Pass Efficient Algorithm for approximating large matrices

Petros Drineas
drineas@cs.yale.edu

Ravi Kannan
kannan@cs.yale.edu

February 5, 2002

1 Summary

In many applications, an $m \times n$ matrix A is stored on disk and is too large to be read into RAM. Our main result is a succinct easily computed approximation A' to A which is also an $m \times n$ matrix. To be precise, A' has the following properties : (s below is a natural number under our choice. It usually will be $O(1)$.)

(i) $A' = CUR$, where C is an $m \times s$ matrix consisting of s (randomly picked) columns of A; R is an $s \times n$ matrix consisting of s (randomly picked) rows of A and U is an $s \times s$ matrix computed from C, R.

(ii) C, U, R can be constructed after making two passes through the whole matrix A from disk,

(iii) using RAM space and additional time (in addition to the two full passes) $O(m + n +$ the number of nonzero entries in $C, R)$ and

(iv)
$$\text{satisfies } \max_{x:|x|=1} |(A - A')x|^2 \leq \epsilon \sum_{i,j} A_{ij}^2.$$

(v) satisfies an upper bound on $||A - A'||_F$ (to be described later.) This upper bound is much smaller than $||A||_F$ when A has a good low-rank approximation, as is the case in many practical applications. (Note $|| \cdot ||_F^2$ is the sum of squares all the entries of the matrix and is called the Frobenius norm.)

We will also give simple information theoretic arguments to show that this is in essence the best we can do. This approximation can be used for "similarity query" problems, (which are widely used in Information Retrieval and other areas) where after A has been preprocessed, we get "query" vectors x and are to find the similarity of x to each row of A - where similarity of two vectors is defined to be their dot product (or normalized dot product, which also, we can handle).

Using an alternative method (based on the ideas of Achlioptas and McSherry [1]) we also show another approximation A'' which can be computed in just one pass, but has the disadvantage that it does not satsify (v) and indeed, always has $||A - A''||_F$ bigger than $||A||_F$.

We formulate a model of computation of "out-of-core" computation, which emphasizes the number of passes through the data from disk. This model has some similarities to the "Streaming Model" as well as older models studied in the context of sorting.

Our algorithm uses adaptive sampling where we take a small sample of the data, but with non-uniform probabilities which reflect the relative sizes of the entries. Uniform sampling has been recently shown to be useful in solving approximately several problems, like the maximum cut problem on dense graphs using only $O(1)$ space. We show some extensions of these results to certain non-dense graphs using adaptive sampling.

Notation $A^{(i)}$ will denote the i th row of matrix A (as a row vector) and $A_{(j)}$ will denote the j th column (as a column vector). $||A||_F, ||A||_2$ will denote the Frobenius and 2-norm as defined above.

2 Introduction

We consider the problem of deriving a succinct approximation A' to an $m \times n$ matrix A stored on disk. It is easy to see by information theory arguments that if one requires $\|A - A'\|_F^2 \leq \epsilon \|A\|_F^2$, then we will in general need Ω(number of non-zero entries in A) space. But in many applications, we only need to compute Ax for "query vectors" x and thus a more natural measure of the approximation in these cases in the 2-norm (denoted $\|\cdot\|_2$) of $A - A'$, namely $\max_{x:|x|=1} |(A - A')x|$ on which we prove a bound of $\sqrt{\epsilon}\|A\|_F$ as stated in the Summary. Obviously, such a bound is only useful for matrices A for which $\|A\|_2^2 >> \epsilon\|A\|_F^2$. This is indeed the case for matrices occurring in many contexts (namely, matrices for which so-called Principal Component Analysis is used.) But, we also prove a good upper bound on $\|A - A'\|_F$ for the more restricted class of matrices A for which, there exists a very good approximation of low rank.

The measure $\max_{x:|x|=1} |(A - A')x|$ is a worst case measure (because of the max). But this is more useful in many contexts than an average case measure, because the relevant query x often comes from a small dimensional subspace and is not random.

Two quick examples are in order - one is the "document-term" matrix, where we have a collection of m documents and n terms (used in the documents) and A_{ij} may represent the number of occurrences of term j in document i or a function of this number. A second example pertains to a collection of images where the rows of the matrix represent images and columns represent pixels and A_{ij} gives the intensity of pixel j in image i. We postpone further discussion of applications to the full version; but in general, these techniques can be used in the broad area of Principal Component Analysis where this may substitute for Singular Value Decomposition.

Our approximation A' is of the form CUR, where R is an $r \times n$ matrix consisting of r $(=\Theta(1/\epsilon^2))$ rows of A picked independently at random and C is an $m \times c$ matrix consisting of c $(=\Theta(1/\epsilon^2))$ columns of A picked independently at random. U is a $c \times r$ matrix which can be computed from C, R. Assuming $\epsilon \in \Omega(1)$, the picture looks like

$$\left(\begin{array}{ccc} & & \\ & & \\ & A & \\ & & \\ & & \end{array} \right) \approx \left(\begin{array}{c} \\ \\ C \\ \\ \\ \end{array} \right) \cdot \left(\begin{array}{c} U \end{array} \right) \cdot \left(\begin{array}{ccc} & R & \end{array} \right). \tag{1}$$

Note that the length of our succinct approximation is $O((m + n)/\epsilon^2)$. In case the matrix A is sparse, with at most m' entries in any column and at most n' entries in any row, then we develop an approximation of length (of representation) at most $O((m'+n')/\epsilon^2)$. [We will also give a simple information theoretical argument giving essentially a lower bound of $\Omega(m+n)$.] The random sampling to get R, C will be according to a carefully chosen probability distribution (not necessarily the uniform.)

We also show that the approximation only requires the input matrix to be presented in a particular general form - which we call the **unordered sparse** representation, in which (only) the non-zero entries are presented as triples (i, j, A_{ij}) in any order. This is suited to applications, where multiple agents may write in parts of the matrix to a central database and we cannot make assumptions about the rules for write-conflict resolution. [One example of this may be the "load" matrix, where each of many routers writes into a central database a log of the messages it routed during a day in the form of triples (source, destination, number of bytes).]

Another aspect of the approximation is that it can be viewed as a way to reconstruct an approximation to the whole matrix A given a randomly chosen subset of columns and a randomly chosen subset of rows. But we caution that as our theorem stands, it needs to know the probabilities that each sampled row and column was picked (up to a scale factor). (But note that these probabilities should be known just for the randomly picked rows and columns, not for all rows and columns.) Our decomposition has already been used for competitive recommendation systems [6] to reconstruct a large matrix. Also, as a by-product of the SR decomposition, we can estimate the singular values of A.

Our approximation may be viewed as a "dimension reduction" technique. The two main known techniques for dimension reduction which have been widely used are Random Projections [[18],[17], [21]] and Singular Value Decomposition (SVD) [[15],[13],[20], [7]]. These methods do not share (i) and (ii) and are thus not suited for very large problems. Our algorithm achieves (i) and (ii) at the cost of some accuracy, namely the $\epsilon\|A\|_F^2$ error.

Our algorithm draws a random sample of the entries of A and analyzes the sample. However, the difference between usual sampling algorithms and ours is that we do not blindly draw a random sample; what we do may be called adaptive sampling, where the sampling probabilities depend on the entries. In the first pass, we pick out a sample of rows and columns; the second pass is used to pick up the required sub matrices which form the whole sample. Then, we compute in RAM with the sample.

We remark that recently, there have been a number of results demonstrating the power of just blind sampling [4],[10], [11],[14],[2]; i.e., sampling where the probabilities are not based on the data. A special case of blind sampling is uniform sampling is where we uniformly at random pick a subset of s rows and columns of A and compute with the $s \times s$ matrix so chosen. The advantage of blind sampling is that we may pick the sample before seeing the data, then in just one pass through the data, we may extract the sample and analyze it. So, these are one-pass algorithms and fit the well-studied "Streaming Data" model (which we discuss later.) For example, one central problem the above papers show can be solved by blind sampling is the following : given the adjacency matrix of a graph G, find the value of maximum cut in G within **additive error** ϵn^2. This is useful only for dense graphs.

We will show here that in our model, with 2 passes, we can tackle some interesting class of non-dense graphs. In a sense, the algorithm we give can be viewed as exploiting the ability to sample a random edge in the graph, rather than a random vertex. What we show is that with 2 passes, and $O(\log n)$ extra RAM space and time, we can find in any graph $G(V, E)$, the max-cut to additive error $\epsilon(|V| - l)^2$, where the l lowest vertex degrees add up to $\epsilon|E|/2$. [Thus, if all but

$r \in o(n)$ vertices in the graph have "low" degrees, then the error is ϵr^2, instead of ϵn^2.] We will tackle a wider class of problems of which max-cut is an example.

3 The Pass Efficient Model

The only access we assume to the matrix is via a pass which is a sequential read of the entire input (from disk). We will now introduce the informal computational model we use. In modern computers, the amount of disk storage (sequential access memory) has increased enormously while RAM and computing speeds have increased, but at a substantially slower pace. Thus, we have the ability to store very large amounts of data, but not in RAM, and also we do not have the ability to process this data with algorithms which may take low polynomial time, or even linear time with large constants. To model this reality, we propose a model of computation in which we allow a small number (for example 2) of passes (sequential reads) through the entire data plus sub-linear computation time and Random Access Memory space. We will call these algorithms PASS EFFICIENT ALGORITHMS.

This model assumes the input data can only accessed by making passes through it, where a pass consists of one sequential read of the entire data plus additional computation time of at most q units after each r bits of data are read. The number r reflects the fact that usually one reads whole blocks of data, not just one bit. The quantity q is there for the following reason - input / output operations (from out of core) take a lot of cycles, so if q is sufficiently small - compared to the number of cycles needed to read in a block - then with a small increase to the running time, we can process the block just read; we will not usually explicitly mention the values of q, r, but the spirit is that a pass is just one read of the data with a small amount of processing. [Note that since additional space is only sub-linear, we really have to allow some processing during a pass, at least to figure out which parts of the input to retain in memory. Otherwise, since we cannot by far store all the data read in a pass, the pass would just be wasted.]

The algorithm is also allowed to use some computation time and Random Access Memory space besides the passes. These are required to be (and often will be substantially) sub-linear. We will measure three parameters of the algorithm - the number of passes, additional time and space. A model where the number of passes was measured as a basic parameter was first used by Munro and Paterson ([19]) in the special case of sorting and selection algorithms; but their definition of a pass allowed additional computation of essentially linear time (which as argued here is not practical for our problems.)

The "Streaming Model" [[16],[5], [3],[9],[17]] on which there is substantial work allows only one pass through the data and restricts the RAM usage to polylogarithmic amount of space. But the model formalized in ([9]) allows polylogarithmic time for processing after reading each bit; so a pass for them is even more generous than the model in ([19]). [We call this a "leisurely" pass.] So, the primary concern of both these models is the additional space (RAM) usage, rather than time. The restriction to one pass in the streaming model allows processing vast amounts of data supplied from other sources which we do not have space to store at all, but which we can "leisurely" look at from a read only 1-way tape taking super-linear time ($O(n)$ times polylog) for the pass. We will compare the models in more detail in the final paper.

3.1 Lower Bound

Lemma 1 *For any m, n positive integers, and $1 > \epsilon > \frac{8}{\sqrt{n}} + \frac{8}{\sqrt{m}}$, there is a set of $\Omega(\epsilon^{-n-m})$ $m \times n$ matrices, such that*

- *Each matrix in the set has Frobenius norm at most 4.*
- *Each entry of each matrix in the set is an integer multiple of $\epsilon/64\sqrt{mn}$.*
- *For two distinct matrices A, A' in the set, we have $||A - A'||_2 \geq \epsilon/80$.*

We give the idea of the simple, but technical proof in the appendix. The lemma supplies essentially a lower bound, since any algorithm which approximates these matrices must output a different approximation to each one, requiring it to output at least $O((n+m)\log(1/\epsilon))$ bits. This is only a lower bound on the number of bits of output.

Remark Random projections will not do : by the lower bound, with $o(m+n)$ description length, we can only achieve an error bound of $\Omega(\epsilon||A||_F)$. If we do a random projection of each row to an s dimensional space, to get a matrix B and say for a particular unit length vector x, its projection is x', the we only get that with high probability, $|A^{(i)}x - B^{(i)}x'| \leq \epsilon|A^{(i)}|$. Squaring and adding over all the rows, we get that

$$|Ax - Bx'| \leq \epsilon||A||_F,$$

holds with high probability for each x. So, for "most" x 's the inequality holds. But this is no use, since, for most x 's, we may have $|Ax| \leq \epsilon||A||_F$. The only x ' s which matter are of small measure.

4 The CUR decomposition

The main technical result of the paper is stated and proved in this section. For any $m \times n$ matrix A, suppose C is a $m \times s$ matrix formed by a random subset of s columns of A, picked in s independent identically distributed trials in each of which one of n columns of A is picked with the probability of picking column j being q_j. [The $\{q_j\}_{j=1}^n$ are nonnegative reals adding up to 1, to be specified later; s is a positive integer. Our upper bounds on the errors will depend on s and will decrease as s increases.] Similarly, suppose R is a $s \times n$ matrix formed by a random subset of s rows of A, picked in s independent identical trials, in each of which one of the m rows of A is picked with the probability of picking row i being p_i. [Again, $\{p_i\}_{i=1}^m$ are nonnegative reals adding up to 1, to be specified later.] The main theorem will say that from C, R, we can compute a $s \times s$ matrix U such that CUR is a good approximation to A provided the $\{p_i\}$ and the $\{q_j\}$ satisfy certain conditions; intuitively, these say heavier rows and columns should have higher probabilities of being picked. To put our theorem in context, it will be useful to contrast it with Singular Value Decomposition (SVD). The SVD of A expresses A as (here ρ is the rank of A)

$$A = \sum_{t=1}^{\rho} \sigma_t(A) u^{(t)} v^{(t)^T}, \qquad \text{where, } \sigma_1(A) \geq \sigma_2(A) \geq \ldots \sigma_\rho(A) > 0,$$

$$\text{where, } \{u^{(t)}\} \text{ and } \{v^{(t)}\} \text{ are each an orthonormal family.}$$

It is well-known that taking the first k terms of this expansion gives us the best rank k approximation to A in both the 2-norm and the Frobenius norm :

$$||A - \sum_{t=1}^{k} \sigma_t(A) u^{(t)} {v^{(t)}}^T ||_2 = \text{Min}_{B:\text{rank}(B)\leq k} ||A - B||_2 = \sigma_{k+1}(A)$$

$$||A - \sum_{t=1}^{k} \sigma_t(A) u^{(t)} {v^{(t)}}^T ||_F^2 = \text{Min}_{B:\text{rank}(B)\leq k} ||A - B||_F^2 = \sum_{t=k+1}^{\rho} \sigma_t(A)^2.$$

Note that we can write : $\sum_{t=1}^{k} \sigma_t(A) u^{(t)} {v^{(t)}}^T$ as $U\Sigma V$, where U is $m \times k$, Σ is a $k \times k$ diagonal matrix and V is a $k \times n$ matrix. So, computing the SVD gives us good succinct approximations, since it only takes space $O(k(m+n))$ to write down U, Σ, V. But the computational problem of finding the SVD is difficult and cannot be carried out by any means in $O(1)$ passes. Instead our theorem will say that weaker bounds (which are however similar in spirit) may be achieved by CUR (which is similar to $U\Sigma V$, but now U is not necessarily a diagonal matrix.) Indeed, we will show that if the rows and columns are picked with probabilities proportional to their length squared, then the bounds given by the following corollary on the errors hold.

Corollary 1 *If we have $p_i = |A^{(i)}|^2/||A||_F^2$ and $q_j = |A_{(j)}|^2/||A||_F^2$ for $i = 1, 2, \ldots m$ and $j = 1, 2, \ldots n$, then we have (E below stands for the expected value)*

$$E(||A - CUR||_2^2) \leq \sigma_{\sqrt{s}+1}(A)^2 + \frac{3}{\sqrt{s}} ||A||_F^2$$

$$E(||A - CUR||_F^2) \leq 16 \frac{2}{s^{1/4}} ||A||_F^2 + \sum_{t=\sqrt{s}+1}^{\rho} \sigma_t(A)^2.$$

A second advantage of our method is that if A is sparse and indeed, each row and column of A has a small number of non-zero entries, then the CUR representation also has a small number of entries,since C, R are just small parts of A and U is only $s \times s$. [s will in general be $O(1)$.] This advantage is not enjoyed by the SVD which in general can destroy sparsity.

The corollary above will follow from a more general Theorem which we presently describe. For the Theorem, we relax the condition that the probabilities be exactly proportional to the row (or column) length squared. Instead, we require :

$$p_i \geq 0 \qquad \sum_i p_i = 1 \qquad p_i \geq \alpha |A^{(i)}|^2 / \|A\|_F^2, \tag{2}$$

where α is a positive constant in $(0\ ,\ 1]$. [If it is 1, the probabilities are exactly proportional. This greater generality will help especially in cases where the exact entries of A are difficult to know.] Similarly, we will require (for a $\beta \in (0\ ,\ 1]$)

$$q_j \geq 0 \qquad \sum_j q_j = 1 \qquad q_j \geq \beta |A_{(j)}|^2 / \|A\|_F^2. \tag{3}$$

Next, we also let C have c columns and R have r rows, where we need not have $r = c$. Finally, there will be another parameter k under our control, which we will specify later. We describe the computation of C, U, R in the following algorithm:

1. **for** $t = 1$ **to** c **independently** pick $j_t \in \{1 \ldots n\}$ in i.i.d. trials with $\mathbf{Prob}(j_t = j) = q_j$ $j = 1 \ldots n$. Let C be the $m \times c$ matrix whose t th column is $A_{(j_t)}$ for $t = 1, 2, \ldots c$.

2. Let D_1 be the $c \times c$ diagonal matrix with $1/\sqrt{cq_{j_t}}$ in the (t,t) th position for $t = 1, 2, \ldots c$. (Note that for this, we need to know q_j only for the sampled columns j.)

3. Let $C' = CD_1$. (C' is C with columns suitably scaled.) Compute $C'^T C'$ which is a $c \times c$ matrix in time $O(c^2 m)$. Find the SVD of $C'^T C'$ (in (c^3) time); suppose it is

$$C'^T C' = \sum_t \sigma_t(C') u^{(t)} u^{(t)^T}.$$

Choose a k such that $\sigma_k(C') > 0$. We will later discuss further the choice of k.

4. **for** $t = 1$ **to** r **independently** pick $i_t \in \{1 \ldots m\}$ with $\textbf{Prob}(i_t = i) = p_i \; i = 1 \ldots m$. Let R be the $r \times n$ matrix whose t th row is $A^{(i_t)}$ for $t = 1, 2, \ldots r$. Let D_2 be the $r \times r$ diagonal matrix with $1/\sqrt{rp_{i_t}}$ as the (t,t) th entry.

5. Let W be a $r \times c$ matrix whose t th row is $C^{(i_t)}$ for $t = 1, 2, \ldots r$. Then,

$$U = D_1 \left(\sum_{t=1}^{k} \frac{1}{\sigma_t(C')^2} u^{(t)} u^{(t)^T} W^T \right) D_2^2.$$

Theorem 1 *Let ρ be the rank of A. Then, assuming $\{p_i\}$ satisfy (2) and the $\{q_j\}$ satisfy (3) , we have*

$$E\left(\|A - CUR\|_F^2\right) \le \sum_{t=k+1}^{\rho} \sigma_t(A)^2 + \left(\frac{2\sqrt{k}}{\beta\sqrt{c}} + \frac{k}{\alpha r}\right) \|A\|_F^2 \tag{4}$$

$$E\left(\|A - CUR\|_2^2\right) \le \sigma_{k+1}(A)^2 + \left(\frac{2}{\beta\sqrt{c}} + \frac{k}{\alpha r}\right) \|A\|_F^2 \le \left(\frac{1}{k+1} + \frac{2}{\sqrt{\beta c}} + \frac{k}{\alpha r}\right) \|A\|_F^2 \tag{5}$$

Remark

Now, Corollary (1) will follow from the Theorem by taking $r = c = s$ and $k = \sqrt{s}$.

If $\epsilon > 0$ is an error parameter, choosing $k = 2/\epsilon$ and $c = 64/(\beta^2\epsilon^2)$ and $r = 8/(\alpha\epsilon^2)$ makes $\|A - CUR\|_2 \le \epsilon\|A\|_F^2$. Thus in essence the theorem says that sampling $\Omega(1/\epsilon^2)$ rows and columns is sufficient for an approximation within 2-norm error at most $\epsilon||A||_F^2$.

Proof: Before proving the theorem , we would like to give some intuition on why CUR is close to A and also introduce some notation. In [7], we proved that $C'C'^T \approx AA^T$ in the sense of Frobenius norm. Let, $v^{(t)} = C'u^{(t)}/\sigma(C')$. $v^{(t)}$ are the left singular vectors of C'. So the projection $\tilde{A}$ of A into the subspace spanned by the top k of them which is

$$\tilde{A} = \sum_{t=1}^{k} v^{(t)} v^{(t)^T} A = YY^T A, \qquad \left(\text{where, } Y = \left(v^{(1)} \; v^{(2)} \; \ldots v^{(k)}\right)\right)$$

can be shown to "capture" "almost" as much of the Frobenius norm of A as A 's projection into the space spanned by its own top k singular vectors would. (using $C'C'^T \approx AA^T$). Indeed it was shown in [7] that $||\tilde{A} - A||_F^2$ was small in expectation. (See Lemma (2 below). Thus, $\tilde{A}$ would have been a fine approximation to A, but for the fact that it is hard to compute - multiplying each $v^{(t)^T}$ by A requires one pass through A for a total of k passes ! Note that we may write

$$CUR = C' \sum_{t=1}^{k} \frac{1}{\sigma_t(C')^2} u^{(t)} u^{(t)^T} W^T D_2^2 R = YXD_2^2 R, \tag{6}$$

$$\text{where } X^T = W \left(\frac{u^{(1)}}{\sigma_1(C')} \; \frac{u^{(2)}}{\sigma_2(C')} \; \cdots \; \frac{u^{(c)}}{\sigma_c(C')} \right)$$

Now, instead of explicitly computing the product Y^TA what we will do is to approximate it by using a technique from [8]. This technique says that if we pick a random subset of columns of Y^T and the corresponding set of rows of A and multiply the resulting matrices, that gives us an approximation to the product Y^TA, provided the probabilities used for the random choices satisfy certain inequalities. (see Lemma (6)) Indeed, the reader may verify that X is obtained precisely by choosing columns $i_1, i_2, \ldots i_r$ of Y^T. This will lead us to the result that $Y^TA \approx XD_2^2R$ in Frobenius norm, giving us then a bound on the Frobenius norm of $A - CUR$. The proof of the 2-norm bound in the Theorem requires more work(Lemma (4).

Lemma 2

$$E(\|A - \tilde{A}\|_F^2) \leq \sum_{t=k+1}^{\rho} \sigma_t(A)^2 + \frac{2\sqrt{k}}{\beta\sqrt{c}}\|A\|_F^2$$

Proof: Complete $\{v^{(1)}, v^{(2)}, \ldots v^{(k)}\}$ to an orthonormal basis $VV = \{v^{(1)}, v^{(2)}, \ldots v^{(m)}\}$ of $\mathbf{R}^m$. Then, we have

$$\begin{aligned}\|A - YY^TA\|_F^2 = \sum_{t=1}^{m} |v^{(t)^T}(A - Y^TA)|^2 &= \sum_{t=k+1}^{m} |v^{(t)^T}A|^2 \\ &= \|A\|_F^2 - \|Y^TA\|_F^2.\end{aligned}$$

$$\begin{aligned}\|Y^TA\|_F^2 = \mathrm{tr}(Y^TAA^TY) &\geq \mathrm{tr}(Y^TC'C'^TY) - \mathrm{tr}(Y^T(AA^T - C'C'^T)Y) \\ &\geq \sum_{t=1}^{k} \sigma_t(C')^2 - \sqrt{k}\|AA^T - CC^T\|_F,\end{aligned}$$

where we get the last inequality because if $AA^T - C'C'^T$ is written as in the basis $VV \to VV$, then $\mathrm{tr}(Y^T(AA^T - C'C'^T)Y)$ is the sum of the first k diagonal entries, which is at most $\sqrt{k}$ times the square root of the sum of squares of these entries (by Cauchy-Schawrtz) which is at most $\sqrt{k}$ times $||AA^T - C'C'^T||_F$ as claimed.

$$\text{So, } \|A - YY^TA\|_F^2 \leq \sum_{t=1}^{k}\left(\sigma_t(A)^2 - \sigma_t(C')^2\right) + \sum_{t=k+1}^{\rho} \sigma_t(A)^2 + \sqrt{k}\|AA^T - C'C'^T\|_F.$$

Now, we have

$$\sum_{t=1}^{k}(\sigma_t(A)^2 - \sigma_t(C')^2 \leq k\left(\sum_{t=1}^{k}(\sigma_t(AA^T) - \sigma_t(C'C'^T))^2\right)^{1/2} \leq \sqrt{k}||AA^T - C'C'^T||_F,$$

the last using the Hoffman-Wielandt inequality. Finally, we bound $E(||AA^T - C'C'^T||_F)$. To this end, first note that

$$\frac{|A_{(j_t)}|^2}{cq_{j_t}} \leq \frac{||A||_F^2}{\beta c}, \text{ thus, } ||C'||_F^2 \leq ||A||_F^2/\beta.$$

Using Lemma (6), we see that

$$E(||AA^T - C'C'^T||_F^2) \leq \frac{1}{\beta c}||A||_F^2||C'||_F^2,$$

$$\text{So, } E(||AA^T - C'C'^T||_F) \leq (E(||AA^T - C'C'^T||_F^2))^{1/2} \leq \frac{1}{\beta\sqrt{c}}||A||_F^2. \tag{7}$$

Plugging this in, we get the lemma.

Lemma 3 $\|Y(Y^TA - XD_2^2R)\|_F^2 = \|Y^TA - XD_2^2R\|_F^2$.

Proof:

$$\|Y(Y^TA - XD_2^2R)\|_F^2 = \mathrm{Tr}((Y^TA - XD_2^2R)^TY^TY(Y^TA - XD_2^2R))$$
$$= \mathrm{Tr}((Y^TA - XD_2^2R)^T(Y^TA - XD_2^2R)) = \|Y^TA - XD_2^2R\|_F^2.$$

Lemma 2 bounds $E(\|A - YY^TA\|_F^2)$. Combining Lemmas 3 and 5 we bound the $E(\|Y(Y^TA - XD_2^2R)\|_F^2)$. Thus, using linearity of expectation, the proof of the first statement of Theorem 1 follows. To prove the second statement we need the following two lemmas:

Lemma 4 *Given the notation of Theorem 1*

$$E(\|A - YY^TA\|_2^2) \leq \sigma_{k+1}^2(A) + \frac{2}{\beta\sqrt{c}}\|A\|_F^2.$$

Proof: Let $V_k = \mathrm{Span}\{v^{(1)}, v^{(2)}, \dots v^{(k)}\}$. V_{m-k} be the orthogonal complement of V_k in $\mathbf{R}^m$.

$$\|A - YY^TA\|_2 = \max_{x\in\mathcal{R}^m, |x|=1} |x^T(A - YY^TA)|.$$

x can be expressed as $a_1 \cdot y + a_2 \cdot z$, such that $y \in V_k$, $z \in V_{m-k}$, $a_1, a_2 \in \mathcal{R}$ and $a_1^2 + a_2^2 = 1$. Thus,

$$\max_{x\in\mathcal{R}^m,|x|=1}(x^T(A - YY^TA) \leq \max_{y\in V_k;|y|=1}(|a_1y^T(A - YY^TA)| + \max_{z\in V_{m-k};|z|=1}(|a_2z^T(A - YY^TA)| \leq$$
$$\max_{y\in V_k|y|=1}(|y^T(A - YY^TA)| + \max_{z\in V_{m-k};|z|=1}(|z^T(A - YY^TA)|$$

But, for any $y \in V_k$, y^TYY^T is equal to y. Thus, $|y^T(A - YY^TA)| = |y^TA - y^TA| = 0$ for all y. Similarly, for any $z \in V_{m-k}$, z^TYY^T is equal to 0. Thus, we are only seeking a bound for $\max_{z\in V_{m-k}} |z^TA|$. To that effect,

$$|z^TA|^2 = z^TAA^Tz = z^T(AA^T - C'C'^T)z + z^TC'C'^Tz$$
$$\leq \|AA^T - C'C'^T\|_F + \sigma_{k+1}^2(C')$$

Thus, $E(||A - YY^TA||_2^2) \leq E(\sigma_{k+1}(C')^2) + E(||AA^T - C'C'^T||_F)$. Now, $AA^T, C'C'^T$ are symmetric matrices and a result of perturbation theory (see e.g. [15], p. 428) states that

$$|\sigma_{k+1}(AA^T) - \sigma_{k+1}(C'C'^T)| \leq ||AA^T - C'C'^T||_2$$

But, $E(||AA^T - C'C'^T||_2) \leq E(\|AA^T - C'C'^T\|_F) \leq \frac{1}{\beta\sqrt{c}}\|A\|_F^2$, from (8). Thus, $|\sigma_{k+1}(AA^T) - \sigma_{k+1}(C'C'^T)| = |\sigma_{k+1}^2(A) - \sigma_{k+1}^2(C') \leq \frac{1}{\sqrt{\beta c}}\|A\|_F^2$ and Lemma 4 follows.

The maximum $|z^TC'|$ over all $z \in V_{m-k}$ appears when z is equal to the $k+1$ left singular vector of C'.

Lemma 5

$$E(\|Y^TA - XD_2^2R\|_F^2) \leq \frac{1}{\alpha r}\|Y^T\|_F^2\|A\|_F^2 = \frac{k}{\alpha r}\|A\|_F^2$$

Proof: The first inequality follows from (6); we defer the details. Then, we simply observe that the columns of Y are singular vectors (thus their length is one) and $\|Y\|_F^2 = k$.

4.1 Sampling

We prove that with the matrix presented in sparse unordered representation , all the sampling necessary to compute S, R can be done in two passes through the matrix. The two claims (1) are simple technical claims, whose proof we give in the Appendix.

Claim 1 *Suppose $a_1, a_2, \ldots a_n$ are n non-negative reals which are read once in this order (streaming). Then with $O(s)$ additional storage, we can pick i.i.d. samples $i_1, i_2, \ldots i_s \in \{1, 2, \ldots n\}$ such that*

$$\mathbf{Prob}(i_t = i) = \frac{a_i}{\sum_{j=1}^n a_j}.$$

Claim 2 *In one pass, plus $O(m+n)$ additional storage, we can pick i.i.d. samples $j_1, j_2, \ldots j_s$ drawn according to probabilities $\{q_j\}$ satisfying $q_j \geq \beta |A_{(j)}|^2 / \|A\|_F^2$ and also pick i.i.d. samples $i_1, i_2, \ldots i_s$ drawn according to probabilities for the rows satisfying $p_i \geq \alpha |A^{(i)}|^2 / \|A\|_F^2$.*

In the second pass : we pick out the entries of the matrix **C** and **R**. (note that we know the scaling factors since we know the probabilities with which we pick each row and column). Since we have $O(s(m+n))$ storage, these can be explicitly computed and so also $\mathbf{C}^T\mathbf{C}$.

4.2 One-pass Approximation

In one pass, we can find an approximation A'' satisfying all the properties as stated in the Summary except for the important bound on the Frobenius norm. This is done as follows (by generalizing an idea in [1]). As we make the pass through A, we pick a subset J of $\lambda = O((m + n)\text{ poly }(1/\epsilon))$ entries of A in λ i.i.d. trials, in each of which an entry of A is picked with the probability P_{ij} of picking the (i,j) th entry being $A_{ij}^2/||A|||_F^2$. By Claim (1) above, this can be done in one pass. Then, we can show that if we set define an $m \times n$ matrix A'' which for each picked (i,j) has entry $A_{ij}/(\lambda P_{ij})$ and has a zero entry for (i,j) which are not picked, then A'' is a good approximation to A in 2-norm. The proof is a modification of the proof of [1] for the special case of uniform sampling. But unfortunately, since λP_{ij} is much smaller than 1, it can be shown that the Frobenius norm of $A - A''$ is very large (larger than $||A||_F$.) We will give the full proof in the final paper.

5 Handling non-dense graphs

In this section, we sketch how in 2 passes plus $O(\log n)$ additional time and RAM space, we can find the value of the maximum cut in a graph $G(V, E)$ to additive error $\epsilon(|V| - l)^2$, where l is chosen so that the sum of the lowest l degrees is $\epsilon|E|/2$. This algorithm will only use elementary ideas. Essentially, we pick vertices in the first pass with probabilities proportional to the degrees (by picking a random edge); we will then discard certain vertices of low degree and then do some rejection sampling after which, we have a uniform random sample from the high degree vertices. Here are a few more details.

Let $G(\{1, 2, \ldots n\}, E)$ be a graph. Let d_i be the degree of the vertex i. We assume that each $d_i \leq \epsilon|E|/4$. (This is a mild assumption that no vertex has more than an $\epsilon/4$ fraction of all edges incident to it, which is obviously true if for example, we have a super-linear number of edges.)

For any $f \in (0\ ,\ 1)$, define $v(f)$ to be the least positive integer such that $\sum_{i:d_i \le v(f)} d_i \ge f|E|$ and let $V(f) = \{i : d_i \le v(f)\}$.

We pick $i_1, i_2, \ldots i_{3s}$ all i.i.d. samples, (where s will be $\Omega(\log n/\text{poly}(\epsilon))$) with $\mathbf{Prob}(i_t = i) = d_i/(2|E|)$ [by picking a random edge] in the first pass; we also find $|E|$ exactly in the first pass. In the second pass, we collect the induced graph on these $3s$ vertices and also compute the degree of each of these $3s$ vertices in the whole graph.

Define L, M to be the minimum positive integers such that

$$|\{t : 1 \le t \le s, d_{i_t} \le L\} \ge 2\epsilon s \quad |\{t : 1 \le t \le s, d_{i_t} \le M\} \ge 5\epsilon s.$$

We have that the probability of a single i_t falling in $V(\epsilon)$ is equal to $\sum_{i \in V(\epsilon)} d_i/|E|$ which is between ϵ and $(5/4)\epsilon$. So using Hoeffding on Bernouli trials, we have that whp :

$$|\{i_1, i_2, \ldots i_s\} \cap V(\epsilon)| \le \frac{3}{2}\epsilon s \text{ which implies that } L \ge v(\epsilon) \tag{8}$$

$$|\{i_1, i_2, \ldots i_s\} \cap V(3\epsilon)| \ge 2.5\epsilon s \text{ which implies that } L \le v(3\epsilon). \tag{9}$$

Similarly we have that with high probability : $v(4\epsilon) \le M \le v(6\epsilon)$.

$$\text{Let } W = \{i : v(\epsilon) \le d_i \le v(6\epsilon) : d_i \le \frac{\epsilon^2|E|}{2\log n|\{j : d_j \ge d_i\}|}\}.$$

$$\sum_{i \in W} d_i \le \epsilon^2|E|\frac{1}{2\log n}\sum_{i \in V(6\epsilon)\setminus V(\epsilon)} (1/|\{j : d_j \ge d_i\}|) \le \frac{\epsilon^2}{2\log n}|E|\left(\sum_{i=1}^{n}\frac{1}{i}\right) \le \epsilon^2|E|/2.$$

Thus the probability that a particular i_t belongs to W is at most $\epsilon^2/2$. Now, we use the second set of s samples. Let $P = \{i_t : s+1 \le t \le 2s, d_{i_t} \in [L\ ,\ M]\}$. Pick uniformly at random an element q of P. By the above, whp, we have that $q \notin W$. Now, we use the third batch of s samples.

For each $i_t, 2s+1 \le t \le 3s$, we independently do the following : if $d_{i_t} < d_q$, then we set all entries in row i_t and column i_t of our sampled sub-matrix to be zero. Otherwise, we accept sample i_t with probability : d_q/d_{i_t}. Now we have that the acceptance probability of a sample is $\sum_{i:d_i \ge d_q} \frac{d_q d_i}{d_i|E|} \ge \frac{\epsilon^2}{2\log n}$. Thus, with $s = \Omega(\log n/\text{poly}(\epsilon))$, we will have the required number of $\text{poly}(1/\epsilon)$ samples surviving the rejection process for us to appeal to the earlier results. Also a simple calculation shows that these are samples drawn uniformly from vertices of degree at least d_q, so appealing to results of say [14], we get the claimed algorithm.

References

[1] D. Achlioptas and F. McSherry, *Fast Computation of Low Rank Approximations*, Proceedings of the 33rd Annual Symposium on Theory of Computing, 2001.

[2] N. Alon, W. F. de-la-Vega, R. Kannan and M. Karpinski, *Random sub-problems of Max-SNP problems*, Preprint, 2001.

[3] N. Alon, Y. Matias, M. Szegedy, "The space complexity of approximating the frequency moments" Symposium on Theoretical Computer Science, 1996. pp 20-29.

[4] S.Arora, D.Karger and M.Karpinski, *Polynomial time approximation schemes for dense instances of NP-hard problems*, Proceedings of the 27th Annual ACM Symposium on Theory of Computing (1995) 284-293.

[5] A. Broder, M. Charikar, A. Frieze, M. Mitzenmacher, "Min-wise independent permutations", in the 30 th ACM Symposium on Theory of Computing, pp 327-336, 1998.

[6] P. Drineas, I. Kerenidis and P. Raghavan, "Competitive Recommendation Systems", ACM Symposium on Theory of Computing, 2002.

[7] P. Drineas, A. Frieze, R. Kannan, S. Vempala and V. Vinay, *Clustering in large graphs and matrices*, Proceedings of the 10th Symposium on Discrete Algorithms, pp. 291-299, 1999.

[8] P. Drineas and R. Kannan, *Fast Monte-Carlo Algorithms to Approximate Matrix Multiplication*, Proceedings of the 42nd Annual Symposium on Foundations of Computing, 2001.

[9] J. Feigenbaum, S. Kannan, M. Strauss and M. Viswanathan, "An approximate L1-difference algorithm for massive data streams" in Foundations of Computer Science, 1999.

[10] W.Fernandez-de-la-Vega, *MAX-CUT has a Randomized Approximation Scheme in Dense Graphs*, Random Structures and Algorithms 8 (1996) 187-199.

[11] A.M.Frieze and R.Kannan, *The Regularity Lemma and approximation schemes for dense problems*, Proceedings of the 37th Annual IEEE Symposium on Foundations of Computing, (1996) 12-20.

[12] A. M. Frieze and R. Kannan *Quick Approximation to matrices and applications*, Combinatorica **19** (2) (1999) pp 175-200.

[13] A. Frieze, R. Kannan and S. Vempala, *Fast Monte-Carlo algorithms for finding low rank approximations*, Proceedings of the 39th Annual Symposium on Foundations of Computing, pp. 370-378, 1998.

[14] O.Goldreich, S.Goldwasser and D.Ron, *Property testing and its connection to learning and approximation*, FOCS 96.

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

[16] M. Henzinger, P. Raghavan, S. Rajagopalan, "Computing on data streams" Technical Note 1998-011, Digital Systems Research Center, Palo Alto, CA May 1998.

[17] P. Indyk, " Stable distributions, Pseudo-random generators, embeddings, and data stream computation" in Foundations of Computer Science, 2000.

[18] J. Kleinberg, "Two Algorithms for nearest neighbor search in high dimensions" , in the Proceedings of the 29 th Symposium on Theory of Computing (1997) pp 599-608.

[19] J. I. Munro and M. S. Paterson, "Selection and sorting with limited storage" in Foundations of Computer Science 1978, pp 253-259.

[20] K. V. Ravi Kanth, D. Agrawal, Amr El Abbadi, "Dimensionality reduction for similarity searching in dynamic databases" in ACM SIGMOD 1998.

[21] S. Vempala, "Random Projection : A new approach to VLSI layout" in 39 th IEEE Foundations of Computer Science, 1998.

6 Appendix

Following the lines of [13] and [8], we analyze the following algorithm:

Given an $m \times n$ matrix A and an $n \times p$ matrix B, we approximate their product by $X \cdot Y$, where X is an $m \times s$ matrix and Y an $s \times p$ matrix, using the following algorithm:

Suppose we have $p_1, p_2, \dots p_m \geq 0$ such that $\sum_{k=1}^{m} p_k = 1$.

- **for** $t = 1$ **to** s **independently**
  - Pick $i_t \in \{1 \dots n\}$ at random with $\mathbf{Prob}(i_t = k) = p_k, \quad k = 1 \dots n$.
  - Include $A_{(i_t)}/\sqrt{sp_{i_t}}$ as a column of X and $B^{(i_t)}/\sqrt{sp_{i_t}}$ as the corresponding row of Y.
- Return $X \cdot Y$ as the approximation to $A \cdot B$.

Lemma 6 *If X and Y are created using the above algorithm and if either $p_k \geq \alpha \frac{|A_{(k)}|^2}{\|A\|_F^2}$ for all $k = 1 \dots n$, or $p_k \geq \alpha |B^{(k)}|^2/||B||_F^2$ for $k = 1, 2, \dots n$, then*

$$E(\|AB - XY\|_F^2) \leq \frac{1}{\alpha s}\|A\|_F^2\|B\|_F^2$$

Proof: Fix attention on one particular i, j. For $t = 1 \dots s$ define the random variable $w_t = \left(\frac{A^{(i_t)}B_{(i_t)}}{sp_{i_t}}\right)_{ij} = \frac{A_{ii_t}B_{i_tj}}{sp_{i_t}}$. So, the w_t's are independent random variables. Also, $(XD_2^2R)_{ij} = \sum_{t=1}^{s} w_t$. Thus, its expectation is equal to the sum of the expectations of the w_t's. But, $E(w_t) = \sum_{k=1}^{n} \frac{A_{ik}B_{kj}}{sp_k} p_k = \frac{1}{s}(AB)_{ij}$. So, $E((XY)_{ij}) = \sum_{t=1}^{s} E(w_t) = (AB)_{ij}$.

Since $(XY)_{ij}$ is the sum of s independent random variables, the variance of $(XY)_{ij}$ is the sum of the variances of these variables. But, using $\mathbf{Var}(w_t) = E(w_t^2) - E^2(w_t)$ we see that $\mathbf{Var}(w_t) = \sum_{k=1}^{n} \frac{A_{ik}^2 B_{kj}^2}{s^2 p_k} - \frac{1}{s^2}(AB)_{ij}^2 \leq \sum_{k=1}^{n} \frac{A_{ik}^2 B_{kj}^2}{s^2 p_k}$. Thus, $\mathbf{Var}(XY)_{ij} \leq s \sum_{k=1}^{n} \frac{A_{ik}^2 B_{kj}^2}{s^2 p_k}$. Using $E((AB - XY)_{ij}) = 0$ and the lower bound for the p_k,

$$E(\|AB - XY\|_F^2) = \sum_{i=1}^{m}\sum_{j=1}^{n} E((AB - XY)_{ij}^2) = \sum_{i=1}^{m}\sum_{j=1}^{n} Var((XY)_{ij}) = \frac{1}{s}\sum_{k=1}^{n}\frac{1}{p_k}(\sum_i A_{ik}^2)(\sum_j B_{kj}^2) =$$
$$\frac{1}{s}\sum_{k=1}^{n}\frac{1}{p_k}|A^{(k)}|^2|B_{(k)}|^2 \leq \frac{\|A\|_F^2}{\alpha s}\sum_{k=1}^{n}\left(|B^{(k)}|\right)^2 = \frac{1}{\alpha s}\|A\|_F^2\|B\|_F^2$$

Proof of Claim 1 We argue that we can pick i_1. The others can be done by running s independent copies of this process. To pick i_1 : suppose we have read $a_1, a_2, \dots a_l$ so far and have a sample i_1 such that

$$\mathbf{Prob}(i_1 = i) = \frac{a_i}{\sum_{j=1}^{l} a_j},$$

and also we keep the running sum $\sum_{j=1}^{l} a_j$. On reading a_{l+1}, we just replace the current i_1 with $l + 1$ with probability

$$\frac{a_{l+1}}{\sum_{j=1}^{l+1} a_j}.$$

It is easy to see by induction that this works.

Proof of Claim (2) : To pick i_1 : just pick (using the previous claim) an entry (i,j) with probabilities proportional to their squares and just take $i_1 = i$. The other i_t and the j_t are also picked by running $2s$ independent experiments simultaneously.

Proof of Lemma (1) The idea of the proof is : - one just picks a "large" set of vectors $u \in \mathbf{R}^n$ such that each u is of length almost exactly 1 and we have $|u - u'|, |u + u'| \geq \epsilon$ for each pair of distinct u, u'. Similarly one picks such a set of $v \in \mathbf{R}^m$ satisfying similar conditions. Then we form all rank 1 matrices of the form uv^T and show that this class has the claimed properties. The detailed proof is deferred to the final paper.