Sampling Techniques for Monte Carlo Matrix Multiplication with Applications to Image Processing

Humberto Madrid[1], Valia Guerra[2], and Marielba Rojas[3]

[1] Center for Applied Mathematics Research (CIMA), Saltillo, Mexico
hmadrid@gmail.com
[2] Institute of Cybernetics, Mathematics and Physics (ICIMAF), Havana, Cuba
vguerra@icimaf.cu
[3] Delft University of Technology, The Netherlands
marielba.rojas@tudelft.nl

Abstract. Randomized algorithms for processing massive data sets have shown to be a promising alternative to deterministic techniques. Sampling strategies are an essential aspect of randomized algorithms for matrix computations. In this work, we show that strategies that are effective or even optimal in the general case, can fail when applied to ill-conditioned matrices. Our experimental study suggests that there exists a relationship between sampling performance and conditioning of the matrices involved. We present an explanation for this behavior and propose a novel, efficient, and accurate sampling strategy for randomized multiplication of affinity matrices in image segmentation.

Keywords: randomized algorithms, massive data sets, image segmentation, Normalized Cuts.

1 Introduction

Unprecedented features and challenges inherent to modern massive data sets have rendered most traditional deterministic algorithms for large-scale matrix computations unsuitable. In this context, strategies that use a degree of randomness have received a lot of attention in recent years. These so-called *randomized algorithms* provide an efficient and reliable means for solving linear algebra problems such as the computation of low-rank approximations to matrices, matrix factorizations, and matrix projections (cf. [4,5,8,12,15]). Randomized versions of basic matrix operations have also been proposed. An example of this is the BasicMatrixMultiplication (BMM) algorithm from [3].

The crucial aspect of randomized algorithms for matrix computations is the random selection criterion or sampling strategy of columns or rows of the matrices. Theoretical results suggest that high-quality approximations can be obtained if judiciously chosen non-uniform sampling is used, and therefore, this should be preferred over more obvious choices such as uniform sampling. In [3], a sampling strategy called *optimal probabilities* was proposed and shown to have some

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2012, LNCS 7329, pp. 45–54, 2012.

optimality properties. A new criterion was recently proposed in [6] which also possesses optimality properties. Interestingly enough, the two criteria coincide for a particular problem in information retrieval.

In this work, we focus on randomized matrix-multiplication with ill-conditioned matrices from inverse problems and in particular, image processing. We present results of an experimental study where we compare different sampling criteria on this kind of matrices in the context of the BMM algorithm. Similar studies have been carried out for other randomized matrix algorithms such as the Nyström method [10]. The performance of uniform and optimal sampling on this kind of problems does no coincide with the theoretical results mentioned above.

Our results seem to indicate that the conditioning of the input matrices determines to a great extend the choice of sampling strategy. We found that for random matrices, optimal sampling performs as predicted by theory. However, for ill-conditioned matrices, uniform sampling yields more accurate approximations. We give an explanation for this behavior that establishes the importance of the linear independence between the vectors in the sample, i.e. the conditioning of the matrix formed by the sampled columns. Based on this observation, we propose *piecewise uniform sampling*, a novel sampling criterion for a particular problem in image processing.

The presentation is organized as follows. In Section 2, we review the basic randomized matrix multiplication algorithm BMM as well as the sampling criterion of optimal probabilities. In Section 3, we present results from our study of the performance of sampling techniques. In Section 4, we describe our new sampling criterion for the multiplication of affinity matrices from image segmentation. Section 5 contains concluding remarks.

2 A Monte Carlo Matrix Multiplication Algorithm

This section describes the BMM algorithm proposed in [3] for computing an approximation to the product AB of an $m \times n$ matrix A and an $n \times p$ matrix B. As in [3], we use $A^{(j)}$ and $A_{(i)}$ to denote, respectively, the jth column and the ith row of a given matrix A. In the remainder of the paper, $\|\cdot\|$ shall denote the Euclidean norm of a vector or a matrix, $\|\cdot\|_F$ shall denote the Frobenius norm of a matrix, and **Pr** shall denote probability.

The main idea behind the BMM algorithm is to consider the outer-product formulation of the matrix product, i.e. write AB as the sum of n rank-one matrices (the product of a column of A and the corresponding row of B), and then select only some terms to approximate the product. This gives:

$$AB = \sum_{t=1}^{n} A^{(t)} B_{(t)} \approx CR = \sum_{t=1}^{s} C^{(t)} R_{(t)} \tag{1}$$

where s is the number of terms selected, and the $m \times s$ matrix C and the $s \times p$ matrix R contain the selected columns of A and corresponding rows of

BasicMatrixMultiplication Algorithm.
Input: $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, $s \in \mathbb{Z}+$ such that $1 \le s \le n$, and $\{p_i\}_{i=1}^n$ such that $p_i \ge 0$ and $\sum_{i=1}^n p_i = 1$
Output: $C \in \mathbb{R}^{m\times s}, R \in \mathbb{R}^{s\times p}$ $C \approx A, R \approx B$
1. **for** t=1:s
1.1. Choose $i_t \in \{1, \ldots, n\}$ with **Pr** $[i_t = k] = p_k$, $k = 1, \ldots, n$, independently and with replacement
1.2. Set $C^{(t)} = A^{(i_t)}/\sqrt{sp_{i_t}}$ and $R_{(t)} = B_{(i_t)}/\sqrt{sp_{i_t}}$
end
2. **return** C, R

Fig. 1. The BasicMatrixMultiplication Algorithm from [3]

B, respectively. In the BMM algorithm, the s columns are selected by random sampling and the number s is referred to as the sample size. Figure 1 shows the complete BMM algorithm from [3]. The algorithm computes submatrices C and R that can be used to approximate the product AB as follows:

$$AB \approx CR = \sum_{t=1}^{s} C^{(t)} R_{(t)} = \sum_{t=1}^{s} \frac{1}{sp_{i_t}} A^{(i_t)} B_{(i_t)}. \tag{2}$$

The scaling factor $1/sp_{i_t}$ guarantees that the algorithm produces an unbiased estimator for the left sum in (1). Other theoretical properties can be found in [3, Lemmas 3 and 4].

Different variants of the BMM algorithm can be obtained by choosing different probabilities p_k in step 1.1. Note that the choice of probabilities determines the sampling strategy. Therefore, these terms are used interchangeably. The following probabilities were proposed in [3]:

$$p_k = \frac{\|A^{(k)}\| \|B_{(k)}\|}{\sum_{k'=1}^{n} \|A^{(k')}\| \|B_{(k')}\|}, \tag{3}$$

and according to [3, Lemma 4], if p_k is the probability of the kth column of A (and thus of the kth row of B) being in the sample, and if the probability vector $\{p_k\}_{k=1}^n$ is defined as in (3), then the expected value of the absolute error $\|AB - CR\|_F^2$ is minimized over all possible choices of sampling probabilities. In [3], (3) are called *optimal probabilities* or *optimal sampling.*

The computational cost of the BMM algorithm depends on the sampling criterion. If uniform sampling is used, the algorithm requires $O(s(m+p))$ time and storage to sample A and B and to construct C and R. Only one pass over the data is necessary in this case. The use of the optimal sampling (3) leads to a requirement of $O(s(m+n+p))$ storage and time, in addition to two passes over the data.

Interestingly enough, when the probabilities derived in [6] are used for sampling term-by-document matrices from information retrieval, they turn out to be identical to the probabilities (3). Note that both sets of probabilities satisfy different optimality criteria (see [6, Section 3.2] for more details). In view of these properties, in general more accurate approximations to the matrix product are obtained when (3) are used as sampling criterion. However, as we shall see in Section 3, these probabilities do not perform as effectively in some cases such as multiplication with ill-conditioned matrices.

3 Sampling Techniques for Ill-Conditioned Matrices

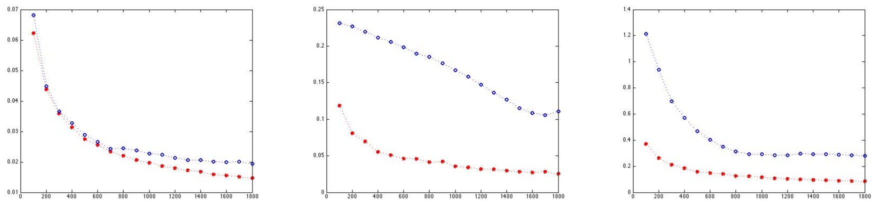
In this section, we present results from a study of the accuracy of the matrix product CR as an approximation to the product AB, when one of the input matrices is ill-conditioned. In order to compute CR, we used the BMM algorithm combined with uniform and optimal sampling. The goal of the study was to evaluate the performance of the sampling criterion. We present an explanation of the results and use it to design a new sampling criterion for a particular application.

Our study was carried out in Matlab R2011b on a MacBookPro with a 2.66 GHz processor and 4 GB of RAM, running Mac OS X version 10.7.3 (Lion). The floating-point arithmetic was IEEE standard double precision with machine precision $2^{-52} \approx 2.2204 \times 10^{-16}$. In all experiments, A was 3600×3600 and B was 3600×50. Accuracy was measured in terms of the relative error in the approximate matrix product defined as $\frac{\|AB-CR\|}{\|AB\|}$. Different sample sizes were used, starting at 100 with 100 increments and with a maximum of $\frac{n}{2} = 1800$. We considered the following three classes of matrices:

Random: A matrix with entries uniformly distributed in $(0, 1)$.
Ill-posed: A matrix resulting from using a quadrature rule to approximate Fredholm or Volterra integral equations of the first kind.
Affinity: A matrix whose entries represent a measure of pairwise affinity or similarity of pixels in an image.

Matrices in the **random** class are in general either well-conditioned or not severely ill-conditioned. Matrices in the **ill-posed** class are ill-conditioned as a consequence of the ill posedness of the underlying continuous problem. **Affinity** matrices are in general ill-conditioned for most similarity measures, such as the ones used for image segmentation purposes. In those applications, pixels must be clustered and this yields affinity matrices with subsets of columns with similar entries and therefore, numerically linearly dependent.

We used the matrix from problem **heat** in [9] as a matrix in the **ill-posed** class. The condition number of this matrix is $O(10^{16})$ This problem is a discretized version of the inverse heat equation described in [1]. Other problems from the set [9] yielded similar results. For the **affinity** class, we used the affinity matrix corresponding to the leftmost image in Figure 4, with condition number of $O(10^{21})$. Both the image and the code that generates the affinity matrix are



Random **Ill-posed** **Affinity**

Fig. 2. Relative error in the matrix product of a matrix A (random, ill-posed, or affinity) and a random matrix B computed with the BMM algorithm with uniform (dot) and optimal (circle) sampling; y-axis: relative error, x-axis: sample size

available from [2]. Random matrices were generated with Matlab's routine `rand`. We used ten different seeds to generate ten different random streams. Since results for all seeds were similar, we report results for one seed only. In our case, the random matrix A had condition number of $O(10^7)$ and the random matrix B had condition number of $O(10)$.

3.1 Uniform and Optimal Sampling for Ill-Conditioned Matrices

Our first experiment was designed to show the accuracy of the approximate matrix product computed with the BMM algorithm when either uniform or optimal sampling were used. The matrix A was of class **random**, **ill-posed**, or **affinity**. The matrix B was **random**. For each sample size, an approximate product CR was computed with the BMM algorithm combined with each of the sampling criteria. The computation was repeated 100 times for each sample size. We report average relative errors. In Figure 2, we show relative error versus sample size for uniform and optimal sampling for the three classes of matrices A considered. We can observe in Figure 2 that uniform and optimal sampling yielded relative errors of similar magnitude when both matrices A and B were random, especially for small sample sizes. As the sample size increased, there seemed to be a small advantage to uniform sampling in this example. The situation changed when A was ill-conditioned. We can observe a greater difference between the relative errors given by each sampling criteria and we also observe that uniform sampling yielded more accurate approximations than optimal sampling. This was the case in particular for small sample sizes. As the sample size increased, the relative errors became closer, although they were still considerably different. This can also be observed in Table 1 where we report the maximum and minimum difference between the relative errors obtained when using uniform and optimal sampling.

3.2 Analysis

In this section, we present an explanation for the results in Figure 2. Note that from (2), we can write the kth column of CR given by the BMM algorithm as a linear combination of the *sampled* columns of A. This gives:

Table 1. Maximum and minimum difference between relative errors in the matrix product computed by the BMM algorithm with uniform and optimal sampling

	Random	**Ill-posed**	**Affinity**
max	6.5e-03	1.6e-01	8.4e-01
min	6.0e-04	7.7e-02	1.7e-01

$$(CR)^{(k)} = \sum_{t=1}^{s} \frac{B_{(i_t,k)}}{cp_{i_t}} A^{(i_t)},$$

i.e., the approximation $(CR)^{(k)}$ is in the subspace spanned by the sampled columns of A. Therefore, the ability of the sample to yield a good approximation to $(AB)^{(k)}$ depends on the ability of the sampled columns of A to represent a new vector. Hence, the linear independence of the sampled columns of A is a crucial factor for the accuracy of the approximation given by the BMM algorithm in the sense that samples that consist of independent columns of A have higher probability of yielding accurate results.

If the matrix A is not too ill-conditioned, then with high probability, any set of sampled columns (with replacement) is a linearly independent set. However, when A is an ill-conditioned matrix, optimal sampling might favor the selection of dependent columns of A. For instance, consider a matrix for which some columns have very similar entries in the same positions. In this case, those columns have similar vector norms and hence, similar components in the probability vector (3). Therefore, if one of those columns is in the sample, then the other similar columns have high probability of being in the sample as well. In contrast to this, uniform sampling does not favor the selection of linearly dependent columns. This analysis explains the behavior of the relative error in Figure 2.

3.3 Distribution of the Relative Error for Different Sampling Criteria

Our second experiment was designed to show the distribution of the relative error in the approximate matrix product over the sample size, for uniform and optimal sampling within the BMM algorithm. The results also show the effect of randomness on the composition of the sample. The matrix A was either **random** or **ill-posed**. The matrix B was **random**. For each sample size, an approximate product CR was computed with the BMM algorithm. The computation was repeated 50 times for each sample size, and each relative error was plotted. In Figure 3, we show the distribution of the relative error over sample size for uniform and optimal sampling for the two classes of matrices A considered. We observe in Figure 3 that the behavior of the relative error differed for the two classes of matrices considered. When A was a random matrix, both uniform and optimal sampling yielded relative errors in the order of 1% which is the expected accuracy when using randomized algorithms (cf. [6]), for sample sizes that are

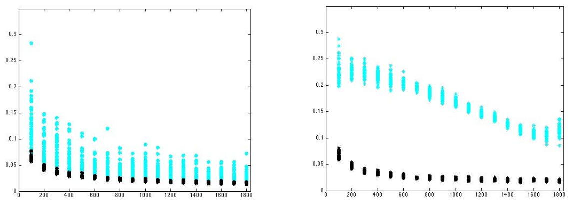


Fig. 3. Relative error vs. sample size for uniform (left) and optimal (right) sampling within the BMM algorithm, when A is either random (dark) or ill-conditioned (clear)

large enough, say 500 and larger in our case. When A was ill-conditioned, the situation was different. Uniform sampling yielded approximations with relative errors in the order of around 12%, starting at $s = 500$, while the relative errors for optimal sampling remained around or above 20% for most reasonable sample sizes, and could only reach 12% for sizes that were close to half the size of the input. We also observed that for ill-conditioned matrices, and in contrast to optimal sampling, uniform sampling yielded relative errors that were around 1% for most sample sizes.

3.4 Discussion

The results in this section, and in particular in Section 3.3, seem to indicate an advantage to uniform over optimal sampling for randomized multiplication with ill-conditioned matrices in terms of accuracy of the approximation. This is somewhat surprising because optimal sampling has a strong theoretical support that suggests otherwise. Results favoring uniform sampling for the Nyström method, another randomized algorithm, were reported in [10].

The observed advantage in terms of accuracy is relevant since uniform sampling also has a number of practical advantages: it only requires one pass over the input (see Section 2); it does not required the actual construction of the approximation matrices C and R; and it allows matrix-free computations, i.e., the matrices A and B do not need to be given explicitly but it suffices to have procedures for computing matrix-vector products with A and B^T. Note that optimal sampling does not possess any of these features.

An interesting fact shown in Section 3.2 is the role of the linear independence of the sampled vectors. The arguments given in that section indicate that an effective sampling criterion for ill-conditioned matrices should be related to a criterion for selecting linearly independent columns of a matrix. Criteria of this kind are in general very expensive and their practical usefulness is limited to cases where additional *a priori* information about the linear dependence of the columns or rows is available. In the next section, we discuss a practical case with this feature, and propose a new sampling criterion that seeks to construct linearly-independent column samples.

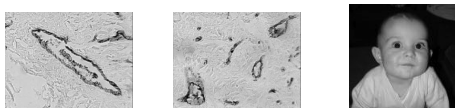


Fig. 4. Test images: medical (left and center); natural (right)

4 Piecewise Uniform Sampling: A Novel Sampling Criterion

In this section, we propose a new sampling criterion geared toward matrices that have sets of columns with similar entries and therefore are likely to be numerically linearly dependent. As discussed in Section 3.1, affinity matrices usually possess this feature.

Our empirical criterion for sampling this kind of matrices is as follows. We first divide the (index) interval $[1, n]$ of possible columns in s subintervals, where $s \leq n$ is the sample size, and then, we uniformly select one column in each subinterval. We call this criterion *piecewise uniform sampling*. The strategy favors samples consisting of non-consecutive columns and has the same cost advantages of uniform sampling (see Section 3.1). To illustrate the performance of the new criterion, we used the affinity matrices associated with the images in Figure 4. The images and the code for computing the affinity matrices are available from [2]. We carried out two experiments in which we compared the accuracy of the approximate matrix product computed with the BMM algorithm in combination with three sampling criteria: uniform, optimal, and piecewise uniform.

In our first experiment, we used the affinity matrices corresponding to the medical images in Figure 4 as matrix A. The matrix B was **random** as in Section 3.1. Matrix dimensions and other settings for the experiments were as in Section 3.1. Figure 5 shows the relative errors in the approximations for each of the sampling strategies. We can observe that uniform and optimal sampling behaved as before, while piecewise uniform sampling yielded approximations with higher accuracy for sample sizes of up to around 25% of the input size. As the sample size increases and the intervals becomes smaller, the criterion loses its random nature. This seems to indicate that smaller sample sizes should be preferred for piecewise uniform sampling. Note that small sample sizes are indeed needed for randomized algorithms to be competitive. In our second experiment, we used randomized matrix-vector multiplication in the context of the Normalized Cut algorithm for image segmentation [13] (see also [7]). This method requires the computation of an eigenvector corresponding to the second smallest eigenvalue of the affinity matrix of the image. In this, as well as in other image processing applications, matrices are usually large or not available explicitly and therefore, so-called matrix-free techniques that rely on matrix-vector multiplications only are the methods of choice. In [2], the eigenvalue problems are solved with Matlab's routine `eigs`, which is an interface to ARPACK [11], an implementation of the very efficient, matrix-free Implicitly Restarted Arnoldi Method [14]. Figure 6 shows the eigenvectors

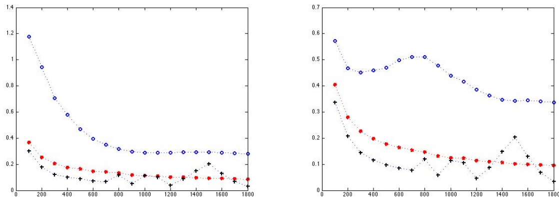


Fig. 5. Relative error vs. sample size for uniform (dot), optimal (circle), and piecewise uniform sampling (cross) of affinity matrices within the BMM algorithm

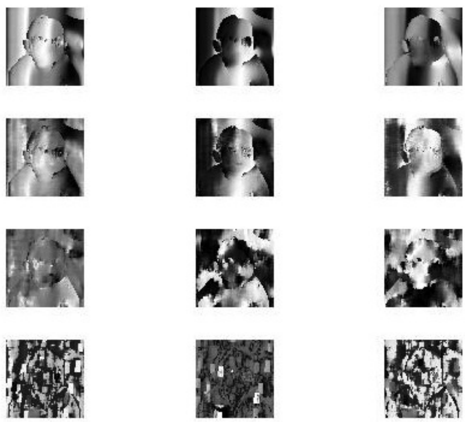


Fig. 6. Approximations of the three leading eigenvectors of an affinity matrix using as matrix-vector product in `eigs` (from the top): deterministic, BMM algorithm with piecewise uniform, uniform, and optimal sampling

computed with `eigs` when the code [2] was used to compute the segmentation of the rightmost image in Figure 4. The affinity matrix was of size 25600×25600 and the sample size was 15% of the problem size, i.e. $s = 3840$. We show results for the deterministic matrix-vector product and for the randomized product computed with the BMM algorithm combined with the three sampling criteria used in the first experiment. We can observe that piecewise uniform sampling yielded an approximation to the second eigenvector (second column in Figure 6) that contains valid information for performing an acceptable segmentation of the image, whereas the approximations obtained by uniform and optimal sampling are not useful for segmentation purposes.

5 Concluding Remarks

We evaluated the practical convenience of using optimal and uniform sampling in a randomized matrix multiplication algorithm. Our experimental results indicate that the optimal criterion often yields approximations with lower accuracy than uniform sampling when one of the matrices involved is ill-conditioned. This is in contrast to what the theoretical background suggests. Ill-conditioned matrices arise in many applications. Examples include discretized operators from ill-posed problems and affinity matrices from image processing. Multiplication with such matrices is required in most efficient numerical methods. We showed experimentally that the optimal distribution might not be suitable in this context and proposed a new sampling criterion that is both efficient and accurate. Our numerical results seem to indicate that the new criterion is a promising sampling strategy for Monte Carlo matrix multiplication in the context of state-of-the-art image-segmentation methods.

References

1. Carasso, A.: Determining surface temperatures from interior observations. SIAM J. Appl. Math. 42, 558–574 (1982)
2. Cour, T., Yu, S., Shi, J.: Normalized Cuts Segmentation Code, Copyright University of Pennsylvania, Computer and Information Science Department (2004), `http://www.seas.upenn.edu/~timothee/software/ncut/ncut.html`
3. Drineas, P., Kannan, R., Mahoney, M.W.: Fast Monte Carlo algorithms for matrices I: approximating matrix multiplication. SIAM J. Comput. 36, 132–157 (2006)
4. Drineas, P., Kannan, R., Mahoney, M.W.: Fast Monte Carlo algorithms for matrices II: computing a low-rank approximation to a matrix. SIAM J. Comput. 36, 158–183 (2006)
5. Drineas, P., Kannan, R., Mahoney, M.W.: Fast Monte Carlo algorithms for matrices III: computing a compressed approximate matrix decomposition. SIAM J. Comput. 36, 184–206 (2006)
6. Eriksson-Bique, S., Solbrig, M., Stefanelli, M., Warkentin, S., Abbey, R., Ipsen, I.C.F.: Importance sampling for a Monte Carlo matrix multiplication algorithm, with application to information retrieval. SIAM J. Sci. Comput. 33, 1689–1706 (2011)
7. Fowlkes, C., Belongie, S., Chung, F., Malik, J.: Spectral grouping using the Nyström method. IEEE T. Pattern Anal. 26, 214–225 (2004)
8. Friedland, S., Mehrmann, V., Miedlar, A., Nkengla, M.: Fast low rank approximations of matrices and tensors. Electron. J. Linear Al. 22, 1031–1048 (2011)
9. Hansen, P.C.: Regularization Tools Version 4.0 for Matlab 7.3. Numer. Algo. 46, 189–194 (2007)
10. Kumar, S., Mohri, M., Talwalkar, A.: Sampling Techniques for the Nyström Method. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, AISTATS (2009)
11. Lehoucq, R.B., Sorensen, D.C., Yang, C.: ARPACK User's Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods. SIAM, Philadelphia (1998)
12. Liberty, E., Zucker, S.W.: The Mailman algorithm: A note on matrix-vector multiplication. Inform. Process. Lett. 109, 179–182 (2009)
13. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. IEEE T. Pattern Anal. 22, 888–905 (2000)
14. Sorensen, D.C.: Implicit application of polynomial filters in a k-step Arnoldi method. SIAM J. Matrix Anal. A. 13, 357–385 (1992)
15. Woolfe, F., Liberty, E., Rokhlin, V., Tygert, M.: A fast randomized algorithm for the approximation of matrices. Appl. Comput. Harmon. Anal. 25, 335–366 (2007)