An Empirical Comparison of Sampling Techniques for Matrix Column Subset Selection

Yining Wang and Aarti Singh
Machine Learning Department, Carnegie Mellonn University
{yiningwa,aartisingh}@cs.cmu.edu

Abstract— **Column subset selection (CSS) is the problem of selecting a small portion of columns from a large data matrix as one form of interpretable data summarization. Leverage score sampling, which enjoys both sound theoretical guarantee and superior empirical performance, is widely recognized as the state-of-the-art algorithm for column subset selection. In this paper, we revisit iterative norm sampling, another sampling based CSS algorithm proposed even before leverage score sampling, and demonstrate its competitive performance under a wide range of experimental settings. We also compare iterative norm sampling with several of its other competitors and show its superior performance in terms of both approximation accuracy and computational efficiency. We conclude that further theoretical investigation and practical consideration should be devoted to iterative norm sampling in column subset selection.**

Index Terms— **Column subset selection, leverage score sampling, iterative norm sampling, rank-revealing QR factorization.**

I. Introduction

Column subset selection (CSS) is the problem of selecting a small portion of columns from a data matrix so that the selected columns serve as a good "snapshot" of the original data matrix. [1] In a broader view, column subset selection can be viewed as a special type of low-rank matrix approximation where a high degree of interpretability is retained by restraining the low-dimensional subspace to be spanned by *actual columns* in the original data matrix. The column subset selection problem has found applications in a number of statistical and machine learning tasks, such as population genetics summarization, electronic circuits testing, recommendation systems, etc [1], [2].

Column subset selection is traditionally solved by rank-revealing QR factorization (RRQR, [3], [4]). Recently there is increasing interest in applying *sampling* based techniques on this problem [5], [6], [7], [8], [9]. Among these methods, the *leverage score sampling* algorithm proposed in [6] is perhaps the most popular one due to its simplicity, good theoretical properties and increased scalability compared to deterministic factorization based methods such as RRQR, making the leverage score sampling particularly suitable for modern data analysis applications where huge data matrices are prevalent. Empirical evidence also suggests that the accuracy of leverage score sampling is comparable with state-of-the-art deterministic methods.

Apart from leverage score sampling, iterative norm sampling proposed in [9] is another sampling scheme designed to tackle the column subset selection problem. The prime motivation of the original paper is to derive an approximation algorithm of the theoretically optimal yet computationally inefficient *volume sampling* algorithm [8]. Compared to leverage score sampling, the iterative sampling algorithm has worse error bounds; in fact, the derived error bound gets exponentially worse when the intrinsic dimension of the data matrix increases [2]. Though [8] also presents another algorithm that has better theoretical guarantees, the algorithm selects too many columns per iteration and hence is highly impractical. Due to these reasons, iterative norm sampling is largely ignored in practical situations.

In this paper, we revisit iterative norm sampling and empirically compare it with leverage score sampling as well as other column subset selection algorithms under a wide range of experimental settings. Quite surprisingly, we demonstrate highly competitive results for iterative norm sampling in terms of both reconstruction (approximation) accuracy and computational efficiency. As a conclusion, we call for more theoretical and practical effort into the iterative norm sampling algorithm.

II. Related work

Rank-revealing QR (RRQR) factorization is a matrix decomposition algorithm based on QR factorization. It is the cornerstone algorithm for column subset selection and is nearly optimal when evaluated in terms of either spectral or Frobenious norm approximation error [4], [1]. [4] provides a nice review and an efficient algorithm for RRQR factorization. However, RRQR typically requires cubic running time, which is too slow for even medium-size data matrices.

Norm sampling proposed in [7] is perhaps the first provably correct sampling based algorithm for matrix column subset selection. The algorithm is embarrassingly simple and also extremely fast ($O(n_1 n_2)$ time complexity, where n_1 and n_2 are the number of rows and columns of the input matrx),. Unfortunately, the algorithm only enjoys an additive error bound and has poor performance over large low-rank matrices. [3]

[1] Formal mathematical formulation of CSS is given in Section III-A.

[2] See Theorem 3 in Section III-D for details.

[3] Section III-A explains the difference between additive and multiplicative (relative) error bounds.

Leverage score sampling was proposed in [6] and is probably the most popular sampling based method for column subset selection. Employing the concept of *incoherence* from the low-rank matrix completion literature [10], [11], the leverage score sampling algorithm bases its sampling distribution on squared ℓ_2 row norm of the top-k truncated right singular vector. Leverage score sampling enjoys good theoretical properties and superb performance in practice. Recent research also shows that for certain matrices sampling according to *square roots* of leverage scores could help obtain tighter error bounds than plain leverage score sampling [12].

Iterative norm sampling was proposed in [9] as an approximation algorithm of the near-optimal but computationally inefficient volume sampling algorithm [8]. It also enjoys multiplicative and relative error bounds as leverage score sampling, yet with worse multiplicative factors. No empirical results are currently known for the iterative norm sampling algorithm, except in [13], [14] where a variant of the iterative sampling algorithm is used for column subset selection with partial observation.

Apart from RRQR and sampling based methods, other optimization formulation such as group Lasso and block orthogonal matching pursuit (OMP) have also been successfully applied to column subset selection [15], [2]. However, no theoretical results are known for these two algorithms.

III. Sampling based CSS algorithms

In this section we first formally define notations and the column subset selection (CSS) problem. We then review three existing and widely applied sampling based column subset selection algorithms. We provide pseudocode for each algorithm and summarize theoretical error bounds as well as practical running time.

A. Notations and problem definition

For any matrix $\mathbf{M}$ we use $\mathbf{M}^{(i)}$ to denote the i-th column of $\mathbf{M}$. Similarly, $\mathbf{M}_{(i)}$ denotes the i-th row of $\mathbf{M}$. For a vector $\boldsymbol{x}$, $\|\boldsymbol{x}\|_2 = \sqrt{\sum_i x_i^2}$ denotes its ℓ_2 norm. For a matrix $\mathbf{M}$, $\|\mathbf{M}\|_2 = \sigma_{\max}(\mathbf{M})$ denotes its spectral norm (i.e., the largest singular value) and $\|\mathbf{M}\|_F = \sqrt{\sum_{i,j}\mathbf{M}_{ij}^2}$ denotes the Frobenious norm of $\mathbf{M}$.

In the column subset selection (CSS) problem, the input is an $n_1 \times n_2$ matrix $\mathbf{M} \in \mathbb{R}^{n_1\times n_2}$. To simplify notations, we also write $n = \max(n_1, n_2)$. An algorithm is expected to output $\mathbf{C} = (\mathbf{C}^{(1)}, \cdots, \mathbf{C}^{(s)}) \in \mathbb{R}^{n_1\times s}$ consisting of $s < n_2$ actual columns of $\mathbf{M}$, so that the approximation error

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_{\xi}$$

is minimized, where $\xi = 2, F$ can be either the matrix spectral norm or Frobenious norm and $\mathbf{C}^{\dagger}$ denotes the Moore-Penrose pseudoinverse of $\mathbf{M}$. Note that here $\mathbf{C}\mathbf{C}^{\dagger}\mathbf{M} = \mathcal{P}_{\mathbf{C}}(\mathcal{M})$ is the projection of each column of $\mathbf{M}$ onto the linear subspace spanned by the selected columns in $\mathbf{C}$.

Algorithm 1 Norm sampling for column subset selection

```
Input: data matrix $\mathbf{M}$, size of column subset $s$.
Norm computation: $\hat{c}_i = \|\mathbf{M}^{(i)}\|_2^2$; $\hat{f} = \sum_i \hat{c}_i$.
Sampling: $\Pr[\mathbf{C}^{(j)} = \mathbf{M}^{(i)}] = \hat{c}_i/\hat{f}$; $i \in [n_2]$, $j \in [s]$.
Output: selected columns $\mathbf{C}$.
```

Algorithm 2 Leverage score sampling

```
Input: $\mathbf{M}$, size of column subset $s$, target rank $k$.
Truncated SVD: $\mathbf{M} = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^{\top} + \mathbf{R}$.
Leverage score computation: $\ell_i = \frac{n_2}{k}\|\mathbf{V}_k^{\top}\boldsymbol{e}_i\|_2^2$.
Sampling: $\Pr[\mathbf{C}^{(j)} = \mathbf{M}^{(i)}] = \ell_i/n_2$.
Output: selected columns $\mathbf{C}$.
```

Usually, the approximation error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_{\xi}$ is compared against $\mathbf{M}_k$, the approximation error of the best rank-k approximation of $\mathbf{M}$. Two types of error bounds are typical:

$$\begin{aligned} \|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_{\xi} &\leq \|\mathbf{M} - \mathbf{M}_k\|_{\xi} + \epsilon\|\mathbf{M}\|_{\xi}; \quad (1)\\ \|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_{\xi} &\leq c\|\mathbf{M} - \mathbf{M}_k\|_{\xi}. \quad (2) \end{aligned}$$

Error bounds in Eq. (2) with $0 < \epsilon < 1$ are usually referred to as *additive* because there is an additive term involved in the bound that does not decrease with the best low-rank approximation error $\|\mathbf{M} - \mathbf{M}_k\|_{\xi}$. On the other hand, Eq. (2) shows an example of a *multiplicative* error bound since the error is multiplicative in terms of $\|\mathbf{M} - \mathbf{M}_k\|_{\xi}$. In addition, when the multiplicative factor c is of the form $c = 1 + \epsilon$ we call Eq. (2) a *relative* error bound. In general, multiplicative and relative error bounds are much preferred to additive ones since in most applications data matrices are approximately low-rank, which means $\|\mathbf{M} - \mathbf{M}_k\|_{\xi}$ could be far smaller than $\|\mathbf{M}\|_{\xi}$.

B. Norm sampling

The norm sampling algorithm was first proposed in [7] as a provably correct algorithm for matrix column subset selection. The idea is extremely simple: each column $\mathbf{M}^{(i)}$ is sampled with probability proportional to its ℓ_2 norm, i.e., $\|\mathbf{M}^{(i)}\|_2^2$. Pseudocode for norm sampling is given in Algorithm 1.

Theorem 1 provides an additive error bound on the results obtained by Algorithm 1. Time complexity of Algorithm 1 is $O(n_1 n_2)$.

Theorem 1 ([7], Theorem 2): Fix input matrix $\mathbf{M} \in \mathbb{R}^{n_1\times n_2}$ and $k < s < n_2$. Let $\mathbf{C} \in \mathbb{R}^{n_1\times s}$ be the output of Algorithm 1. Then with probability at least 0.9 the following holds:

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_F^2 \leq \|\mathbf{M} - \mathbf{M}_k\|_F^2 + \frac{10k}{cs}\|\mathbf{M}\|_F^2, \quad (3)$$

where $\mathbf{M}_k$ is the best rank-k approximation of $\mathbf{M}$ and c is a universal constant.

C. Leverage score sampling

Leverage score sampling for matrix column subset selection and CUR/CX approximation was first proposed in [6] and was shown to achieve *multiplicative* or even *relative* error bounds, which improves previous additive results like

Algorithm 3 Iterative norm sampling

```
Input: data matrix $\mathbf{M}$, size of column subset $s$.
Initialize: $\mathbf{C} = \mathbf{0}$, $\mathbf{X} = \mathbf{M}$.
for $i = 1$ to $s$ do
    Norm computation: $\hat{c}_j = \|\mathbf{X}^{(j)}\|_2^2; \hat{f} = \sum_j \hat{c}_j$.
    Sampling: $\Pr[\mathbf{C}^{(i)} = \mathbf{M}^{(j)}] = \hat{c}_j/\hat{f}$.
    Back projection: $\mathbf{X} \leftarrow \mathbf{X} - \mathbf{C}\mathbf{C}^\dagger\mathbf{X}$.
end for
Output: selected columns $\mathbf{C}$.
```

[7] and [5]. In particular, the *row space leverage scores* of a matrix $\mathbf{M}$ is defined as

$$\ell_i := \frac{n_2}{k}\|\mathbf{V}_k^\top \boldsymbol{e}_i\|_2^2, \ \ i = 1, \cdots, n_2; \tag{4}$$

for some target rank k, where $\mathbf{M} = \mathbf{U}_k\boldsymbol{\Sigma}_k\mathbf{V}_k^\top + \mathbf{R}$ is the top-k truncated singular value decomposition of $\mathbf{M}$ and $\boldsymbol{e}_i \in \mathbb{R}^{n_2}$ is the unit vector with only the ith index non-zero. The leverage score sampling algorithm samples columns of $\mathbf{M}$ with probability proportional to the leverage score of each column, as shown in Algorithm 2.

Theorem 2 shows that leverage score sampling achieves relative error bounds if slight over-sampling (i.e., $s > k$) is allowed. The computational copmlexity of the leverage score sampling algorithm is $O(n_1 n_2 k)$, with truncated SVD step being the computational bottleneck.

Theorem 2 ([6], Theorem 3): Let $\mathbf{M} \in \mathbb{R}^{n_1\times n_2}$ *be the input matrix and* $\epsilon \in (0, 1)$ *be an accuracy parameter. Suppose* $s \geq 3200k^2/\epsilon^2$ *and let* $\mathbf{C}$ *be the selected columns output by Algorithm 2. Then with probability at least 0.7 the following holds:*

$$\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F \leq (1+\epsilon)\|\mathbf{M} - \mathbf{M}_k\|_F, \tag{5}$$

where $\mathbf{M}_k$ *is the best rank-k approximation of* $\mathbf{M}$.

In this work we also consider an alternative of sampling according to *square roots* of leverage scores; that is, $p_j \propto \sqrt{\ell_j}$. Though such schemes do not have an error bound under standard column subset selection setting yet, similar approaches have been adopted recently in several statistical machine learning settings, including low-rank approximation [12] and graph signal recovery [11].

D. Iterative norm sampling (approximate volume sampling)

In [9] an iterative norm sampling algorithm (or equivalently an approximation of volume sampling [8]) was proposed as another column subset selection algorithm that enjoys multiplicative error bounds. Unlike norm sampling (Algorithm 1) that selects all columns at a time from a norm-dependent distribution, iterative norm sampling selects fewer columns at each iteration and subtracts the projection of the input matrix onto column space spanned by selected columns between iterations. The number of columns selected per iteration is tunable and slightly over-sampling is shown to yield better approximation bounds [9]. Nevertheless, empirical evidence suggests that choosing one column per iteration is already sufficient for providing high-quality results and we will focus mainly on this variant of iterative norm sampling in this paper. Pseudocode for iterative norm sampling are listed in Algorithm 3 and error bounds are presented in Theorem 3.

Theorem 3 ([9], Proposition 2): Let $\mathbf{M} \in \mathbb{R}^{n_1\times n_2}$ *be the input matrix and* $s = k$ *in Algorithm 3. We then have*

$$\mathbb{E}_{\mathbf{C}}\left[\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F^2\right] \leq (k+1)!\|\mathbf{M} - \mathbf{M}_k\|_F^2, \tag{6}$$

where $\mathbf{M}_k$ *is the best rank-k approximation of* $\mathbf{M}$.

The $(k+1)!$ factor may look appalling in Eq. (6). However, this is mainly due to the fact that we are selecting *exactly* k columns from $\mathbf{M}$, with k the target rank. If oversampling is allowed (e.g., $s = \Omega(k^2\log k + k/\epsilon)$), it can be shown that a slight variant of Algorithm 3 that allows sampling more than one columns per iteration achieves similar relative-error bounds ($\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_F^2 \leq (1+\epsilon)\|\mathbf{M} - \mathbf{M}_k\|_F^2$) as in Theorem 2 [9].

On the computational side, a brute-force implementation of Algorithm 3 requires $O(n_1 n_2 k^2)$ operations due to the back projection step. However, note that at each iteration we are only removing a 1-dimensional component from $\mathbf{X}$. Consequently, the algorithm can be trivially accelerated to run in $O(n_1 n_2 k)$ operations, the same time complexity as leverage score sampling.

Finally, we remark that iterative norm sampling is similar in principle to a block OMP algorithm proposed in [2]. Both involve iterative column selection and back projection between iterations. The major difference between the two algorithms is that iterative norm sampling is based on residue norm of column vectors while the block OMP algorithm considers the product of the original column vector with the residue vector.

IV. Experiments

In this section we compare the empirical performance of norm sampling, leverage score sampling and iterative norm sampling on both synthetic and real-world datasets. Computational efficiency is also compared on data at different scales. For reference purposes, we also include the group Lasso formulation [15] and classical rank-revealing QR (RRQR) factorization [4] for comparison. All algorithms are implemented in Matlab except in Section IV-B where methods are re-implemented in C++ for fair efficiency comparison. For RRQR we use the implementation in [16], which is a C++ implementation with a Matlab wrapper.

A. Synthetic data

We first test column subset selection algorithms on synthetic datasets. We generate matrices of dimension 50×50, with the intrinsic rank r ranging from 10 (low-rank matrices) to 50 (high-rank matrices). To generate a rank-k matrix we first sample an $n \times k$ random Gaussian matrix ($n = 50$) A; set $\mathbf{M} = \mathbf{A}\mathbf{A}^\top$ and then normalize each entry of $\mathbf{M}$ so that $\|\mathbf{M}\|_F = 1$. For low-rank matrices we also impose small random Gaussian noise on each entry of the matrix to better distinguish reconstruction error among different algorithms.

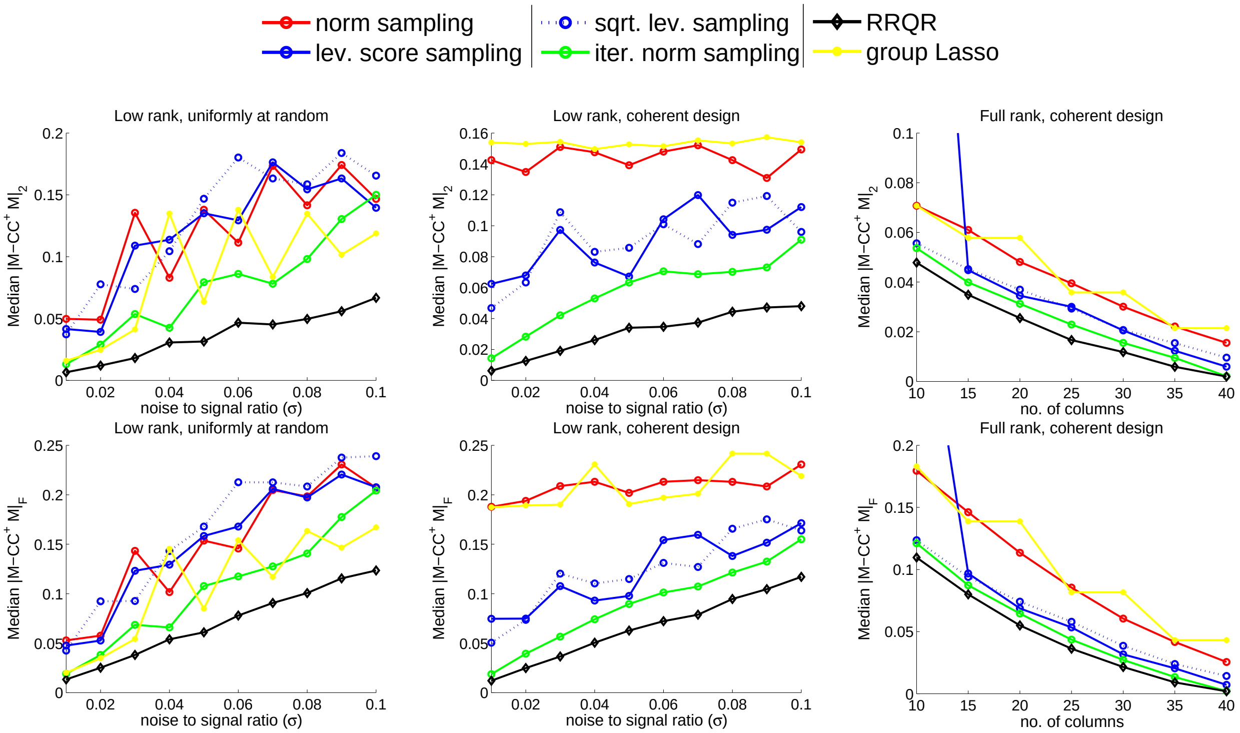

Fig. 1. Empirical comparison of reconstruction error $\|\mathbf{M}-\mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_{\xi}$ ($\xi = 2$ or F) of sampling based algorithms summarized in Section III. The blue dotted line displays leverage score sampling with probability proportional to *square roots* of leverage scores. Top: spectral reconstruction error; bottom: Frobenious reconstruction error. From left to right: low-rank matrices sampled uniformly at random; low-rank matrices with coherent columns; full-rank matrices with coherent columns. For sampling based methods we run the algorithms for 10 times and report the median of the reconstruction error. Because norm and leverage score sampling is quite variable, we report the median instead of the mean to make result more stable.

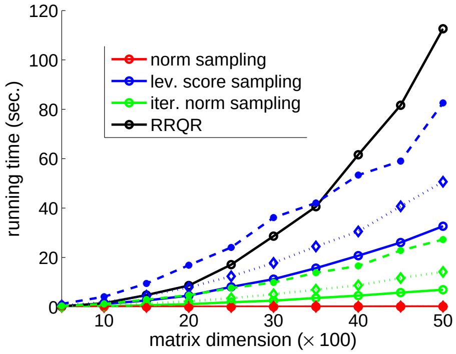

Fig. 2. Running time (seconds) of norm sampling, leverage score sampling, iterative norm sampling and RRQR factorization with respect to different matrix sizes. Solid line: $k = 25$; dotted line: $k = 50$; dashed line: $k = 100$.

Due to the nature of column subset selection, uniformly random matrices alone are not sufficient for evaluation since no column in a matrix sampled uniformly at random is significantly more representative than the other columns. Therefore, we also synthesize data matrices that are highly coherent. To do this, we simply pick a random column in $\mathbf{M}$, enlarge its ℓ_2 norm by 10 times and repeat the same column for 5 or 10 times. In this way the data matrix consists of 10 identical important columns. A satisfactory column subset selection algorithm is expected to choose one and only one from the repeated columns.

In Figure 1 we show comparison of reconstruction error $\|\mathbf{M}-\mathbf{C}\mathbf{C}^{\dagger}\mathbf{M}\|_{\xi}$, $\xi = 2, F$ on both low-rank and full-rank synthetic matrices. Note that for low-rank matrices the number of columns selected (s) is set to be equal to the intrinsic rank (k) of $\mathbf{M}$. The first observation is that both relative-error algorithms (leverage score sampling and iterative norm sampling) outperforms additive-error algorithms (norm sampling) by a large margin, especially when input data matrix has highly coherent columns. In addition, our result also shows that iterative norm sampling consistently outperforms leverage score sampling under almost all scenarios. This is quite surprising as leverage score sampling is usually considered to be the state-of-the-art method for column subset selection (at least when Frobenious norm reconstruction error is considered) and also dominates iterative norm sampling in terms of theoretical error bounds.

We further comment on the two non-sampling based methods. The rank-revealing QR (RRQR) factorization method is clearly the best algorithm for column subset selection regardless of data matrix properties (intrinsic rank or column coherence) and/or evaluation metric (spectral or Frobenious norm). This also consolidates the near-optimal theoretical properties of RRQR factorization [1]. However, RRQR requires cubic running time, which is computationally heavier than all the three sampling based algorithms. We demonstrate this difference empirically in Section IV-B. Finally, group Lasso formulation behaves much like an additive-error algo-

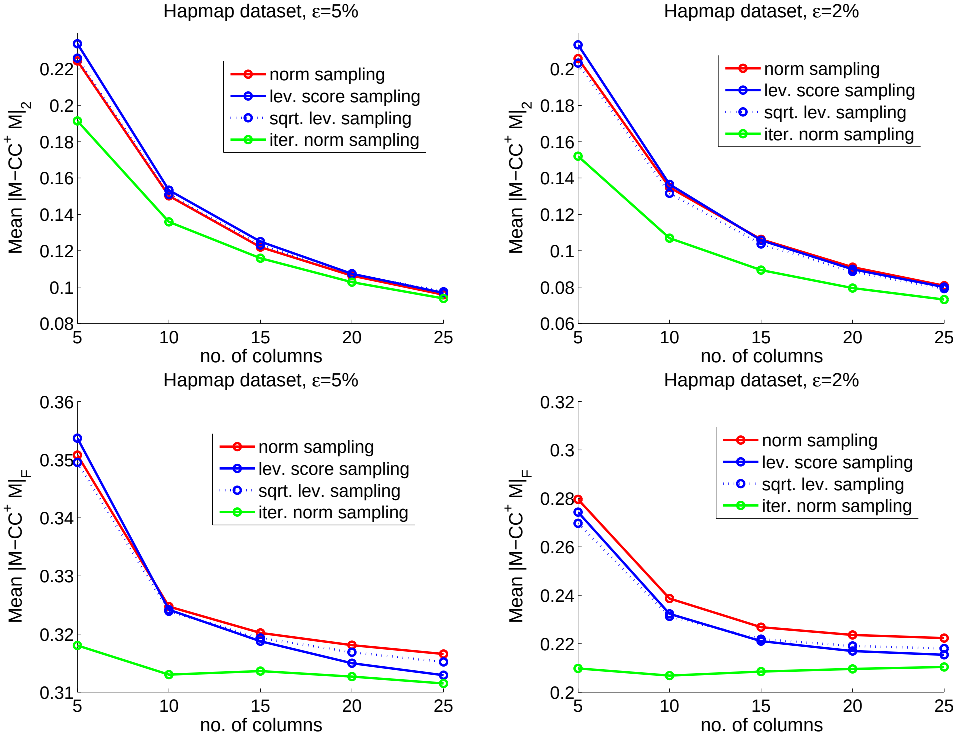

Fig. 3. Comparison of mean reconstruction error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_\xi$, $\xi = 2$ or F on the Hapmap phase 2 dataset. RRQR and group Lasso are not included due to efficiency reasons. Top: spectral norm reconstruction error; bottom: Frobenious norm reconstruction error. ε denotes the best low-rank approximation error for each sliding window. See [17] or Section 6.2 in [13] for details.

rithm [4] while being equally or even more computationally expensive than the other methods.

B. Running time

In this section we compare the running time of norm sampling, leverage score sampling, iterative nom sampling and the optimal RRQR factorization algorithms on data matrices of different sizes and intrinsic dimensions. For fair comparison we re-implement the three sampling-based algorithms in C++ and use [16] as the C++ implementation (with a Matlab wrapper) for RRQR factorization. Experiments were conducted on a laptop with an Intel Core i5-4200U@1.60GHz CPU, 12 GB main memory and Windows 8.1 operating system installed. Only one computing thread is allowed for all algorithms.

Figure 2 shows that RRQR factorization scales poorly with data matrix dimension n. [5] This is perhaps unsurprising as RRQR has $O(n^3)$ regardless of the intrinsic dimension k or the number of columns selected. On the other hand, norm sampling requires the least computational resource since it only takes $O(n^2)$ operations to scan through the data matrix once and compute per-column ℓ_2 norm. The comparison between leverage score sampling and iterative norm sampling is interesting: both algorithms share the same $O(n^2k)$ computational complexity, yet leverage score sampling is considerably slower than iterative norm sampling. This is mainly because in leverage score sampling one needs to compute a truncated SVD of the data matrix. In contrast, iterative norm sampling only performs Gram-Schmidt orthogonalization per iteration and computes back projection onto a single vector, which is nothing more than a matrix inner product evaluation. Consequently, iterative norm sampling has less computational overhead and is faster in practice than leverage score sampling. This observation reveals another potential advantage of the iterative norm sampling algorithm.

C. Human genetic dataset

The Hapmap Phase II database [18] is a major database that contains genetic data of sampled individuals across the globe. Previously people have used low-rank methods such as principle component analysis (PCA) and column subset selection to analyze the genetic dataset [17], [19]. In this section we test the reconstruction accuracy of norm sampling, volume sampling and iterative norm sampling on this real-world dataset. RRQR factorization and group Lasso are excluded from the comparison for computational efficiency reasons. We use the gene data of the first chromosome in the joint east Asian population database (CHB and JPT) for demonstration. The dataset can be represented as a data matrix with 89 rows (individuals) and 311,854 columns (gene snippets). Standard pre-processing is applied to further transform the data matrix into a series of a few hundred smaller matrices. Readers can refer to [17] or Section 6.2 in

[4] The original paper in which it was proposed did not provide an error bound.

[5] We use square input matrices in this experiment; that is, $n = n_1 = n_2$.

[13] for details about the pre-processing steps.

Figure 3 shows the average reconstruction error $\|\mathbf{M} - \mathbf{C}\mathbf{C}^\dagger\mathbf{M}\|_\xi$, $\xi = 2, F$ of the three sampling based column subset selection algorithms. It is clear that iterative norm sampling outperforms both the other algorithms by a large margin under all experimental settings. On the other hand, leverage score sampling only shows notable improvement over the baseline norm sampling algorithm when the number of selected columns is large and the Frobenious reconstruction error is evaluated.

V. Concluding remarks

Experimental results in Section IV shows that iterative norm sampling outperforms leverage score sampling in terms of approximation accuracy under almost all testing scenarios. This does not agree with theoretical results for both algorithms, as presented in Section III. This raises an interesting question of whether error bounds for iterative norm sampling can be further improved. In particular, we feel the error bound in Eq. (6) is extremely loose because the exponential $(k+1)!$ factor does not show up at all in practice even when k is as large as 50. We think it is a promising future direction to explain theoretically the superior empirical performance of iterative norm sampling.

On the theoretical side, though iterative norm sampling has comparable or worse theoretical error bounds compared to leverage score sampling, under certain settings iterative norm sampling could have better theoretical guarantee. One example is [13] where the authors consider a *missing data* setting; that is, only a small portion of the input matrix is observed. Under such settings, the authors showed that if $O(k^2)$ columns and feedback-driven sampling schemes are allowed, iterative norm sampling could achieve $(1+\epsilon)$-type relative error; while on the other hand approximate leverage score sampling is only able to achieve a constant-factor multiplicative error bound.

Computational efficiency is another interesting issue arising from our experimental results. In Section IV-B we show that iterative norm sampling is actually faster than exact leverage score sampling. In [20] a fast algorithm was proposed to approximately compute leverage scores of a large matrix. The algorithm has $O(n_1 n_2 \log n)$ time complexity when $n_1 \ll n_2$. However, on near square matrices the time complexity of the proposed algorithm reduces to $O(n^2 k)$. It is an interesting question to further speed-up iterative norm sampling so that it can handle large-scale data matrices.

References

[1] C. Boutsidis, M. Mahoney, and P. Drineas, "An improved approximation algorithm for the column subset selection problem," in *SODA*, 2009.

[2] L. Balzano, R. Nowak, and W. Bajwa, "Column subset selection with missing data," in *NIPS Workshop on Low-Rank Methods for Large-Scale Machine Learning*, 2010.

[3] T. F. Chan, "Rank revealing QR factorizations," *Linear Algebra and Its Applications*, vol. 88, pp. 67–82, 1987.

[4] M. Gu and S. C. Eisenstat, "Efficient algorithms for computing a strong rank-revealing QR factorization," *SIAM Journal on Scientific Computing*, vol. 17, no. 4, pp. 848–869, 1996.

[5] P. Drineas, R. Kannan, and M. W. Mahoney, "Fast monte carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition," *SIAM Journal on Computing*, vol. 36, no. 1, pp. 184–206, 2006.

[6] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, "Relative-error CUR matrix decompositions," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 844–881, 2008.

[7] A. Frieze, R. Kannan, and S. Vempala, "Fast monte-carlo algorithms for finding low-rank approximations," *Journal of the ACM*, vol. 51, no. 6, pp. 1025–1041, 2004.

[8] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang, "Matrix approximation and projective clustering via volume sampling," *Theory of Computing*, vol. 2, pp. 225–247, 2006.

[9] A. Deshpande and S. Vempala, "Adaptive sampling and fast low-rank matrix approximation," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2006, pp. 292–303.

[10] B. Recht, "A simpler approach to matrix completion," *The Journal of Machine Learning Research*, vol. 12, pp. 3413–3430, 2011.

[11] S. Chen, R. Varma, A. Singh, and Kovačević, "Signal recovery on graphs: Random versus experimentally designed sampling," *arXiv:1504.05427*, 2015.

[12] T. Yang, L. Zhang, R. Jin, and S. Zhu, "An explicit sampling dependent spectral error bound for column subset selection," in *ICML*, 2015.

[13] Y. Wang and A. Singh, "Provably correct active sampling algorithms for matrix column subset selection with missing data," *arXiv:1505.04343*, 2015.

[14] ——, "Column subset selection with missing data via active sampling," in *AISTATS*, 2015.

[15] J. Bien, Y. Xu, and M. Mahoney, "CUR from a sparse optimization viewpoint," in *NIPS*, 2010.

[16] I. Houtzager, "Rank-revealing QR factorization," http://www.mathworks.com/matlabcentral/fileexchange/18591-rank-revealing-qr-factorization, [Online; accessed 30-June-2015].

[17] A. Javed, P. Drineas, M. Mahoney, and P. Paschou, "Efficient genomewide selection of PCA-correlated tSNPs for genotype imputation," *Annals of Human Genetics*, vol. 75, no. 6, pp. 707–722, 2011.

[18] T. international HapMap consortium, "The international HapMap project," *Nature*, vol. 437, pp. 789–796, 2003.

[19] P. Paschou, M. Mahoney, A. Javed, J. Kidd, A. Pakstis, S. Gu, K. Kidd, and P. Drineas, "Intra- and interpopulation genotype reconstruction from tagging SNPs," *Genome Research*, vol. 17, no. 1, pp. 96–107, 2007.

[20] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff, "Fast approximation of matrix coherence and statistical leverage," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3475–3506, 2012.