RANDOM SAMPLING FOR ESTIMATING THE PERFORMANCE OF FAST SUMMATIONS

Balaji Vasan Srinivasan, Ramani Duraiswami

Department of Computer Science, University of Maryland, College Park, MD, USA.
[balajiv,ramani]@umiacs.umd.edu

ABSTRACT

Summation of functions of N source points evaluated at M target points occurs commonly in many applications. To scale these approaches for large datasets, many fast algorithms have been proposed. In this technical report, we propose a Chernoff bound based efficient approach to test the performance of a fast summation algorithms providing a probabilistic accuracy. We further validate and use our approach in separate comparisons.

1. INTRODUCTION

There are a number of applications where summation of source kernels at a number of target points need to evaluated. These computations have quadratic complexity (O(N^2)) thus hindering its scalability to large datasets. Fast algorithms and parallelization address this issue, but however, they come with a error bound. However to test these error bounds for larger data, it is not possible to evaluate the direct approach for larger datasets. To overcome this, we propose a random sampling approach using which evaluate the error only at K evaluation points. We derive a bound to the sample size K from Chernoff bounds for a desired accuracy.

In section 2 we show the Chernoff bound for a random variable X with mean μ. In section 3 we derive a bound for sampling by adapting it to evaluate the performance of a summation algorithms. In section 4, we validate the proposed approach with a popular summation algorithm, Improved Fast Gauss Transform and then extend it to a parallel algorithm for Gaussian summation. We conclude the paper in section 5

2. CHERNOFF BOUNDS

Chernoff bound gives the upper tail bound ($Pr\left[X \geq \mu(1+\delta)\right]$) and lower tail bound ($Pr\left[X \leq \mu(1-\delta)\right]$). The upper tail bound is given by,

$$\begin{aligned} Pr\left[X \geq \mu(1+\delta)\right] &= \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\mu} \\ &\leq e^{-\mu\delta^2/3} \end{aligned} \tag{1}$$

Similarly, the lower tail bound can be given by,

$$Pr\left[X \leq \mu(1-\delta)\right] \leq e^{-\mu\delta^2/2}. \tag{2}$$

Eqs 1 and 2 gives the Chernoff bound.

3. SAMPLING PROBLEM

The goal in sampling is to select a subset of the original data at random. Let us analyze the property of the resulting subset, let X_i be the random variable corresponding to the i^{th} sample of the subset such that, X_i is 1 if a desired property is satisfied, 0 otherwise. If K is the size of the subset, then it is desired that,

$$Pr\left[\left|\frac{\sum_i X_i}{K} - \frac{M}{N}\right| \geq \epsilon\right] \leq \eta, \tag{3}$$

where M is the number of datapoints satisfying the desired property in the original dataset. Let us denote $\frac{M}{N}$ as p and $\sum_i X_i$ as Y, thus Eq. 3 can be rewritten as,

$$\begin{aligned} Pr\left[\left|\frac{Y}{K} - p\right| \geq \epsilon\right] &= Pr\left[|Y - Kp| \geq K\epsilon\right] \\ &= Pr\left[Y \geq Kp + K\epsilon\right] \\ &\quad + Pr\left[Y \leq Kp - K\epsilon\right] \end{aligned} \tag{4}$$

To summarize, we want to select K sized subset from a large data, such that, if M points of the N total points in the full data have a certain property, the property is preserved by a certain number L points in the subset, such that $\frac{L}{K} = \frac{M}{N}$ with high probability given by Eq. 4.

3.1. Adaptation

Before applying the Chernoff bound here, we adapt this to the problem of testing a summation algorithm. The summation algorithm would give the sum at evaluation points in an efficient fashion. We want to test the accuracy of the fast algorithm. So we shall sample evaluation points at random K points and would evaluate the sum directly at these points. We would then check the accuracy with respect to the fast algorithm to be tested. We expect the error evaluated at all K points to be below a certain threshold.

Let us assume that the fast algorithm assures an error bound of ς. Let us define the property that we shall look for in the data as the error between the direct and fast approach $\leq \varsigma$. Because the algorithm assures such an error bound, the p in Eq. 4 is 1. The expected number of the points in the subset that will hold the error property is K, thus $\mu = K$. Applying Chernoff bound and substituting $p = 1$ in Eq. 4,

$$\begin{aligned} Pr\left[Y \geq K(1+\epsilon)\right] + \quad Pr &\quad \left[Y \leq K(1-\epsilon)\right] \\ \leq &\quad 0 + e^{K\epsilon^2/2} \\ \leq &\quad e^{K\epsilon^2/2} \leq \delta \\ \Rightarrow K \quad \geq &\quad \frac{2}{\epsilon^2}\log\left(\frac{1}{\delta}\right) \end{aligned} \tag{5}$$

Thus, setting the parameters ϵ,δ and ς, we can choose K points uniformly at random from the original data set, evaluate the sum directly and test for the desired error bound. If all the points satisfy

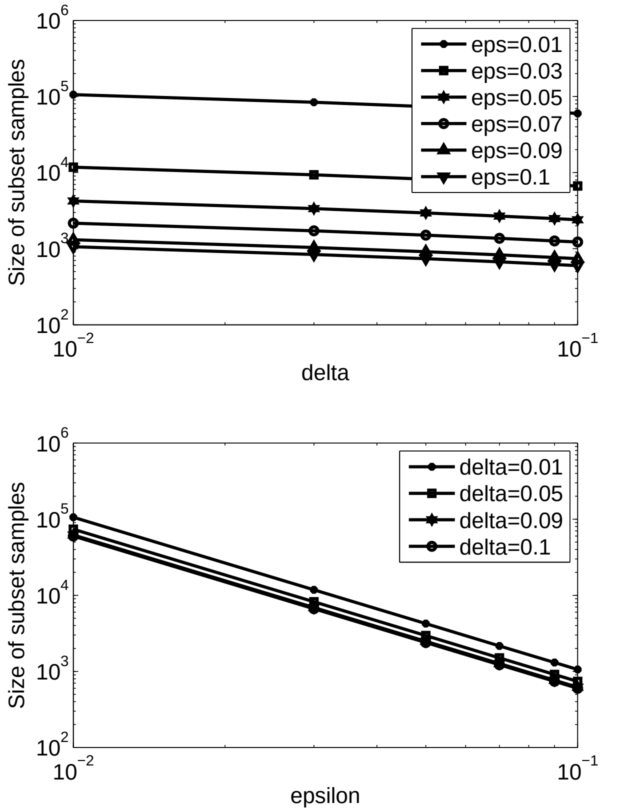

Fig. 1. *Variation of the size of the sample set with ϵ and δ*

the required error bound ς, algorithm can be declared accurate within confidence interval ϵ and probability $1 - \delta$.

Figure 1 and Table 1 show the size of sample set for various values of ϵ and δ. It can be seen that for guaranteeing that at least 90% of the computations by the fast algorithms lies within the declared error bound ς with a confidence of 90%, we need to check at least 600 samples.

4. EXPERIMENTS

In this section we first test the proposed algorithm with a popular Gaussian summation algorithm, Improved Fast Gauss Transform (IFGT) [1]. We further extend our approach to provide a probabilistic bound for a GPU-accelerated algorithm.

IFGT: IFGT is a popular algorithm which provides a linear acceleration for Gaussian summations of the form,

$$f_j = \sum_{i=1}^{N} q_i \exp\left(-\frac{\|x_i - y_j\|^2}{h^2}\right) \tag{6}$$

It is evident that the direct summation takes O(N^2). The IFGT algorithm provides a loose error bound. We evaluated the summation of $100,000$ Gaussian kernels at $100,000$ points. The supplied a required error bound of 10^{-6}. We evaluated the direct sum at $2,952$ points for $\epsilon = 0.05$ and $\delta = 0.05$. We observed that the error of the Gaussian summation evaluated at the randomly chosen sample points was less than 10^{-8}. Thus according to our algorithm, the error bound is 10^{-8} for 95% of the samples with a high probability of 0.95. This is in conjunction with the results in [1], thus validating our approach.

ϵ	δ	Size of sample
0.01	0.01	105967
0.01	0.02	92104
0.01	0.05	73778
0.01	0.1	59915
0.02	0.01	26492
0.02	0.02	23026
0.02	0.05	18445
0.02	0.1	14979
0.05	0.01	4239
0.05	0.02	3685
0.05	0.05	2952
0.05	0.1	2397
0.1	0.01	1060
0.1	0.02	922
0.1	0.05	738
0.1	0.1	600

Table 1. Value of the sample set size, for various values ϵ and δ

Graphics processor based algorithm: Graphics processing units (GPU) are highly capable set of data-parallel processors and are evolving in to highly capable compute coprocessors. Many algorithms have been accelerated using this latest trend. In a graphics processor, floating point operations are cheaper than double precision operations, hence most of the algorithms on a GPU use floating point operations. However, this results in lesser accuracy than the corresponding CPU versions, which use double precision operations.

In this experiment, we evaluate the accuracy of a GPU-based floating point algorithm, by comparing it with the corresponding direct-double-precision version. We evaluated the sum of $10,000$ Gaussian kernels for $10,000$ points. We used our approach to provide a bound for the relative absolute error. The faster GPU version was used to evaluate the summation in a parallel fashion. The direct CPU approach was used at $2,952$ points and the relative absolute error was evaluated at these points. It was found that all the error were less than 10^{-5}. Our approach would result in the claim that at least 95% of the samples have a relative error $\leq 10^{-5}$ with probability 0.95. To further test our claim, we evaluated the relative error at all points and found that in fact 100% of the samples have a relative error $\leq 10^{-5}$.

5. CONCLUSION

In this report, we have explored the use of random sampling with the aid of Chernoff bound to come up with a size of the subset which can guarantee a desired property with a confidence and probability, that can be used to test the performance of a fast summation algorithm. Although the size of K for a reasonable ϵ and δ might be large that it cannot be used with smaller data sizes, it will be handy for very large data sizes. Our approach was further validated to provide bounds for the performance of Improved fast Gauss transform [1] and a GPU based Gauss transform.

6. REFERENCES

[1] V. C. Raykar and R. Duraiswami, “The improved fast Gauss transform with applications to machine learning,” in *Large Scale Kernel Machines*, L. Bottou et al., Eds., MIT Press, 175-201, 2007.