

An Elastic and Scalable FPGA-based Accelerator for Bayesian Network Structure Learning (ベイジアンネットワーク構造学習のための 可塑性と拡張性を備えたFPGAアクセラレータ)

宮城竜大
高木研

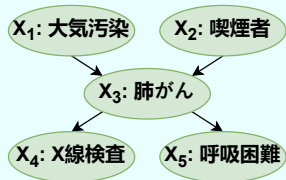
- ▶ 計算機・通信技術の高性能化・低コスト化
- ▶ インターネットを介して膨大な量のデータが蓄積される (ビッグデータ)
- ▶ 機械学習を用いて既知のデータを学習し，未知のデータを予測したい！

ベイジアンネットワーク

- ▶ 確率変数の条件付独立関係を DAG に符号化したモデル
- ▶ DAG の探索空間が膨大になるため，データから構造を学習することが難しい

id	大気汚染	喫煙	肺がん	X線検査	呼吸困難
1	High	True	False	True	False
2	High	False	False	True	True
3	Low	False	True	False	False
⋮	⋮	⋮	⋮	⋮	⋮
N	Low	False	True	False	True

構造学習



推論



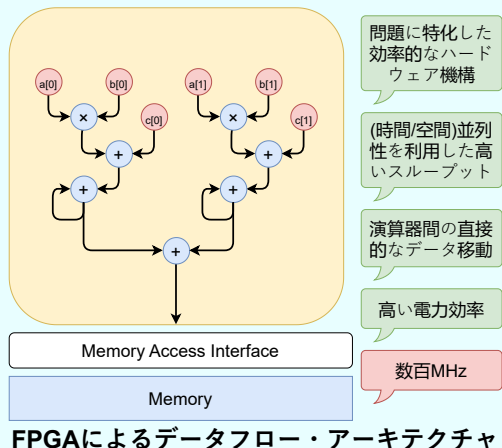
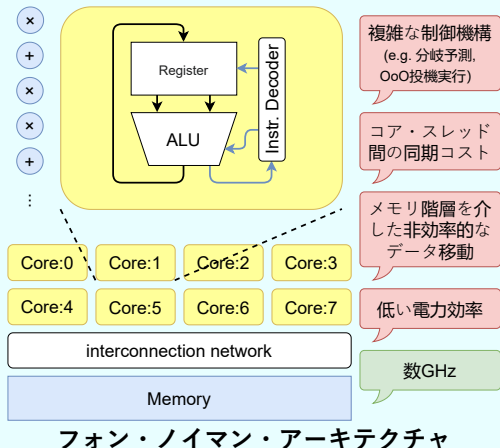
$$p(X2=T \mid X3=T, X4 = T)$$

$$p(X1=F \mid X3=T, X4 = F)$$

- ▶ 本研究の提案: 構造学習を FPGA を用いたデータフロー計算によって高速化する

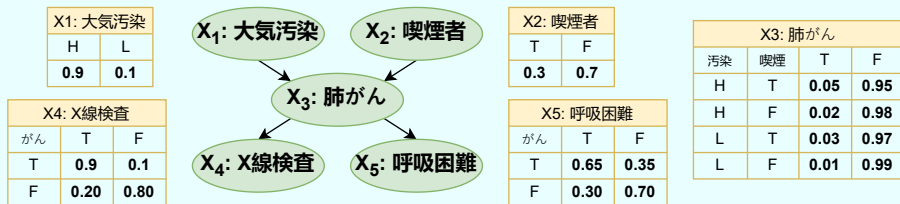
準備: FPGA

- ▶ 任意の論理回路を繰り返し書き換えて実行できるデバイス
- ▶ 問題に特化したデータフロー・アーキテクチャによって…



準備: ベイジアンネットワーク

- ▶ 離散確率変数間の **条件付き独立関係** を DAG に符号化した確率モデル
- ▶ 確率変数 $V = \{X_1, X_2, \dots, X_n\}$ 上のベイジアンネットワークは以下で定義.
 - 各変数 X に対する親変数の集合 Pa_X^G (つまり, DAG G を規定)
 - 条件付き確率パラメータ集合 $\Theta = \{p(X \mid \text{Pa}_X^G)\}$ (条件付き確率テーブル)



- ▶ 確率変数の推論に利用 (医療診断など)
 - 例えば, 喫煙習慣のある人 ($X_2 = T$) が, 肺がんであるのかかわらず ($X_3 = T$), 胸部 X 線検査で見逃される ($X_4 = F$) の確率: $p(X_4 = F \mid X_2 = T, X_3 = T)$

準備: ベイジアンネットワークの嬉しさ

- ▶ ベイジアンネットワークは符号化された条件付き独立性を利用して、同時確率分布をコンパクトな条件付き確率の積に因数分解する。

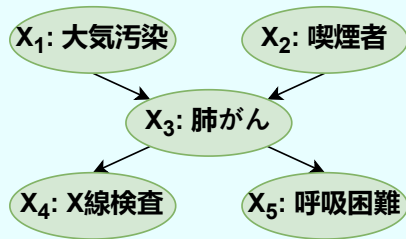
$$p(X_1, X_2, \dots, X_n) = \prod_{X \in \mathbf{V}} p(X \mid \mathbf{Pa}_X^G). \quad (1)$$

- ▶ (左辺) 離散確率変数の同時確率分布を直接推定すると…
 - △ 組み合わせ爆発によって指数的な数のパラメータが必要
 - △ 各パラメータの妥当な推定値を得るために必要なサンプルのも膨大
 - ▶ (右辺) 因数分解、各条件付き確率は非常に小さい変数集合の関係のみに注目
 - 少ないパラメータで記述可能
 - 比較的少数・疎なサンプル集合からも頑健な推定が可能
- ベイジアンネットワークは離散確率変数の同時確率分布の圧縮表現
- ▶ ただし、このような DAG/因数分解の候補の数は膨大…

準備: ベイジアンネットワークの構造学習

- ▶ N 個のサンプルをもとに適切な DAG 構造 \mathcal{G} を推定

id	大気汚染	喫煙	肺がん	X線検査	呼吸困難
1	High	True	False	True	False
2	High	False	False	True	True
3	Low	False	True	False	False
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
N	Low	False	True	False	True



- ▶ **score-based:** サンプル集合 S からスコアを最大化する DAG \mathcal{G} を探索 ← target
 - 少ないデータからも優良な構造が得られる
 - △ NP 困難, 組合せ爆発による膨大な時間・空間計算量が問題
- ▶ **constraint-based:** 条件付き独立性検定 (CI test) による
 - サンプル数の増加に伴って漸近的に真の条件付き独立構造を推定できる
 - △ 必要なサンプル数が膨大となる可能性が実験によって示唆

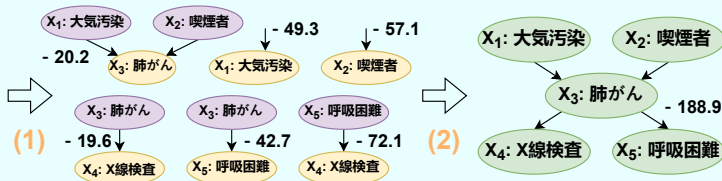
準備: ベイジアンネットワークの score-based 構造学習

- ▶ DAG 全体のスコアは局所スコア (Local Score, LS) に分解できる.

$$p(\mathbf{S} \mid \mathcal{G}) = \prod_{X \in \mathbf{V}} \text{LS}(X, \text{Pa}_X^{\mathcal{G}}), \quad (2)$$

- ▶ あらかじめ, 局所スコアによって局所構造の候補を評価・メモ化しておくとか…?
 - 異なる大域構造の探索において再利用できる
 - ◎ 有望でない候補を除いておくことで, 大域構造の探索空間を小さくできる
- ▶ 結果として, 構造学習は以下のように2ステップに分割できる.
 - (1) 親集合同定: 膨大な局所構造 (X, Pa_X) を評価し, そのうち有望なものを抽出
 - (2) 大域構造学習: 有望な局所構造候補を組み合わせて, 大域構造を学習する

id	大気汚染	喫煙	肺がん	X線検査	呼吸困難
1	High	True	False	True	False
2	High	False	False	True	True
3	Low	False	True	False	False
⋮	⋮	⋮	⋮	⋮	⋮
N	Low	False	True	False	True



準備: 親集合同定

- 一般的に, すべての可能な局所構造を評価するのは難しいので, 小さな正定数 m を定めて $|\mathbf{Pa}_X| \leq m$ なる局所構造のみ評価する
- これらの局所スコアの計算には $|\mathbf{X}| \leq m + 1$ となるような全ての確率変数集合 $\mathbf{X} \subseteq \mathbf{V}$ についての頻度表の構築が必要である。

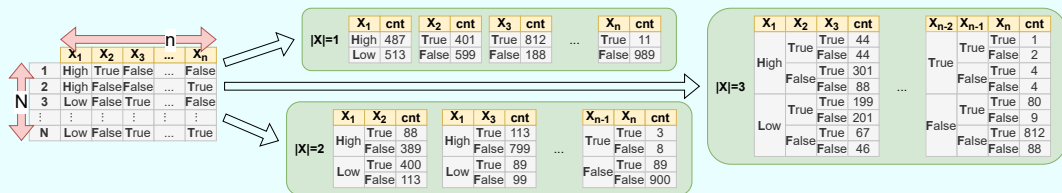


Figure: $m = 2$ の場合の例. この場合, $|\mathbf{X}| \leq 3$ の頻度表の構築が必要となる

→ このような頻度表の構築が構造学習全体のボトルネックとなっている

目的: 「 $|\mathbf{X}| \leq m + 1$ となる全ての確率変数集合 $\mathbf{X} \subseteq \mathbf{V}$ についての頻度表の構築」を高速化する

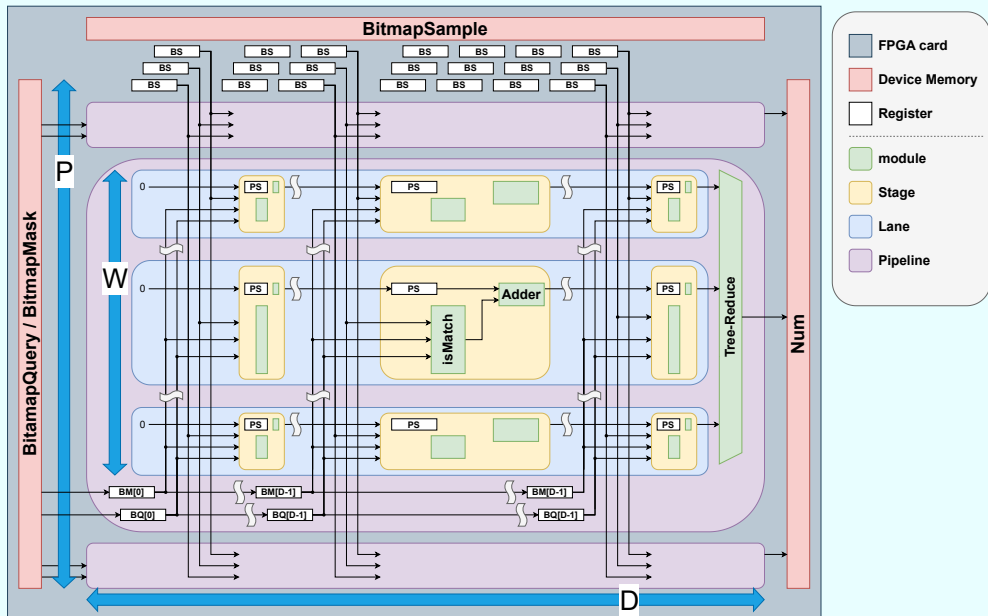
- ▶ 必要な頻度表およびセルの数は組み合わせ爆発的
- ▶ 十分な精度を得るためのサンプル数 N は比較的少数（数千から数万）

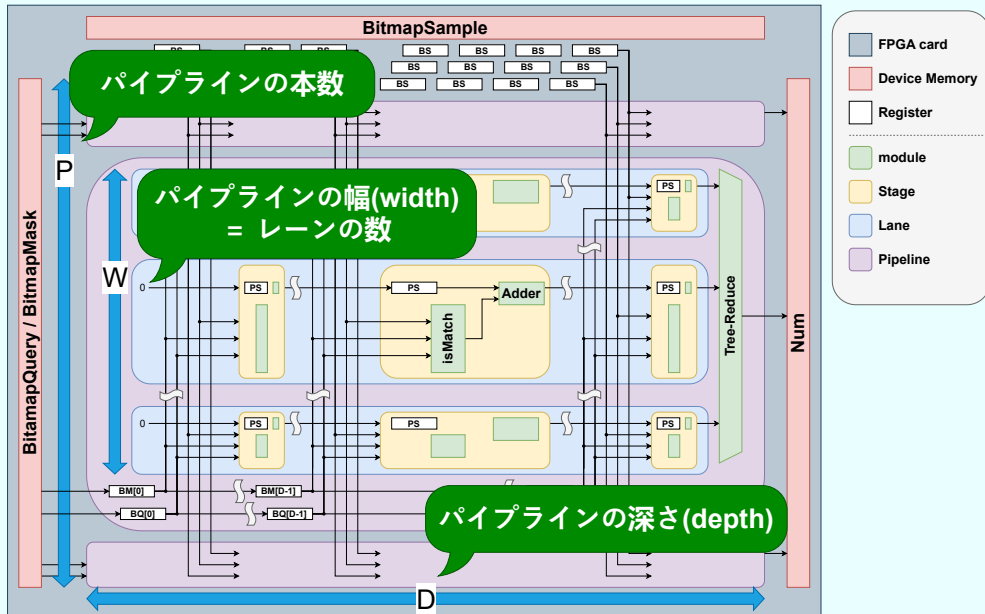
提案: データフロー計算に適した形に問題をシフト

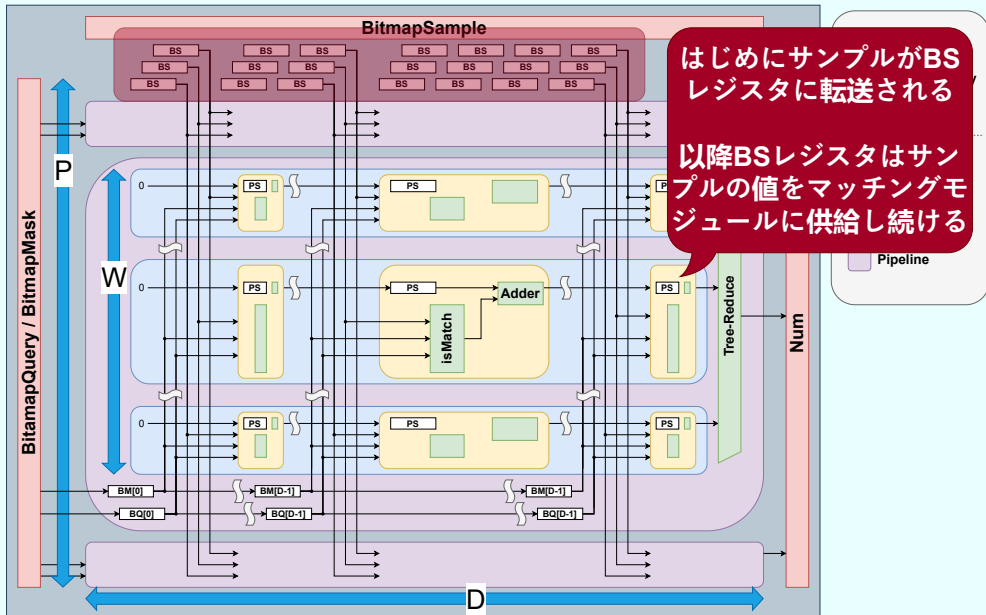
- ▶ 計算が必要な頻度表のセルごとにクエリを発行
- ▶ サンプルおよびサンプルとクエリのマッチングモジュールを展開
- ▶ マッチングモジュールの並列パイプラインにクエリのデータフローを形成

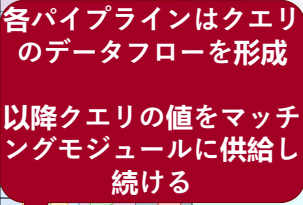
→ サンプルの数 N による複雑さを抑制

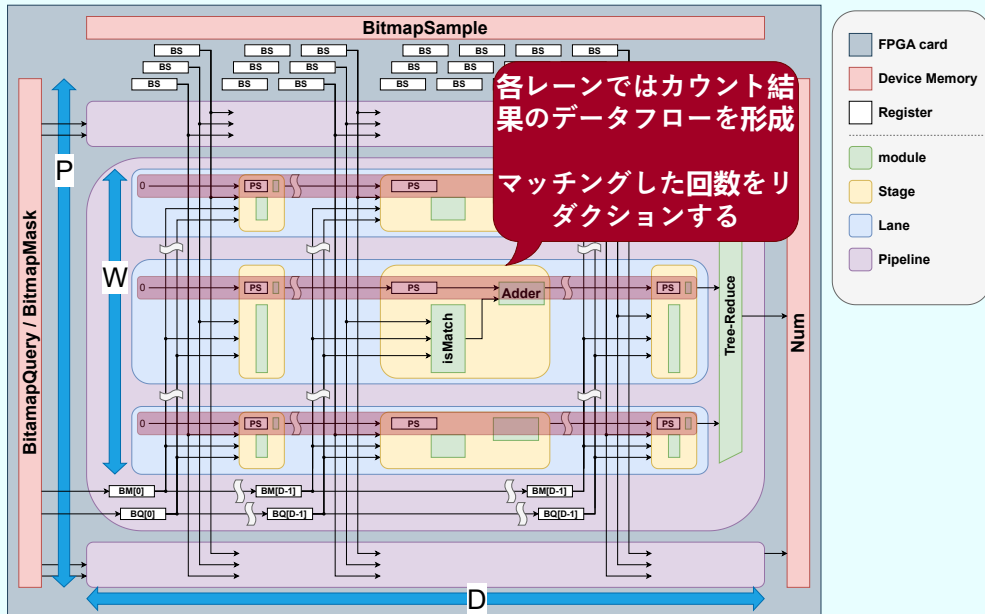
→ サンプルの数 N に比例する数のモジュールが用意できた場合、あるパターンを満たすサンプルの個数を定数クロックで数えることができる

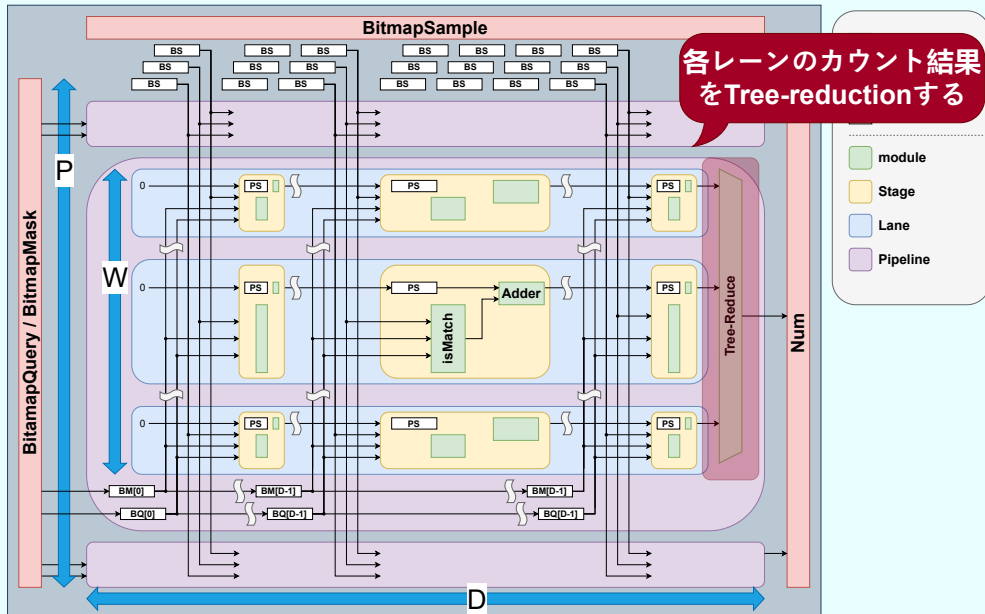












提案手法: Elastic Sample Filter

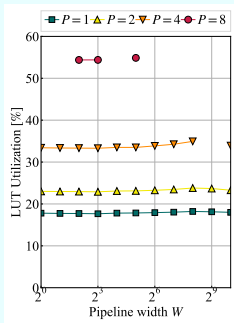
- ▶ 各クエリはパイプラインを通過する間に、 $W \times D$ 個のサンプルとマッチング
- ▶ さらに、 P 個のクエリが並列に処理される
- ▶ ただし、高い並列度はリソース使用量および配置配線の複雑さとトレードオフ
 - マッチングモジュールや加算器は LUT を消費する
 - D が大きくなるとパイプラインレジスタが FF を消費する
 - サンプルのレジスタは P のファンアウトを持つ
 - クエリのレジスタは W のファンアウトを持つ

これらが、動作周波数の低下を招く可能性がある

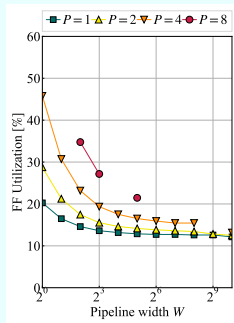
→ 並列度構成を可塑的に制御することで高い並列度と動作周波数を両立できる

評価: ESF の設計空間探索

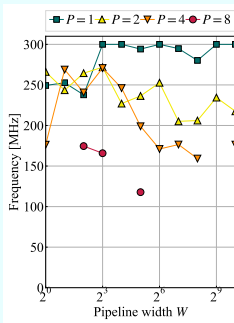
► Xilinx Alveo U50 / Vitis 2020.2 / 高位合成 (C++) / $W \times D = N = 1000$



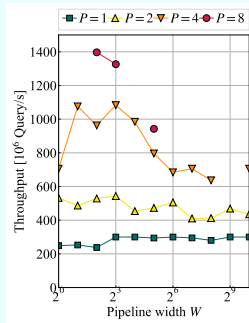
(a) LUT Utilization



(b) FF Utilization



(c) Frequency

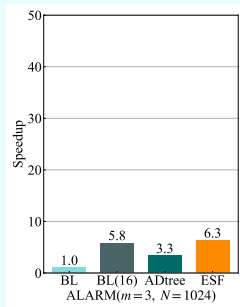


(d) Throughput

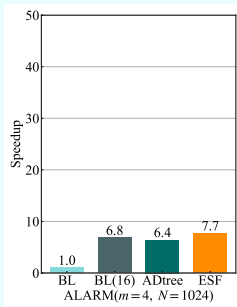
- D によって W をバランスして動作周波数の低下を抑制
- 最も高い性能を引き出す構成を同定

評価: ESF の性能比較

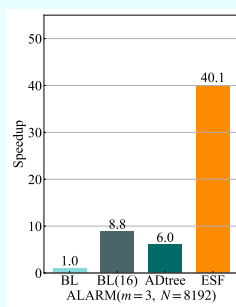
- ▶ 37 変数 / 親変数数の上限 m / サンプル数 N
- ▶ ベースライン手法, マルチスレッド実装, ADtree, ESF がすべての局所スコアを求める時間を比較.



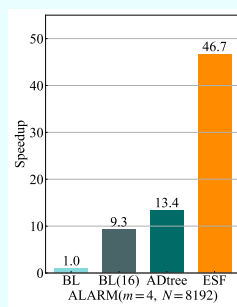
(e) $m = 3, N = 1024$



(f) $m = 4, N = 1024$



(g) $m = 3, N = 8192$

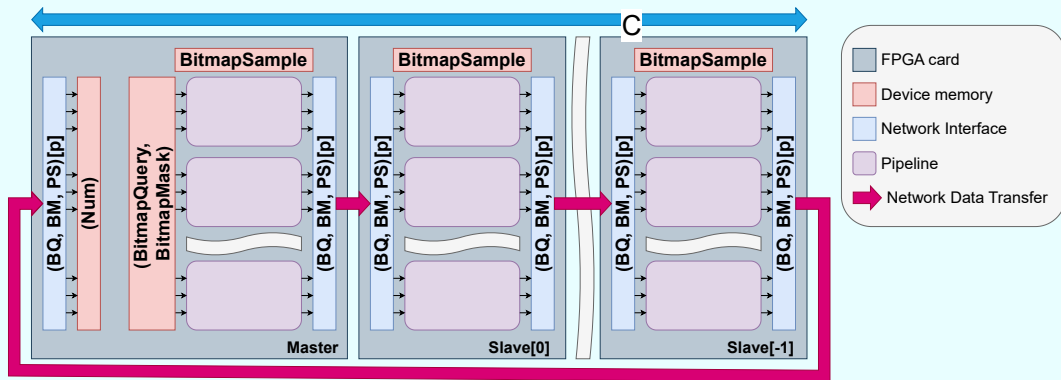


(h) $m = 4, N = 8192$

→ サンプル数 N による複雑さを抑制できていることを実証

提案手法: ESSF (Elastic Scalable Sample Filter)

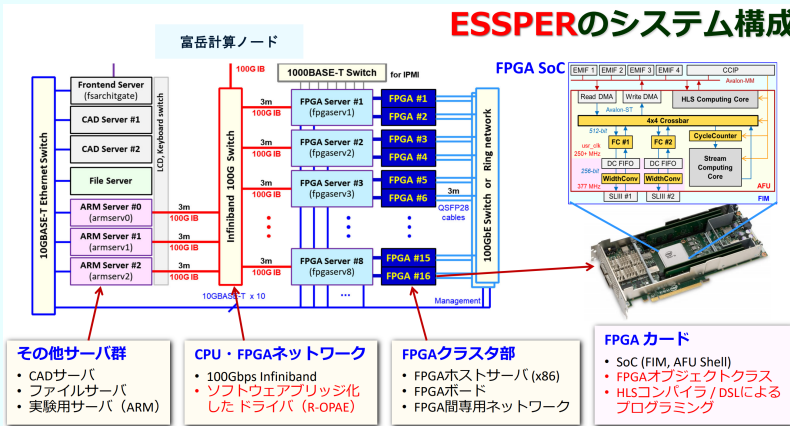
- ▶ 基本アイデア: ESF を直列に接続, データフローを一つに統合
 - ▶ C 枚の ESF を直列にリング型のトポロジーで接続
- 複数 FPGA カードを用いてより性能を向上できる (拡張性)



富岳 FPGA クラスタ ESSPER

- ▶ 高性能計算のための柔軟かつスケーラブルな FPGA クラスタの試作システム
- ▶ FPGA サーバ 8 台, それぞれに Intel FPGA PAC D5005 が 2 枚搭載
- ▶ D5005 は Intel Stratex 10 SX280 FPGA が搭載

ESSPERのシステム構成



評価: 富岳 FPGA クラスタ ESSPER での帯域幅のモデリング

- ▶ FPGA カード同士は専用ネットワークによって任意のトポロジで接続可能
- ▶ FPGA はネットワークとの接点にクレジットベースのフロー制御機構を搭載
- ▶ ネットワーク帯域幅 (256-bit@390MHz = 100Gbps) の 96%程度が利用可
- ▶ ESSF はデータの bit 幅 w , 動作周波数 f とすると, 必要な帯域幅は

$$(2 \times w + 32) \times P \times f [Mbps].(4) \quad (3)$$

- ▶ → 実用的に支障のない制約で, ほぼストールなく稼働できる.
 - $w = 64, f = 200$ だと, $P \leq 3$ が必要
 - $w = 64, f = 300$ だと, $P \leq 2$ が必要
 - $w = 64, f = 400$ だと, $P \leq 1$ が必要

ベイジアンネットワークのスコアベース構造学習におけるボトルネックとなる親集合同定を高速化

- ▶ データフロー計算に適した形に問題をシフトし，それを体現する FPGA ベースのアクセラレータ Elastic Sample Filter (ESF) を提案
- ▶ 提案する ESF の設計空間探索により，最も高い性能を導く設計指針を同定
- ▶ マルチスレッドベースライン手法に対して最大 5.0×，ADtree に対して最大 6.7× の高速化を達成
- ▶ さらにスケーラビリティを備えた Elastic Scalable Sample Filter (ESSF) を提案

Future Work

- ▶ ESSPER での大規模実験評価

Thank you for your attention.

有望な局所構造とは

- ▶ 最適な構造に絶対に現れない局所構造は存在する
- ▶ 例えば,

$$\mathbf{Pa}_X \subseteq \mathbf{Pa}'_X$$

かつ

$$\text{LS}(X, \mathbf{Pa}_X) > \text{LS}(X, \mathbf{Pa}'_X)$$

なる変数 X の親集合候補 $\mathbf{Pa}_X, \mathbf{Pa}'_X$ があるとき, 局所構造 (X, \mathbf{Pa}'_X) を局所構造 (X, \mathbf{Pa}_X) に置き換えることで, 巡回を作らずに全体のスコアを大きくすることができる. これにより, 変数 X の親集合の候補から \mathbf{Pa}'_x を除くことができる.

- ▶ 有望の定義は他にも考えうる

局所スコアの性質

- 変数集合 $X \sqcup \mathbf{Pa}_X$ による頻度表から $LS(X, \mathbf{Pa}_X)$ が計算できる

Case:0 $\{X_2, X_3\}$ の観測が X_1 の予測に寄与しない ... スコア低 ↘

Case:1 $\{X_2, X_3\}$ の観測が X_1 の予測に寄与する ... スコア高 ↗

X_2, X_3 の観測が
 X_1 の予測に寄与しない

$$LS(X_1, \{X_2, X_3\}) \\ = -28.95$$

Case:0

		X1	
X2	X3	1	2
1	1	10	10
1	2	10	10
2	1	10	10
2	2	10	10

X_2, X_3 の観測が
 X_1 の予測に寄与する

$$LS(X_1, \{X_2, X_3\}) \\ = -1.92$$

Case:1

		X1	
X2	X3	1	2
1	1	20	0
1	2	0	20
2	1	0	20
2	2	20	0

事象 X_3 の観測は
 X_2 の予測に寄与しない

$$LS(X_1, \{X_3\}) \\ = -26.32$$

		X1	
X3		1	2
1		20	20
2		20	20

事象 X_2 の観測は
 X_1 の予測にしない

$$LS(X_1, \{X_2\}) \\ = -26.32$$

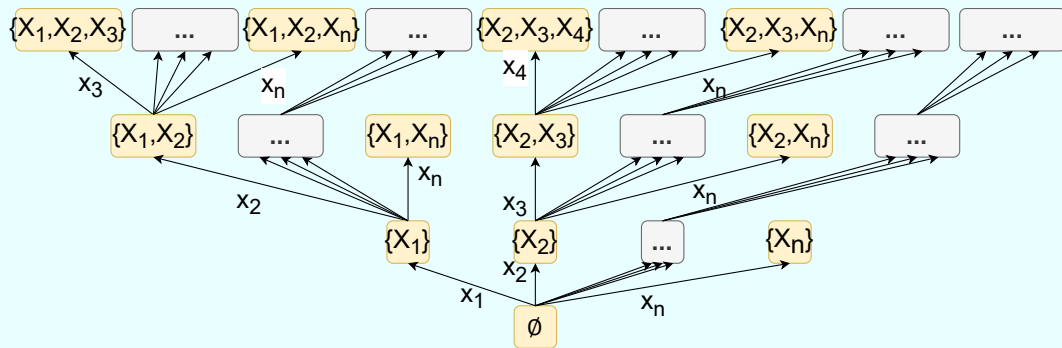
		X1	
X2		1	2
1		20	20
2		20	20

- 有望な局所構造を取りこぼさないために、できるだけ大きな m まで計算できることが望ましい

頻度表の構築 (ADtree)

all-dimensions tree, ADtree

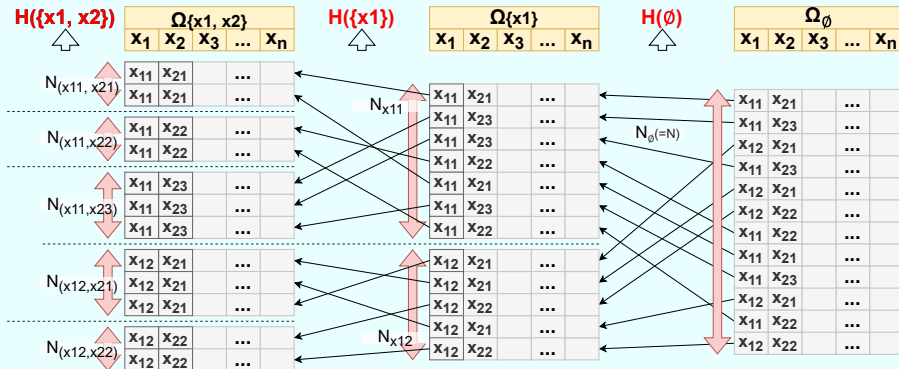
- ▶ ノード \mathbf{X} は $\forall X_i \in \mathbf{X}, j > i$ を満たすノード $\mathbf{X} \sqcup X_j$ へのエッジをもつ (\emptyset が根).
- ▶ ノード \mathbf{X} は $\Omega(\mathbf{X})$ の分割表を持つ.
- ▶ そのため, $m + 1$ の深さまでの ADtree を構築すれば, 親変数数が m 以下の局所スコアをすべて求められる.



頻度表の構築 (ADtree)

all-dimensions tree, ADtree

- ▶ ADtree は根からエッジに沿って、サンプルをバケットソートのように割り振っていくことで、効率的に構築することができる。
- ▶ 変数集合 \mathbf{X} の分割表の構築は $X \in \mathbf{X}$ に対して $O(|X(\Omega)|N)$



頻度表の構築に関する競合手法との比較

- ▶ **ベースライン手法** (サンプル全体を走査して、数え上げる)
 - △ 変数集合 \mathbf{X} の分割表の構築は $O(|\mathbf{X}(\Omega)|N)$
 - 容易に並列化可能
- ▶ **ADtree**
 - 変数集合 \mathbf{X} の分割表の構築は $X \in \mathbf{X}$ に対して $O(|X(\Omega)|N)$
 - △ 親集合の大きさに制限があるとき、ADtree は十分に深くない (すなわち、 $|\mathbf{X}(\Omega)|$ が $|X(\Omega)|$ になることによる計算量の削減は小さい)
 - △ ランダムアクセスとデータの依存性から並列化が困難
- ▶ **ESF** (サンプルの個数に比例するモジュールを用意できる場合)
 - ◎ 変数集合 \mathbf{X} の分割表の構築は $O(|\mathbf{X}(\Omega)|)$