Performance studies of GooFit while estimating the global statistical significance of a new physical signal

Alexis Pompili and Adriano Di Florio
(on behalf of CMS Collaboration)

Dipartimento Interateneo di Fisica di Bari and I.N.F.N.-Sezione di Bari
via G. Amendola 173, 70126 Bari, Italy
E-mail: alexis.pompili@ba.infn.it, adriano.diflorio@ba.infn.it

Abstract. Graphical Processing Units (GPUs) represent one of the most sophisticated and versatile parallel computing architectures that are currently entering the High Energy Physics field. GooFit is an open source tool interfacing ROOT/RooFit to the CUDA platform on nVidia GPUs that acts as an interface between the MINUIT minimization algorithm and a parallel processor which allows a Probability Density Function to be evaluated in parallel.

In order to test the computing capabilities of GPUs with respect to traditional CPU cores, a high-statistics pseudo-experiment method has been implemented both in ROOT/RooFit and GooFit frameworks with the purpose of estimating the local statistical significance of an already known signal. The optimized GooFit application running on GPUs provides striking speed-up performances with respect to the RooFit application parallelized on multiple CPU workers by means of the PROOF-Lite tool.

This method is extended to situations when, dealing with an unexpected signal, a global significance must be estimated. The Look-Elsewhere-Effect is taken into account by means of a scanning technique in order to consider - within the same background-only fluctuation and everywhere in the relevant mass spectrum - any fluctuating peaking behavior with respect to the background model. The execution time of the fitting procedure for each MC toy can considerably increase, thus the RooFit-based approach gets so time-expensive that may become unreliable while GooFit is an excellent tool to carry reliably out this p-value estimation method.

1. Introduction to GooFit

The word GPU-accelerated computing refers to an enhancement of application performances that can be obtained by offloading compute-intensive portions to the GPU, while the remaining code still runs on the CPUs. The computing capabilities are enhanced once a sequence of elementary arithmetic operations are performed in parallel on a huge amount of data. In the context of High Energy Physics (HEP) analysis applications, GooFit is an under development open source data analysis tool that is being optimized to allow the HEP analyst to perform complex fit tasks on GPUs or multithreaded CPU backends. Its description and details can be found elsewhere [1]; it has been very recently updated to a version 2.0 [2] that is characterized by a set of new features. However the results shown here have been obtained by using the previous main version. GooFit is used in applications for parameter estimation, that interfaces ROOT [3]/RooFit [4] to the CUDA [5] parallel computing platform on nVidia’s GPUs (it also supports OpenMP). Parameter estimation and interpolation are a crucial part of many physics analyses. The evaluation of Probability Density Function (PDF) on large datasets is
usually the bottleneck in the MINUIT [6] algorithm. GooFit acts as an interface between the MINUIT minimization algorithm and a parallel processor which allows a PDF to be evaluated in parallel. Specifically the FitManager object [1] forms the interface between MINUIT (running on CPU) and a GPU which enables the evaluation in parallel of the PDF representing the physical model (GooPdf object). Fit parameters are estimated at each negative-log-likelihood (NLL) minimization step on the host side (CPU) while the PDF/NLL is evaluated on the device side (GPU) [7] until fit convergence and consequently a cyclical intense memory transfer takes place between CPU and GPU. The FitControl object [1] allows to switch between Maximum Likelihood (ML) and $\chi^2$ fits, either unbinned or binned.

Preliminary tests performed with an unbinned ML fit, either by using a single CPU and by using an additional GPU, indicate amazing speed-up values of several hundreds when the involved number of events is greater than few thousands and the considered task is enough compute-intensive (a convolution of a Breit-Wigner and a Gaussian resolution function was considered in our tests). As expected, for a binned ML fit, the speed-up ranges from few units to few dozens (increasing with the number of bins). Applications using, even recursively, a series of several fits with complicated PDFs, can evidently take advantage of the GPU acceleration by using GooFit. Once implemented within GooFit, Monte Carlo pseudo-experiments represent a good example of an application with these characteristics, as discussed in the next section.

2. GooFit performances for Monte Carlo toys

Monte Carlo pseudo-experiments (MC toys) are used to estimate the probability ($p$-value) that background fluctuations would - alone - give rise to a signal as much significant as that seen in the data. To test the computing capabilities of GPUs with respect to CPU cores, a high-statistics MC toys technique was implemented both in ROOT/RooFit and GooFit frameworks [8] with the aim to estimate a $p$-value and specifically the local statistical significance of the structure observed by CMS close to the kinematical threshold of the $J/\psi\phi$ invariant mass in the $B^+ \rightarrow J/\psi\phi K^+$ decay [9].

The used hardware setup consists in two servers, one equipped with two nVidia TeslaK20 and 32 cores (16 + 16 by Hyper-Threading) and the other with one nVidia TeslaK40 and 40 (20 + 20) cores [10]. To efficiently run RooFit MC toys on the 72 CPUs available on the two servers hosting the GPUs, the PROOF-Lite [11] tool is used. On the other hand the nVidia Multi Process Service tool [12] allows the execution of - up to 16 - simultaneous processes on the same GPU acting as a scheduler and allowing a balanced full usage of the GPU. The optimized

![Figure 1. Comparison for the elapsed time employed with two TeslaK20 and one TeslaK40 together as a function of the number of MC toys; GooFit/MPS runs 48 concurrent processes while RooFit/PROOF-Lite runs on 72 CPUs. For 1M toys the red diamond point shows the extrapolated time (about 11days) for the RooFit application.](image-url)
**GooFit** application running on GPUs has provided [8] striking speed-up performances with respect to the RooFit application parallelized on multiple CPUs by means of PROOF-Lite tool. In particular, from the point of view of the end-user analyst having at its own disposal all the 72 CPU cores and the three GPUs, it has been measured that 1M of MC toys can be produced in about 11 days with RooFit/PROOF-Lite and in about 6 hours only with GooFit/MPS (Fig.1).

The extension of this method when a new unexpected signal is reconstructed will be presented in the next section.

### 3. Clustering approach to address the LEE

When a new unexpected signal is reconstructed in HEP, the *global* significance of the associated peak needs to be estimated and the Look Elsewhere Effect [13] must be taken into account. This implies to consider, within the same background-only fluctuation and everywhere in the relevant mass spectrum, any random peaking behaviour with respect to the expected background model. Thus a scanning technique based on a clustering approach has been developed.

Beforehand a pseudo-data invariant mass distribution of 15K candidates in a generic region of interest, namely [1,18]GeV, has been generated according to a fictitious 7th order polynomial background model on the top of which any desired amount of a *significant* signal, mimicked by a Voigtian model, can be artificially added close to 8GeV (as for instance in Fig.2). At this mass value a 60MeV mass resolution is considered. The fits to the pseudo-data distribution of Fig.2 are performed accordingly: the background-only model (the *Null Hypothesis* $H_0$) is a 7th order polynomial function whereas the background+signal model (the *Alternative Hypothesis* $H_1$) is obtained by adding a Voigtian function. The resolution values in the latter are reasonably increased as a function of the increasing invariant mass while satisfying the 60MeV constraint at 8GeV. By performing the $H_0$ and $H_1$ fits, the (local) statistical significance of this peak is $Z \sigma = 5.5 \sigma$ with $Z$ approximately estimated by:

$$Z \approx \sqrt{-2[\ln(L_{H_1}) - \ln(L_{H_0})]}$$

(1)

where $L_{H_0}$ ($L_{H_1}$) is the likelihood evaluated for $H_0$ ($H_1$) hypothesis [14].

The MC toys method is configured as follows. As first step of each toy iteration, a distribution based on the background-only model is generated over the whole mass spectrum and the $H_0$ fit is performed. As a second step the clustering technique acts on each generated pseudo-experiment as follows:

(i) search for a *seed* bin, namely for a bin whose content fluctuates more than $x \sigma$ strictly above the value of the background function in the center of that bin ($\sigma$ is the statistical error associated to the considered bin).

(ii) Add any *side* bin to the *seed* bin if it holds a content that fluctuates more than $z \sigma$ strictly above the value of the background function in the center of that bin, otherwise the *seed* bin forms a 1-bin cluster.

(iii) Check also for *light seeds*, namely bins that fluctuate more than $y \sigma$ with $z < y < x$ and with at least a *side* bin fluctuating more than $z \sigma$. In case of positive result a cluster is formed.

In the third step, a series of independent $H_1$ fits is performed by cycling on the clusters collected in the clustering step. At the end of this step the fit with the best $\Delta NLL$ (the test statistic) is chosen. On the whole a $\Delta NLL$ distribution is obtained over all the processed MC toys.

A set of *baseline* clustering parameters ($x, y, z$) = (2.25, 1.50, 1.00) has been chosen in order to satisfy two concurrent requirements: not missing any possible interesting fluctuation and avoiding selecting too many irrelevant fluctuations. This *baseline* configuration has been run on about 76M pseudo-experiments and the $\Delta NLL$ distribution is shown in Fig.3, with the superimposed red line indicating the $\Delta NLL_{data}$ value for the pseudo-data.
Table 1. Mean number of alternative hypothesis fits per toy \(<fit_{H_1}>) and fraction of toys with no fit \(f_{no\text{fit}}\) for the three different clustering configurations described in the text.

| Clustering configs. | \(<fit_{H_1}>\) | \(f_{no\text{fit}}\) |
|---------------------|----------------|----------------|
| Tight (3.00, 1.75, 1.00) | 2.2 | \(~10\%\) |
| Baseline (2.25,1.50, 1.00) | 4.5 | \(~1\%\) |
| Loose (2.00, 1.25, 1.00) | 6.6 | 0.1% |

The global \(p\)-value is then estimated by:

\[
p = \int_{\Delta NLL_{data}}^{\infty} f(\Delta NLL)d(\Delta NLL) \simeq \frac{9.820 \cdot 10^2}{7.584 \cdot 10^7} \simeq 1.295 \cdot 10^{-5}
\]

This corresponds to the global statistical significance \(Z\sigma = \Phi^{-1}(1 - p)\sigma \simeq 4.22\sigma\), through the inverse function of the cumulative distribution of the standard Gaussian. As expected by considering the LEE, the global significance is relevantly lower than the estimated local one.

4. Evaluation of the possible systematic uncertainty

In order to test the behavior of the method and to estimate the possible systematic uncertainty associated to the clustering technique, three sets of configuration parameters, i.e. three values for the \((x, y, z)\) parameters, have been carefully considered. After some tests with different cuts two further configurations are chosen besides the baseline clustering cuts: a set of tighter values (3.00, 1.75, 1.00) and a set of looser values (2.00, 1.25, 1.00). The Tab.1 reports details about these three clustering configurations such as the average number of \(H_1\) fits per toy and the fraction of toys with no fit.

These three configurations have been run on a same common set of 45M fluctuations and the three corresponding \(\Delta NLL\) distributions are shown superimposed in Fig.4.

**Figure 2.** Simulated invariant mass distribution (pseudo-data). \(H_0(H_1)\) fit is in red (blue); in the top right box the best values for the estimated parameters of the \(H_1\) model are given.

**Figure 3.** \(\Delta NLL\) distribution for about 76M toys for the baseline configuration of clustering technique. The red line indicates the \(\Delta NLL_{data} \simeq 30.27\) value for the pseudo-data distribution in Fig.2.
Table 2. Estimated global significances for the 3 clustering configurations with respect to different local significance values estimated by Eq.(1).

| Local Significance   | 4.0σ | 4.5σ | 5.0σ | 5.5σ | 6.0σ |
|----------------------|------|------|------|------|------|
| Tight (3.00, 1.75, 1.00) | 2.21 | 2.91 | 3.58 | 4.22 | 4.87 |
| Baseline (2.25,1.50, 1.00) | 2.20 | 2.91 | 3.58 | 4.22 | 4.87 |
| Loose (2.00, 1.25, 1.00)   | 2.20 | 2.91 | 3.58 | 4.22 | 4.87 |

By focusing on the region of interest for the estimation of the statistical significance, i.e. the tail of the $\Delta NLL$ distribution ($\Delta NLL > 20$), it is evident that there is no relevant difference (Fig.5) among the three configurations. This can be appreciated by inspecting, in Figg.4, 5, the normalized deviations $(x - y)/(x + y)$ of the other two distributions with respect to the baseline distribution. This is also confirmed by examining the estimated global significances for the p-values corresponding to different values of local significances, as reported in Tab.2. It can be concluded that the systematic uncertainty on the p-values associated to the method is negligible.

5. Conclusions
A high-statistics pseudo-experiment method, based on a scanning and clustering approach, has been implemented and tested within GooFit framework with the purpose to estimate the global statistical significance of an unexpected new signal. The presented results clearly indicate that the systematic uncertainty associated to the method is negligible and that the p-value estimation is not affected by the clustering configuration. This kind of validation studies has been performed by exploiting the high performance of an optimized GooFit application running on GPU-equipped servers.

Figure 4. $\Delta NLL$ distributions for 45M of common fluctuations for the 3 configurations: baseline (black), tight (red) and loose (blue).

Figure 5. The same $\Delta NLL$ distributions of Fig.4 once zoomed into the range 20.0-45.0 to inspect their tail behaviour.
Aknowledgements
We are grateful for valuable support from G. Donvito, manager of the High Performance Cluster hosted by the ReCas [15] Data Center.

References
[1] Andreassen R, Meadows B T, de Silva M and Sokoloff M D J. Phys.: Conf. Series 513, 052003 (2014)
[2] Schreiner H F et al. arXiv:1710.08826 (2017)
[3] Brun R and Rademakers F Nucl. Instrum. Meth. A 389, 81-86 (1997)
[4] Verkerke W and Kirkby D P eConf C0303241 MOLT007 (Preprint physics/0306116) (2003)
[5] https://docs.nvidia.com/; for these studies CUDA version 6.5(7.0) were used for Tesla K20(40) boards.
[6] James F and Roos M CERN Program Library routine D506 (1989) (long write-up)
[7] This can be checked by nVidia Visual Profiler (nvvp)).
[8] Pompili A and Di Florio A, J. Phys. Conf. Ser. 762, 012044 (2016) [ACAT 2016]
[9] CMS Collaboration, Phys. Lett. B 734, 261 (2014)
[10] Tesla K20(K40) has 5(12)GB GDDRs; the 16(20) cores are E5-2640 v2 @ 2.0GHz with 64(256)GB of RAM
[11] https://root.cern.ch/proo; see for specific PROOF-Lite applications on multi-core servers: Barbone L, Donvito G and Pompili A, J. Phys.: Conf. Series 396 042017 (2012)
[12] https://docs.nvidia.com/deploy/pdf/CUDA_Multi_Process_Service_Overview.pdf
[13] Gross E and Vitells O Eur. Phys. J. C70, 552 (2010)
[14] I. Narsky and F. C. Porter, Statistical Analysis Techniques in Particle Physics: Fits, Density Estimation and Supervised Learning, Wiley, 2013.
[15] ReCas is a project financed by the italian MIUR (PONa3_00052, Avviso 254/Ric.); its web page is http://www.recas-bari.it/index.php/en/.