

8) V.R. J.Gehrke and W.Loh, "Boat optimistic decision tree construction," in *SIGMOD,*, 1999.

9) P.E.Utgoff, "Incremental induction of decision trees," *Machine Learning*, vol.4, pp. 161–186, 1989.

10) K.Tumer and J.Ghosh, "Error correlation and error reduction in ensemble classifiers," *Connection Science*, vol.8, pp. 385–403, 1996.

11) J.P. P. S.Y. HaixunWang, JianYin and J.X. Yu, "Suppressing model overfitting in mining concept-drifting data streams," in *SIGKDD,*, 2006, pp. 736–741.

12) Y. Yang, X. Wu, and X. Zhu, "Combining proactive and reactive predictions for data streams." in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2005, pp. 710–715.

13) J. R. Quinlan, *C4.5: programs for machine learning.* Morgan Kaufmann Publishers Inc., 1993.

14) I.H. Witten and E.Frank, *Data Mining: Practical Machine Learning Tools.* Morgan Kaufmann Publishers Inc., 2005.

15) L. G.Hulten and P.domingos, "Mining time-changing data streams." in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 97–106.