

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

PROJECT N°2 REPORT

RANDOMIZED NYSTRÖM

Authors:

Tara FJELLMAN

Amal SEDDAS

Professor:

Laura GRIGORI

The EPFL logo is rendered in a bold, red, sans-serif typeface. The letters are thick and blocky, with the 'E' and 'F' featuring a distinctive stepped or 'staircase' design on their right-hand sides.

Contents

1	Introduction	1
2	Algorithms	1
2.1	Sketching Matrices	1
2.2	Randomized Nyström low rank approximation	1
3	Stability analysis	1
4	Parallelisation	1
5	Experimental procedure	1
6	Algorithm Performance	2
6.1	Sequential Performance	2
6.2	Parallel Performance	3
7	Conclusion	4

1 Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2 Algorithms

2.1 Sketching Matrices

2.2 Randomized Nyström low rank approximation

3 Stability analysis

- bullet points about comments to include

4 Parallelisation

5 Experimental procedure

- averaging to increase precision of results - when running in parallel, was limited by number of cores $p = 2^{2s}$ with $s \in \mathbb{N}$ (due to perfect square constraint of the parallelised matrix multiplication and the SHRT algorithm using the Hadamard transform), so we ran results for 1,4,16,64 - ran results on the Helvetios cluster - k was taken often around $n/20$ as we deemed this value relevant for real applications - n was made vary as powers of 2 due to the Hadamard transformation while l was taken logarithmically spaced just to make the visuals more easy to interpret

6 Algorithm Performance

6.1 Sequential Performance

- in general (both plots) observe that k rank approximation represents minority of runtime (as expected as of order $k^3 + nk^2$). Indeed for shown examples represents at most 10%. This would be different if one were to take big values of k . Also does not really seem to depend on sketching method which makes sense given the computations are the same regardless of which sketching method was used to obtain the B and C matrices

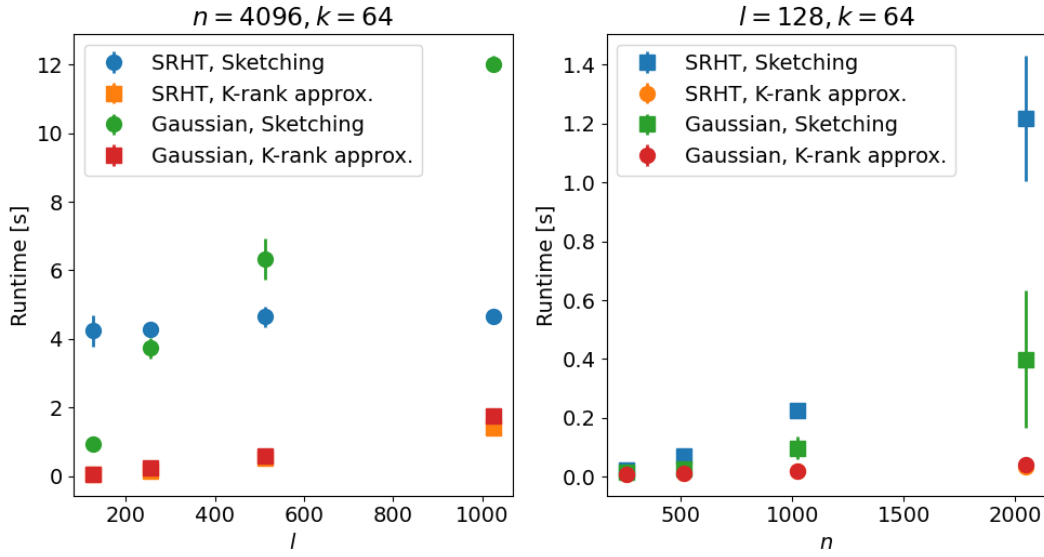


Figure 1: Runtimes associated to the tested sketching methods : as a function of l (left) and as a function of n (right). Runtimes are broken down in that associated to the computation of $A\Omega, \Omega^T A\Omega$ and that associated to the computation of the k rank approximation.

- different behaviours as a function of l and n : as a function of l SRHT looks almost flat, having really slow increase, while gaussian increases steadily given its complexity is of order $nl^2 + ln^2$. This gives advantage for SRHT if we are embedding "dense spaces" as it scales much better than gaussian as function of l . For n : both algos show significant increase, but SRHT shows faster runtime growth. This makes sense due to extra $\log_2(n)$ term in its complexity. This would give advantage to gaussian sketching if we are dealing with sparse information within really high dimensional spaces. -

6.2 Parallel Performance

- selected $l = n/64$ given we are using TSQR for the QR factorisation and we need the matrix to be tall-skinny to actually observe a speed-up. To have a good speed-up for a bigger l , while keeping numerical stability for ill-conditioned matrices another algo should be used.

- for small example : some speed-up is observed as a function of number of cores but communication costs are non neglectable and even dominate the total runtime for $p=64$. This is expected as the computation is rather small to be run in parallel. The runtimes for the k-rank are the most affected as these represented the smallest part of the computation to start with. for the specific values selected the gaussian sketching is faster (indeed l is taken only as a small fraction of n).

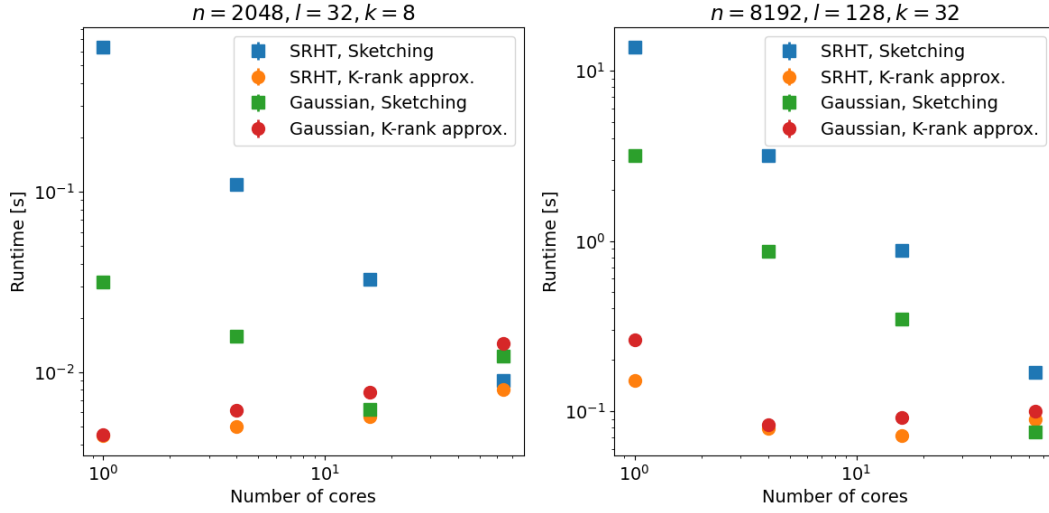


Figure 2: Runtimes associated to the tested sketching methods as a function of the number of cores : for a “small” example (left) and a “large” one (right). Runtimes are broken down in that associated to the computation of $A\Omega, \Omega^T A\Omega$ and that associated to the computation of the k rank approximation.

- for the large : communication is not much of an issue, and big speed ups (quasi linear) are observed for the sketching computation. This means the parallelization is successful and scales well for big computations. We can see the speedup is not as good for k-rank partly because the computation was smaller to start with. The fact that the matrix is not that tall-skinny is also expected to have contributed to the low speed-up of that part. -

7 Conclusion

Aknowledegments

References