Generative and Multi-phase Learning for Computer Systems Optimization

Yi Ding, Nikita Mishra, Henry Hoffmann
Department of Computer Science, University of Chicago

1. Introduction
Optimizing modern computer systems requires tradeoff:
• Deliver reliable performance
• Minimize energy consumption

2. Problem Formulation
Meet latency constraints with minimal energy via config:

Resource management via system configurations:
• Resources have complex, non-linear effects on performance and energy
• Resource interactions create local optima

2.2. Problem Formulation
How to find the optimal system configuration?

• Resource management via config

Optimizing modern computer systems requires tradeoff:
• Deliver reliable performance
• Minimize energy consumption

Hardware: resources to an allocation of application configs
Software: an application profile for hardware

Asymmetric benefits: only configs on the optimal frontier useful

3. Motivational Example: SRAD on big.LITTLE system

4. Our Solution
Key insight: High accuracy ≠ good system results

Machine learning to the rescue, but:
• Scarce data: expensive collection, limited range behavior
• Asymmetric benefits: only configs on the optimal frontier useful

5. Recommender Systems —> Learn by Example

6. Generating Data for Accuracy
Goal: different enough but still realistic, to be plausible

How:
• Random number generator — different but not plausible
• Gaussian Mixture Model (GMM) — plausible but not different

7. Generating Data with a GMM

8. Multi-phase Sampling

9. Experimental Setup

10. Improve Prediction Accuracy w/ GM

11. Improve Energy Savings w/ MP

12. Conclusions

• Generative model improves prediction accuracy.
• Multi-phase sampling method improves energy savings.
• Improving accuracy does not necessarily improve energy consumption.
• We advocate to de-emphasize accuracy but incorporate system structure into learners.