Ph.D. Project: FCCM for Bayesian Network Structure Learning

Ryota Miyagi (2nd year of 3-year PhD), Hideki Takase, The University of Tokyo, Japan

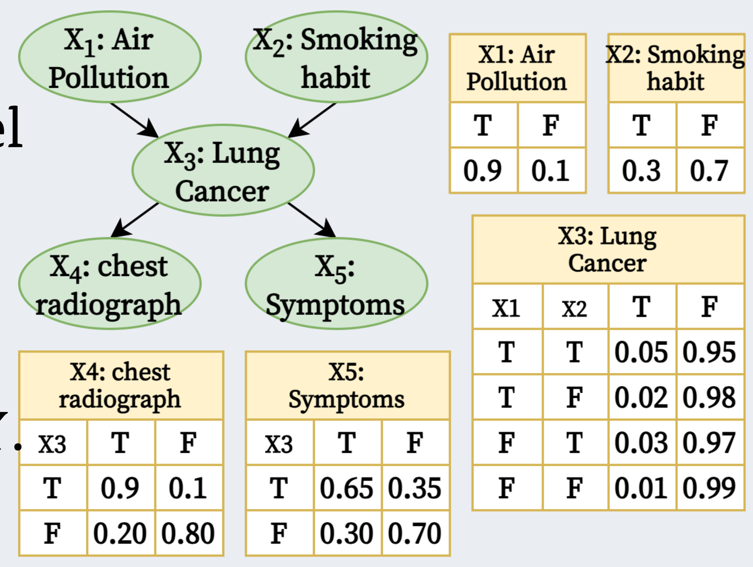
Problem and Motivation

- **Bayesian network:** probability model using a Directed Acyclic Graph (DAG)

$$p(\boldsymbol{V}) = \prod_{X \in \boldsymbol{V}} p(X \mid \boldsymbol{Pa}_X)$$

where, $\boldsymbol{Pa}_X$ is **parent set** of X.

- 🥰 Compact representation of $p(\boldsymbol{V})$



- **Structure learning**: Identifying an appropriate DAG from data;
- 😨 Complexity grows exponentially with variable count.
- 😨 Structure learning for a large Bayesian network is challenging.

🤔 We need FCCM for scalable acceleration

Achievement

1. FCCM acceleration for local score computation
2. Performance modeling of Streaming Computation in ESSPER FPGA Cluster

Ph.D. Projects

3. Scaling our accelerator using multiple FPGAs
4. Scalable acceleration of global structure learning
5. Unified Software for score-based structure learning

Contribution

Scalable Acceleration:

- Establish a methodology for score-based structure learning of large Bayesian networks

Unified Software:

- Easy Benchmarking
- Accelerate further research
- Direct Comparison vs. constraint-based structure learning

On a broader scale,

- Competitive with decision trees?
- Applications in AI/ML, robotics

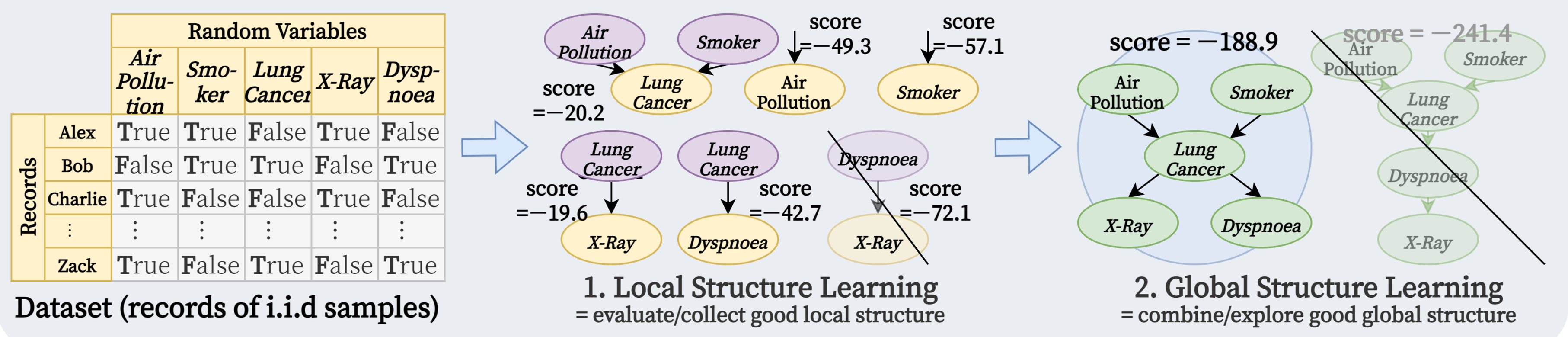
Score-based Structure Learning

- Combinatorial optimization to identify DAGs that maximize scores
- NP-hard for datasets of discrete variables of sufficient size
- The score is decomposed into a product of **local scores** (LSs):

$$p(\boldsymbol{S} \mid G) = \prod_{X \in \boldsymbol{V}} \mathrm{LS}(X, \boldsymbol{Pa}_X)$$

where $\boldsymbol{Pa}_X$ is parent set of X.

- **Local structure learning** evaluates numerous local structures and filters out unpromising ones to narrow the pool of global structures.
- **Global structure learning** constructs the complete graph structure from these promising local structure candidates.



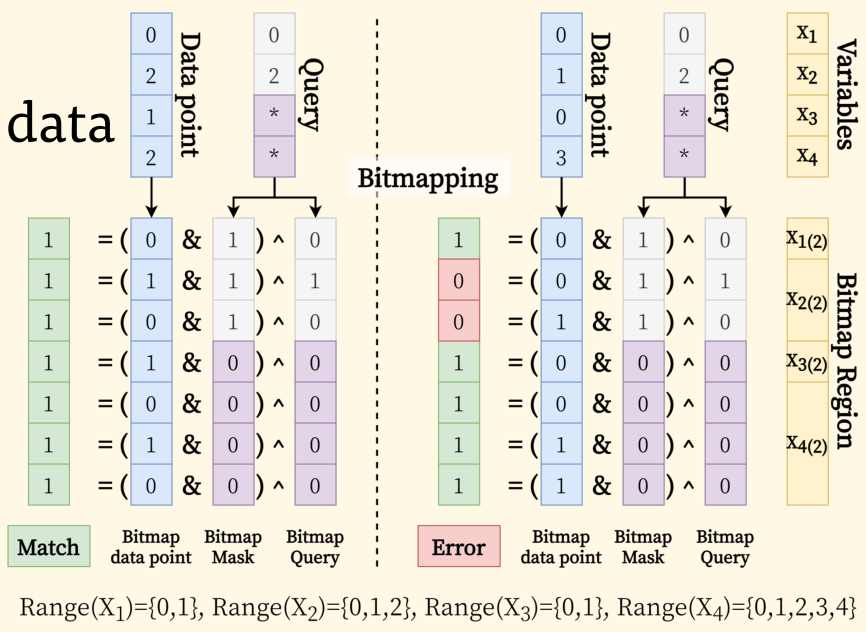
1. FCCM acceleration for local score computation [1]

- Local scores depend on numerous **counting queries**: (e.g. How many records have *Smoker*=T, *LungCancer*=F, *others*=*).
- **ADtree**: sophisticated recursive tree data structure
  👍 High data reusability; 👎 Data dependence prevent parallelization
- **Our method**: scanning dataset for each query with FCCM accelerator
  👎 Lower data reusability; 👍 No data dependency, high parallelism

😮 **Insight**: data reusability benefit << parallelization benefit?

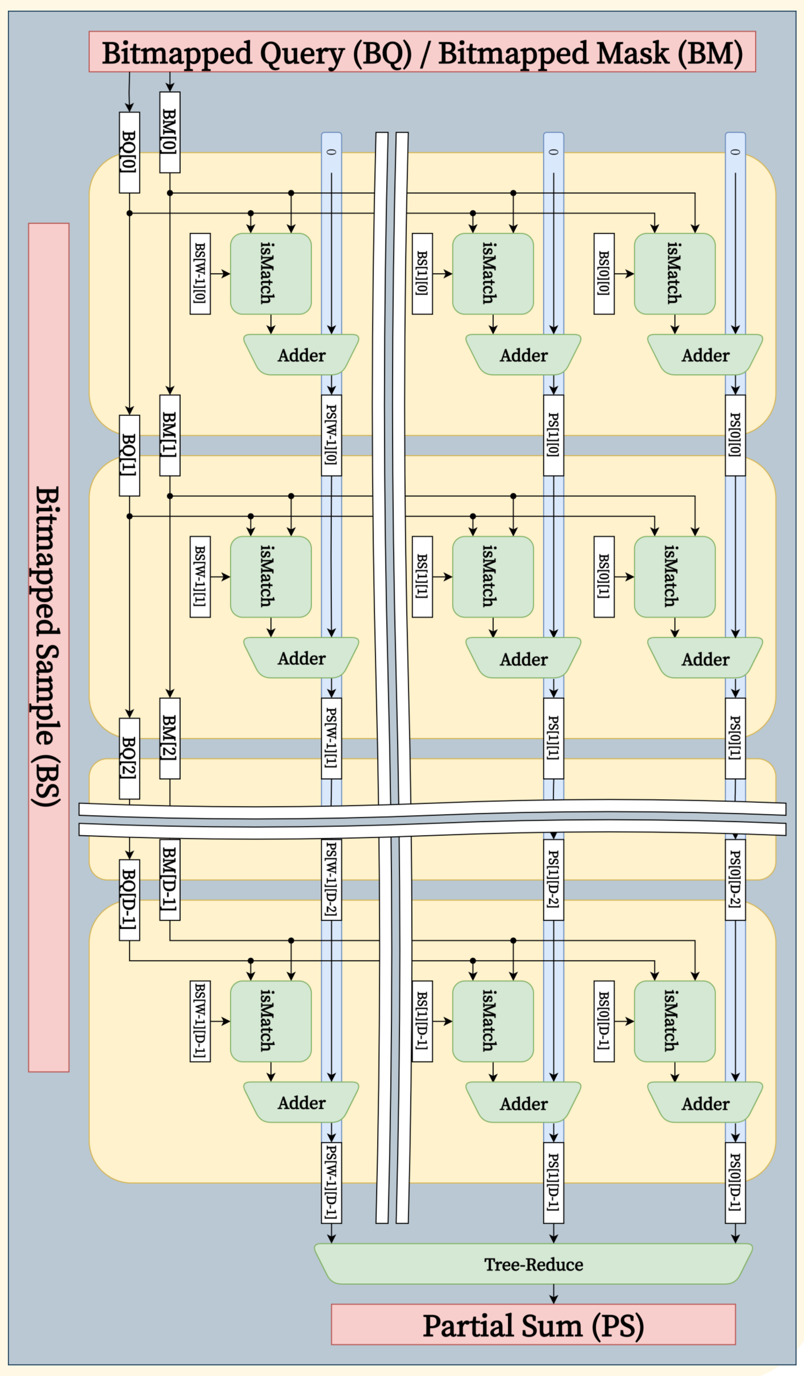
Bitmap representation

- Bitmap both counting queries and data
- 🥰 Simple, FPGA-friendly logic for multiple implementations.
- 🥰 Flattened operation cycle, independent of query conditions.
- 🥰 Uniform handling of different problem instances



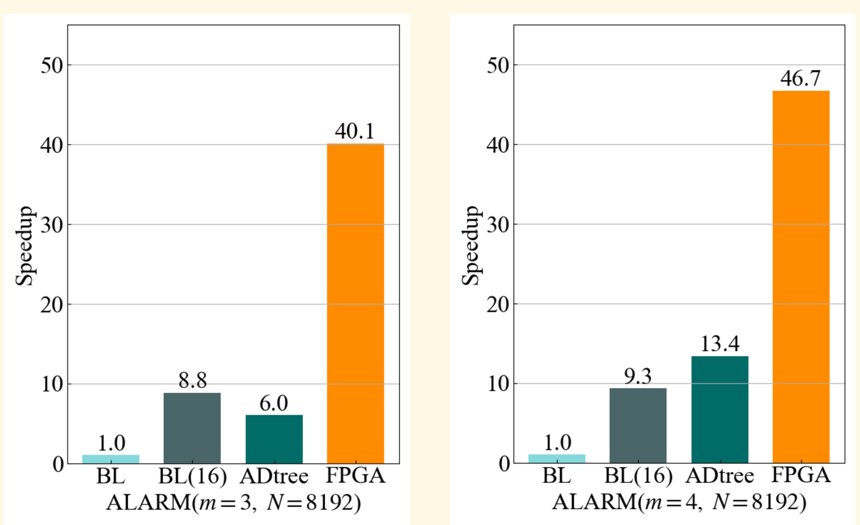
Accelerator design

- Placing PEs of matching operation in parallel / series.
  => Spatial/Temporal Parallelism ↑
- Data pre-cached; queries supplied in dataflow manner
  => Occupancy ~100%



Evaluation

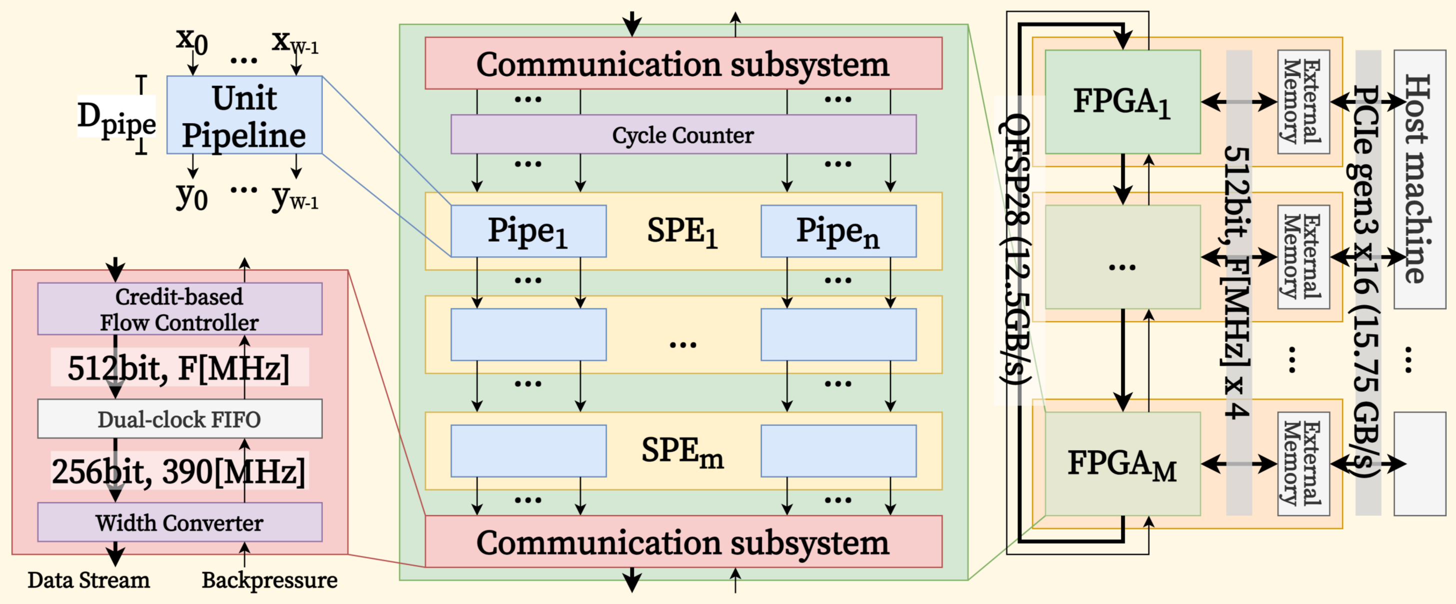
- computation time of all local score
- Up to 6.7x faster than ADtree



v.s. GPU

- Results obtained, but unpublished

2. Performance modeling of stream computing in ESSPER [2]



Model Setting:

- 1D-ring of M-cascaded FPGAs, each with n-parallel m-cascaded PEs
- Back pressure to stall the pipeline under insufficient bandwidth

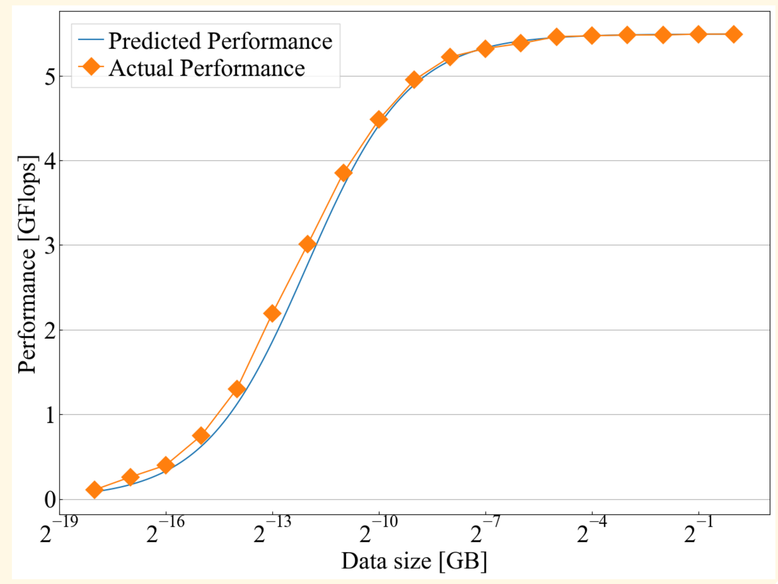
Performance Model:

- Orange: pipeline overhead
- Blue: utilization

$$P(n, m, M) = n \cdot m \cdot M \cdot \mathrm{F} \cdot O_{\mathrm{pipe}} \cdot \frac{1}{1 + \frac{\mathrm{PipeDepth}}{\mathrm{StreamLength}}} \cdot \min\left(\frac{B_{\mathrm{link,\,available}}}{B_{\mathrm{link,\,required}}}, \frac{B_{\mathrm{mem,\,available}}}{B_{\mathrm{mem,\,required}}}, 1\right)$$

Model Validation

- ESSPER[3]: FPGA-cluster in RIKEN
- 1D-ring of 2-cascaded FPGA
- 16 parallel 4B floating-point add



3. Scaling our accelerator using multiple FPGAs

- Cascading the accelerators scales pipeline parallelism.
- Scalable according to the performance model
- When our computation kernel is embedded, inter-FPGA communication suffers, e.g., packet loss.
- Potential causes: Low Frequency (200MHz or lower)

😨 **Concern**: How do we get high frequency? HLS to RTL?

4. Scalable acceleration of global structure learning

- Previous work has focused on algorithmic aspects (i.e., computational time, space, accuracy, etc.).
- We need FCCM to ensure scalable acceleration
- It has many computational factors hindering Parallelization (i.e., dynamic resource allocation, irregular memory access patterns, and recursive and pointer-based operations along graph structures)

😨 **Concern**: How to make it suitable for parallelization?

5. Unified Software for score-based structure learning

- Not directly related to the FCCM, but as a necessary
- Previous software include only elementary heuristics (i.e., hill climbing, tabu search, exhaustive search)
- Difficult to benchmark new methods in a consistent environment
- Also, quantifying acceleration requires a baseline.

😨 **Concern**: How to make time for the implementation?

Acknowledgments

A part of this work was supported by JST CREST (JPMJCR21D2), JST SPRING (JPMJSP2108) and Grant-in-Aid for JSPS Fellows (24KJ0578).

Personal website

References

[1] **R. Miyagi**, R. Yasudo, K. Sano, and H. Takase, “Elastic Sample Filter: An FPGA-based Accelerator for Bayesian Network Structure Learning,” FPT 2022.
[2] **R. Miyagi**, R. Yasudo, K. Sano, and H. Takase, "Performance Modeling and Scalability Analysis of Stream Computing in ESSPER FPGA Clusters." FPT 2023.
[3] K. Sano, A. Koshiba, T. Miyajima, and T. Ueno, “ESSPER: Elastic and scalable FPGA-cluster system for high-performance reconfigurable computing with supercomputer Figaku,” HPC Asia 2023.