Solving Classification problems using Genetic Programming Algorithms on GPUs

Alberto Cano, Amelia Zafra and Sebastián Ventura

Knowledge Discovery and Intelligent Systems Research Group

University of Córdoba, Spain

HAIS'10

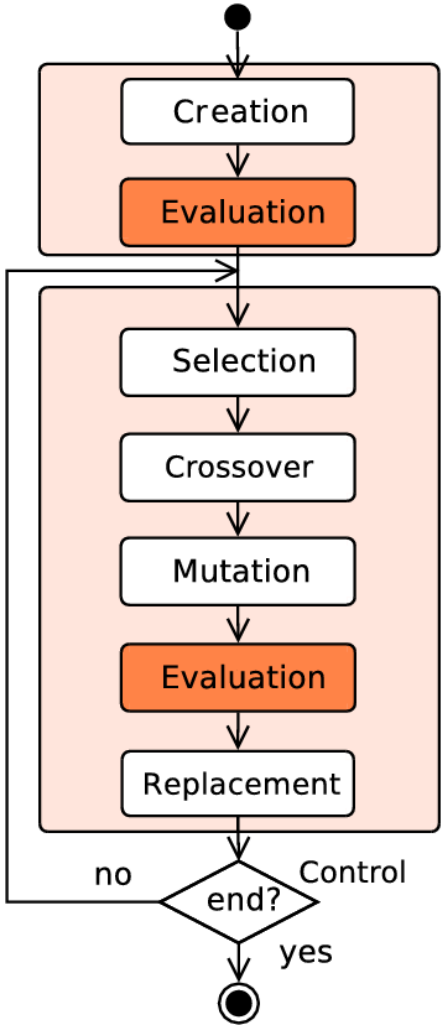
OUTLINE

- Introduction
- Genetic Programming Evolution Model
- GPU Programming Model
- Experiments
- Results
- Conclusions

INTRODUCTION

- Classification rules for Data Mining
- Genetic Programming
- Grammar-Guided Genetic Programming (G3P)
- High computational time

Genetic Programming Evolution Model



Phase	Time (ms)	Percentage
Initialization	8647	8,96%
Creation	382	0,39%
Evaluation	8265	8,57%
Generation	87793	91,04%
Selection	11	0,01%
Crossover	13	0,01%
Mutation	26	0,03%
Evaluation	82282	85,32%
Replacement	26	0,03%
Control	5435	5,64%
Total	96440	100 %

EVALUATION

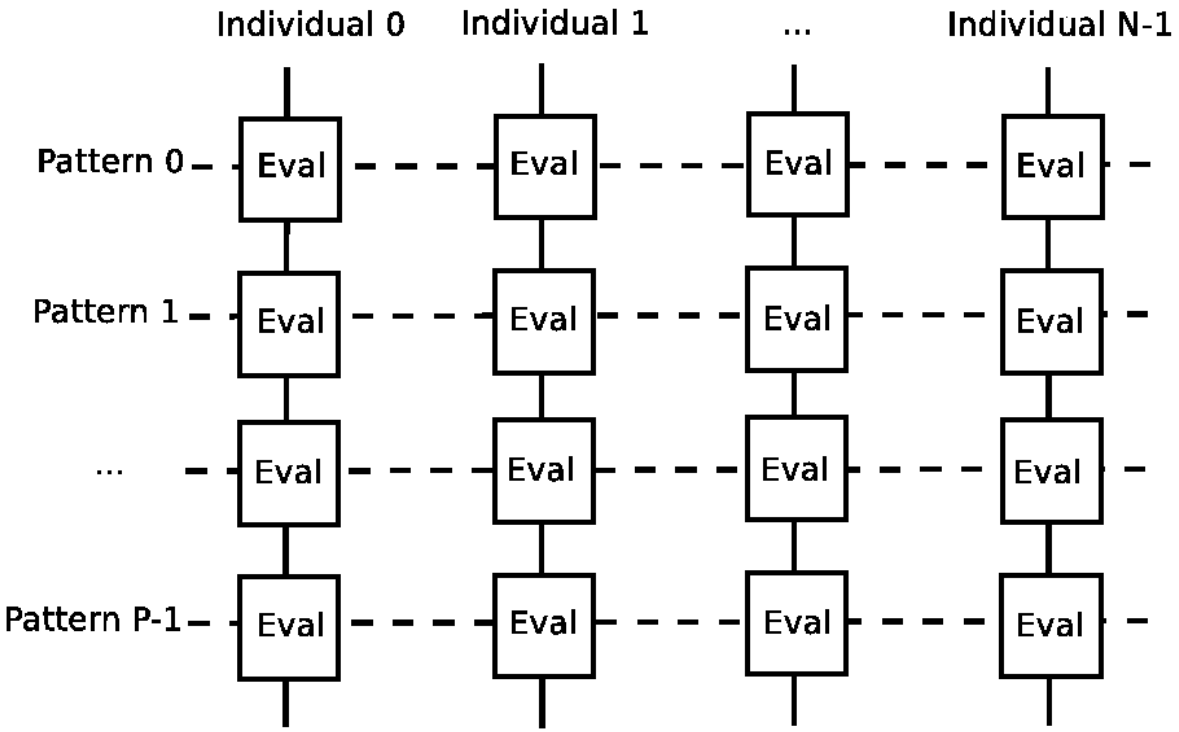
- The fitness function calculates a fitness value for each individual
- Each individual must be tested over every pattern
- The fitness value is a quality index of the individual

$$fitness = hits - fails$$

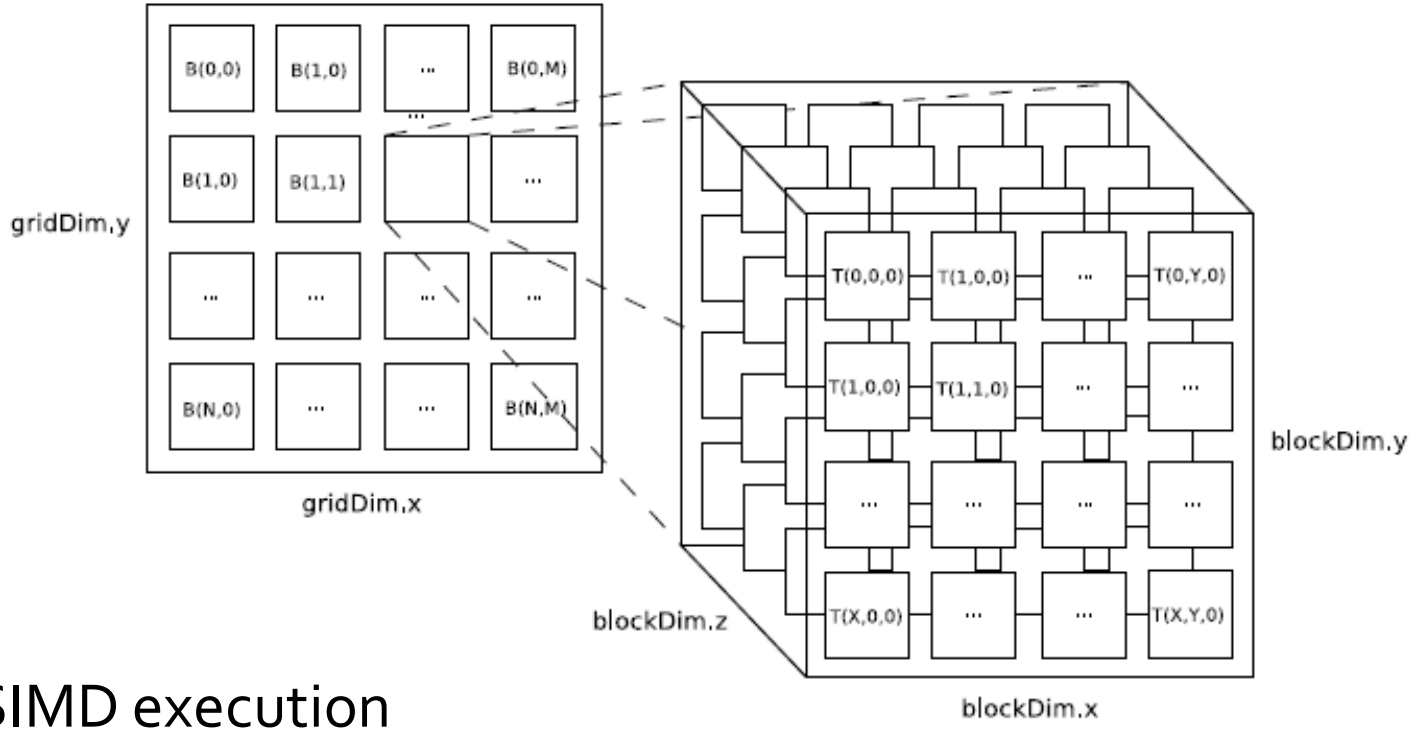
- The performance slows as the population or the patterns size is increased

PARALLELIZATION

- The fitness function can be computed for each individual concurrently
- The test of a rule over a pattern is self-dependent



GPU MODEL



- SIMD execution
- Up to 65536 x 65536 x 512 = $2*10^{12}$ threads
- Many core architecture: 240 cores NVIDIA GTX 285
- Large high-bandwidth device memory

EVALUATION ON GPU

1. Evaluation of the patterns

   A thread performs the test of one individual over one pattern.

   The result is stored: *result[individual][pattern] = hit | fail ;*

   *Threads count = patterns count * population size*

   These millions of evaluations can be performed concurrently.

2. Reduction

   A function that counts the evaluation results per individual.

   These values are employed to build the confusion matrix and then the fitness of the individual is calculated.

CLASSIFICATION ALGORITHMS

- Falco, Della and Tarantino

$$fitness = I - ((t_p + t_n) - (f_p + f_n)) + \alpha * N$$

- Tan, Tay, Lee and Heng

$$Se = \frac{t_p}{t_p + f_n} \qquad Sp = \frac{t_n}{f_p + t_n} \qquad fitness = Se * Sp$$

- Bojarczuk, Lopes and Freitas

$$Sy = \frac{maxnodes - 0.5 * numnodes - 0.5}{maxnodes - 1} \qquad fitness = Se * Sp * Sy$$

EXPERIMENTS

- UCI machine learning datasets

  Shuttle: 9 attributes, 58000 instances and 7 classes

  Poker hand: 11 attributes, 10^6 instances and 10 classes

- Hardware setup

  Intel i7 920 @ 2.6 GHz

  2 NVIDIA GTX 285 2GB

- How do the population size and the number of instances influence the speed-up ?

RESULTS

Shuttle

		Tan Model				Falco Model				Bojarczuk Model			
	Pop	100	200	400	800	100	200	400	800	100	200	400	800
Speed up	Java	1	1	1	1	1	1	1	1	1	1	1	1
	C1	2,8	3,1	3,2	2,9	5,4	8,1	5,2	5,0	18,8	12,5	11,2	9,5
	C2	5,5	6,1	6,3	5,7	10,6	15,9	10,5	10,1	35,9	24,6	22,1	18,1
	C4	10,1	11,5	12,5	10,7	19,7	30,3	20,5	19,8	65,2	47,3	40,1	33,7
	C8	11,1	12,4	13,4	10,3	19,9	30,1	21,2	20,6	65,7	46,8	40,5	34,6
	GPU	218	267	293	253	487	660	460	453	614	408	312	269
	GPUs	436	534	587	506	785	1187	899	867	1060	795	621	533

Poker hand

		Tan Model				Falco Model				Bojarczuk Model			
	Pop	100	200	400	800	100	200	400	800	100	200	400	800
Speed up	Java	1	1	1	1	1	1	1	1	1	1	1	1
	C1	2,7	3,2	3,1	3,0	4,6	5,0	5,6	4,9	5,5	5,7	5,8	4,7
	C2	5,5	6,5	6,7	5,5	9,0	9,8	11,1	9,7	10,6	11,0	11,6	11,3
	C4	10,3	11,1	12,8	10,5	16,8	18,9	21,6	18,9	20,3	20,7	22,5	22,3
	C8	11,2	12,9	14,0	10,2	18,5	20,5	23,3	26,4	21,8	22,0	24,2	23,8
	GPU	155	174	234	221	688	623	648	611	142	147	148	142
	GPUs	288	336	500	439	1275	1200	1287	1197	267	297	288	283

CONCLUSIONS

- ✓ GPUs are best for massive multithreading tasks
- ✓ Speed-up is great even for small datasets
- ✓ The execution time of high dimensional problems is lowered from a week to less than an hour
- ✓ The GPU model scales to multiple devices
- ✓ Next stop: a parallel and distributed evolution model

UNIVERSIDAD DE CÓRDOBA

KDIS Lab
KNOWLEDGE DISCOVERY AND INTELLIGENT SYSTEMS

Thank you!

Solving Classification problems using Genetic Programming Algorithms on GPUs

Alberto Cano, Amelia Zafra and Sebastián Ventura

Knowledge Discovery and Intelligent Systems Research Group

University of Córdoba, Spain

HAIS'10