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

Resource management via system configurations:

- Resources have complex, non-linear effects on performance and energy
- Resource interactions create local optimal

How to find the optimal system configuration?

$$\mathcal{C} \leftarrow \{\text{Core assignment}\} \times \{\text{Clock speed assignment}\} \times \{\text{Memory controller}\}$$

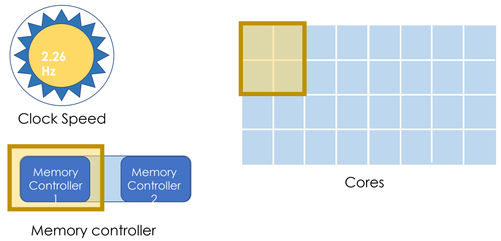


Figure 1. Example of a Configuration Space

2. Problem Formulation

Meet latency constraints with minimal energy via configs

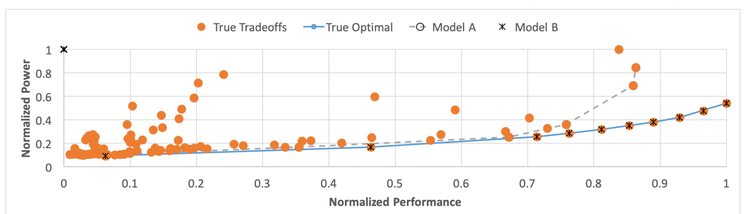
Config: an allocation of hardware resources to an application

Require the power and performance profile for applications

Expensive

Learn to estimate these values

3. Motivational Example: SRAD on big.LITTLE system



	Model A	Model B
Optimal points	Just far enough	True data
Non-optimal points	True data	Very far
Goodness of fit	99%	0
Energy over optimal	22% ❌	0 ✅

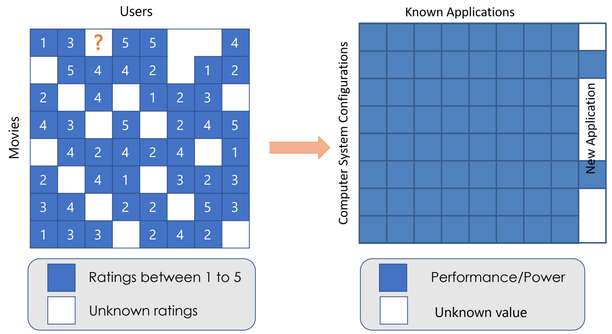
4. Our Solution

Key Insight:
High accuracy ≠ good system results

Machine learning to the rescue, but:

- Scarce data: expensive collection, limited range behavior → Generative Model
- Asymmetric benefits: only configs on the optimal frontier useful → Multi-phase Sampling

5. Recommender Systems —> Learn by Example

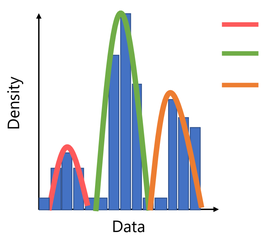


6. Generating Data for Accuracy

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



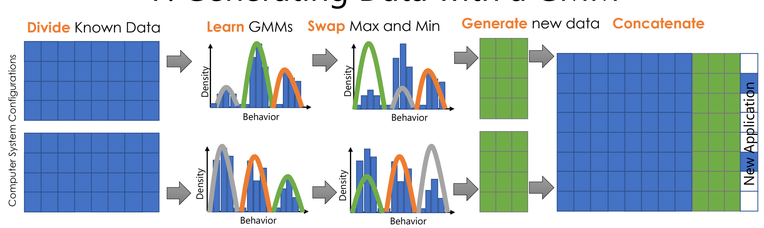
K: number of components
x_i : data points, i=1,…,N
w_k: weight of k-th component

Probability that x_i belongs to k-th comp:

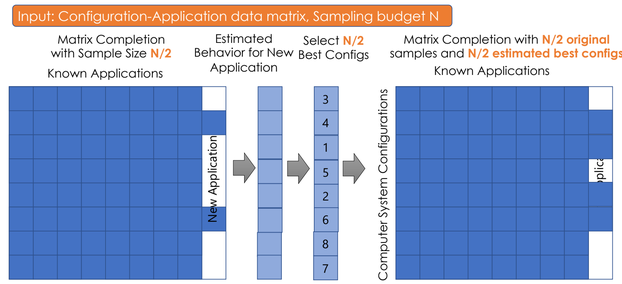
$$p(\mathbf{x}_i) = \sum_{k=1}^{K} w_k g(\mathbf{x}_i|\mu_k, \Sigma_k)$$

$$g(\mathbf{x}_i|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \mu_k)^\top \Sigma_i^{-1}(\mathbf{x}_i - \mu_k)\right)$$

7. Generating Data with a GMM



8. Multi-phase Sampling



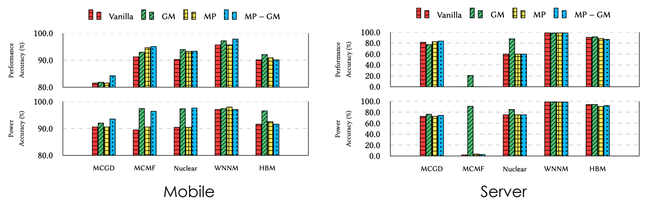
9. Experimental Setup

	Mobile	Server
System	Ubuntu 14.04	Linux 3.2.0 system
Architecture	ARM big.LITTLE	Intel Xeon E5-2690
# Applications	21	22
# Configurations	128	1024

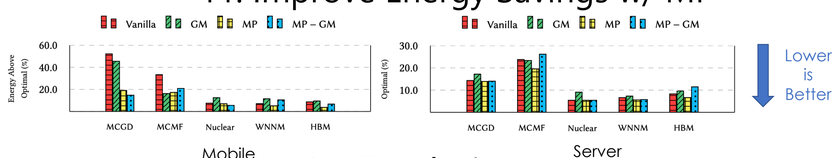
Learning Models	Category
MCGD	MC
MCMF	MC
Nuclear	MC
WNNM	MC
HBM	Bayesian

Frameworks	Definitions
Vanilla	Basic learners
GM	Generative model
MP	Multi-phase sampling
MP-GM	Combine GM and MP

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.

• Yi Ding, Nikita Mishra, Henry Hoffmann. Generative and Multiphase Learning for Computer Systems Optimization. In The 46th Annual International Symposium on Computer Architecture (ISCA '19)