

確率推論 VLSI プロセッサの構成と高安全知能自動車への応用

佐藤伸也 亀山充隆 (東北大学)

Design of a Probabilistic Inference VLSI Processor and Its Application to Highly-Safe Intelligent Vehicles

*Shinya Sato, Michitaka Kameyama (Tohoku University)

Abstract— This article presents a VLSI-oriented probabilistic inference algorithm for real-world applications. We use a Bayesian network for probabilistic inference, and a sampling method for VLSI oriented inference algorithm. To solve a data transfer bottleneck between a memory and a processing element, logic-in-memory architecture is proposed.

Key Words: probabilistic inference, Bayesian network, sampling method, logic-in-memory architecture

1 まえがき

近年, リアルワールドにおいて予測に基づき自律的動作を行う知能システムの実現が望まれている [1]. その一例として, 自律的に環境を認識し, 障害物に衝突することなく目的地まで走行する高安全知能自動車システムが検討されている (Fig.1). 一般に, 機械的運動の動作速度は環境の変化速度と比較してかなり遅いため, 機械的な先行動作をするために, 自車の軌道計画の前処理として, 他車や人などの動的障害物の軌道を予測する処理が必須となる. このような予測を行うため, 対象とする問題を確率モデルとして表現し, 観測された情報から起こりうる各状態の確率を求める方法が提案されている [2][3].

本稿では, 確率モデルとしてベイジアンネットワークに注目し, リアルワールドにおける予測処理に向けたアルゴリズムとしてサンプリング法を採用して, その評価および専用 VLSI プロセッサ化のための考察をしている [4]. また, 高安全知能自動車への応用として, 対向車の将来の位置を予測するネットワークを構築し, シミュレーションを行った. しかし, これらの確率計算には膨大な計算が必要となるため, 高速な応答が求められるリアルワールドでの処理においては, 専用 VLSI あるいはシステム LSI としての実現が望まれる. このような観点から, メモリ転送ボトルネックを解消した高性能確率推論 VLSI プロセッサの提案を行っている.

2 ベイジアンネットワークを用いた確率推論

2.1 ベイジアンネットワーク

リアルワールドには様々な不確実性が存在するため, それをいかに扱うかが非常に重要になる. そのために確率という概念を導入し, 様々な情報を確率変数で表現して, それらの因果関係に注目したグラフィカルなモデリング手法がベイジアンネットワークである (Fig.2). ベイジアンネットワークでは, 確率変数をノード, 変数間の定性的な依存関係をグラフ構造, 変数間の定量的な

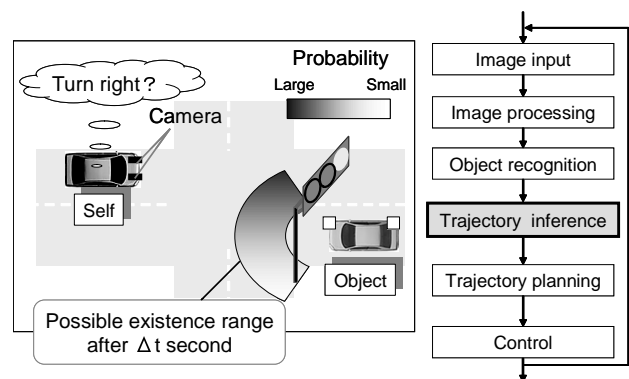


Fig. 1: System of highly-safe intelligent vehicles

依存関係を親ノードに依存した条件付き確率によって表す. 確率変数 X_i の親ノードの集合を $\pi(X_i)$ とすると, X_i は $P(X_i|\pi(X_i))$ で表される条件付確率表 (CPT) を持つ.

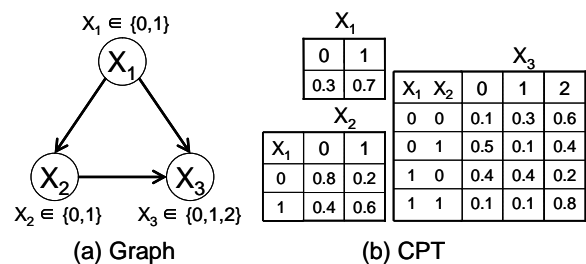


Fig. 2: Example of a Bayesian network

ベイジアンネットワークでは, 一部の変数の値が観測できたときに, その情報を入力することで他の未観測の変数の事後確率分布を計算することができる. すなわち, 観測されたノードを証拠ノード E , 求めたい未観測ノードを X とすると, $P(X|E)$ を求めることが目的となる.

2.2 サンプリング法を用いた確率計算

全ての確率変数の結合確率を全て計算すれば、その中から与えられた証拠値に合致するものだけを抜き出して正規化することで、求めたい変数の確率分布を厳密に計算することが可能である。ノード数が n 個のとき、全ての変数の結合確率は以下のように与えられる。

$$P(X_1, \dots, X_n) = P(X_1|\pi(X_1)) \cdots P(X_n|\pi(X_n)) \quad (1)$$

しかし、一般にノード数、状態数が増えると結合確率の数は爆発的に増加するために、全ての結合確率を計算するのは困難である。そこで、結合確率の小さいものは計算を省略することにより計算量を削減した近似計算がサンプリング法である。このアルゴリズムの特徴は、サンプル数を変えることで計算時間と解の精度のトレードオフが調整できることである。リアルワールド応用では高速応答性が要求されるため、予測時間にある一定以上時間がかかってしまうことは許されない。

そのため、予測精度をある程度犠牲にしても、おおまかな確率分布を一定時間内に知ることができる必要が求められる。ベイジアンネットワークの確率計算アルゴリズムはこれまで多数提案されているが、この点においてサンプリング法はリアルワールドでの確率計算アルゴリズムとして適している。例として、サンプル数を変化させたときの解の精度、及び計算時間を Fig.3 に示したベイジアンネットワークにおいて求めた。計算精度は、全ての未観測のノードの各具現値において厳密解との誤差を求め、その中の最大誤差として評価した。サンプル数と最大誤差、及びサンプル数と計算時間の関係を Fig.4 に示す。

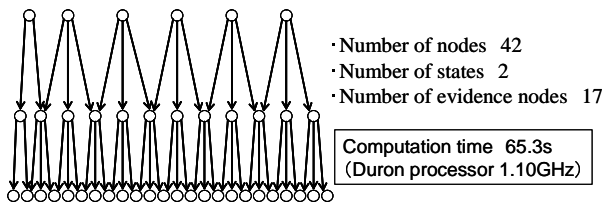


Fig. 3: Simulation of a Bayesian network

一般に、同じノード数でもグラフ構造が変化すれば、設定される CPT が変化するために、同一のサンプル数でも計算精度、計算時間は変わってくる。また同じグラフ構造でも証拠ノードが変化すれば同一のことが起こる。しかし、サンプル数を増やせば理論的に厳密解に収束することは証明されている。

実際にサンプリング法で計算するときの問題設定は、(1) 計算時間の制限が設定されたときに、その制限時間内で解の精度を最大化する、(2) 解の精度の制限が設定されたときに、その制限精度を満たした条件で計算時間を最小化する、といった2通りがあると考えられる。

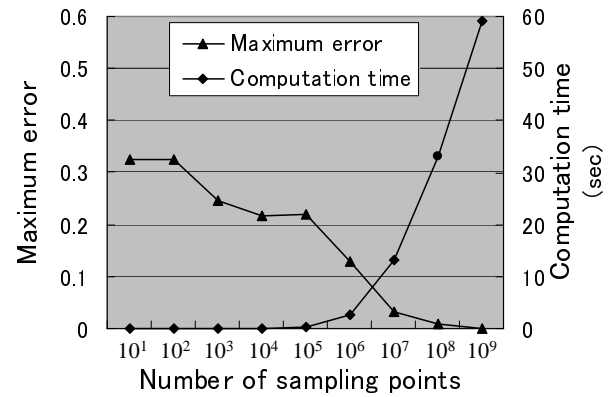


Fig. 4: Maximum error and computation time

これら2つの場合のどちらにおいても、「あるサンプル数が設定されたときに、そのサンプリング処理を高速に実行する」ことにより目的が実現できる。しかしながら、汎用プロセッサの計算速度では、リアルワールドにおける大規模な推論をするにあたっては、高速性・高精度性を両方とも兼ね備えた予測は困難であると Fig.4 の例からも推測できる。そこで、サンプリング処理を高速に実行するために、そのサンプリング処理に特化した専用プロセッサの実現が望まれる。

実際の計算手順は、まず式 (1) の右辺におけるそれぞれのノードの CPT を観測ノードと未観測ノードの2つに分けて、(i) 未観測ノードのみの CPT の積を結合確率としてサンプリングを行い、(ii) 観測ノードの CPT の積を得られたサンプルの結合確率にかけることにより式 (1) を計算し (Scoring)、(iii) この値を計算したい未観測ノードの確率として累積加算する。(i) ~ (iii) の処理を全てのサンプルが得られるまで繰り返し、最後に正規化することにより確率分布が計算できる。これらのフローチャートを Fig.5 に示す。

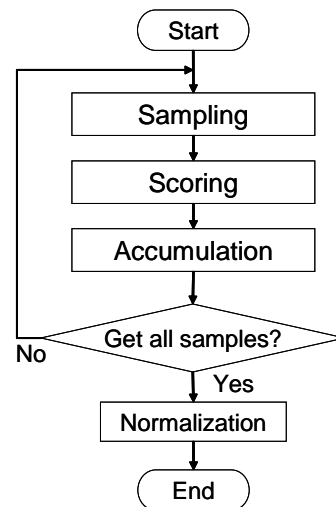


Fig. 5: Flow chart of a sampling method

3 ロジックインメモリ構造 VLSI プロセッサ

3.1 並列処理を指向した処理フロー

専用プロセッサ化を考える上では、様々な処理の依存関係を考慮し、並列性を活用することが有用である。ここでは Fig.5 に示したフローチャートにおいて、繰り返し処理となるサンプリング、Scoring、累積加算のそれぞれの処理間の依存関係に注目し、並列処理を指向した処理フローについて考察する。

まず、Scoring は観測ノードの CPT の積を求め、さらにサンプリングで得られた未観測ノードの CPT の積に掛け合わせる処理であるため、サンプリングの結果に依存する。ただし、観測ノードがネットワークのどの位置に入るかによって依存度が異なるといった性質があり、その依存度によってはサンプリング処理との並列計算は可能である。観測ノードがネットワークの最下層（子を持たないノード）に集中する場合は最も依存度が高くなるため、その場合は並列計算は全くできない。ここではその最悪の場合を想定してサンプリングと Scoring の並列計算は考えない。累積加算については、Scoring の結果を累積加算する処理であるため、Scoring との並列計算は不可能である。よって同一サイクルにおけるサンプリング、Scoring、累積加算の並列計算は困難であると考えられる。しかし、ここで連続したサイクル間の依存関係も考慮すると、サンプリングは 1 つ前のサイクルで計算されたサンプリングの処理結果には依存するが、1 つ前のサイクルの Scoring 及び累積加算には依存しないため、Fig.6 のような処理フローが可能になる。これにより全体の処理時間を短縮できる。

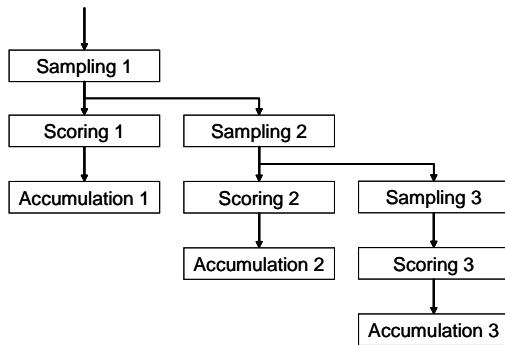


Fig. 6: Sampling method

3.2 サンプル探索空間の分割による並列処理

Fig.6 に示したように、サンプリングは 1 つ前のサイクルでのサンプリング処理の結果に依存するため、直列処理しかできないといった問題がある。サンプリングの処理では、未観測ノードの CPT の結合確率空間において、サンプル数に応じたしきい値を設定して、そのしきい値に対応したサンプルを探索する処理を行うことになる。そこで、その確率空間を適切に分割することでサ

ンプリング処理の並列計算が実現できる。よって、Fig.6 に示す処理フローを計算する PE(Processing Element) を複数使用し、並列実行可能なアーキテクチャが可能になる。しかし、複数の PE の並列処理には複数のメモリからの並列アクセスが必要であり、相互結合回路網上での転送ボトルネックが問題となる。そこで、各 PE で使用される中間結果はその PE 内でのみ利用されるという中間結果の局所性といった特徴と、CPT の適切なメモリアロケーションにより、メモリ-PE 間の相互結合網を完全局所化できるロジックインメモリ構造を提案する。Fig.7 に示すように、それぞれの PE に割り与えられた分割の中で、その時点ですでに決定している値によって CPT を分割することにより、冗長性の少ないメモリアロケーションが可能になる。

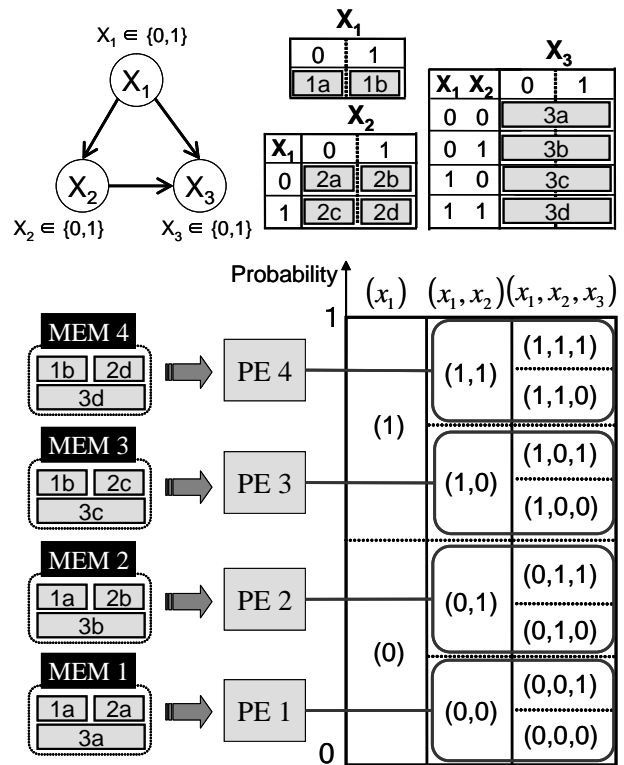


Fig. 7: Logic-in-memory architecture

4 高安全知能自動車への応用

ベイジアンネットワークを用いた確率推論を、高安全知能自動車システムで必要不可欠な処理である、動的障害物の軌道予測に応用することを考える。一例として、Fig.8(a) に示す環境においてシミュレーションを行った。同図は信号機のある十字路において、前方から対向車が接近してくる際に、自車が交差点内で右折しようとしている状況を表している。自車がすぐに右折すべきか対向車が通過するまで待つべきかの判断のために、対向車の t 秒後の位置の予測を行う必要がある。今回使用したノードは以下の通りである。

観測ノード

- 対向車の情報（速度，位置，向き，ウィンカー）
- 交差点までの距離
- 信号の色

未観測ノード

- t 秒後の対向車の情報（速度，位置，向き）
- 人間のモデル（対向車の意思・制御）

これらのノードを用いて Fig.8(b) に示したネットワークを構築し，観測ノードの値を変化させたときに望まれる予測ができていないか確認した。

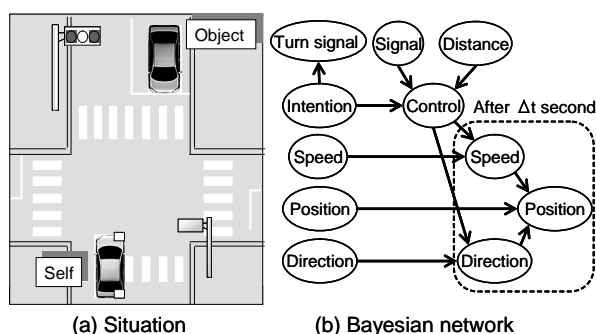


Fig. 8: Prediction example

一例として，対向車の速度のみを変更して t 秒後の対向車の位置の予測を行った結果を Fig.9 に示す．確率は扇状に表されており，色が濃い所ほど確率が高いことを示している．Fig.9(a) は，停止線付近の確率が高く，Fig.9(b) は，交差点内の確率が高いことが分かる．このことは信号が黄色の状況で対向車の速度が遅い場合は停止線で停止する確率が高く，速度が速い場合は交差点に進入する確率が高いという予測を表している．これらの結果は実際に運転する際の状況を表現しており，適切な推論がなされていると言える．さらに，道路の形状，標識や表示，他の自動車等に関する情報を増やし，繊細にモデル化することで，ベイジアンネットワークの予測精度の向上が期待できる．

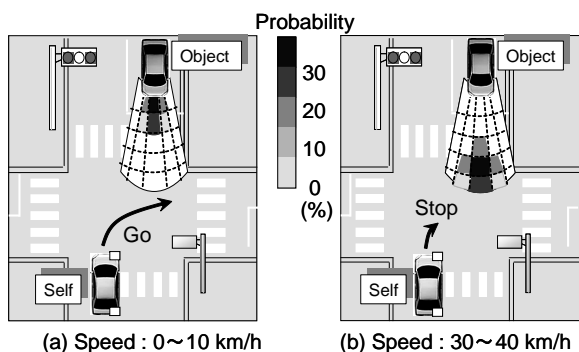


Fig. 9: Simulation result

5 むすび

リアルワールドにおける確率推論アルゴリズムとしてサンプリング法が適していることに着目し，その評価を行った．また，適切なメモリアロケーションを行うことで，並列処理における転送ボトルネックを解消できるロジックインメモリ構造に基づく確率推論 VLSI アーキテクチャを提案した．各 PE の処理時間は結合確率の値に依存するため，最初の段階で最適なメモリアロケーションを静的に決定することが困難となる．そこで，今後はリアルタイムで各 PE の計算終了信号を発生させ，メインメモリからアクセスすべきデータを各 PE に動的に再配分し計算負荷の均等化を図る階層的メモリ構造のアーキテクチャを検討中である．

参考文献

- [1] 亀山，張山，“リアルワールド応用知能集積システムの展望”，計測と制御，Vol.40，No.12，pp.841-847(2001).
- [2] Castillo,Gutiérrez and Hadi,“ Expert Systems And Probabilistic Network Models ”, Springer(1997).
- [3] 本村陽一，“ベイジアンネットワークによる確率推論技術”，計測と制御，Vol.42，No.8，pp.649-654，(2003).
- [4] 佐藤，亀山，“ベイジアンネットワーク用ロジックインメモリ構造 VLSI プロセッサの構成”，電子情報通信学会エレクトロニクスソサイエティ大会，C-12-11(2004).