

データストリーム処理へのベイジアンネットワークの導入

佐藤 亮[†] 川島 英之^{†,††} 北川 博之^{†,††}

[†] 筑波大学大学院システム情報工学研究科 〒305-8573 茨城県つくば市天王台 1-1-1

^{††} 筑波大学計算科学研究センター

E-mail: [†]punisiro@kde.cs.tsukuba.ac.jp, ^{††}{kawasima,kitagawa}@cs.tsukuba.ac.jp

あらまし 本研究では確率推論の一技法であるベイジアンネットワークと関係ストリーム処理を統合するシステムについて述べる．ベイジアンネットワークはオブジェクト関係データモデルにおける ADT として格納される．ベイジアンネットワークへの入力イベント生起である．これを取得するため、入力リレーションをイベントに変換する ASSOC 処理を提案する．次に、BN 演算子を提案する．BN 演算子はデータストリームに対して ASSOC 処理を実行し、確率伝播を行った後、問合せ内容に応じて確率的タプルを生成する．問合せではメソッドを用いることでグラフ構造のベイジアンネットワークから確率的タプルが抽出される．BN 演算子は集約、結合演算子と同様に状態を有し、高い計算コストを必要とする．そして ASSOC 処理を記述するための構文および問合せ言語について述べた後、ネットワークカメラを用いて作成した 6 ノードから構成される実験用ベイジアンネットワークについて述べる．

キーワード ベイジアンネットワーク、ストリーム処理、確率的データストリーム、オブジェクト関係データモデル．

Introduction of Bayesian Networks for Data Stream Processing

Ryo SATO[†], Hideyuki KAWASHIMA^{†,††}, and Hiroyuki KITAGAWA^{†,††}

[†] Graduate School of Systems and Information Engineering, University of Tsukuba 1-1-1 Tennodai, Tsukuba, 305-8573

^{††} Center for Computational Sciences, University of Tsukuba, University of Tsukuba

E-mail: [†]punisiro@kde.cs.tsukuba.ac.jp, ^{††}{kawasima,kitagawa}@cs.tsukuba.ac.jp

Abstract This paper describes a system that integrates a probabilistic reasoning technique Bayesian networks and relational stream processing. Bayesian networks are stored in the form of ADT in the object relational data model. Inputs to a Bayesian network is event occurrences. To obtain them, we propose an operator that converts an input relation to events, and refer it to ASSOC. Next, we propose BN operator. The BN operator firstly conducts ASSOC and then probability propagation and then it outputs probabilistic tuples with regard to a query. On the query, methods extract probabilistic tuples from graph structured Bayesian networks. The BN operator is a stateful operator like aggregation and join operator, and it requires high computational cost since it may generate more tuples than inputs. And, we describe a syntax to describe ASSOC and query language. Then finally we describe a Bayesian network consisted of 6 nodes.

Key words Bayesian Networks, stream processing, probabilistic data stream, object relational data model.

1. はじめに

近年、無線センサノードや GPS や RFID、ネットワークカメラのような様々なセンサデバイスが使われている．これらのセンサデバイスからは逐次大量のデータが生成されており、このようなデータをデータストリームと呼ぶ．データストリームを用いて実世界で生じるイベントを検知したり、あるいは検知結果から状況を推論する研究も行われている．この推論技法の代表的なものに、ベイジアンネットワーク [2] がある．ベイジアンネットワークを用いて実世界の状況を認識するシステムは多数開発されている．その例には状況推定 [1][17]、麻酔行為表

現、電子メールエージェント等がある．これらは時間的変動のない静的な事象を対象としているため、動的な事象であるデータストリームを対象にしていない．

そこで本研究は、ベイジアンネットワークを用いて不確実な状況を表す確率的データストリームを連続的に処理するシステムを提案する．

ベイジアンネットワークを用いたデータストリーム処理システムを構築するためには 3 つの課題がある．1 つ目は、ベイジアンネットワークから生成されるデータとデータベース内に格納されているデータとの関係演算処理を実行するために、グラフ構造であるベイジアンネットワークを関係データモデル内で

表現する方法である。

2つ目は、データストリーム処理システムに対する入力であるデータストリームをベイジアンネットワークの各イベントに対応させる方法である。

最後の3つ目は、グラフ構造であるベイジアンネットワークから、ベイジアンネットワーク内の各イベントの確率値等の情報を取得する方法である。本研究では、以上の3課題を解決し確率的データストリーム処理システムの構築を行う。

本論文の構成は以下の通りである。2節ではベイジアンネットワークとデータストリームについて概説する。3節では提案システム構築における3つの課題についての解決方法を説明する。4節では提案システムの評価実験について述べる。5節で関連研究と本研究を比較する。最後に6節では結論を述べる。

2. ベイジアンネットワークとデータストリーム

本節では、確率を用いるために使用しているベイジアンネットワークの説明と本研究で扱うデータストリームについて述べる。

2.1 ベイジアンネットワーク

不確実な状況を表現するためにベイジアンネットワークが広く用いられている。ベイジアンネットワークとは、確率的な因果関係をモデル化する手法であるグラフィカルモデリングの一つである。因果関係のある事象の確率変数をノードとするグラフ構造と、各ノードに割り当てられた条件付き確率分布群によってベイジアンネットワークは定義される。因果関係の有無を有向グラフで表す。また、因果関係の強さを条件付き確率で表す。各変数の条件付き確率分布は、本研究で対象とする離散的な確率変数の場合には、親ノードと子ノードの各変数がとる具体的な値ごとに割り当てられた条件付き確率を離散的な表、すなわち条件付き確率表(Conditional Probability Table(CPT))として表現する。図1にベイジアンネットワークの例を示す。このベイジアンネットワークはA, B, C, Dの4ノードで構成されている。本研究で扱うベイジアンネットワークは、それぞれのノードがtrueとfalseの2値だけを扱うだけでなく、ノードAのような複数イベント(eventX, eventY, eventZ)も扱う。

2.1.1 確率推論アルゴリズム

ベイジアンネットワークの各ノードの確率値を求めるアルゴリズムは大きく分けて2種類存在する。それらは、近似推論アルゴリズムと厳密推論アルゴリズムである。近似推論アルゴリズムは、過去にどのくらいイベントが生じたなどのサンプル数に依存する計算方法なので、サンプル数が多いと精度が高くなるがサンプル数が少ないと精度が低くなる。そこで、本研究では正確な確率値を求めるために、厳密推論アルゴリズムを使用し、そのアルゴリズムである Pearl's Message-Passing [9] のアルゴリズム(以下、メッセージパッシング)を採用した。

2.1.2 Pearl's Message-Passing アルゴリズム

本研究で使用するメッセージパッシングは、あるイベントが生じたらその情報を全てのノードに伝播し全ての確率を計算

するアルゴリズムである。このアルゴリズムの処理を図1を使用し説明する。図1のベイジアンネットワークにおいてノードAのeventXが生起すると、その情報をノードBとノードCに伝播し確率計算を実行し、さらにノードCはノードDに情報を伝播し確率計算を実行する。全てのノードに伝播し終わったら確率計算が終了するアルゴリズムである。この確率計算方法だとノード数の増加や、生起イベントの増加に比例して計算量が増大する欠点がある。

2.2 データストリーム

近年、大量に流れ来るデータを効率的に処理する技術としてデータストリーム処理が注目されている。データストリームとは、センサデバイスなどの情報源から逐次配信され、刻一刻と変化していくデータのことである。データストリームに関する研究は2002年頃[14]から始まった。当初は関係データ演算をストリームモデルに適応させ、処理系において性能を追求する研究が行われたが、現在では多様なアプリケーションに特化した専用問合せ言語を持つ専用システムに関する研究が多数行われている。本システムでは、センサデバイスなどから配信されるデータストリームを受信しイベント生起を表すタプルに変換して、確率的データストリームという各タプルに確率値が付与されているデータストリームをユーザに配信する。

3. データストリーム処理システムに対するベイジアンネットワークの導入

本節では、データストリーム処理システムに対するベイジアンネットワークの導入における3つの課題の解決方法と提案システムの処理について述べる。

3.1 DBにおけるベイジアンネットワークの表現

本節では、グラフ構造であるベイジアンネットワークのDB内における表現方法について述べる。

3.1.1 オブジェクト関係データモデル

ベイジアンネットワークの推論結果と、DB内に格納され

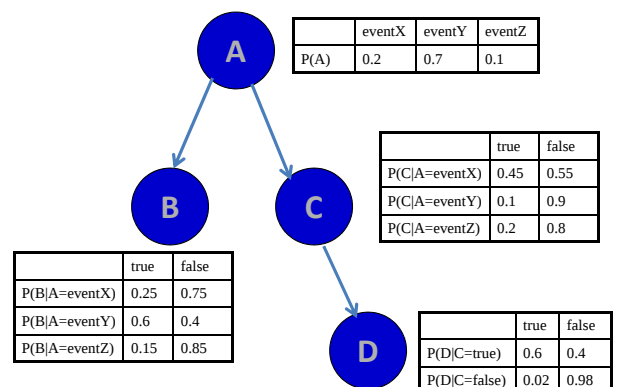


図1 ベイジアンネットワークの例

ているデータを関係演算処理を用いて統合利用するために、ベジアンネットワークを関係データベースに格納する。

ベジアンネットワークと関係データベースを自然に統合するため、オブジェクト関係データモデルを採用し、オブジェクトにベジアンネットワークを格納する。ベジアンネットワークはグラフ構造を有するため、そのままでは関係データベース内で扱うことができない。この問題を解決するためにオブジェクト関係データベースのオブジェクトとして、ベジアンネットワークの拡張データ型 (ADT) を作成する。オブジェクト関係データベースは Java 言語などのオブジェクト指向のプログラミングで使用するオブジェクトの形式で表されるデータをデータベースシステムに取り入れたものである。そして、ADT として BN-Obj 型を定義し、ベジアンネットワークのオブジェクトを格納する。BN-Obj を作成することにより、今まではデータベースに蓄えられていなかったベジアンネットワークをデータベース内に格納可能になる。データベースにベジアンネットワークを格納することで、関係テーブルに格納されているデータと同じようにベジアンネットワークを扱うことができ、関係演算子 (選択, 射影, 結合等) を使用可能になる。

3.2 ベジアンネットワークに対する入力

提案システムは、センサデバイス等から生成されるデータストリームを受信し、そのデータに基づきベジアンネットワークの各イベントの確率計算を実行する。そのとき、データストリームがどのベジアンネットワークのイベントと対応しているかわからないためデータストリームをイベント生起を示すデータに変換しなければならない。そこで、データストリームとベジアンネットワークのイベントを対応付けるために、ASSOC 処理を提案する。これは入力データストリームを有限の関係データに変換する window 演算 [18] の直後に適用される。また、本システムが扱うデータモデルには、関係データモデル (R)、データストリームモデル (S)、確率的関係データモデル (PR)、そして確率的データストリームモデル (PS) の 4 種類がある。これら異なるデータモデルを統一的に扱うためには、モデル変換演算子が必要になるため、データモデルを変換する演算子を導入する。

3.2.1 ASSOC 処理: 入力データストリームとベジアンネットワークとの関連付け

window 演算により関係データに変換されたデータに対して、ASSOC 処理を適用すると、入力データ 1 タプルにつき 1 タプルのイベント生起情報を含む関係データが生成される。なお、window 演算における window 幅はこの ASSOC 処理を定義する ASSOC 構文に記述される。構文の詳細は 3.5 節で述べる。

3.2.2 データモデル変換演算子

本提案システムで扱うデータモデルは 4 種類であるから、全ての組合せを考えると $16 (= 4 \times 4)$ 種類のモデル変換演算子が考えられるが、本研究では 7 つのモデル変換演算子のみを扱う。その他 9 つのモデル変換演算子は、本研究で扱う 7 つのモデル変換演算子の組合せや、PS は入力としては出現せず出力のみ出現する事などを考慮し、定義しない。そして、本研究ではベジアンネットワークと関係データベースとの統合を行なうた

めに、ベジアンネットワークが有する確率的データモデルを確率的関係データモデルへと変換する演算子を導入する。

これら 7 つのモデル変換演算子は 3 種類の状況で用いられる。それらの状況は、提案システムへの入力時、提案システムからの出力時、そして提案システム内での処理時である。

- 提案システムへの入力時

- S-to-R (Stream to Relation)

入力データストリームを有限の関係データに変換する。S-to-R には window 演算がある。データストリームとベジアンネットワークを関連付けする ASSOC 構文は、ここで適用する。このモデルは [18] の window 演算に基づいている。

- 提案システム内での処理時

- R-to-R (Relation to Relation)

有限長の関係データから有限長の関係データを導出する。PR-to-PR 演算子とは異なり、問合せ処理において、Safe、即ちタプル独立性保存を考慮して問合せ計画を立てる必要はない。R-to-R は通常の関係演算子を全て含む。

- PR-to-PR (Probabilistic Relation to Probabilistic Relation)

有限長の確率的関係データから、有限長の確率的関係データを導出する。R-to-R 演算子とは異なり、問合せ処理において Safe、即ちタプル独立性を保つ問合せ計画が立案される必要がある [16]。PR-to-PR は Safe である関係演算子を全て含む。さらに、ベジアンネットワーク上のイベント生起確率を確率的関係データへ変換する。この詳細は 3.3.2 節で述べる。

- R-to-PR (Relation to Probabilistic Relation)

関係データを確率的関係データへと変換する。確率的関係データと関係データとの結合の時に、関係データは本演算子を適用する必要がある。

- PR-to-R (Probabilistic Relation to Relation)

確率的関係データを関係データへと変換する。

- 提案システムへの出力時

- PR-to-PS (Probabilistic Relation to Probabilistic Stream)

有限の確率的関係データを無限長の確率的データストリームに変換する。

- R-to-S (Relation to Stream)

有限の関係データを無限長のデータストリームに変換する。

3.3 ベジアンネットワークからの推論結果抽出

DB 内に格納したベジアンネットワークから確率計算を実行後の確率値を抽出するとき、BN-Obj 型はオブジェクト型であるため、通常の関係演算子に適用できない。これを解決するために、本提案システム特有のベジアンネットワーク演算子 (以下 BN 演算子) を BN-Obj 型に用意し、BN-Obj をタプルに変換することで、問合せに対応する。この詳細を下記で述べる。

3.3.1 BN 演算子

本提案システムでは、リレーショナル演算子や window 演算子の他に、リレーションを受信し、ASSOC 処理を実行してイベント生起のデータに変換し、その情報を基に確率計算を実行してタプルを生成するメソッド群を BN 演算子として扱う。

BN 演算子の入力タプルは WINDOW 演算により生成される有限長のリレーションである。BN 演算子は、window 内に入っ

ているタプルを受信し，ASSOC 処理を実行してタプルをイベント生起のタプルに変換する．イベント生起タプルと 3.3.3 節で説明する lifespan により現在生起しているイベントを決定し確率計算を実行する．確率計算を実行後のベイジアンネットワークから 3.3.2 節で述べるメソッドを使用し指定したタプルを生成する．BN 演算子は複数タプルを生成することもある．複数イベントを扱うベイジアンネットワークのノードでは，1 つのノードで同時に複数のイベントが生起することは有り得ないため，同じノードのイベント生起のタプルを複数個受け取った場合には，最新のタプルのイベント生起のみ適用する．

3.3.2 タプル生成メソッドの定義

グラフ構造である BN-Obj 型から確率値を有するタプルを抽出するためのメソッド，getNodeContent() を本節で定義する．

a) getNodeContent(output-attr, nodeset)

getNodeContent() は nodeset で指定したベイジアンネットワークのノード集合から output-attr で指定した属性で構成するタプルに変換する．

b) output-attr

output-attr は **Time**，**Id**，**Node**，**Ev**，**Pr** から好きな属性を指定して出力する．出力する属性は複数でも可能である．**Time** は時間，**Id** はノードの ID 番号，**Node** はノードの名前，**Ev** はイベントの名前，**Pr** は確率値の属性である．例えば，全ての属性を出力したい時には，getNodeContent(*, nodeset) と記述すれば良い．又，**Ev**，**Pr** の 2 つの属性が欲しい時は，getNodeContent((Ev,Pr), nodeset) と記述すれば良い．

c) nodeset

nodeset は，ベイジアンネットワークオブジェクトから，任意のノード集合を選択する．ノード集合を抽出する構文を図 2 の BNF に示す．get は，nodeset により抽出されたノードの子や親や兄弟ノードを抽出する．そして select は，条件に一致するノードを抽出する．引数の value は任意の数値である．

```
nodeset ::= monadic | dyadic
dyadic ::= nodeset aggregate nodeset
monadic ::= getNode | selectNode
get ::= getChild(nodeset) |
      getParent(nodeset) |
      getSibling(nodeset)
select ::= selectTime(compare, value) |
          selectId(compare, value) |
          selectEv(compare, value) |
          selectPr(compare, value) |
          selectNode(compare, value)
aggregate ::= | ∩ | -
compare ::= < | <= | = | >= | > | <>
```

図 2 BNF of nodeset

3.3.3 ベイジアンネットワークモデルの拡張

2.1 節で定義したベイジアンネットワークのモデルには生起期間の概念が無い．あるイベントが生起したら，その一瞬だけイベントが生起することは不自然なので，Lifespan を導入する．

Lifespan はイベントの生起期間を表す．Lifespan の例を図 3 に示す．図 3 中の下向矢印はノードの生起を示し，右向矢印の長さはイベント生起期間を示す．図 3 において，時刻 t_2 においてはイベント A のみが生起している状態を表す．時刻 t_3 ではイベント A とイベント B が生起している状態を表す．そして時刻 t_4 ではイベント B が生起している状態を表す．すなわち，問合せが時刻 t_2, t_3, t_4 で実行されたとすると，それぞれイベント A，イベント A とイベント B，イベント B が生起しているとみなされてベイジアンネットワーク中で確率伝播が実行される．

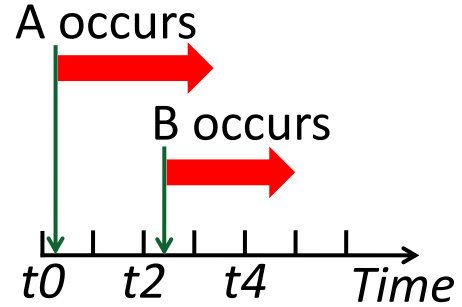


図 3 Lifespan

3.4 システムの処理の流れ

以上のようにしてベイジアンネットワークをデータストリーム処理システムに導入した．ここで，本提案システムの処理の流れを説明する．

提案するシステムは，データストリームを入力として受信し，ベイジアンネットワークを用いて確率的関係データを生成した後，それと既存の関係データと統合し，その結果を確率的データストリームとして出力する．図 4 に本システムの処理の流れを示す．図 4 では，ユーザは本システムに対して「部屋 103 の火災の確率と，火災時の問合せ先の情報が欲しい」という連続的問合せを登録している．システムはセンサからのデータストリームを window 演算子を用いて有限長の関係データに変換する．そのデータに対して ASSOC 処理を適用し，イベント生起の情報を示すタプルに変換する．ASSOC 構文の詳細は 3.2.1 節で述べた．図 4 の例では，温度の数値データを「高温」というイベントの生起に変換している．図 4 中 BN(ベイジアンネットワーク) 演算子は，この window 内のタプル 1 つを受け取り，ASSOC 処理を実行して，その情報に基づいて「火災」の確率値を計算し，確率的関係データを生成する．BN 演算子の詳細は 3.3.1 節で述べた．上記のように生成した確率的関係データと既存の関係データを統合し，確率的データストリームとしてユーザに配信する．このようにシステムはユーザが欲しい不確実な情報を既存の関係データと統合し配信できる．

3.5 問合せ言語

最後にユーザがシステムに対してどのような問合せを記述するかについて述べる．提案システムに対する問合せは，ユーザが容易に記述できるよう SQL ライクにした．この例を図 5，図 7 に示す．

図 5 の問合せ例は，tableBN というテーブルから，RoomN =

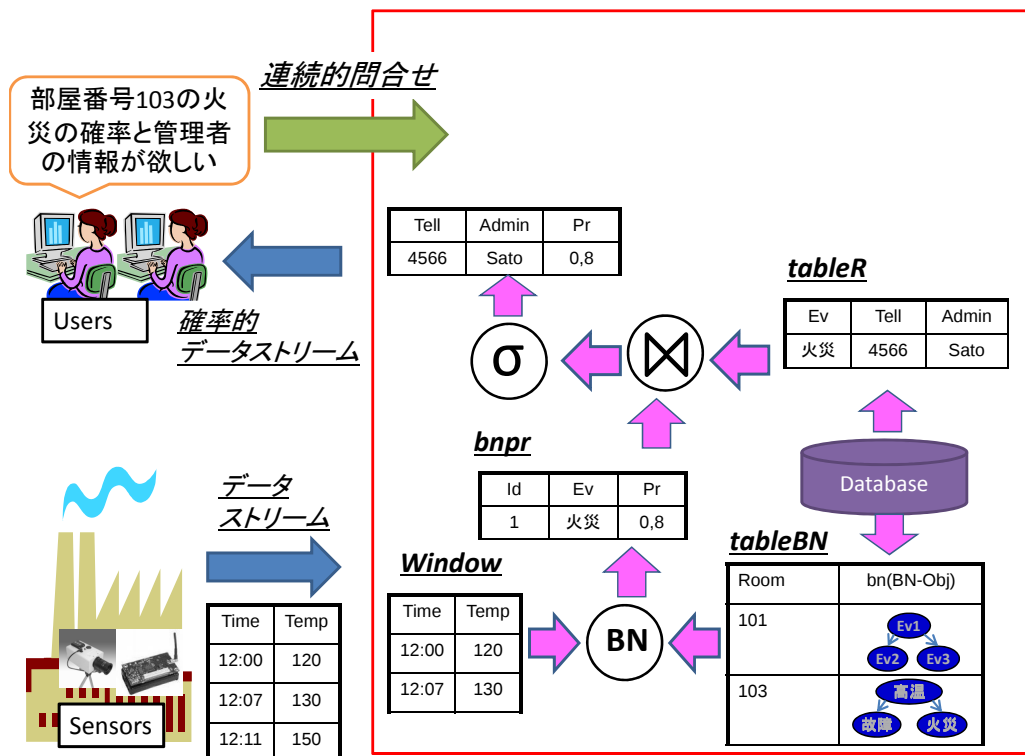


図 4 処理の流れ

```

MASTER 10sec
SELECT bn.getNodeContent
(*,selectPr(>,0.5))
FROM tableBN
WHERE tableBN.Room = '103'
ASSOC FROM sensor1[5]
TO tableBN.BN
WHERE tableBN.Room = '103'
CONDITION WHEN sensor1.temp > 50 AND
BN.Node = 'NodeA' THEN 'eventX'

```

⇒

Time	Id	Node	Ev	Pr
T1	1	NodeA	eventX	1
T2	2	NodeB	eventB	0.8

図 5 問合せ例 1

'103' であるタプルに格納されているベイジアンネットワークに対して、 $Pr > 0.5$ のイベントの情報を 10 秒間隔で要求している。このとき、ASSOC 節以下で sensor1 からの属性名 temp というデータを window 幅 5 タプルで NodeA の 'eventX' というイベントに変換している。この時 $Pr > 0.5$ のイベントが eventX と eventB だとする。この問合せ結果として図 5 の下に示す確率的データストリームがユーザに配信される。

3.5.1 各節の説明

図 5 を用いて各節の説明をする。

a) MASTER 節

MASTER 節では、提案システムに対する問合せを実行する

タイミングを定義する。Master は時間幅のパラメタにより決定される。例えば 10 秒毎や、1 時間毎などである。

b) SELECT 節

SELECT 節では、射影演算、集約演算を記述する。BN-Obj 型に対する処理である getNodeContent() メソッドはここで記述する。また、ベイジアンネットワークから生成されるタプルが複数存在する可能性があるため、メソッドから生成されたタプルは UNNEST 操作を用いて平坦化を行うことにより、第一正規形に変換する。

c) FROM 節

FROM 節は、問合せ対象とするテーブル名を記述する。

d) WHERE 節

WHERE 節では、FROM 節で選択したテーブルに対しての選択演算処理を記述する。

e) ASSOC 節

ベイジアンネットワークと対応付けを行う情報源を FROM 節で記述する。TO 節と WHERE 節で対応するベイジアンネットワークのイベントが格納されているテーブル名と属性名を記述する。CONDITION 節で対応するイベント名と条件を記述する。情報源を選択するとき window 幅も指定する。又、ASSOC 構文は図 6 のように予め登録して、図 7 のようにして ASSOC 構文を使用することも可能である。この時 WINDOW 幅は ASSOC を使用するとき指定する。

3.6 BN 演算子の挙動

図 4 にあるように、ASSOC 処理は WINDOW 演算子の出力であるリレーション R を、 R 内のタプル数と同数のイベントを有するリレーション E に変換する。

```

CREATE ASSOC sensor1
FROM sensor1
TO tableBN.BN
WHERE tableBN.Room = '103'
CONDITION WHEN sensor1.temp > 50 AND
BN.Node = 'NodeA' THEN 'eventX'

```

図 6 ASSOC 構文の登録

```

MASTER 10sec
SELECT bn.getNodeContent
(*,selectPr(>,0.5))
FROM tableBN
WHERE tableBN.Room = '103'
USING ASSOC sensor1[5]

```

図 7 問合せ例 2

ストリーム処理における集約演算子や結合演算子は状態付演算子 (Stateful Operator) である [19], [20]。我々が導入した lifespan により, BN 演算子も同様に状態付演算子となっている。

BN 演算子は E を受け取ると, アルゴリズム??に示すように動作する。まず, E 中の全イベントに関する lifespan をベイジアンネットワークで設定する。次に, 異なる状態を全て取得する。例えば入力 E 中に 3 つのイベント A, B, C が有り, 図 8 のように lifespan が設定された場合には, アルゴリズム 1 中の $getDifferentStates$ は $\{A\}$, $\{A, B\}$, $\{B, C\}$ という 3 つの異なる状態を返す。そして各状態について, 確率伝播, ノード選択, 確率的タプル生成を終えた後, BN 演算子は各回に生成されたタプルの和を出力キューに出力する。

頻繁にイベントが発生するような環境では, $getDifferentStates$ の計算コストは高くなる可能性がある。なぜなら E に n 個のイベントがある場合, ナイーブな計算量は $\sum_{i=1}^n n C_i$ となるからである。この理由は全ての状態の組合せを数え上げる必要があるからである。例えば図 8 の場合には A, B, C の 3 イベントがある。このとき, 下記の 7 状態 ($\sum_{i=1}^3 3 C_i$) が成り立つか調べる必要がある。

$\{A \wedge \neg B \wedge \neg C\}$, $\{\neg A \wedge B \wedge \neg C\}$, $\{\neg A \wedge \neg B \wedge C\}$, $\{A \wedge B \wedge \neg C\}$, $\{A \wedge \neg B \wedge C\}$, $\{\neg A \wedge B \wedge C\}$, $\{A \wedge B \wedge C\}$ 。

Algorithm 1 Evaluate BN

```

1: INPUT:  $BN_x$  (a Bayesian network),  $E$  (a set of events)
2: OUTPUT: Probabilistic tuples;
3:
4: for all  $e_i \in E$  do
5:   set a lifespan  $\in BN_x$  expressed by  $e_i$ ;
6: end for
7:  $DS \leftarrow getDifferentStates(BN_x)$ ;
8: for all state  $ds_i \in DS$  do
9:   propagate( $ds_i$ ,  $BN_x$ );
10:  generate probabilistic tuples from  $BN_x$ 
11:  output the tuples to the output queue;
12: end for

```

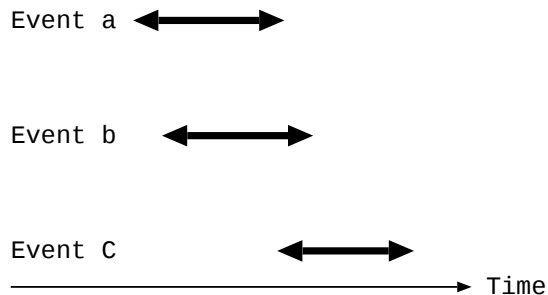


図 8 異なる状態

BN 演算子は関係演算子に比べて高いコストを有すると下記理由から考えられる。(1) 結合演算子のように, BN 演算子は入力タプルよりも出力タプル数の方が多くなる可能性がある。(2) 状態数に比例して確率伝播 (propagate) 実行回数が増加し, 計算コストを高める可能性がある。

4. 実環境への適用

提案システムを評価する為に, 実データを用いてベイジアンネットワークを構成した。確率推論を実行するベイジアンネットワークは図 13 を用いた。このベイジアンネットワークはカメラにより撮影された図 9 や図 10 等の比較画像と現在撮影した画像との差分画像の絶対値ノルムをノードのイベントとして扱う。上記の差分値を用いることにより現在の部屋の状況を推論する。部屋の状況は, 人が机で作業しているイベントである “Working”, 人が動いているイベントである “Moving”, そして何もイベントが生起していない “None” の 3 つである。

各ノードの説明は以下の通りである。

- Event : 現在部屋で起きている状況 (イベント数 3: Working, Moving, None)
- Background : 背景画像 (図 9) と現在の画像との差分値 (イベント数 10)
- Before : 一つ前に撮影した画像と背景画像との差分値 (イベント数 8)
- Working1 : Working 時の画像 1 (図 10) と現在の画像との差分値 (イベント数 12)
- Working2 : Working 時の画像 2 (図 11) と現在の画像との差分値 (イベント数 12)
- Working3 : Working 時の画像 3 (図 12) と現在の画像との差分値 (イベント数 10)

教師データとして 1400 枚の画像データを用いてベイジアンネットワークを構築した。

5. 関連研究

本システムと同じように, データストリームを連続的に処理するシステムは多数存在する。

代表的なものでは, STREAM [4], Aurora [5], TelegraphCQ [6], Borealis [7], WaveScope [8], そして StreamSpinner [3] 等がある。これらのシステムではデータストリームに対する連続的の問合せ処理を行うが, 不確実な状況を表す確率的データスト

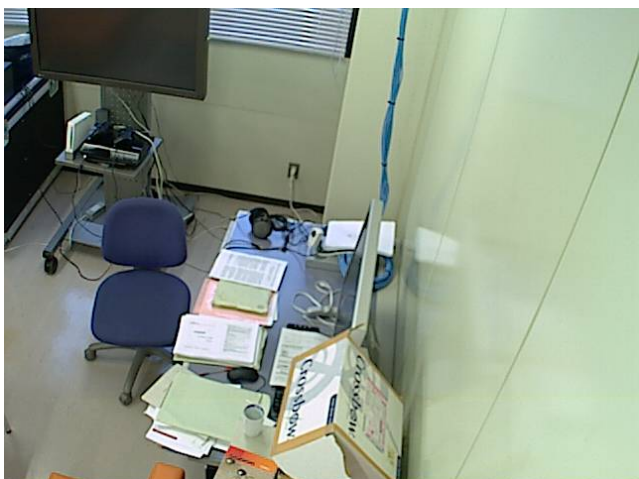


図9 背景画像

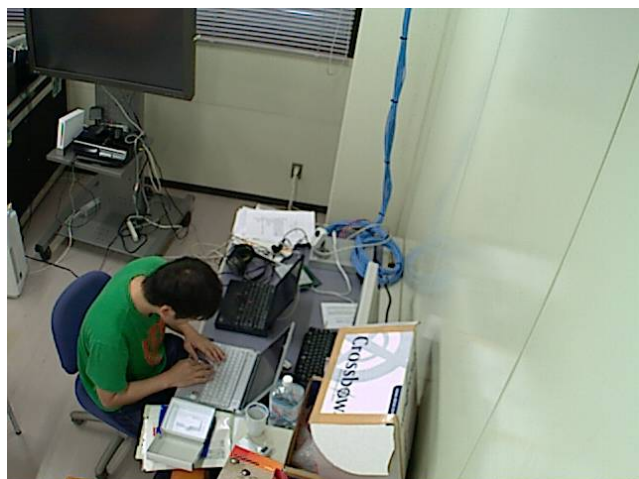


図11 Working 時の画像 2

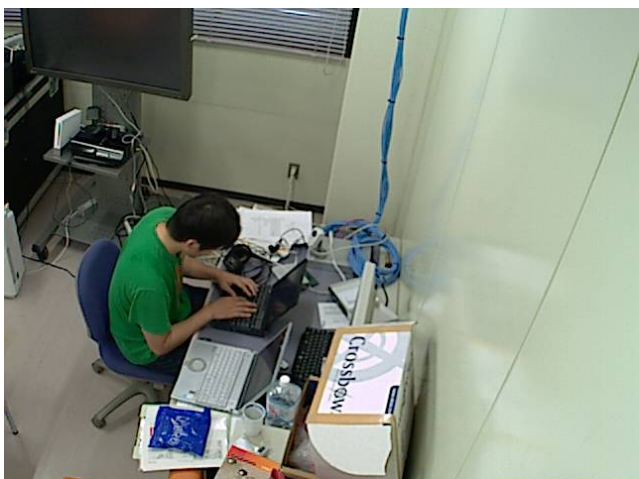


図10 Working 時の画像 1



図12 Working 時の画像 3

リームについては考慮していない．ストリーム処理と関係データ処理の違いに、演算子が Stateless か Stateful かという点がある．これまでの研究では結合演算子と集約演算子[19][20]の Stateful 化が研究されてきた．我々が導入した lifespan により BN 演算子は Stateful となるが、このような演算子を研究している例は、我々の知る限り他に存在しない．

確率的データベースを用いて不確実性を扱う研究には、Stanford 大学 Trio プロジェクトの ULDB[15]が存在する．ULDB は来歴を用いて確率計算を実行する．しかし、本研究のようにベジアンネットワークなどの手法を用いてデータベース内で推論し、確率値を含むタプルを新しく生成することについては考慮されていない．さらに ULDB は関係データ処理を対象とし、ストリームは扱わない．

ベジアンネットワークを使用し、データストリームを扱う研究には、D-JENGA[1]がある．D-JENGA はセンサデバイスから得られる信号データを対象に確率計算をするが、本研究はセンサデバイスから得られるデータを解析した後に生成される、イベントストリームを対象に確率計算を実行する．さら

に、本研究は確率推論のみならず関係演算子全般を対象とするが、D-JENGA はベジアンネットワークしか計算対象としていない．

ベジアンネットワークを用いた、状況推定を目的とした研究に[17]がある．この研究は、ベジアンネットワークを用いて確率計算を実行し、状況推定を行っているが、データストリームを対象にしておらず、他のデータとの統合利用についても言及していない．

6. 結論と今後の課題

実世界状況を連続的に監視するシステムを支援する基盤技術の開発を目指し、本研究では研究課題として、確率的表現と連続的問合せモデルを有する技術の開発を取り上げた．そこで本研究ではベジアンネットワークを用いたデータストリーム処理モデル及び、データベースとの統合を提案した．提案モデルでは、関係演算処理をするためにグラフ構造であるベジアンネットワークを DB に格納した．さらに、データストリームとベジアンネットワークの各イベントを対応付けるための

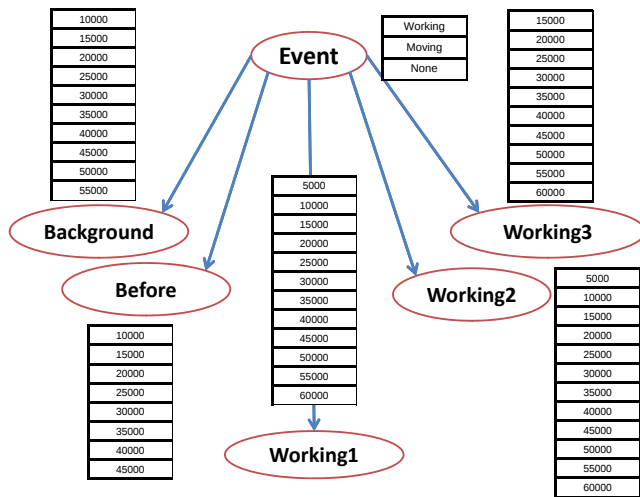


図 13 構成したベイジアンネットワーク

ASSOC 処理を提案し、イベント生起のデータによりベイジアンネットワークの確率計算を実行し、確率的関係データを生成する BN 演算子を提案した。さらに、確率的データとデータベースのデータとの統合を可能にした。

以上より、本研究ではベイジアンネットワークと関係データベースとの連携を可能にするモデルを提案したと結論する。

今後の課題は以下の通りである。(1) 提案システムを実装する。すでに選択、射影、結合、集約、和、窓の演算子を処理可能な μ SPE を準備済である。同 μ SPE への BNM 演算子導入を速やかに行う。(2) 実験で用いたベイジアンネットワークを使用してカメラデータによる実環境での状況推論の実験を行う。また、本研究ではカメラを 1 台のみ用いてベイジアンネットワークを構成したが、我々は既に 30 台程度のネットワークカメラを室内に敷設している。それゆえ複数台のカメラから得られるデータを元にして大規模なベイジアンネットワークを構成し、研究室環境における状況推定アプリケーションを稼働させる。(3) 3.6 節で述べたように、ナイーブな技法では異なる状態を取得する計算コストが高いため、効率的な技法を開発する。

謝辞 本研究の一部は科学研究費補助金基盤研究 (#18200005)、筑波大学 VBL 研究プロジェクトによる。

文 献

- [1] Masaya Kadota, Hiroto Aida, Jin Nakazawa, and Hideyuki Tokuda. D-jenga: A parallel distributed bayesian inference mechanism on wireless sensor nodes. Proc. of the International Conference on Networked Sensing Systems, 2006.
- [2] Richard E. Neapolitan, Learning Bayesian Networks, Prentice Hall, 2004
- [3] Shinichi Yamada, Yousuke Watanabe, Hiroyuki Kitagawa, and Toshiyuki Amagasa. Location-based information delivery using stream processing engine streamspinner. Proc. of the International Conference on Mobile Data Management p. 57, 2006.
- [4] The STREAM Group, STREAM: The Stanford Stream Data Manager, IEEE Data Engineering Bulletin, 2003.
- [5] Daniel J. Abadi and Donald Carney and Ugur Cetintemel and Mitch Cherniack and Christian Convey and Sangdon Lee and Michael Stonebraker and Nesime Tatbul and Stanley B. Zdonik, Aurora: a New Model and Architecture for Data Stream Management, VLDB

- Journal, 2007,
- [6] Sirish Chandrasekaran and Owen Cooper and Amol Deshpande and Michael J. Franklin and Joseph M. Hellestein and Wei Hong and Suresh Krishnamurthy and Sam Madden and Vijayshankar Raman and Fred Reiss and Mehul Shah, TelegraphCQ: Continuous Dataflow Processing for an Uncertain World, Proc. of the Conference on Innovative Data Systems Research, 2003.
- [7] D. Abadi and Y. Ahmad and M. Balazinska and U. Cetintemel and M. Cherniack and J.-H. Hwang and W. Lindner and A. Maskey and A. Rasin and E. Ryvkina and N. Tatbul and Y. Xing and Stan Zdonik, The Design of the Borealis Stream Processing Engine, Proc. of the Conference on Innovative Data Systems Research, 2005.
- [8] Lewis Girod and Yuan Mei and Ryan Newton and Stanislav Rost and Arvind Thiagarajan and Hari Balakrishnan and Samuel Madden, The Case for a Signal-Oriented Data Stream Management System, Proc. of the Conference on Innovative Data Systems Research p. 397-406, 2007.
- [9] J. Pearl, Probabilistic Reasoning in Intelligent Systems; Network of Plausible Inference. Morgan Kaufmann, 1988.
- [10] G Piriou, et al, Recognition of Dynamic Video Contents with Global Probabilistic Models of Visual Motion, IEEE Transactions on Image Processing, 2006.
- [11] Yanlei Diao, Neil Immerman, and Daniel Gyllstrom. SASE+: An Agile Language for Kleene Closure over Event Streams. UMass Technical Report 07-03.
- [12] E. Wu, Y. Diao, and S. Rizvi. High-performance complex event processing over streams. In SIGMOD, 407-418, 2006.
- [13] Nodira Khousainova, Magdalena Balazinska, Dan Suciu. Probabilistic RFID Data Management. UW CSE Technical Report UW-CSE-07-03-01. March 1, 2007.
- [14] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and Issues in Data Stream Systems. In ACM Symposium on Principles of Database Systems, 2002.
- [15] O. Benjelloun, et al. Uldbs: Databases with uncertainty and lineage. Proc. VLDB, 2006.
- [16] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In Proc. of VLDB, 2004.
- [17] 石川詔三, 本村陽一, 河田諭志, 西田佳史, 原一之. 日常生活行動における確率的因果構造モデルの構築と行動推論. The 22nd Annual Conference of the Japanese Society for Artificial Intelligence, 2008.
- [18] A. Arasu, S. Babu, and J. Widom. The cql continuous query language: semantic foundations and query execution. The VLDB Journal, 15(2): 121-141, 2006.
- [19] Mayur Datar, Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Maintaining Stream Statistics over Sliding Windows. SIAM Journal on Computing, Volume 31, Issue 6, Pages 1794 - 1813, 2002.
- [20] Sirish Chandrasekaran and Michael J. Franklin. Streaming Queries over Streaming Data.