DIMACS Technical Report 2006-04
March 2006

Polynomial Time Algorithm for Column-Row Based Relative-Error Low-Rank Matrix Approximation [1]

by

Petros Drineas [2] Michael W. Mahoney [3] S. Muthukrishnan [4]

[1] This DIMACS Technical Report is identical to our manuscript of November 3, 2005 with the same title that has already been distributed.

[2] Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York 12180, drinep@cs.rpi.edu.

[3] Yahoo Research Labs, Sunnyvale, California 94089, mahoney@yahoo-inc.com. Part of this work was performed while at the Department of Mathematics, Yale University, New Haven, Connecticut, USA 06520.

[4] Department of Computer Science, Rutgers University, New Brunswick NJ 08854, e-mail: muthu@cs.rutgers.edu.

ABSTRACT

Given an $m \times n$ matrix A and an integer k less than the rank of A, the best – with respect to the Frobenius norm – rank k approximation to A is A_k, which is obtained by truncating the Singular Value Decomposition (SVD) of A. While A_k is routinely used in data analysis, it is difficult to interpret and understand it in terms of the *original data*, namely the rows and columns of A which come from the application domain.

In this paper, we address the problem of obtaining low-rank approximations that are directly expressible in terms of the original rows and columns of A. Our main results are as follows. We present a randomized algorithm to determine a set C of columns, whose size is polynomial in $k, \log(1/\delta), 1/\varepsilon$, such that the matrix A' expressly written in terms of C satisfies

$$\|A - A'\|_F \leq (1+\varepsilon)\, \|A - A_k\|_F$$

with probability at least $1-\delta$. This is the first polynomial time algorithm for low-rank matrix approximation that gives relative error guarantees; all previously known methods including the seminal work of Frieze, Kannan and Vempala [14] yield approximations with a large additive term of $\varepsilon \|A\|_F$ and are improved by our result. We further extend this result to obtain a randomized algorithm to determine a set C of columns and a set R of rows, both polynomial in $k, \log(1/\delta), 1/\varepsilon$ in size, such that the matrix A' expressly written in terms of C and R satisfies

$$\|A - A'\|_F \leq (1+\varepsilon)\, \|A - A_k\|_F$$

with probability at least $1-\delta$. This is again the first polynomial time algorithm of this form with relative error guarantees; previously, even existence of such C and R were not known.

All our algorithms employ random sampling, but rather than sampling rows and columns of A as in prior work, we carefully use information from the top few singular vectors of A. This technique was recently introduced in [13] for the l_2 regression problem, and is significantly extended here. Our algorithms are quite simple, taking time of the order of the time needed to compute the SVD of A, and will likely be useful in practical applications.

1 Introduction

1.1 Overview

In many applications, the data are represented by a real $m \times n$ matrix A. Such a matrix may arise if the data consists of n objects, each of which is described by m features. Examples of objects include documents, genomes, stocks, hyperspectral images, and web groups, while examples of the corresponding features are terms, environmental conditions, temporal resolution, frequency resolution, and individual users. In each of these application areas, practitioners spend vast amounts of time analyzing the data in order to understand, interpret, and ultimately use this data. Often the central task in this analysis is to develop a compressed representation of A that may be easier to analyze and interpret. By far, the most common compressed representation is a low-*rank* approximation to the data matrix.

The *rank* ρ of a matrix $A \in \mathbb{R}^{m \times n}$ (WLOG $m \leq n$) is the maximum number of linearly-independent rows (or columns). Clearly, $\rho \leq m$. A *low-rank* approximation of A is any matrix B of rank $k \ll \rho$ that approximates A in some norm. The "best" such approximation is obtained using Singular Value Decomposition (SVD). The SVD of A is given by $A = U\Sigma V^T$, where $U \in \mathbb{R}^{m \times \rho}$, $\Sigma \in \mathbb{R}^{\rho \times \rho}$, and $V \in \mathbb{R}^{n \times \rho}$. The traditional method for low-rank approximation of A is to truncate the SVD to some $k \ll \rho$. That is, define the rank-k matrix $A_k = U_k \Sigma_k V_k^T$ where $U_k \in \mathbb{R}^{m \times k}$ is the first k columns of U, $\Sigma_k \in \mathbb{R}^{k \times k}$ is the top left $k \times k$ portion of Σ (kth principal submatrix of Σ), and $V_k \in \mathbb{R}^{n \times \rho}$ is the first k columns of V. It is a classical result that A_k is the best rank k approximation for A in any unitarily invariant norm; throughout this paper, we will work with the Frobenius norm ($||A||_F = \sqrt{\sum_{i,j} A_{ij}^2}$) which is one such norm. The SVD and hence the best rank-k approximation of a general matrix A can be computed in $O(n^2 m)$ time [16]. Henceforth, we generically use $SVD(A)$ to denote the time to compute the SVD of A.

This SVD-based low-rank approximation A_k is routinely used by data analysts. However, there is a fundamental difficulty with it: the new "dimensions" (the so-called eigencolumns and eigenrows) of A_k are linear combinations of the original dimensions. As such, they are difficult to interpret in terms of the original columns and rows. For example, the vector [0.3 age + 0.7 height] being one of the significant independent "feature" or "factor" of a dataset of people's features is not informative. This problem is fundamental to all applications of SVD. In addition, alternatives including weighted, structured, generalized SVD low-rank approximations suffer from the same problem. From an analyst's point of view, it would be highly preferable to have a low-rank approximation that is nearly as good as that provided by the SVD but that is expressed in terms of a small number of *actual* columns and *actual* rows of a matrix, rather than linear combinations of those columns and rows. In addition, in some applications, there are technical reasons why SVD based low-rank approximation is not desirable. For example, if the input data matrix is sparse, as is common with term-document matrices, then the orthogonalization provided by the SVD may destroy sparseness [22, 23, 3].

In this paper, we address the problem directly and focus on choosing columns and rows for dimensionality reduction of matrices via low-rank approximation. In the next subsection, we describe the two parts to our main result.

1.2 Problems and Our Main Results

Column-based k-rank approximation. Let A be an $m \times n$ matrix, and let $C = C(A)$ be an $m \times c$ matrix whose columns consist of a small number $c = c(k)$ of columns of the matrix A, for a given k. A *column-based low-rank approximation* to A is a matrix $A' = CX$, i.e., a matrix approximation that is explicitly written in terms of some columns of A. Note that, in general, A_k will not satisfy this condition. Notice that if $c = k$ then a lower bound for $||A - A'||_F$ is $||A - A_k||_F$. Rather than attempting to choose the "best" set of k columns, we will consider the problem of using $c \geq k$ columns, c small, so that $||A - A'||_F$ is close to this lower bound. A can be written explicitly as C and X for a total size of $O(mc + cn)$ saving over A's size of $O(nm)$. Our first result is as follows.

Theorem 1 *Let $\epsilon \in (0, 1]$. Given any $m \times n$ matrix A and any positive integer $k \ll$ **rank**(A), let A_k be the best rank k approximation to A. There exists a randomized algorithm that runs in $O(SVD(A))$ time and selects $c = O(k^2 \ln(3/\delta)/\varepsilon^2)$ columns of A such that, with probability at least $1 - \delta$:*

$$\|A - CX\|_F \leq (1 + \varepsilon) \|A - A_k\|_F ,$$

where C is the $m \times c$ matrix whose columns are the selected columns of A, and X is the associated $c \times n$ matrix $C^+ A$, where C^+ is the Moore-Penrose generalized inverse. [1]

This result improves two lines of research in theoretical computer science. First, the seminal work of Frieze, Kannan and Vempala from [14, 15], in our parlance, can be thought of as sampling columns from a matrix A to form a matrix C such that $||A - CX||_F \leq ||A - A_k||_F + \varepsilon ||A||_F$. Here the approximation is damped by a rather large additive term that depends on the Frobenius norm of the entire matrix A; hence, relative error approximation such as our result above is a significant improvement, and has been sought by the community for the past few years. A series of papers have followed [14] in the past 6 years including [1, 9, 10, 11]. All of them yield additive error approximations only. These previous results were motivated by massive data applications and sought pass-efficient algorithms, typically algorithms that made a single pass over the data. There has been recent progress in decreasing the magnitude of the additive error: [21, 7] and [12] show that by performing the random sampling in several "passes" over the matrix the additional additive error drops exponentially with the number of passes. Still, these results do not provide a relative error approximation such as the one we provide.

The second line of research that is relevant is the recent work by Deshpande, Rademacher, Vempala and Wang [21, 7] that can be interpreted as showing the *existence* of a roughly k^2 sized set of columns C with the relative error approximation in our theorem above. Their existence proof is obtained by a novel volume-sampling approach which relies on estimating

[1] The Moore-Penrose generalized inverse, or pseudoinverse, of A may be expressed in terms of the SVD as $A^+ = V_A \Sigma_A^{-1} U_A^T$ [20].

the volume of the simplex formed by each of the k-sized subsets of the columns. It is not clear how their methods can be modified to obtain an effective algorithm beyond enumerating all the exponential sets of a given size. Our result provides the first-known relative error column-based low-rank approximation in polynomial time. "Feature selection" is a broad area that addresses the choice of columns explicitly for dimension reduction, but the metrics there are typically optimization-based [6] or machine-learning based [5]. These formulations tend to have set-cover like solutions and are incomparable with the linear-algebraic structure such as the low-rank criteria we consider here that is the standard among data analysts.

Column-Row k-rank approximation Let A be an $m \times n$ matrix, let $C = C(A)$ be an $m \times c$ matrix whose columns consist of a small number $c = c(k)$ of columns of the matrix A, and let $R = R(A)$ be an $r \times n$ matrix whose rows consist of a small number $r = r(k)$ of rows of the original matrix A, for a given k. A *column-row based k-rank approximation* to A is a matrix of the form $A' = CUR$, for a matrix U of appropriate dimensions, i.e., a matrix approximation that is explicitly written in terms of a small number of columns and rows of A. The combined size of C, U and R is $O(mc + rn + cr)$ that is, as before, an improvement over A's size of $O(nm)$. As before, $c, r \ll n, m, \rho$, and we want to choose C and R and define a matrix U such that $||A - CUR||_F$ is as close as possible to $||A - A_k||_F$. Our second result is as follows:

Theorem 2 *Let $\epsilon \in (0, 1]$. Given any $m \times n$ matrix A and any positive integer $k \ll \rho$, let A_k be the best rank k approximation to A. There exists a randomized algorithm that runs in $O(SVD(A))$ time and constructs an $m \times c$ matrix C consisting of $c = O\left(k^2 \ln(6/\delta)/\epsilon^2\right)$ columns of A, an $r \times n$ matrix R consisting of $r = O\left(k^4 \ln^3(6/\delta)/\epsilon^6\right)$ rows of A, and a $c \times r$ matrix U such that, with probability at least $1 - \delta$: $\|A - CUR\|_F \leq (1 + \epsilon) \|A - A_k\|_F$.*

The line of research initiated by Frieze, Kannan and Vempala [14] has lead in successive papers to additive error approximations for matrix decompositions of the form $A \approx CUR$, but no relative error approximations. See [8] and also [11] for additive error approximations. This is the first known algorithm that achieves the relative error approximation. Indeed, unlike in the column-based low-rank approximation case, it was not even known if such C and R exist!

Our result also has technical applications. For example, the choice of columns and rows we present may be thought of as forming a "core set" [2] for approximate matrix computations which will have other applications. Also, the effective algorithm we provide for relative error approximation may also help improve algorithms for projective clustering. For example, in [7], the authors use an exhaustive enumeration algorithm for relative error matrix approximation to projective clustering, which can be replaced by our faster, polynomial time algorithm in certain cases.

Finally, we should note that in linear algebra community, there are several heuristics [22, 23, 3, 18, 17] to get C, U, R like ours for low-rank approximation, but none that is comparable in algorithms or in its proven guarantees.

1.3 Technical Overview of Our Analysis

Our goal is to obtain low-rank matrix approximation A' that is expressed in terms of the columns and/or rows of A and that are almost as good in a relative-error sense as A_k, the optimal approximation given by the SVD. We will choose C by sampling and rescaling columns. More precisely, $C = AS_CD_C$, where S_C is a column sampling matrix and D_C is a diagonal rescaling matrix; similarly, $R = D_RS_R^TA$ where S_R^T is a row sampling matrix and D_R is a diagonal rescaling matrix. We will set W to be the intersection between the chosen columns and rows, i.e., $W = D_RS_R^TAS_CD_C$. Then the column-based approximation will comprise $A' = CX$ where $X = C^+A$, and will satisfy our Theorem 1. Likewise, $A' = CW^+R$ will be the column-row approximation with $U = W^+$ and satisfy Theorem 2. The crux will be the random sampling method we will use and the proofs of the results above.

Random Sampling Method. Our algorithm relies on randomly sampling rows and columns. All previous work on low-rank matrix approximations beginning with the seminal [14] rely on random sampling too, and they typically sample based on row/column norms. Instead, we will sample, somewhat unusually, based on the singular vectors of A. The precise form of sampling is somewhat complicated and is shown in the proofs. However, it is similar to the sampling method we developed recently to solve the l_2 regression problem [13]. In order to construct sampling probabilities satisfying the conditions in [13] and the ones we need here which may be thought of as generalizations of the conditions in [13], it suffices to perform $O(\text{SVD}(A))$ computations, and faster procedure is not known. In the ℓ_2 regression problem, this is not interesting as an algorithm since $O(SVD(A))$ time suffices to solve the entire problem. In the present paper, however, sampling in $O(SVD(A))$ time will translate into a substantial improvement because there was no previously polynomial time algorithm for column-based rank approximation with relative error, and even existence result was not known for the column-row low-rank approximation.

Outline of Proofs. Given a matrix A and a set of its columns C, those columns clearly form a basis for the column space of C. If we want to get the best fit for all of the columns of A in terms of that basis, we want to solve the $CX \approx A$ for the matrix X. More precisely, we would like to solve the optimization problem:

$$\mathcal{Z} = \min_{X \in \mathbb{R}^{c \times n}} \|A - CX\|_F . \tag{1}$$

It is well-known that the matrix $X = C^+A$ is the "smallest" matrix among those that solve this problem. In this case, we are approximating the matrix A as $A' = CX = CC^+A = P_CA$ where $P_C = CC^+$ and by keeping only the columns C, we are incurring an error of $\|A - CC^+A\|_F$. Two questions arise: the first is how to choose the columns C so that $\|A - CC^+A\|_F$ is within relative error ϵ of $\|A - A_k\|_F$; the second is how to choose rows R such that $\|A - CW^+R\|_F$ is within relative error ϵ of $\|A - CC^+A\|_F$.

For the first, we will show that, given a matrix A, we can choose a set of columns C such that CC^+A captures almost as much of A as does A_k in a relative error sense.

We can also view it from an optimization perspective above, in which case we are given a matrix A and we are relaxing it by inserting a projection matrix P_{A_k} at the appropriate spot and we are choosing the column sampling matrix to get the right results. More precisely, let us make the following approximation: $X = C^+A \approx \left(P_{A_k}C\right)^+ P_{A_k}A$. This matrix X is clearly suboptimal with respect to solving (1) since $X = C^+A$ is optimal, i.e., $\|A - CC^+A\|_F \leq \left\|A - \left(P_{A_k}C\right)^+ P_{A_k}A\right\|_F$, but we can show that by choosing C properly, i.e., by choosing S_C and D_C (the column sampling and rescaling matrices) properly, we have that $\|A - CC^+A\|_F \leq (1+\epsilon)\,\|A - A_k\|_F$.

For the second, we will show that, given a matrix A and a set of its columns C chosen as above, we can choose a set of its rows R such that CW^+R captures almost as much of A as does P_CA in a relative error sense. This is a technical extension of our earlier ℓ_2-regression result [13] and is obtained by generalizing the ℓ_2 regression problem with a matrix of right hand side vectors (rather than just the one vector in the ℓ_2 regression problem). We can also view it from the optimization perspective, in which case we are given A and C and we are relaxing the exact solution by inserting a row-sampling matrix at the appropriate spot. More precisely, let us make the following approximation: $X = C^+A \approx \left(D_RS_R^TC\right)^+ D_RS_R^TA$, and note that $\left(D_RS_R^TC\right)^+ D_RS_R^TA = W^+R$. This matrix X is clearly suboptimal with respect to solving (1) since $X = C^+A$ is optimal, i.e., $\|A - CC^+A\|_F \leq \|A - CW^+R\|_F$, but we can show that by choosing D_R and S_R (the row sampling and rescaling matrices) properly we have that $\|A - CW^+R\|_F \leq (1+\epsilon)\,\|A - CC^+A\|_F$.

In the proof of both of these steps, the main technical challenge is to sample in a manner that captures *entirely* a certain subspace of interest. This is required since we will be performing operations such as pseudoinversion that are not well-behaved to missing a dimension, no matter how insignificant its singular vlaue is. This is different than sampling to capture coarse statistics upto additive $\varepsilon||F||_F$, and it necessitates the use of more complex probabilities and more sophisticated analysis. Our algorithms however are quite simple, and easy to implement since the main part is SVD computation which is a tried-and-tested routine.

2 Preliminaries

Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. For any matrix $A \in \mathbb{R}^{m\times n}$, let $A_{(i)}, i \in [m]$ denote the i-th row of A as a row vector, and let $A^{(j)}, j \in [n]$ denote the j-th column of A as a column vector. For any orthogonal matrix $U \in \mathbb{R}^{m\times\ell}$, let $U^\perp \in \mathbb{R}^{m\times(m-\ell)}$ denote an orthogonal matrix whose columns are an orthogonal basis spanning the subspace of $\mathbb{R}^m$ that is orthogonal to the subspace spanned by the columns of U. For more details on linear algebra, see [19, 16, 4]. A result on approximating the product of two matrices by random sampling will be used in an essential manner in this paper; it is described in detail in [9]. See Appendix A for a description relevant to this paper.

3 The Column-Based Low-Rank Approximation

We consider selecting columns for relative-error column-based low-rank matrix approximation.

3.1 The Algorithm for Column Selection

We describe a randomized algorithm that satisfies Theorem 1.

1. The algorithm computes the SVD of A

$$A = U_A \Sigma_A V_A^T = U_A \left[\begin{array}{cc} \Sigma_k & \mathbf{0} \\ \mathbf{0^T} & \Sigma_{\rho-k} \end{array} \right] \left[\begin{array}{c} V_k^T \\ V_{\rho-k}^T \end{array} \right]. \quad (2)$$

   In the above $\rho \leq m$ denotes the rank of A, $U_A \in \mathbb{R}^{m\times\rho}$, Σ_A is a $\rho \times \rho$ diagonal matrix, and $V_A \in \mathbb{R}^{n\times\rho}$. Also, $\mathbf{0}$ denotes a $k \times (\rho - k)$ matrix of zeros, Σ_k denotes the $k \times k$ diagonal matrix containing the top k singular values of A, $\Sigma_{\rho-k}$ denotes the $(\rho - k) \times (\rho - k)$ matrix containing the bottom $\rho - k$ singular values of A, V_k denotes the $n \times k$ matrix whose columns are the top k right singular vectors of A, and $V_{\rho-k}$ denotes the $n \times (\rho - k)$ matrix whose columns are the bottom $\rho - k$ right singular vectors of A.

2. The algorithm computes p_i, for all $i \in [n]$:

$$p_i = \frac{(1/3)\left|(V_k)_{(i)}\right|^2}{\sum_{j=1}^n \left|(V_k)_{(j)}\right|^2} + \frac{(1/3)\left|\left(\Sigma_{\rho-k}V_{\rho-k}^T\right)^{(i)}\right| \left|(V_k)_{(i)}\right|}{\sum_{j=1}^n \left|\left(\Sigma_{\rho-k}V_{\rho-k}^T\right)^{(j)}\right| \left|(V_k)_{(j)}\right|} + \frac{(1/3)\left|\left(\Sigma_{\rho-k}V_{\rho-k}^T\right)^{(i)}\right|^2}{\sum_{j=1}^n \left|\left(\Sigma_{\rho-k}V_{\rho-k}^T\right)^{(j)}\right|^2}. \quad (3)$$

   Notice that $\sum_{i\in[n]} p_i = 1$.

3. The algorithm picks c columns of A in c i.i.d. trials with replacement with respect to the probabilities p_i and returns the $m \times c$ matrix C containing the sampled columns. Equivalently, a sampling matrix S is constructed using Algorithm 1 with inputs the p_i of eqn. (3) and c. (Algorithm 1 also constructs a rescaling matrix D which is not necessary for the construction of C but is crucial for proving Theorem 1.) X can now be computed as C^+A in $O(SVD(A))$ time easily.

3.2 Proof of Theorem 1

We seek to bound the Frobenius norm of $A - CC^+A = A - AS\,(AS)^+\,A$. Toward that end, we introduce the diagonal rescaling matrix D in the above expression. Notice that scaling the columns of a matrix (equivalently, post-multiplying the matrix by a diagonal matrix) does not change the subspace spanned by the columns of the matrix. This simple observation holds for *any* possible rescaling; our careful choice for the rescaling matrix D will allow us to repeatedly apply Theorem 4 to bound the approximation error. Formally,

$$\begin{aligned}
\left\|A - CC^{+}A\right\|_F^2 &= \left\|A - (AS)(AS)^{+}A\right\|_F^2 = \left\|A - (ASD)(ASD)^{+}A\right\|_F^2 \\
&= \left\|U_A\Sigma_A V_A^T - \left(U_A\Sigma_A V_A^T SD\right)\left(U_A\Sigma_A V_A^T SD\right)^{+} U_A\Sigma_A V_A^T\right\|_F^2 \\
&= \left\|\Sigma_A - \left(\Sigma_A V_A^T SD\right)\left(U_A\Sigma_A V_A^T SD\right)^{+} U_A\Sigma_A\right\|_F^2 = \left\|\Sigma_A - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^{+} \Sigma_A
\end{aligned}$$

In the above we used the unitary invariance of the Frobenius norm and $\left(U_A\Sigma_A V_A^T SD\right)^{+} = \left(\Sigma_A V_A^T SD\right)^{+} U_A^T$. We now exploit the careful "disentangling" of the "top" singular subspace of A from the "bottom" one in eqn. (2). Thus,

$$\begin{aligned}
\left\|\Sigma_A - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^{+} \Sigma_A\right\|_F^2 &= \left\|\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix} - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^{+}\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix}\right\|_F^2 \qquad (4) \\
&+ \left\|\begin{bmatrix}\mathbf{0} \\ \Sigma_{\rho-k}\end{bmatrix} - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^{+}\begin{bmatrix}\mathbf{0} \\ \Sigma_{\rho-k}\end{bmatrix}\right\|_F^2 \qquad (5)
\end{aligned}$$

We bound the above two terms. Since $I - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^{+}$ is a projector matrix and may be dropped without increasing a unitarily invariant norm, and so the bound on eqn. (5) follows:

$$\left\|\left(I - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^{+}\right)\begin{bmatrix}\mathbf{0} \\ \Sigma_{\rho-k}\end{bmatrix}\right\|_F^2 \le \left\|\begin{bmatrix}\mathbf{0} \\ \Sigma_{\rho-k}\end{bmatrix}\right\|_F^2 = \|A - A_k\|_F^2 . \qquad (6)$$

We next bound the right-hand side term in eqn. (4). We seek an upper bound of the form $\epsilon\,\|A - A_k\|_F^2$, namely an upper bound that does not depend at all on *any* of the top k singular values of A. The observation that allows us to get such a bound is:

$$\left\|\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix} - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^{+}\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix}\right\|_F^2 = \min_{X \in \mathbb{R}^{c\times k}} \left\|\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix} - \left(\Sigma_A V_A^T SD\right)X\right\|_F^2 . \qquad (7)$$

This follows from standard least squares, since $\left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^{+}\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix}$ is the exact projection of the matrix $\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix}$ on the subspace spanned by the columns of $\Sigma_A V_A^T SD$. We consider a suboptimal – but very convenient for our purposes – choice for X, namely $X = \left(\Sigma_k V_k^T SD\right)^{+}\Sigma_k \in \mathbb{R}^{c\times k}$. From eqn. (7)

$$\left\|\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix} - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_A V_A^T SD\right)^{+}\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix}\right\|_F^2 \le \left\|\begin{bmatrix}\Sigma_k \\ \mathbf{0}\end{bmatrix} - \left(\Sigma_A V_A^T SD\right)\left(\Sigma_k V_k^T SD\right)^{+}\Sigma_k\right\|_F^2 . \qquad (8)$$

We want to emphasize that our suboptimal choice of X leads to a right hand side in the above equation that is easier to manipulate and bound. Indeed, our arguments do not seem to provide any bound for the left hand side of eqn. (8) directly.

Let the rank of the $k \times c$ matrix $V_k^T SD$ be $\tilde{k}$, and let its SVD be $V_k^T SD = U_{V_k^T SD} \Sigma_{V_k^T SD} V^T_{V_k^T SD}$, where $U_{V_k^T SD} \in \mathbb{R}^{k \times \tilde{k}}$, $\Sigma_{V_k^T SD} \in \mathbb{R}^{\tilde{k} \times \tilde{k}}$, and $V_{V_k^T SD} \in \mathbb{R}^{c \times \tilde{k}}$. Clearly $\tilde{k} \leq k$. The following lemma argues that, given our choice for the p_i, the rank of $V_k^T SD$ is equal to k and, even more, all the singular values of $V_k^T SD$ are close to 1.

Lemma 1 *Let $\epsilon \in (0,1]$. Using the sampling probabilities of eqn. (3), if $c \geq 576 k^2 \ln(3/\delta)/\epsilon^2$, then with probability at least $1-\delta/3$: $\tilde{k} = k$ i.e., $rank(V_k^T SD) = rank(V_k)$, $\left\| \Sigma_{V_k^T SD} - \Sigma^{-1}_{V_k^T SD} \right\|_2 \leq \epsilon/\sqrt{2}$, and $\left(\Sigma_k V_k^T SD\right)^+ = \left(V_k^T SD\right)^+ \Sigma_k^{-1}$.*

The proof of this lemma is adaptation of earlier work in [13] and is omitted. We manipulate the right hand side of eqn. (8) using eqn. (2) and lemma 1. $\left\| \begin{bmatrix} \Sigma_k \\ \mathbf{0} \end{bmatrix} - \left(\Sigma_A V_A^T SD\right) \left(\Sigma_k V_k^T SD\right)^+ \Sigma_k \right\|_F^2$ equals

$$
\begin{aligned}
&= \left\| \begin{bmatrix} \Sigma_k \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \Sigma_k & \mathbf{0} \\ \mathbf{0} & \Sigma_{\rho-k} \end{bmatrix} \begin{bmatrix} V_k^T \\ V_{\rho-k}^T \end{bmatrix} SD \left(V_k^T SD\right)^+ \right\|_F^2 \\
&= \left\| \begin{bmatrix} \Sigma_k \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \Sigma_k V_k^T \\ \Sigma_{\rho-k} V_{\rho-k}^T \end{bmatrix} SD \left(V_k^T SD\right)^+ \right\|_F^2 \\
&= \left\| \Sigma_k - \Sigma_k \underbrace{V_k^T SD \left(V_k^T SD\right)^+}_{=I_k} \right\|_F^2 + \left\| \Sigma_{\rho-k} V_{\rho-k}^T SD \left(V_k^T SD\right)^+ \right\|_F^2 \qquad (9) \\
&= \left\| \Sigma_{\rho-k} V_{\rho-k}^T SD \left(V_k^T SD\right)^+ \right\|_F^2 . \qquad (10)
\end{aligned}
$$

The first term of eqn. (9) is essentially the most important point of the proof. The p_i's are carefully constructed to guarantee that the $k \times c$ matrix $V_k^T SD$ has full rank and thus its columns – which are k-dimensional vectors – span $\mathbb{R}^k$. As a result, the projection of Σ_k on the subspace spanned by the columns of $V_k^T SD$ is equal to Σ_k. Notice that Σ_k does not appear in eqn. (10); at this point in the proof, we have removed any dependency of the error term on the top k singular values. We combine eqns. (4), (5), (6), (8), and (10) to get

$$
\left\| A - CC^+ A \right\|_F \leq \left\| A - A_k \right\|_F + \left\| \Sigma_{\rho-k} V_{\rho-k}^T SD \left(V_k^T SD\right)^+ \right\|_F . \qquad (11)
$$

(Note that we extracted the square root of both sides to get the above result.) We further manipulate the second term in the right hand side of eqn. (11) by replacing the pseudoinverse of $V_k^T SD$ by the transpose of the same matrix and introducing a small additional error. The following lemma, combined with lemma 1, motivates this manipulation by providing a small bound for the additional error.

Lemma 2 $\left\|\left(V_k^T SD\right)^+ - \left(V_k^T SD\right)^T\right\|_2 = \left\|\Sigma_{V_k^T SD}^{-1} - \Sigma_{V_k^T SD}\right\|_2$

The proof of this lemma is adaptation of earlier work in [13] and is omitted. Using this lemma, the triangle inequality, and for any two matrices A and B, $\|AB\|_F \leq \|B\|_2 \|A\|_F$, we have $\left\|\Sigma_{\rho-k} V_{\rho-k}^T SD \left(V_k^T SD\right)^+\right\|_F$

$$\begin{aligned} &\leq \left\|\Sigma_{\rho-k} V_{\rho-k}^T SD \left(V_k^T SD\right)^T\right\|_F + \left\|\Sigma_{\rho-k} V_{\rho-k}^T SD \left(\left(V_k^T SD\right)^+ - \left(V_k^T SD\right)^T\right)\right\|_F \\ &\leq \left\|\Sigma_{\rho-k} V_{\rho-k}^T SDDS^T V_k\right\|_F + \left\|\Sigma_{V_k^T SD}^{-1} - \Sigma_{V_k^T SD}\right\|_2 \left\|\Sigma_{\rho-k} V_{\rho-k}^T SD\right\|_F . \end{aligned} \qquad (12)$$

To bound eqn. (12) we will apply lemma 3. In words, the lemma states that we can accurately approximate the Frobenius norm of an arbitrary matrix Q by looking at a carefully sampled and appropriately rescaled subset of its columns. The lemma's proof is via a martingale argument. It is straightforward, but technical, and is omitted here.

Lemma 3 *Let* $\epsilon \in (0,1]$. *For any* $Q \in \mathbb{R}^{m \times n}$, *construct the sampling matrix* S *and the rescaling matrix* D *using Algorithm 1 with inputs* Q, c *and sampling probabilities* p_i *s.t.* $\sum_{i\in[n]} p_i = 1$ *and* $p_i \geq \frac{\beta \left|Q^{(i)}\right|_2}{\sum_{i\in[n]} \left|Q^{(i)}\right|_2}$, *for some* $\beta \in (0,1]$. *If* $c \geq 16 \ln(3/\delta)/\left(\beta^2 \epsilon^2\right)$, *then with probability at least* $1 - \delta/3$:

$$\|QSD\|_F \leq (1 + \sqrt{\epsilon}) \|Q\|_F .$$

Lemma 4 *Let* $\epsilon \in (0,1]$. *If* $c \geq 144k \ln(3/\delta)/\epsilon^2$ *then, with probability at least* $1 - \delta/3$: $\left\|\Sigma_{\rho-k} V_{\rho-k}^T SD\right\|_F \leq (1 + \sqrt{\epsilon}) \|A - A_k\|_F$.

To prove this lemma simply apply lemma 3 with $Q = \Sigma_{\rho-k} V_{\rho-k}^T$ and notice that our choice of the p_i (eqn. (3)) satisfies the constraint of the lemma with $\beta = 1/3$. (Also, we use $\left\|\Sigma_{\rho-k} V_{\rho-k}^T\right\|_F = \|A - A_k\|_F$.) Finally, lemma 5 bounds the first term in eqn. (12).

Lemma 5 *Let* $\epsilon \in (0,1]$. *If* $c \geq 144k \ln(3/\delta)/\epsilon^2$ *then, with probability at least* $1 - \delta/3$:

$$\left\|\Sigma_{\rho-k} V_{\rho-k}^T SDDS^T V_k - \underbrace{\Sigma_{\rho-k} V_{\rho-k}^T V_k}_{=\mathbf{0}}\right\|_F \quad \leq \quad \epsilon \|A - A_k\|_F . \qquad (13)$$

Proof. We apply Theorem 4 for approximating the product of two matrices to the matrices $\Sigma_{\rho-k} V_{\rho-k}^T$ and V_k, noticing that the chosen p_i satisfy the constraint of equation (20) with $\beta = 1/3$. The lemma follows by our choice of c and $\left\|\Sigma_{\rho-k} V_{\rho-k}^T\right\|_F = \|A - A_k\|_F$, $\|V_k\|_F = \sqrt{k}$. ∎

Let $c \geq 576k^2 \ln(3/\delta)/\epsilon^2$ and notice that with this choice of c, lemma 1, lemma 4, and lemma 5 hold simultaneously with probability at least $1 - 3(\delta/3) = 1 - \delta$. We condition the following derivations on the event that all three lemmas hold. First, from eqn. (12) using lemmas 1, 4, and 5 we get

$$\begin{aligned}\left\|\Sigma_{\rho-k}V_{\rho-k}^T SD\left(V_k^T SD\right)^+\right\|_F &\leq \left\|\Sigma_{\rho-k}V_{\rho-k}^T SDDS^T V_k\right\|_F + \left\|\Sigma_{V_k^T SD}^{-1} - \Sigma_{V_k^T SD}\right\|_2 \left\|\Sigma_{\rho-k}V_{\rho-k}^T SD\right\|_F \\ &\leq \left(\epsilon + \frac{\epsilon}{\sqrt{2}}(1+\sqrt{\epsilon})\right)\left\|A - A_k\right\|_F \leq 2.5\epsilon \left\|A - A_k\right\|_F.\end{aligned}$$

(Notice that $\epsilon \leq 1$.) Combining with eqn. (11) we get $\left\|A - CC^+A\right\|_F \leq (1+2.5\epsilon)\left\|A - A_k\right\|_F$ which concludes the proof of Theorem 1 by letting $\epsilon' = \epsilon/2.5$ and adjusting c accordingly.

4 The Column-Row-Based Low-Rank Approximation

In this section, we consider selecting rows for relative-error column-row-based low-rank matrix approximation. As a preliminary step, we present and analyze an algorithm for choosing a good set of rows, i.e. a good matrix R, assuming that a set of columns, i.e., a matrix C has been given.

4.1 Algorithm for Selecting Rows, given columns C

Consider the following idea for approximating a matrix $A \in \mathbb{R}^{m \times n}$. Assume that we are given any set of c columns of A, forming a matrix $C \in \mathbb{R}^{m \times c}$. Consider the columns of C as a set of "basis vectors" that are, in general, neither orthogonal nor normal. We will express all the columns of A as linear combinations of the columns of C. If m and n are large and $c = O(1)$, then this is an overconstrained least-squares fit problem. Thus, for all columns $A^{(j)}, j \in [n]$, we can solve

$$\min_{x_j \in \mathbb{R}^c} \left|A^{(j)} - Cx_j\right|_2 \tag{14}$$

in order to find a c-vector of coefficients x_j and get the optimal least-squares fit for $A^{(j)}$. Equivalently, we seek to solve

$$\left\|A - CX_{opt}\right\|_F = \min_{X \in \mathbb{R}^{c \times n}} \left\|A - CX\right\|_F \tag{15}$$

in order to express A as $A \approx CX_{opt}$. Notice that $X_{opt} \in \mathbb{R}^{c \times n}$ is a matrix whose columns are the coefficient vectors $x_j, j \in [n]$ that minimize equation (14); also, $\left\|A - CX_{opt}\right\|_F = \left\|A - CC^+A\right\|_F$.

We will now modify the above approach to get a CUR decomposition for the matrix A, given C. Toward that end, we will use a generalization of the ideas in [13]. Instead of solving the generalized least squares problem of eqn. (15) we will solve a *sampled* version of the problem, constructed as follows.

1. We compute the SVD of C, $C = U_C \Sigma_C V_C^T$, where $U_C \in \mathbb{R}^{m \times \rho}$, $\Sigma_C \in \mathbb{R}^{\rho \times \rho}$, $V_C \in \mathbb{R}^{c \times \rho}$, and ρ is the rank of C.

2. We compute p_i for all $i \in [m]$:

$$p_i = \frac{(1/3)\left|(U_C)_{(i)}\right|_2^2}{\sum_{j=1}^n \left|(U_C)_{(j)}\right|_2^2} + \frac{(1/3)\left|(U_C)_{(i)}\right|_2 \left|\left(U_C^{\perp} U_C^{\perp T} A\right)_{(i)}\right|_2}{\sum_{j=1}^n \left|(U_C)_{(j)}\right|_2 \left|\left(U_C^{\perp} U_C^{\perp T} A\right)_{(j)}\right|_2} + \frac{(1/3)\left|\left(U_C^{\perp} U_C^{\perp T} A\right)_{(i)}\right|_2^2}{\sum_{j=1}^n \left|\left(U_C^{\perp} U_C^{\perp T} A\right)_{(j)}\right|_2^2}. \tag{16}$$

   (Notice that $\sum_{i\in[n]} p_i = 1$.)

3. We create the sampling matrix S and the rescaling matrix D by running Algorithm 1 with inputs a positive integer $r \leq m$ (to be specified in Theorem 3) and the p_i of eqn. (16).

4. We return as output the matrices $R = S^T A \in \mathbb{R}^{r\times n}$ and $U = \left(DS^T C\right)^+ D \in \mathbb{R}^{c\times r}$.

The time required to compute the SVD of C is $O(c^2 m)$; computing the probabilities p_i of eqn. (16) takes an additional $O(cmn)$ time. Overall, the running time of the algorithm is $O(mn)$ since c, r are constants independent of m, n.

We now provide some intuition on the construction of U and R. Consider the following "sampled and rescaled" version of eqn. (15):

$$\left\|DS^T A - DS^T C\tilde{X}_{opt}\right\|_F = \min_{X\in\mathbb{R}^{c\times n}} \left\|DS^T A - DS^T CX\right\|_F. \tag{17}$$

We will essentially demonstrate that solving the "sampled and rescaled" problem and substituting its solution back to the original problem provides a simple CUR decomposition with a provable error bound. First, notice that $\left\|DS^T A - DS^T C\tilde{X}_{opt}\right\|_F = \left\|DS^T A - DS^T C\left(DS^T C\right)^+ DS^T A\right\|_F$ and thus we let $\tilde{X}_{opt} = \left(DS^T C\right)^+ DS^T A$. We will use $\tilde{X}_{opt}$ as an approximation to X_{opt} (which achieves the optimal value for the full problem of eqn. (14)) and bound the error

$$\left\|A - C\tilde{X}_{opt}\right\|_F = \left\|A - C\underbrace{\left(DS^T C\right)^+ D}_{U}\underbrace{S^T A}_{R}\right\|_F = \left\|A - CUR\right\|_F. \tag{18}$$

Let $W = S^T C$ be the $r \times c$ matrix that corresponds to rows of C that are in R – equivalently, W contains the common elements of C and R. C consists of a few columns of A, R consists of a few rows of the matrix A, W consists of the elements of A that are both in C and R, and D is an $r \times r$ diagonal rescaling matrix. In general, $(DW)^+ D \neq W^+$.

The following theorem is our main quality-of-approximation theorem for computing a CUR decomposition to A given C. Its proof is a generalization of [13] and is deferred to the full version of the paper.

Theorem 3 *Let $\epsilon \in (0, 1]$. Given a $m \times n$ matrix A and an $m \times c$ matrix C consisting of c columns of A, there exists a randomized algorithm that runs in $O(mn)$ time and constructs an $r \times n$ matrix R consisting of $r = 1440c^2 \ln(3/\delta)/\epsilon^2$ rows of A and a $c \times r$ matrix U such that, with probability at least $1 - \delta$: $\|A - CUR\|_F \leq (1+\epsilon)\|A - CC^+ A\|_F$.*

4.2 Proof of Theorem 2

We now combine the results of Theorems 1 and 3 in order to obtain a relative-error approximation for a matrix that is explicitly described in terms of a small number of columns and rows of the matrix. Equivalently, we run the algorithm of Section 3 to construct a matrix C consisting of c columns of A such that $\left\|A - CC^{+}A\right\|_F \leq (1+\epsilon) \left\|A - A_k\right\|_F$, and then we run the algorithm of Section 4.1 to construct matrices U and R, given the computed C. If we let $c = \frac{1440k^2 \ln(3/\delta)}{\epsilon^2}$ and $r = \frac{1440^2 k^4 \ln^3(3/\delta)}{\epsilon^6}$ then, with probability at least $1 - 2\delta$,

$$\left\|A - CUR\right\|_F \leq (1+\epsilon)^2 \left\|A - A_k\right\|_F \leq (1+3\epsilon) \left\|A - A_k\right\|_F . \qquad (19)$$

The proof of Theorem 2 follows by letting $\delta' = \delta/2$, $\epsilon' = \epsilon/3$, and adjusting c and r accordingly.

5 Concluding Remarks

To the best of our knowledge, this is the first result addressing the problems of selection of columns and rows to get a low-rank matrix approximation expressed in terms of those columns and rows in polynomial time. We conclude with several open problems.

- To what extent do the results of the present paper generalize to other matrix norms.
- What hardness results can be established for the optimal choice of columns and/or rows.
- Does there exist a deterministic any factor approximation algorithm to any of these problems.

References

[1] D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 611–618, 2001.

[2] P.K. Agarwal, S. Har-Peled, and K.R. Varadarajan. Geometric approximation via coresets - survey. *Manuscript.*

[3] M.W. Berry, S.A. Pulatova, and G.W. Stewart. Computing sparse reduced-rank approximations to sparse matrices. Technical Report UMIACS TR-2004-32 CMSC TR-4589, University of Maryland, College Park, MD, 2004.

[4] R. Bhatia. *Matrix Analysis.* Springer-Verlag, New York, 1997.

[5] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[6] M. Charikar, R. Kumar V. Guruswami, S. Rajagopalan, and A. Sahai. Combinatorial feature selection problems. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 631–642, 2000.

[7] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 000–000, 2006.

[8] P. Drineas and R. Kannan. Pass efficient algorithms for approximating large matrices. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 223–232, 2003.

[9] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. Technical Report YALEU/DCS/TR-1269, Yale University Department of Computer Science, New Haven, CT, February 2004.

[10] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. Technical Report YALEU/DCS/TR-1270, Yale University Department of Computer Science, New Haven, CT, February 2004.

[11] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. Technical Report YALEU/DCS/TR-1271, Yale University Department of Computer Science, New Haven, CT, February 2004.

[12] P. Drineas and M.W. Mahoney. A randomized algorithm for a tensor-based generalization of the Singular Value Decomposition. Technical Report YALEU/DCS/TR-1327, Yale University Department of Computer Science, New Haven, CT, June 2005.

[13] P. Drineas, M.W. Mahoney, and S. Muthukrishnan. Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 000–000, 2006.

[14] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 370–378, 1998.

[15] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.

[16] G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1989.

[17] S.A. Goreinov and E.E. Tyrtyshnikov. The maximum-volume concept in approximation by low-rank matrices. *Contemporary Mathematics*, 280:47–51, 2001.

[18] S.A. Goreinov, E.E. Tyrtyshnikov, and N.L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and Its Applications*, 261:1–21, 1997.

[19] R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, New York, 1985.

[20] M.Z. Nashed, editor. *Generalized Inverses and Applications.* Academic Press, New York, 1976.

[21] L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via iterative sampling. Technical Report MIT-LCS-TR-983, Massachusetts Institute of Technology, Cambridge, MA, March 2005.

[22] G.W. Stewart. Four algorithms for the efficient computation of truncated QR approximations to a sparse matrix. *Numerische Mathematik*, 83:313–323, 1999.

[23] G.W. Stewart. Error analysis of the quasi-Gram-Schmidt algorithm. Technical Report UMIACS TR-2004-17 CMSC TR-4572, University of Maryland, College Park, MD, 2004.

A Review of Matrix Multiplication

The following is a result on approximating the product of two matrices by random sampling will be used in an essential manner in this paper; it is described in detail in [9]. For simplicity of presentation we have slightly modified the statement of the theorem for the purposes of this paper; see Theorem 2.1 of [13] for details.

```
Data   : $p_i \geq 0, i \in [n]$ s.t. $\sum_{i \in [n]} = 1$, positive integer $c \leq n$.
Result : sampling matrix $S$, rescaling matrix $D$.
$S = \mathbf{0}_{n \times c}$;
$D = \mathbf{0}_{c \times c}$;
for $t = 1, \ldots, c$ do
    Pick $i_t \in [n]$, where $\mathbf{Pr}(i_t = i) = p_i$;
    $S_{i_t t} = 1$;
    $D_{tt} = 1/\sqrt{c p_{i_t}}$;
end
```

Algorithm 1: Creating the sampling matrix S and the rescaling matrix D.

Theorem 4 *Given $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{n\times p}$, $c \in \mathbb{Z}^+$ such that $c \leq n$, and sampling probabilities $\{p_i\}_{i=1}^n$ such that $\sum_{i\in[n]} p_i = 1$ and*

$$p_i \geq \beta \frac{\left|A^{(i)}\right|_2 \left|B_{(i)}\right|_2}{\sum_{j=1}^n \left|A^{(j)}\right|_2 \left|B_{(j)}\right|_2} \qquad (20)$$

(for some $\beta \in (0,1]$) construct the sampling matrix S and the rescaling matrix D using Algorithm 1. Assume that $\delta \in (0,1/3)$. Then, with probability at least $1-\delta$:

$$\left\|AB - ASDDS^TB\right\|_F \leq \frac{4\sqrt{\ln(1/\delta)}}{\beta\sqrt{c}} \left\|A\right\|_F \left\|B\right\|_F .$$