Acceleration of Machine Learning Models based on GPGPU technology

for fast data mining in multidisciplinary physical environments

Mauro Garofalo
University of Naples Federico II

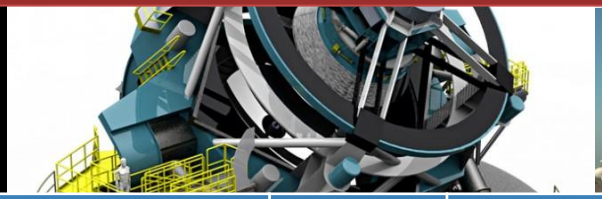
Guarino, D.; Brescia, M.; Cavuoti, S.; Pescapè, A.; Longo, G.; Ventre G.

I've seen things you people wouldn't believe. *Attack ships on fire off the shoulder of Orion. I've watched c-beams glitter in the dark near the Tannhäuser Gate.* ***All those ... moments will be lost in time, like tears...in rain.***
Time to die...

ROY EFFECT: (*Blade Runner*)

MOST DATA WILL **NEVER** BE SEEN BY HUMANS!!!

Data quantity and complexity

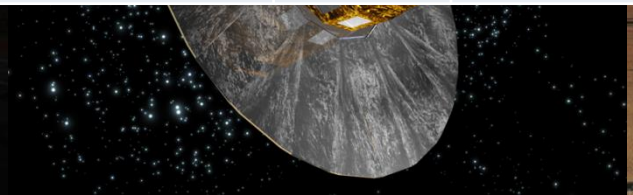




	TB	Total	Epochs	Parameters
VST	0.15 TB/day	100 TB	tens	S>100
HST		120 TB	few	>100
PANSTARRS		600 TB	Few-many	>>100
LSST	30 TB/day	> 10 PB	hundreds	>>100
GAIA		1 PB	many	>>100 heterogeneous
SKA	**1.5 PB/day**		>> 10^2	hundreds










Data, Data everywhere, yet ...

I can't ***find*** the data I ***need***

- Data is scattered over the network
- many versions and formats

I can't ***get*** the data I ***need***

- need an expert to get the data

I can't ***understand*** the data I ***found***

- available data poorly documented

I can't ***use*** the data I ***found***

- results are unexpected
- data needs to be transformed from one form to other
- classic tools can't handle data dimension

Machine Learning

Machine learning: Field of study that gives computers the ability to learn without being explicitly programmed.

Arthur Samuel (1959).

February 24, 1956, Arthur Samuel's Checkers program, which was developed for play on the IBM 701, was demonstrated to the public on television.

In 1962, self-proclaimed checkers master Robert Nealey played the game on an IBM 7094 computer…

May 11, 1997 – Deep Blue defeats Kasparov

…the computer won!

How to Accelerate?

3 Ways to Accelerate Applications

Maximum Flexibility	"Drop-in" Acceleration	Easily Accelerate Applications
Programming Languages	Libraries	Compiler Directives
NVIDIA CUDA	Thrust	OpenACC Directives for Accelerators

Application

Scientific Problem		Model	CUDA Platform
Astrophysics	Globular Cluster *Classification*	GA	Thrust
	Photometric Redshift *Estimation*	MLPGA	CUDA C
Computer Network	Network Traffic *Classification*	MLPGA	OpenACC
Medical	Alzheimer Disease *Prediction*	SVM	Hybrid

Development Environment: DAME Program

Multi-purpose data mining with machine learning Web App REsource

Extensions

- **DAME-KNIME**
- **ML Model plugin**

Specialized web apps for:
Transient classification (STraDiWA

- **text mining (VOGCLUSTERS)**
- **EUCLID Mission Data Quality**

Web Services:

- **GAME (GPU-CUDA ML model)**
- **WFXT Time Calculator**
- **SDSS mirror**

http://dame.dsf.unina.it/

Science and management
Documents
Science cases
Newsletters

Brescia, M.; Cavuoti, S.; Nocella, A.; ***Garofalo, M****. et al 2014 PASP. Vol. 126, No. 942 pp 783-797*

Genetic Algorithm

- Search algorithms that mimics natural selection;
- A Population of individuals evolves to promoting survival and reproduction to better fit the properties of a given environment;
- GAME has been designed to solve optimization problems for classification and regression;
- Their functional structure naturally lends itself to be implemented on parallel architectures.

Correspondence between biological and mathematical model

Biology	Math
Individual	Vector **x**
Adaptation to Environment	Fitness Function **f(x)**
Competition	Selection Function
Reproduction	Crossover and Mutation
Survival of fittest	Better Solution

Accelerating with Thrust

Maximum Flexibility

"Drop-in" Acceleration

Easily Accelerate Applications

Programming Languages

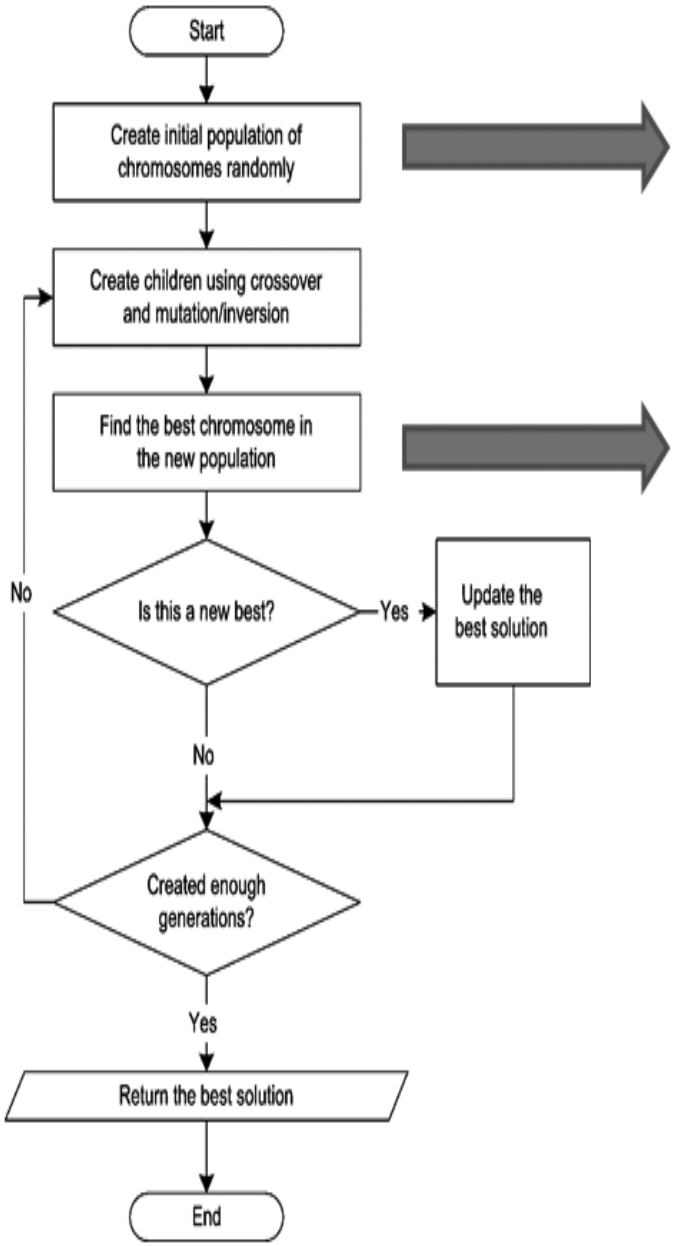
Libraries

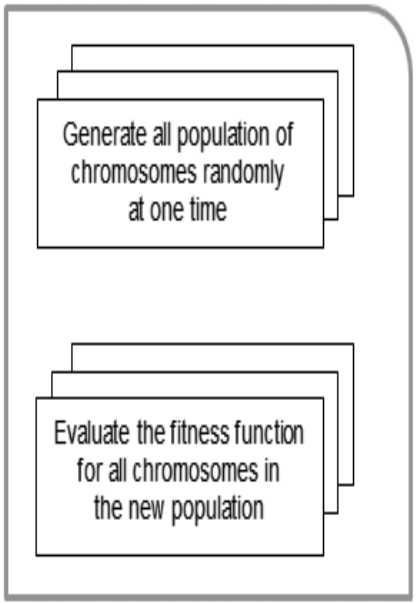
Thrust

cuBLAS

Compiler Directives

GAME Algorithm





C++

```
for (int i = 0; i < num_features; i++) {
   for (int j = 1; j <= poly_degree;
j++) {
      ret +=  v[j] * cos(j * input[i]) +
         v[j + poly_degree] * sin(j *
input[i]);
   }
}
```

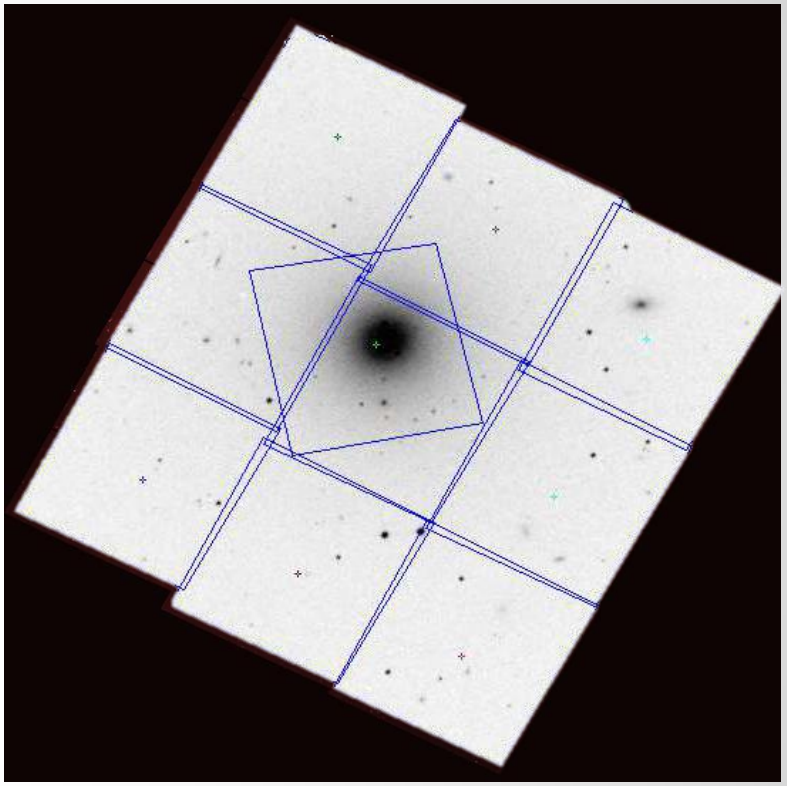
Thrust

```
struct sinFunctor {
  __host__ __device__
 double operator()(tuple <double, double> t) {
  return sin(get < 0 > (t) * get < 1 > (t));
 }
};

thrust::transform
 (thrust::make_zip_iterator
  (thrust::make_tuple(j.begin(), input.begin())),
 thrust::make_zip_iterator
  (thrust::make_tuple(j.end(), input.end()),
 vSin.begin(),
 sinFunctor());

double sinComp= reduce(vSin.begin(), vSin.end());
```

Globular Cluster Recognition

NGC1399 Dataset

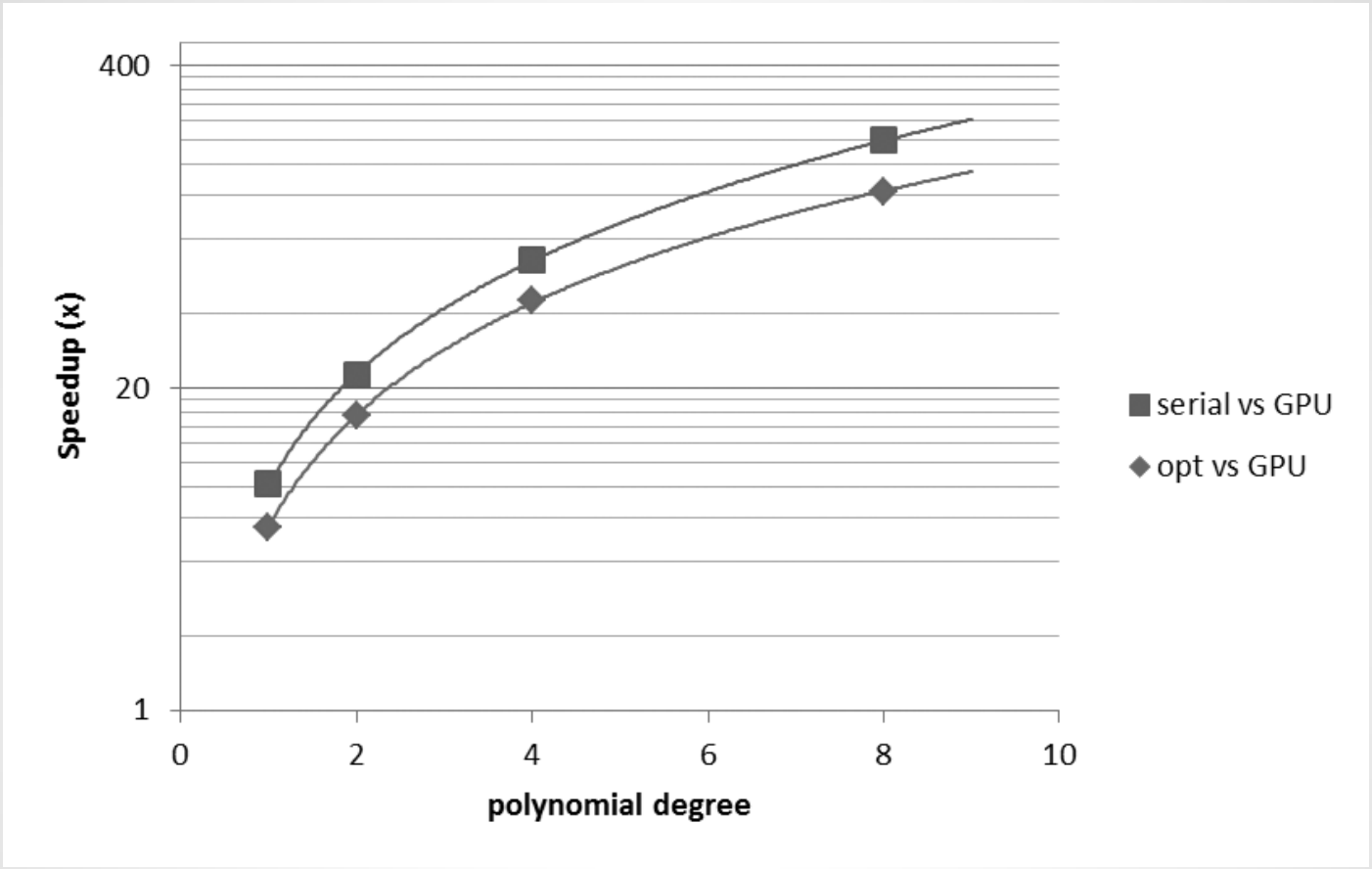
- 7 optical parameters (Magnitude at various apertures and image FWHM) (feature 1-7);
- 4 structural parameters (radii and brightness) (features 8-11);
- The label of the class, 0 (no GC), 1 (yes GC) (last column);



	Classification Accuracy
CPU	86%
CPU+GPU	84%

Brescia, M.; Cavuoti, S.; Paolillo, M. et al.; 2012, ***MNRAS, 421, 2, 1155-1165***

GAME Performance Test

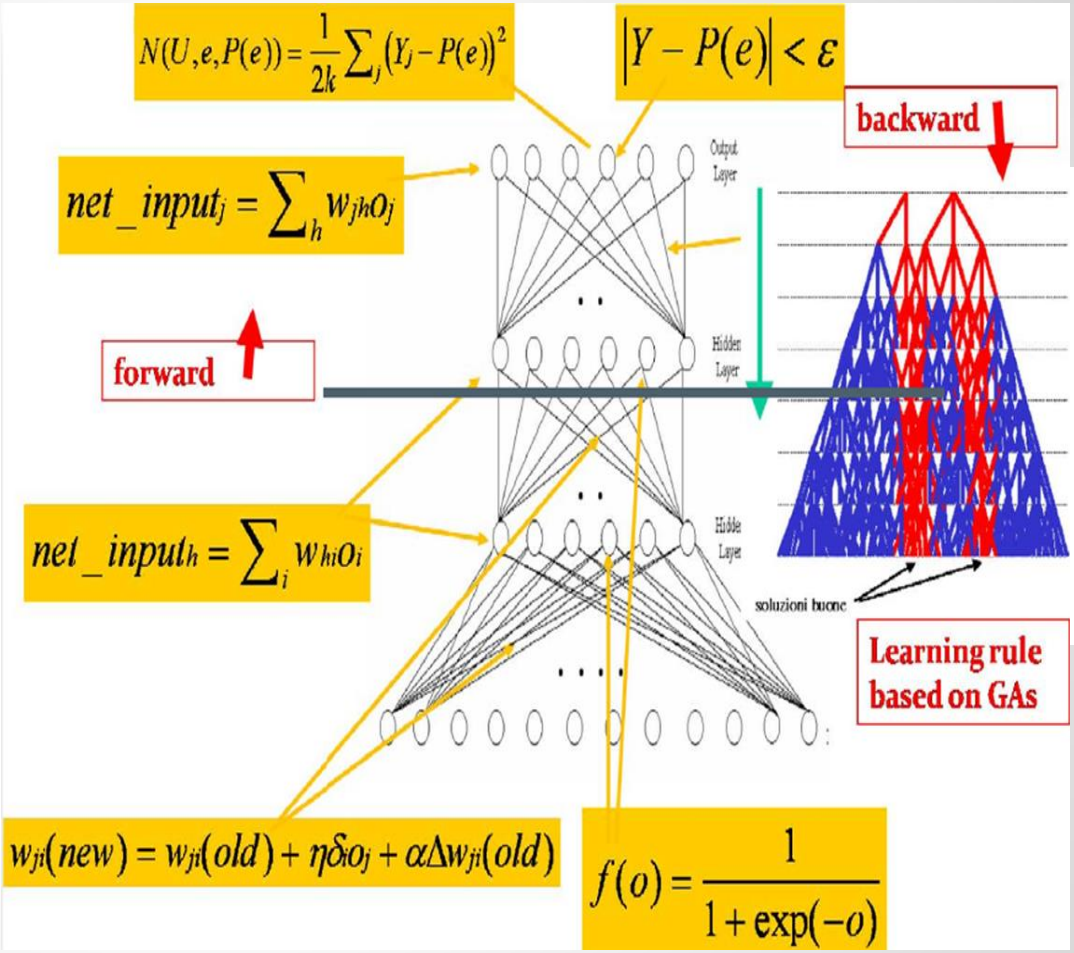


Speedup: up to 200x

Cavuoti, S.; ***Garofalo, M.;*** *Brescia, M.; et al. 2012, Proceedings of WIRN,* ***Springer*** *Vol. 19*
Cavuoti, S.; ***Garofalo, M.;*** *Brescia, M.; et al. 2014,* ***New Astronomy*** *Vol. 26 pp 12-22*

MLPGA Model

- Hybrid model NN (MLP) + GA
- Supervised machine learning model (provides a training phase with input data + known targets)
- Weights evolution using GA instead of Back Propagation
- High generalization capability on unknown data
- Solve classification and regression problems



The training phase is very slow and the model does not scale with the input data.

Accelerating with CUDA C

Maximum Flexibility

"Drop-in" Acceleration

Easily Accelerate Applications

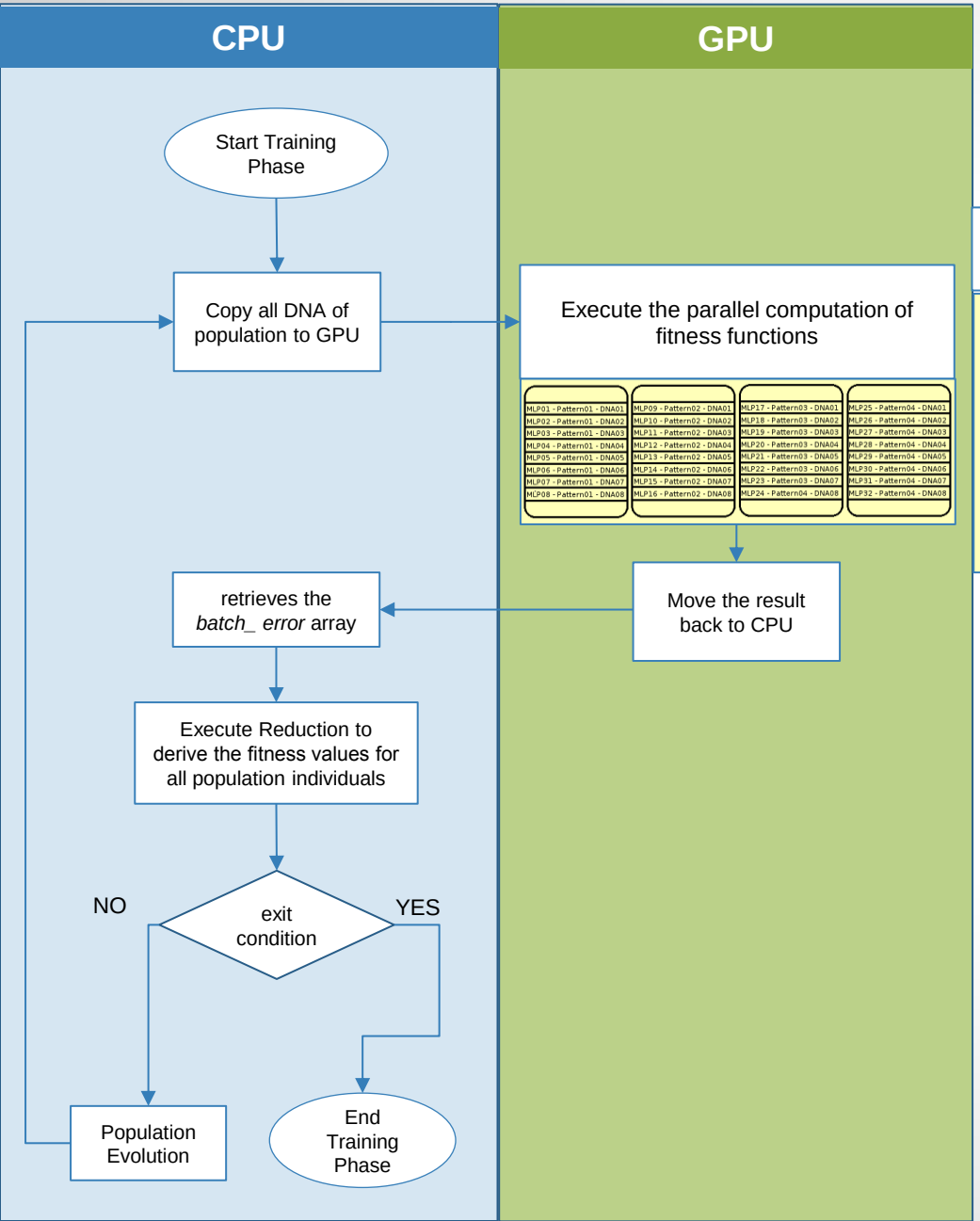
Programming Languages

CUDA C

Libraries

Compiler Directives

FMLPGA



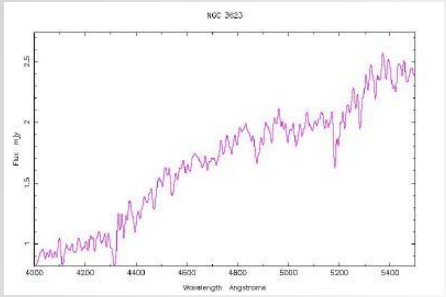
FMLPGA - CUDA GRID

MLP01 - Pattern01 - DNA01	MLP09 - Pattern02 - DNA01	MLP17 - Pattern03 - DNA01	MLP25 - Pattern04 - DNA01
MLP02 - Pattern01 - DNA02	MLP10 - Pattern02 - DNA02	MLP18 - Pattern03 - DNA02	MLP26 - Pattern04 - DNA02
MLP03 - Pattern01 - DNA03	MLP11 - Pattern02 - DNA03	MLP19 - Pattern03 - DNA03	MLP27 - Pattern04 - DNA03
MLP04 - Pattern01 - DNA04	MLP12 - Pattern02 - DNA04	MLP20 - Pattern03 - DNA04	MLP28 - Pattern04 - DNA04
MLP05 - Pattern01 - DNA05	MLP13 - Pattern02 - DNA05	MLP21 - Pattern03 - DNA05	MLP29 - Pattern04 - DNA05
MLP06 - Pattern01 - DNA06	MLP14 - Pattern02 - DNA06	MLP22 - Pattern03 - DNA06	MLP30 - Pattern04 - DNA06
MLP07 - Pattern01 - DNA07	MLP15 - Pattern02 - DNA07	MLP23 - Pattern03 - DNA07	MLP31 - Pattern04 - DNA07
MLP08 - Pattern01 - DNA08	MLP16 - Pattern02 - DNA08	MLP24 - Pattern04 - DNA08	MLP32 - Pattern04 - DNA08

At each iteration all N MLP networks (GA chromosomes) are created and processed in parallel on GPU

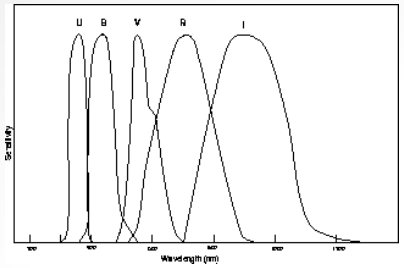
Photometric Redshifts: as an Inverse Problem

Spectral Energy Distribution convolved with band filters



Galaxy spectrum - F(l)

X



Photometric system - S_i(l)

=

$$m_U = -2.5\log_{10}\frac{\int F(\lambda)S_U(\lambda)d\lambda}{\int S_U(\lambda)d\lambda} + c_u$$

$$m_B = -2.5\log_{10}\frac{\int F(\lambda)S_B(\lambda)d\lambda}{\int S_B(\lambda)d\lambda} + c_B$$

Etc…

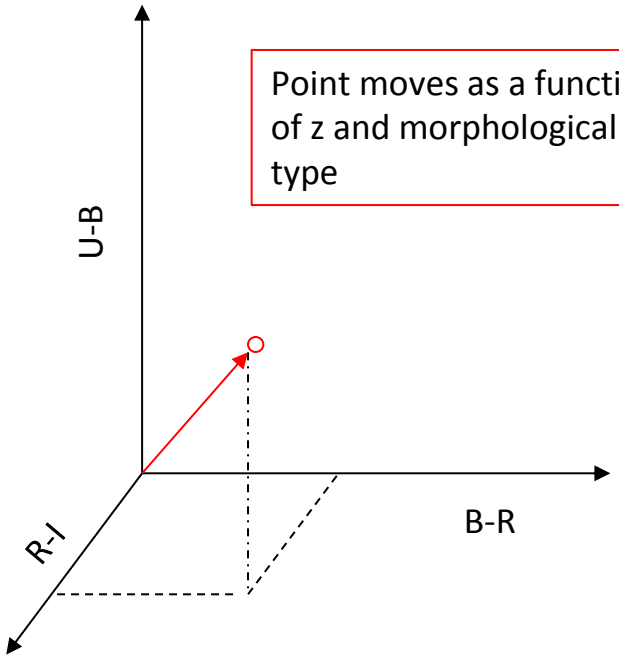
Color indexes

$$U - B \equiv m_U - m_B$$

$$B - R \equiv m_B - m_R$$

$$etc.$$

Photo-z are an inverse problem

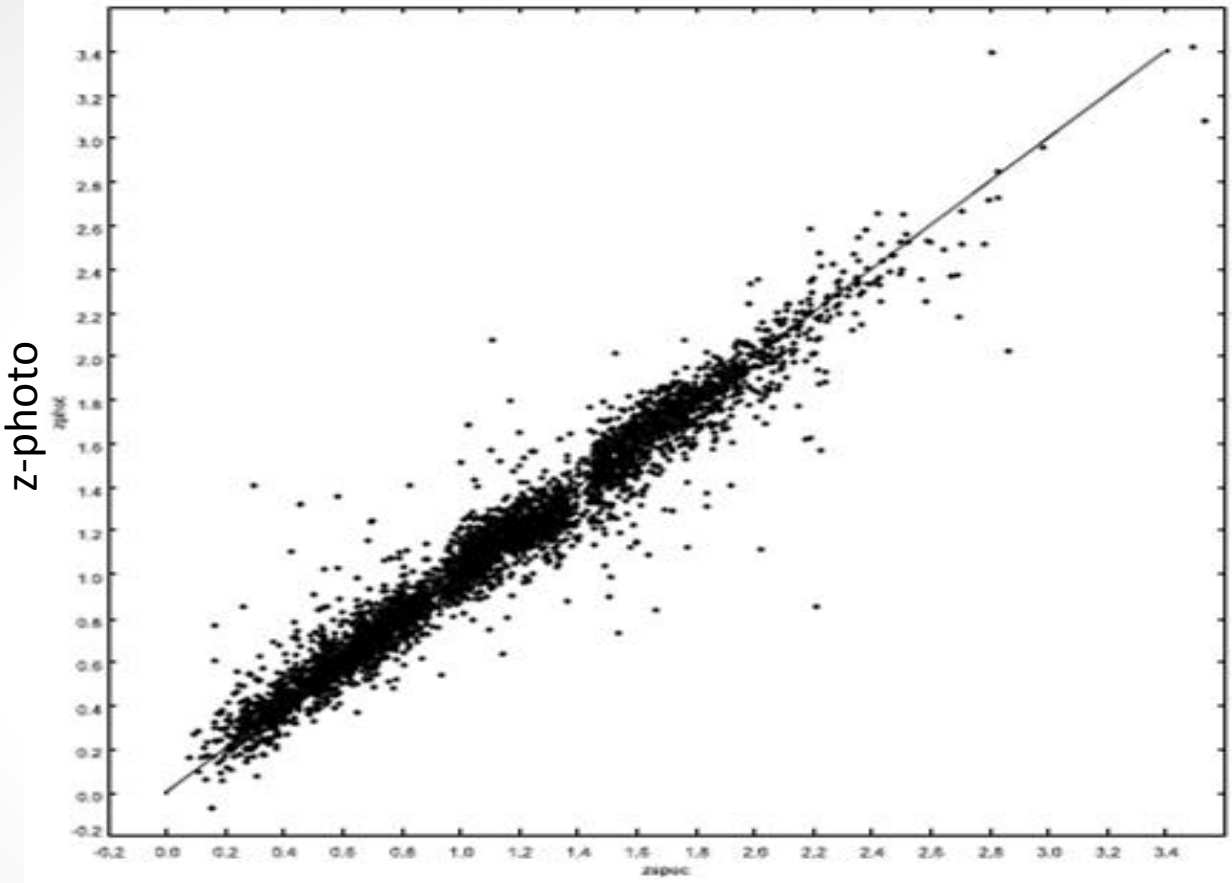


FMLPGA Scientific Validation

Dataset:
1000 patterns - 11 features

Epochs:
from 1000 to 50000

Selection functions:
Roulette, Ranking and Fitting



zspec vs zphoto scatter plot best result with ROULETTE selection function

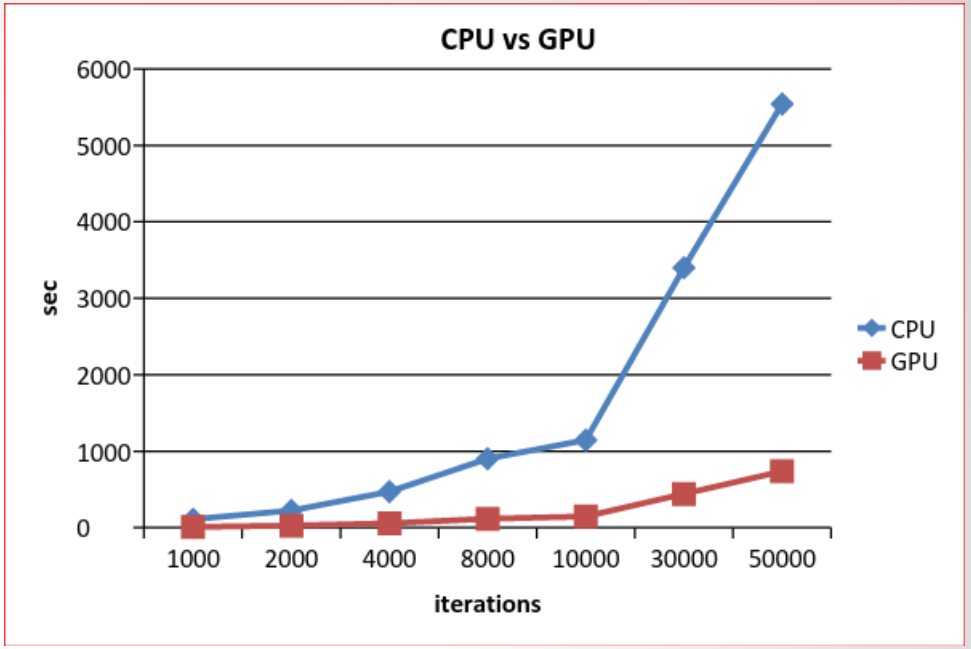
Standard deviation: < 0.02

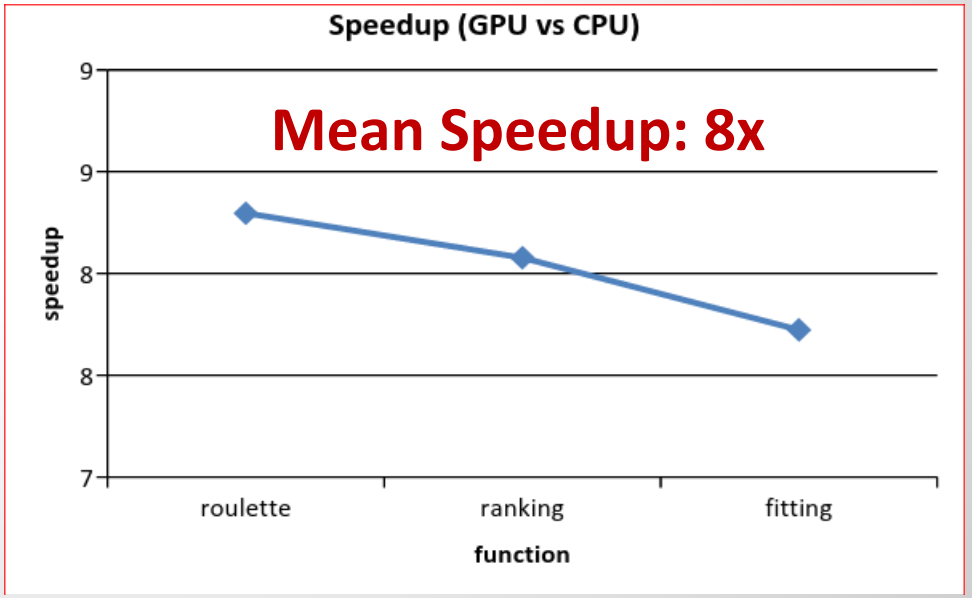
FMLPGA Performance tests

Dataset:
1000 patterns - 11 features

Epochs:
from 1000 to 50000

Selection functions:
Roulette, Ranking and Fitting





Accelerating with OpenACC

Maximum Flexibility

"Drop-in" Acceleration

Easily Accelerate Applications

Programming Languages

CUDA C

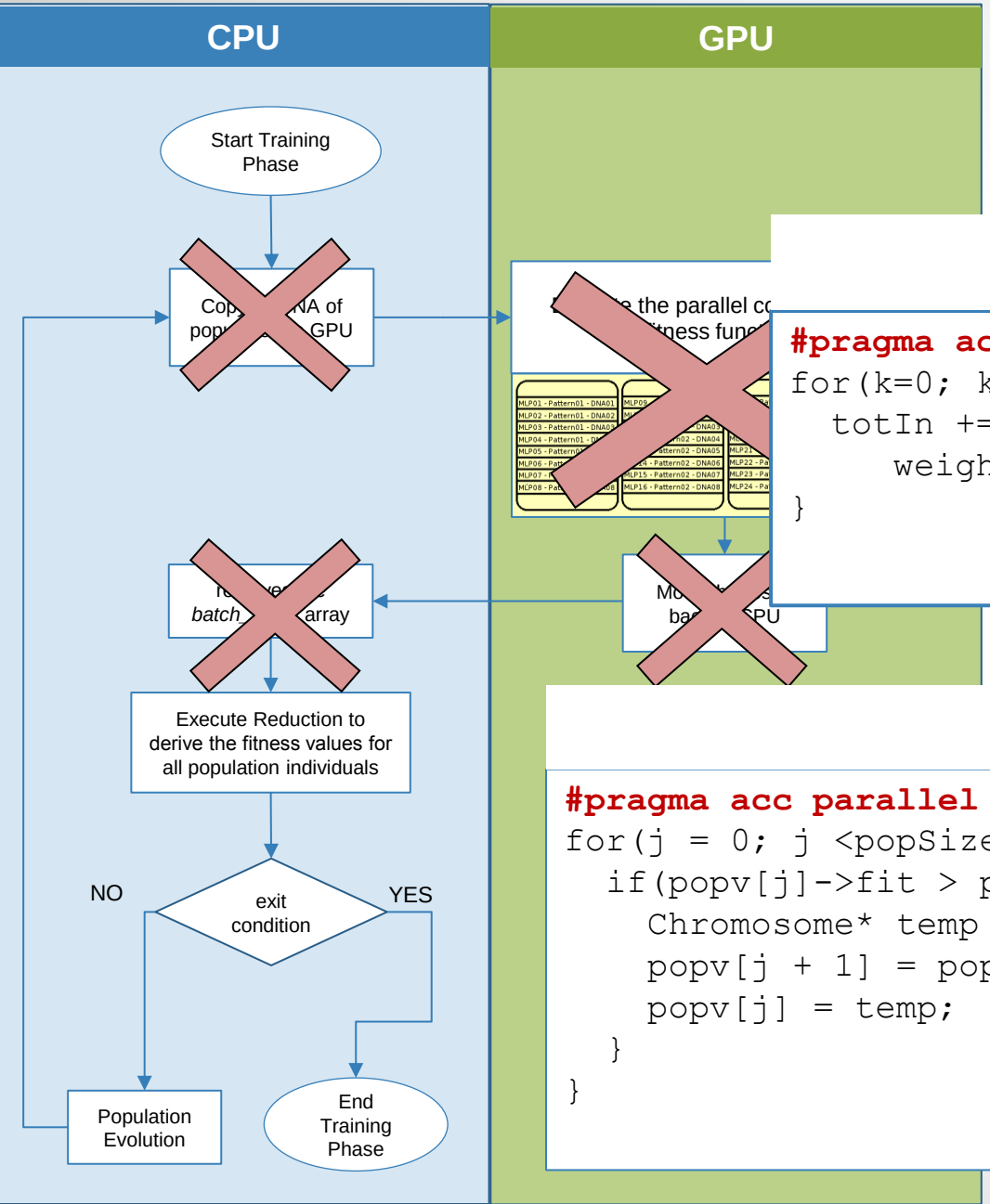
Libraries

Thrust
cuBLAS

Compiler Directives

OpenACC

MLPGA-Acc



MLP Forward function

```
#pragma acc parallel loop reduction(+:totIn)
for(k=0; k<wSize) -1; k++){
  totIn += n->layers[0]->neurons[j] -
     weights[k]*input[k];
}
```

Bubble Sort

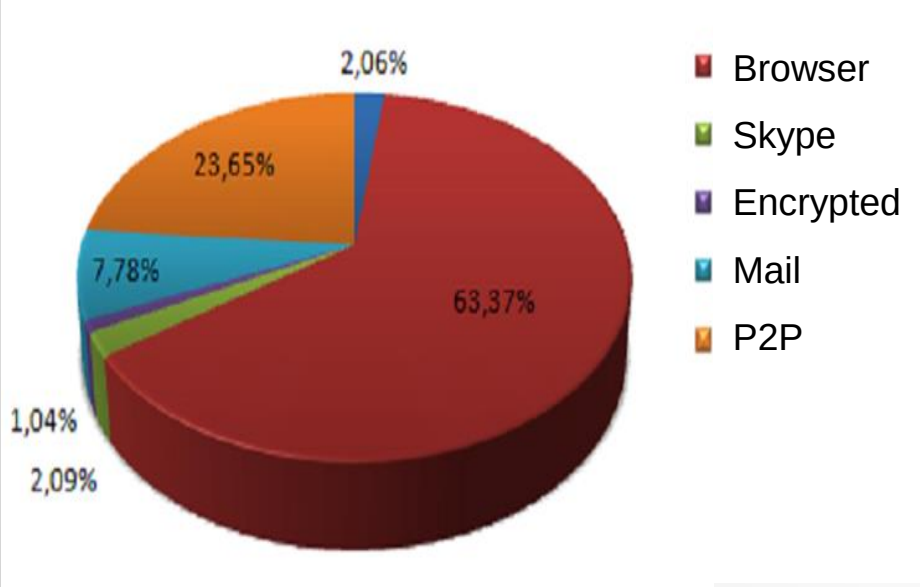
```
#pragma acc parallel private(temp) copy(popv[:popSize-1])
for(j = 0; j <popSize-1; j++) {
  if(popv[j]->fit > popv[j + 1]->fit){
    Chromosome* temp = popv[j + 1];
    popv[j + 1] = popv[j];
    popv[j] = temp;
  }
}
```

Network Traffic Classification

	Port-based	Payload inspection	Flow-based	ML-based
Method	Port number Inspection	Protocol Signature search	Header Inspection	Association trained by data
Pro	Simple	Precision	Privacy	model-data independence
Con	• IANA Standard Ports • Tunneling	• Privacy • Cryptography • New Application	Requires all flow's packets	ground truth needed

Network Traffic Classification

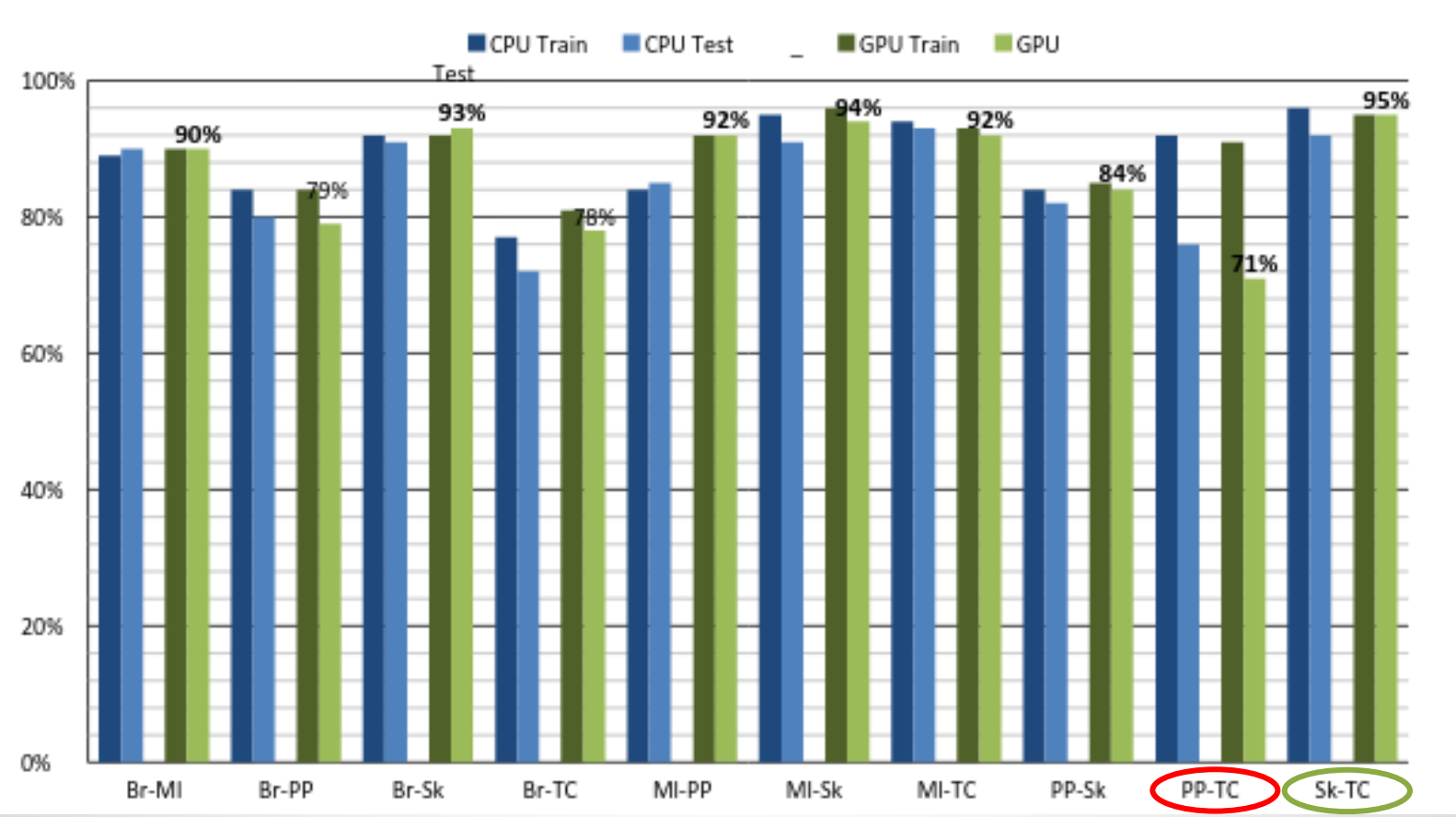
- 20251 patterns: bi-flows;
- 5 Target Classes: 17 applications grouped by class
- 4 features : Time Elapsed, Byte, UpPackets, DownPackets;



Class	Application
Browser	firefox-bin, firefox, safari, opera, firefox.exe
P2P	Bittorrent.exe, Emule, Transmission, Btdna.exe
Mail	Mail, thunderbird-bin, Thunderbird.exe
Encrypted	Proxy, ssh, ocspd
Skype	Skype, Skype.exe

MLPGA-Acc – Scientific Validation

Mean Accuracy: CPU vs GPU



Worst Accuracy: 71%

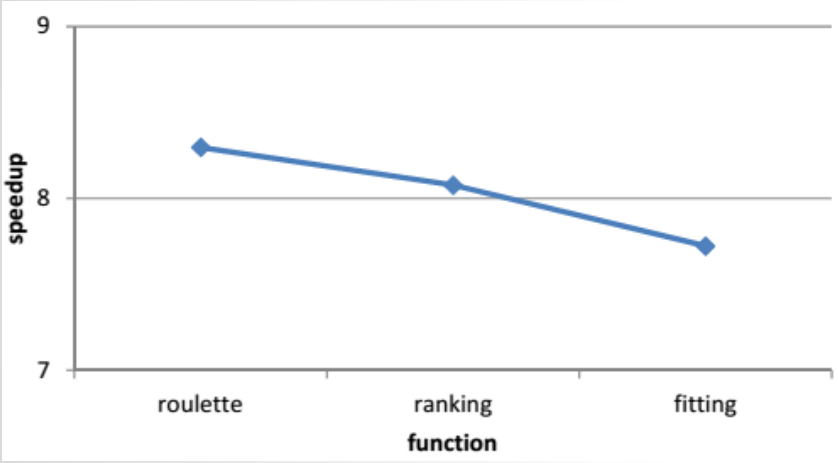
Better Accuracy: 95%

MLPGA-ACC – Performance Test

Test platform: AMD Opteron 6220 1.4Ghz 8-core NVIDIA TESLA K20c 2496 core

FMLPGA

1000 patterns and 11 features



Speedup: 7.6 - 8.3

Line of code: 1000

MLPGA-ACC

270-2107 patterns and 4 features

NETWORK TRAFFIC CLASSIFICATION											
80% train - 20% test		Speed up									
U.C.	Iter	Br-Ml	Br-PP	Br-Sk	Br-TC	Ml-PP	Ml-Sk	Ml-TC	PP-Sk	PP-TC	Sk-TC
Speed up	2500	1,4	1,6	1,6	1,9	1,5	1,4	1,9	1,6	1,7	1,8
	5000	1,6	1,5	1,5	1,6	1,4	1,5	1,5	1,6	1,6	1,6
	10000	1,5	1,3	1,5	1,7	1,5	1,5	1,5	1,6	1,6	1,5
	20000	1,6	1,5	1,5	1,6	1,3	1,4	1,5	1,5	1,5	1,5
NUM PATTERNS		1192	921	300	197	2107	619	306	392	270	328

Speedup: 1.4 - 1.9

Lines of code: 2

Reduced development time paid in terms of loss of performance

SVM

$$\min_{\alpha} \frac{1}{2}\bar{\alpha}^T \hat{Q}\,\bar{\alpha} - \bar{e}^T\bar{\alpha}$$

subject to $\bar{y}^T\bar{\alpha} = 0$

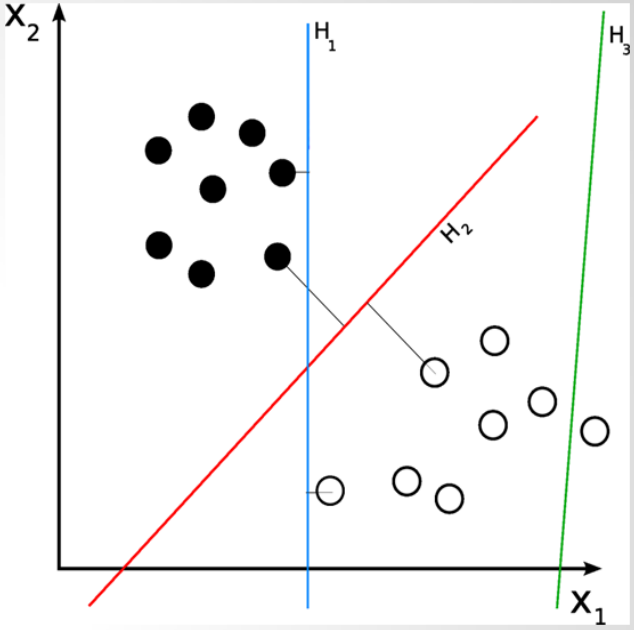
$0 \le \bar{\alpha} \le C \quad i = 1,\ldots l$

$\bar{\alpha} = [1,\ldots 1]^T$

Kernel function

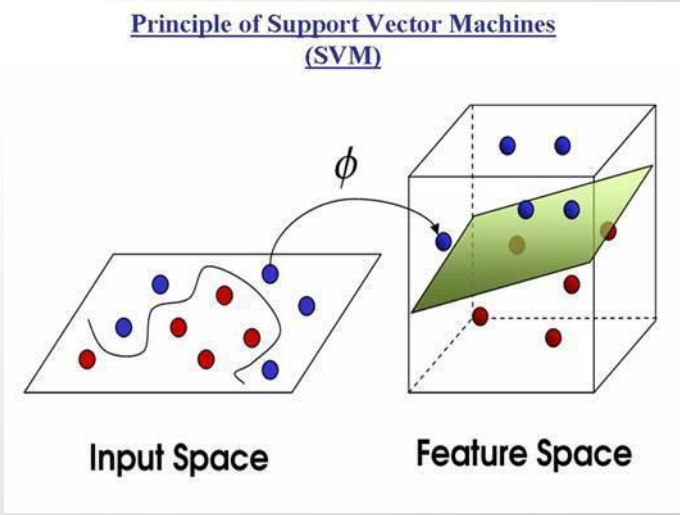
$$Q_{i,j} \equiv y_i y_j K(\bar{x}_i, \bar{x}_j)$$

$$K(\bar{x}_i, \bar{x}_j) \equiv \phi(\bar{x}_i)^T \phi(\bar{x}_j)$$



- linear: $K(x_i, x_j) = x_i^T x_j$
- polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- radial basis function (RBF): $K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2), \gamma > 0$
- sigmoid: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$

The input space can always be mapped to some higher-dimensional feature space where the training set is separable:



Accelerating with ALL

Maximum Flexibility

Programming Languages

CUDA C

"Drop-in" Acceleration

Libraries

cuBLAS

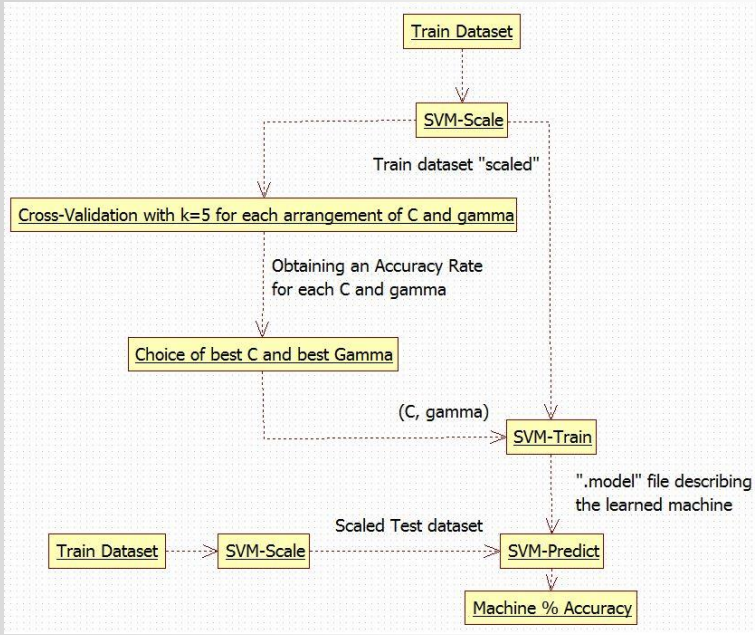
Easily Accelerate Applications

Compiler Directives

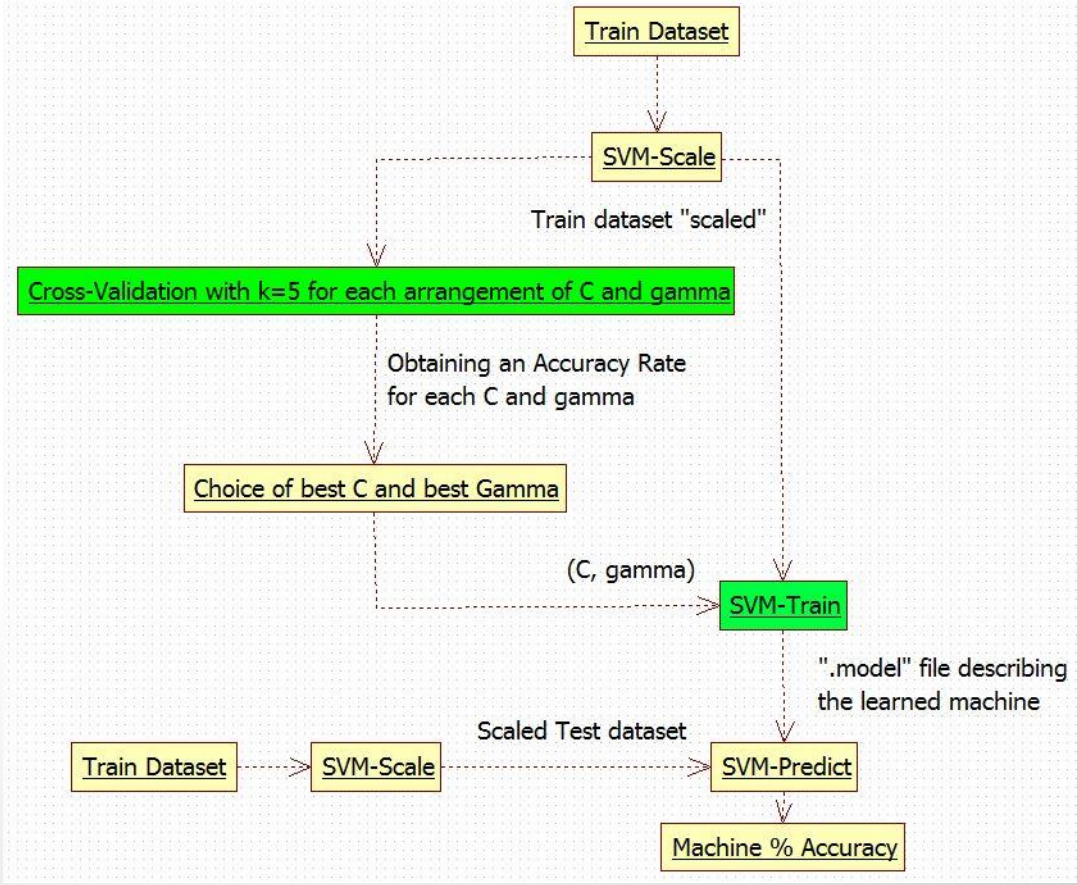
OpenACC

Implementazione Fast SVM

LIBSVM - SVM

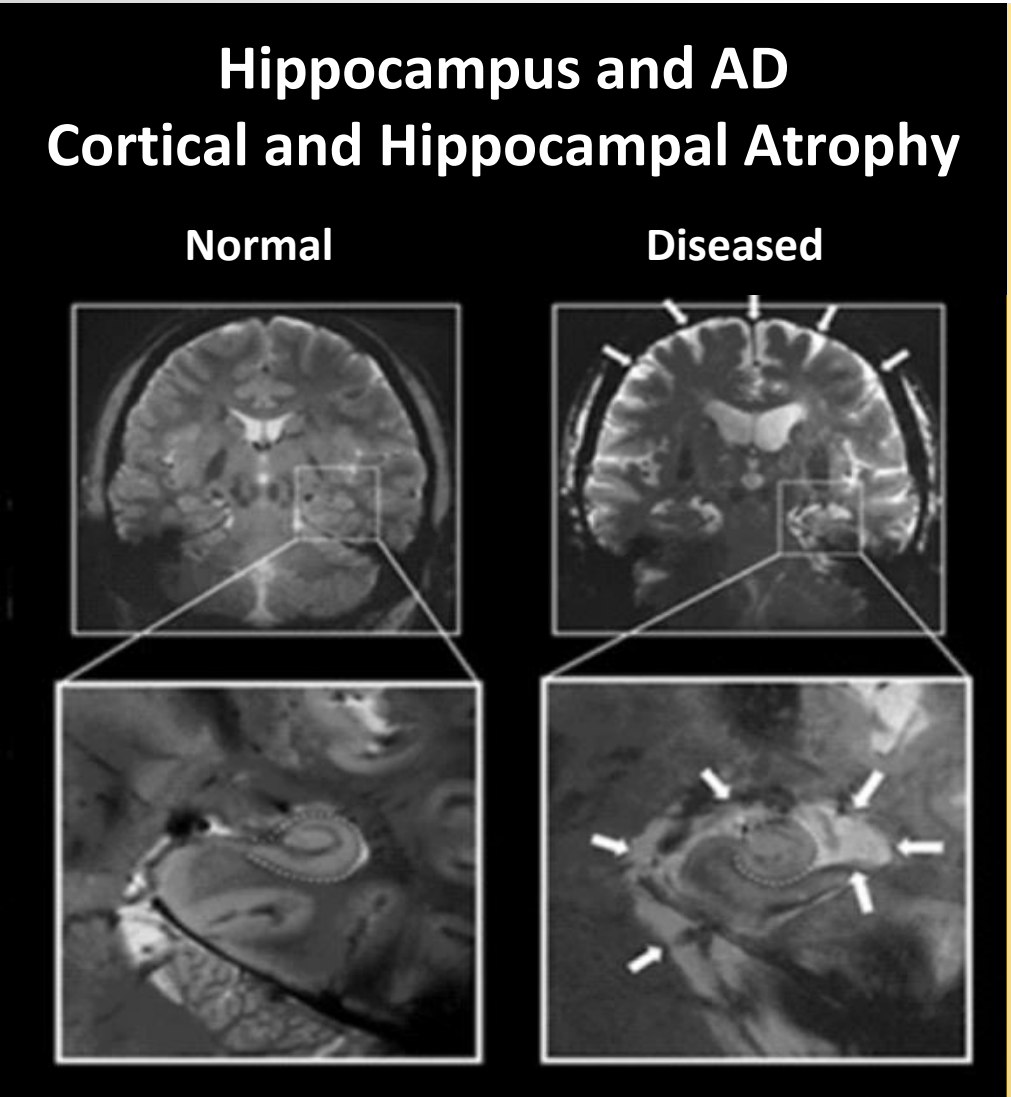


Fast SVM



Alzheimer Disease Prediction

Hippocampus volume classification through MRI of human brain



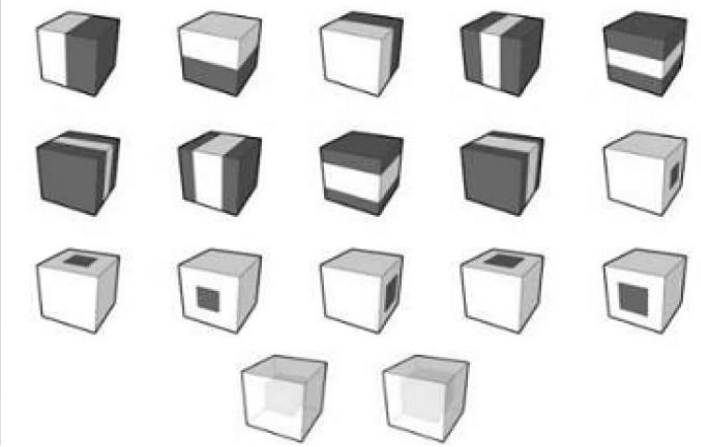
The growing volume of the hippocampus is statistically correlated to the pathology of Alzheimer's disease.

Fully define the volume of interest (VOI) from 3D MRI is important for early diagnosis.

The manual analysis is very time consuming and highly dependent on the experience of the specialist and the machinery used.

The idea is to eliminate the human intervention, delegating the classification in automated systems.

Dataset

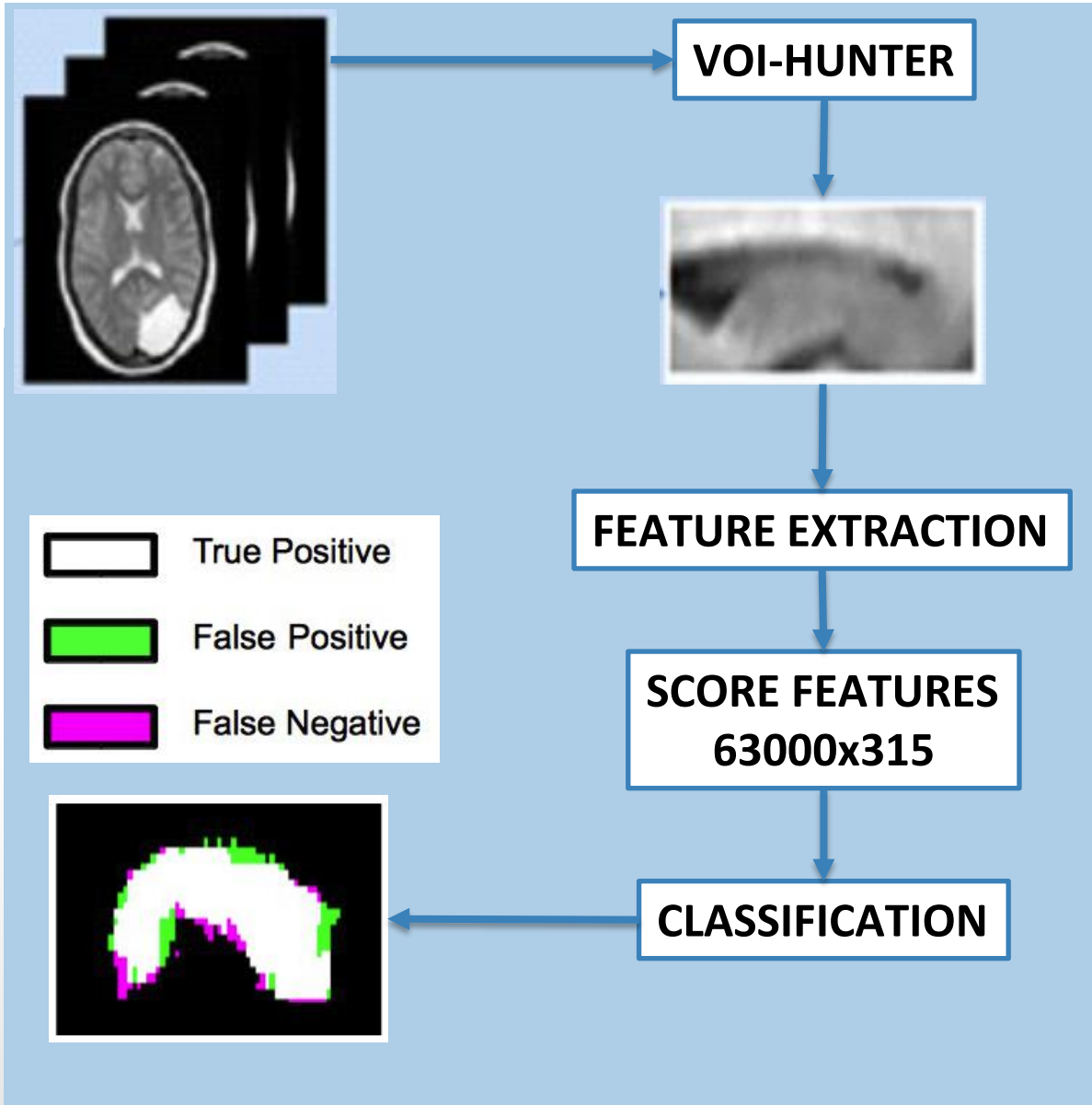


Haralik and Haar-Like Features:

1	-	position
2	-	gray level
66	-	gradients – 3x3 mask
66	-	gradients – 5x5 mask
66	-	gradients – 7x7 mask
66	-	gradients – 9x9 mask
49	-	texture 3D features

Total: 315 feature

WORKFLOW



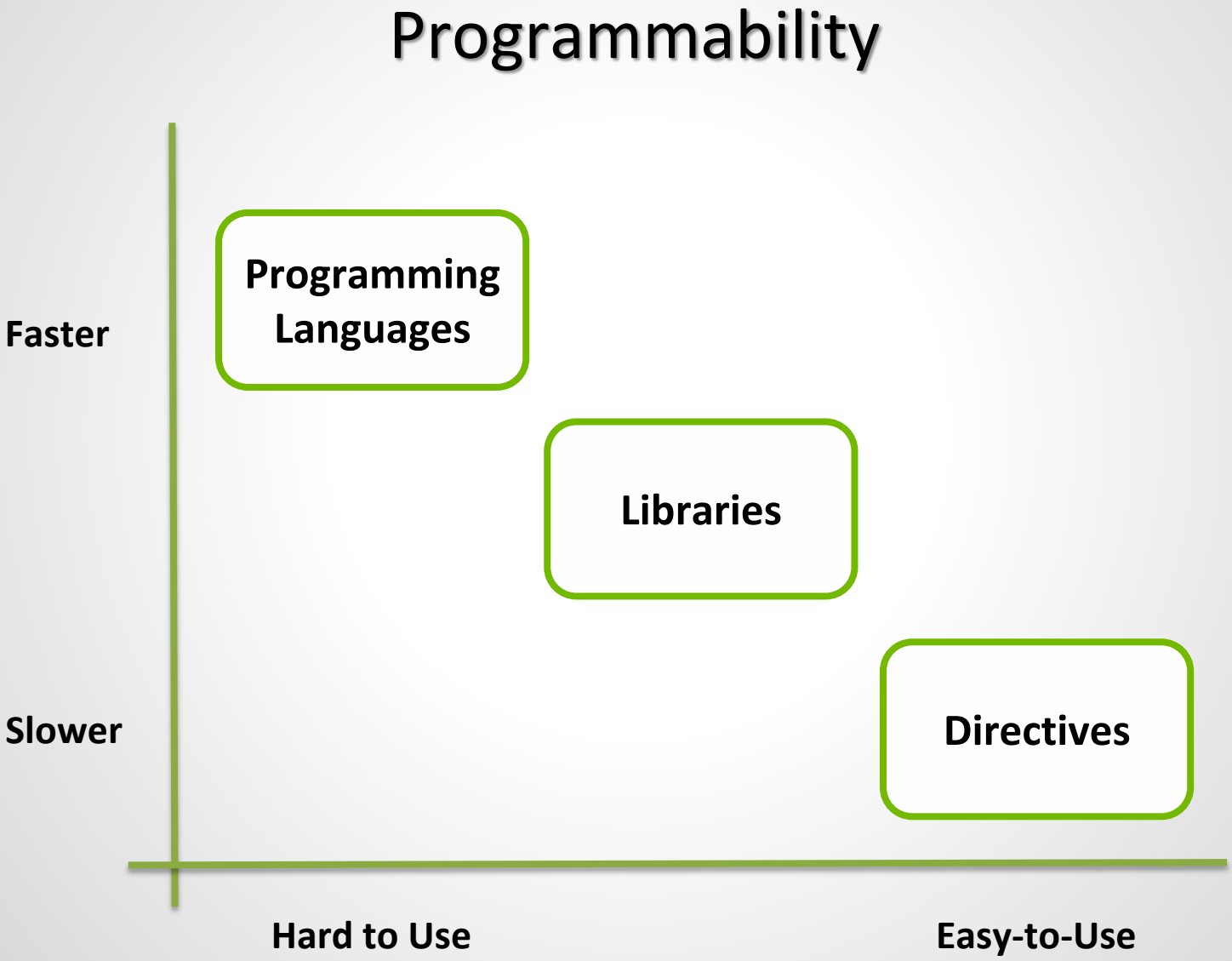
FEATURE SELECTION

315 Features	Completeness	Purity	Contamination
Positive	81,80%	77,40%	22,60%
Negative	83,50%	86,90%	13,10%
Accuracy	82,80%		

36 Features	Completeness	Purity	Contamination
Positive	79,60%	72,70%	27,30%
Negative	80,50%	85,80%	14,20%
Accuracy	80,20%		

Mean Speedup = 3.4x

Comparing Acceleration Approaches



Conclusion and Future Work

We obtained good results about redshift estimation and other issues

We aim to improve all models in Dameware exploiting the power and flexibility of the GPGPU

So, in conclusion we have not yet concluded, actually just started!