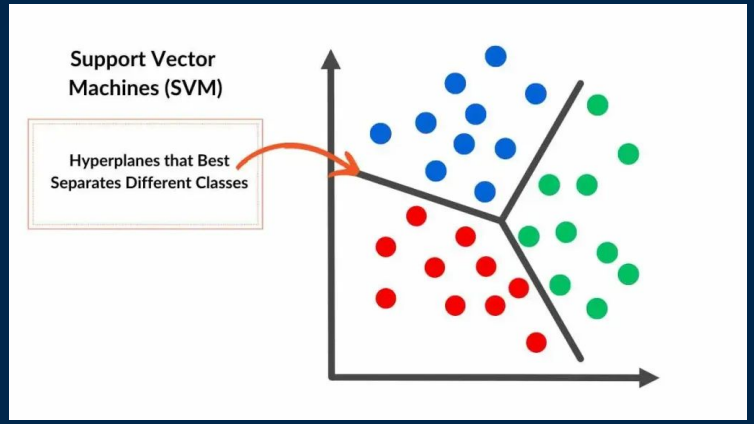
MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Predicting Vectorization Profitability Using Binary Classification

Antoine Trouve, Arnaldo J. Cruz, Dhouha Ben Brahim, Hiroki Fukuyama, Kazuaki J. Murakami, Hadrien Clarke, Masaki Arai, Tadashi Nakahira, Eiji Yamanaka

Joshua Hsueh, Olaf Dsouza,
Joshua Weingarden, Jimmy Dai

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN



The paper discusses…

- Profitability of vectorization
- Machine learning techniques for classification
- Performance improvements
- Distinction between other ML-compiler research

Scalar

```
a[0] = b[0] + c[0]
a[1] = b[1] + c[1]
a[2] = b[2] + c[2]
a[3] = b[3] + c[3]
```

- Independent but seemingly similar instructions…
- Repeated work to load, add, then store

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Vectorization

One SIMD Instruction

```
a[0..3] = b[0..3] + c[0..3]
```

- Parallelization depends on…
  - Dependencies?
  - Memory alignment?
- Performance improvement depends on vector width
- Automatic vectorization by compiler
- Automatic Basic-Block Vectorization (ABBV)
- Overhead work:
  - Packing/unpacking vectors
  - Register pressure
  - Misalignment

Related Work

- Machine Learning for compiler optimizations
- Reduce execution time or compilation time
- Originality:
  - Classification problem instead of regression
  - Focus on vectorization profitability
  - Complementary to other traditional optimizations
  - Stock et al. did ML for automatic vectorization but for perfectly nested, independent loops

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Feature Inputs

Identifier	Level	Range	Description
AST1	AST	$\mathbb{N}$	The increment of the innermost `for` loop
AST2	AST	$\{0, 1\}$	Will the address of the first access to the array be always aligned with the machine's vector size ?
AST3	AST	$\{0, 1\}$	In array accesses, is the induction variables involved in the last dimension the one of the innermost loop ?
AST4	AST	$\mathbb{N}$	Number of array accesses in the body of the loop
AST5	AST	$\mathbb{N}$	The number of arrays accessed inside the loop
AST6	AST	$\{0, 1\}$	Does the benchmark involve any restrict keyword ?
IR1	IR	$\mathbb{N}$	The size of the dataset
IR2	IR	$\mathbb{N}$	Estimation of the dynamic instruction count
IR3	IR	$\mathbb{N}$	The depth of the innermost loop
IR4	IR	$\mathbb{N}$	The estimated trip-count of the innermost loop
IR5	IR	$\mathbb{N}$	Number of IR statements in the innermost loop
IR6	IR	$\mathbb{N}$	The number of variables used in the innermost loop, for the code in single static assignment (SSA) representation

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

AST Features (Source Structure)

ID	Feature	Effect on Profitability
AST1	Loop increment (stride)	Contiguous = good
AST2	Memory alignment	Aligned = good
AST3	Induction variable on last array dim	Improves locality
AST4	# array accesses	Too many = alias risk
AST5	# arrays accessed	↑ → register pressure
AST6	`restrict` keyword	Helps vectorize

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

IR Features (Compiler Level)

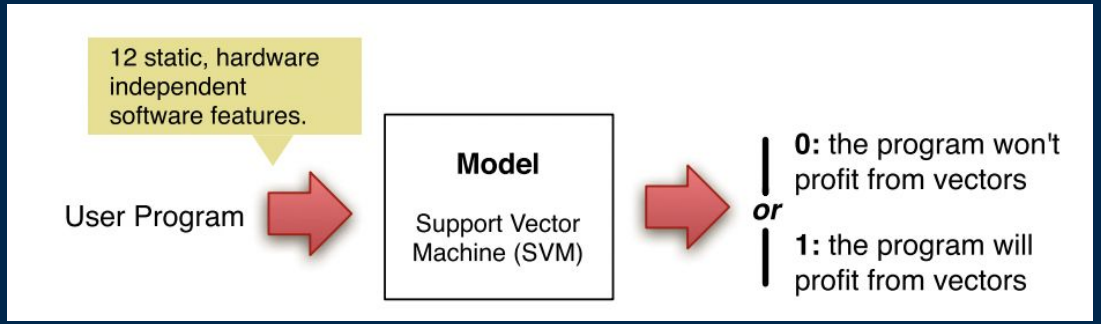
ID	Feature	Effect
IR1	Dataset size	Large loops amortize overhead
IR2	Dynamic instruction count	Too small → not worth SIMD
IR3	Loop depth	Shallow = less benefit
IR4	Trip count	Low trip count = bad
IR5	# IR statements	More ops = higher gain
IR6	# SSA variables	Too many = spills

TSVC Dataset

- 151 base loops × unroll factors (1–20) → ≈ 3020 programs
- Compilers: icc 12.1.5, gcc 4.6.3, LLVM 3.3
- CPU: Intel Core2 Duo 2.66 GHz
- Each program compiled with ABBV ON/OFF
- Speedup = scalar runtime / vectorized runtime
- Labeling:
  - ≥ 1.05 → Profitable
  - ≤ 0.95 → Non-profitable
  - 0.95–1.05 → Ignored

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

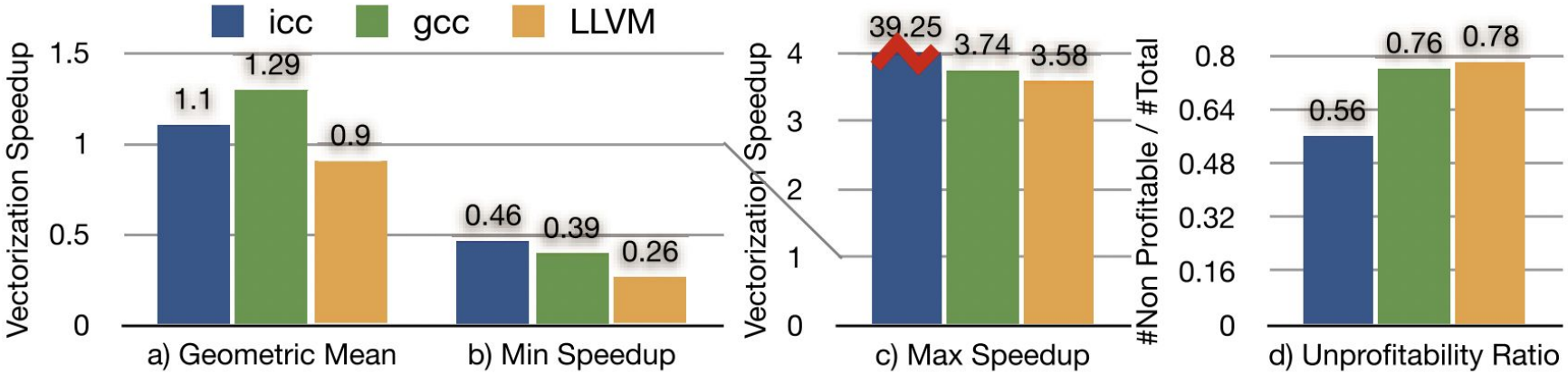
High-Level Model



- Validation: LOOCV (Leave-One-Out Cross-Validation)
- Train on N – 1 samples, test on 1; repeat N times. Maximizes use of small dataset (≈3 k samples)

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Early Results



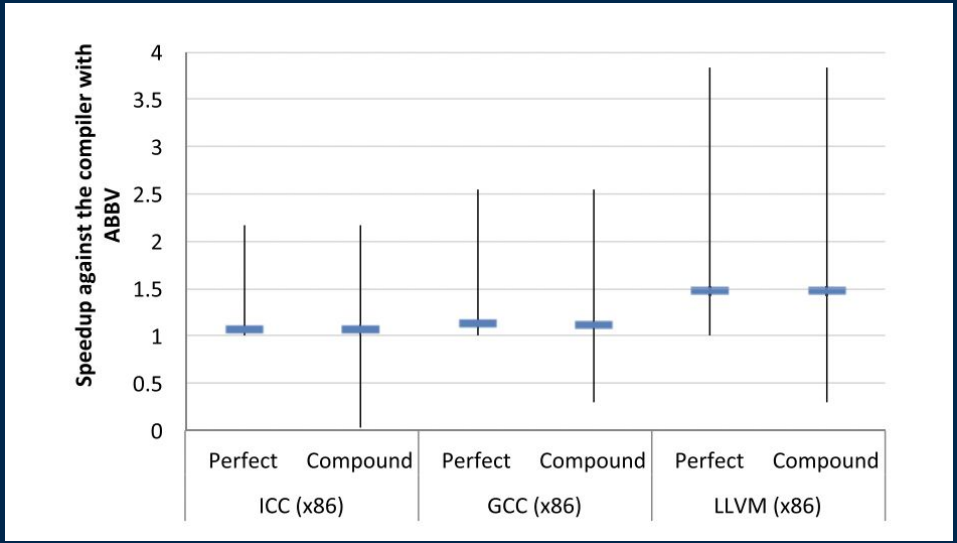
MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Improvements

- Imbalance of data (73% in class 1)
  - Train another SVM on balanced data
  - Compound Predictor
- Model Evaluation:
  - Add Class 2 - within 5% margin, ABBV has minimal effect
  - Model still predicts 0 or 1, but disregard class 2 as part of accuracy
- Comparison to other feature sets
  - GCC compared to random numbers had no difference
  - Keep select 12, more applicable

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Final Results



MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Final Results

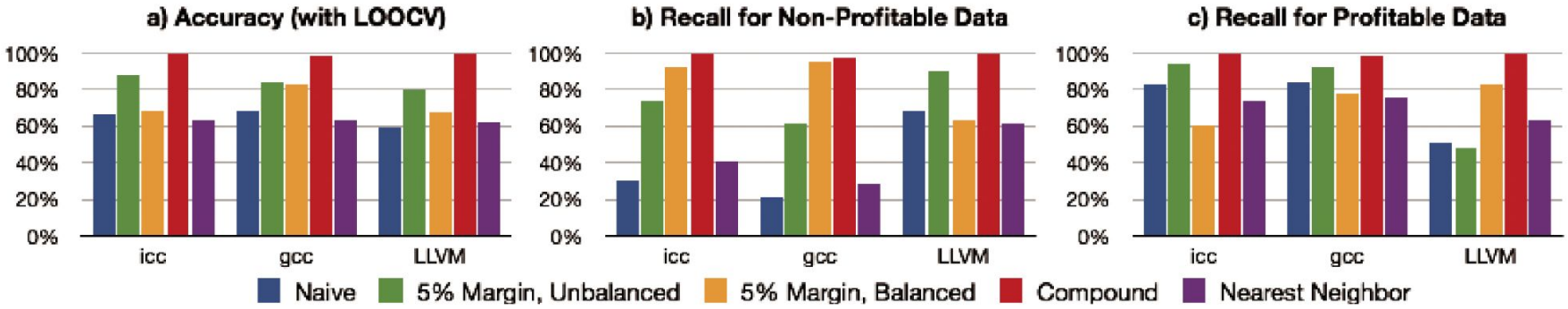


Fig. 3 Prediction accuracies for all methods and compilers: (a) the LOOCV total accuracies (b) the LOOCV accuracies per class.

Final Results

- Unprofitability ratio drops down to <1% for all compilers
- Mispredicts only 2 programs
- Up to 3.8 times speedup
- On average, 47.3% faster in LLVM
- Low compilation overhead (<1 second)
- One time training on user machine (several hours)

Future Challenges

- 2 Cases of Misprediction
  - Loop unroll factor of 12
  - 12 is a multiple of the SIMD datapath width
  - Singularity, edge case
- Non-linear expressions
  - Hard to measure software characteristics in these cases
  - These better express control flow-graph of programs

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Thank You!

Questions?