Randomized algorithms and matrix decompositions

Luis Rademacher

1 Introduction (L1)

These notes discuss novel randomized algorithms for some tasks in numerical linear algebra, such as matrix multiplication and certain matrix decompositions (SVD, interpolative decomposition). One of the main motivations is the need for fast algorithms for the analysis of massive data. The emphasis will be more on algorithms and algorithmic tools rather than complexity or linear algebra.

Randomized algorithms are frequently faster or simpler that deterministic counterparts. For example, Karger's algorithm [**?**, **?**] for the minimum cut of a graph is simpler than deterministic algorithms. The only theoretically efficient ways of estimating the volume of high dimensional convex bodies are based on random walks [**?**, **?**].

While randomized algorithms commonly have a chance of failure, this chance can be made arbitrarily small (say, smaller than the chance that the computer fails for any other reason) with a modest computational cost.

In the case of massive data or massive computation, they frequently provide a tradeoff between use of computational resources and approximation quality of the output. This is the idea behind approximation algorithms, many of which are randomized. For example, random projection [**?**, **?**], a building block in the design of many efficient randomized algorithms, is based on the idea of randomly embedding high-dimensional data into a space of lower dimension, with guaranteed low distortion, where the choice of the target dimension gives the tradeoff between computational cost and approximation quality.

We will discuss some basic "phenomena in high dimension", while discussing the analysis and design of some randomized algorithms, such as those based on random projection. For example: What is the "typical" angle between two random unit vectors in $\mathbb{R}^n$? For an answer, it is easy to show $\mathbb{E}((X \cdot Y)^2) = 1/n$. What is the typical length of a random Gaussian vector in n dimensions? How concentrated is it?

The two main algorithmic problems that we will discuss:

1. (Subspace approximation) Given m points in $\mathbb{R}^n$ and $k \leq m, n$, find a k-dimensional subspace that minimizes the sum of squared distances of the points to the subspace.

2. (Subset selection) Given m points in $\mathbb{R}^n$ and $k \leq m, n$, find a k-dimensional subspace spanned by k input points that minimizes the sum of squared distances of the points to the subspace.

In a linear algebraic language, if the points are arranged as rows of an $m \times n$ matrix A, the first problem corresponds to finding the best rank-k approximation (in the squared Frobenius norm, the sum of squares of entries), and the second problem corresponds to an "interpolative decomposition", $A \approx XR$, where the rows of R are k rows of A.

The first problem is fundamental in modern data analysis. For example, the points may correspond to input data that is intrinsically low dimensional (i.e. the coordinates are not independent), but that has been corrupted by moderate noise. Finding the optimal subspace and projecting the noisy data to it would essentially remove the noise. Solving the subspace approximation problem and picking an orthonormal basis of it can give the most relevant k-dimensional basis in which to analyze the data, acting as a sort of dimensionality reduction and hopefully highlighting the relevant features of the data. Unfortunately this basis is made of arbitrary vectors in space and can be difficult to interpret. The second problem, subset selection, tries to overcome this as a refinement of subspace approximation: The desired subspace is spanned by k of our input points and therefore gives a basis that can be immediately interpreted in terms of the original data, as well as giving a representative set of k input data points. Both problems work as a form of data compression, as a special case of dimensionality reduction.

We will discuss two ways in which randomization has been used to speed-up linear-algebraic computations: sampling (of rows or columns of a given matrix according to non-uniform distributions) and random projection. Frequently, the randomized algorithms do not provide a completely new way of solving the problem, but instead replace it with a much smaller problem while incurring a small error. The smaller problem can then be solved by classical techniques.

There are other problems that have been approached with randomized algorithms that we may discuss briefly, such as matrix multiplication and linear regression.

The emphasis will be on algorithms and tools from discrete geometry and probability rather than complexity or linear algebra.

A thorough understanding of these algorithmic problems would involve some additional topics that we may discuss only briefly: concentration bounds for the quality of randomized matrix decomposition, streaming algorithms (that work with only one pass over the input data), and stability.

2 Randomized algorithms (L1)

2.1 The union bound

An elementary probabilistic bound that is surprisingly useful. If $A_1, \ldots, A_m$ are "bad" events in a common probability space, each occurring with probability at

most p, then the probability that at least one of them happens is

$$\Pr(\bigcup A_i) \leq \sum \Pr(A_i) \leq mp.$$

Thus, the probability that none of them happen is at least $1 - mp$. Here is an example:

In $\mathbb{R}^d$, how large can a set of orthonormal vectors be? Answer: d. What if we relax the condition to almost orthogonality, say, for all distinct x, y in the set we have $|x \cdot y| \leq 1/100$? Then the size of the set can be $e^{\Omega(d)}$ (note that the relaxation does not seem to help much in low dimension). We will need the following fact:

Lemma 1 (from Ball's notes)**.** *For any $0 < \epsilon < 1/2$, the $(d-1)$-dimensional volume of $\{x \in S^{d-1} : x_1 \geq \epsilon\}$ (the cap at distance ϵ from the origin) as a fraction of the volume of S^{d-1} is at most $e^{-d\epsilon^2/2}$.*

Proof. It is easier to compare the volume of the convex hull C of the origin and the cap with the volume of the unit ball, which is the desired ratio. But C is contained in a ball of radius $\sqrt{1-\epsilon^2}$ so that the ratio is at most $(\sqrt{1-\epsilon^2})^d \leq e^{-d\epsilon^2/2}$ □

In particular, given any hyperplane, a constant fraction of the mass of S^{d-1} lies within distance $O(1/\sqrt{d})$ of the hyperplane.

Here is a simple argument for almost orthogonal vectors: For m to be fixed later, pick m random points $P_1, \ldots, P_m$ from S^{d-1}. By Lemma 1, the probability that $|P_i \cdot P_j| \geq 1/100$ for $i \neq j$ is at most $2e^{-d/(2\times 10^4)}$. Thus, the probability that all pairs are almost orthogonal is at least $1 - 2\binom{m}{2}e^{-d/(2\times 10^4)}$ (the union bound). This is positive for some $m = e^{\Omega(d)}$, which implies that in a random set of points of exponential size, *with high probability* all pairs are nearly orthogonal.

Exercise 1. *Show a matching upper bound up to exponential factors for the problem of nearly orthogonal vectors: Show that any set of unit vectors such that all pairwise inner products are at most $1/100$ in absolute value has cardinality $e^{O(d)}$.*

2.2 The method of conditional expectation

Example from Alon-Spencer 3rd, 16.1

3 Random projection (L2)

Random projection is the following basic idea: That given a set of n points in $\mathbb{R}^d$ one can map (or embed or project) this set of points to a set of n points in $\mathbb{R}^k$ for $k \ll d$ while preserving the metric structure approximately, say, pairwise distances are preserved to within a constant factor. More precisely, Johnson and Lindenstrauss proved the following result. We need a definition to state it: For a

pair of metric spaces (X, d_X), (Y, d_Y), a map $f : X \to Y$ is a $(1+\epsilon)$-embedding iff for all $x, y \in X$ we have

$$(1-\epsilon)d_X(x,y) \leq d_Y(f(x), f(y)) \leq (1+\epsilon)d_X(x,y)$$

Theorem 2 (Johnson-Lindenstrauss lemma)**.** *Let X be an n-point set in an Euclidean space, and let $\epsilon \in (0, 1/2]$ be given. Then there exists a $(1+\epsilon)$-embedding of X into l_2^k, where $k = O(\epsilon^{-2} \log n)$.*

So, if an algorithm takes as input a set of n points in $\mathbb{R}^d$, the algorithm only cares about pairwise distances between the points and an approximate answer is acceptable, then we could improve the running time of the algorithm by first embedding the input points in a space of dimension $O(\epsilon^{-2} \log n)$. For this to be possible, we need an embedding that is efficiently computable. Initially, Theorem 2 was proven by projecting the points to a random k-dimensional subspace and showing that with high probability such a projection gives a $(1+\epsilon)$-embedding. Later [Achlioptas] showed that a random $\{-1, 1\}$ matrix works as the embedding map, and [Indyk and Motwani, Dasgupta and Gupta] showed it by using a random Gaussian matrix. The Gaussian case leads to a simple proof that we will see in a moment. A projection as in the Johnson-Lindenstrauss lemma is sometimes called a Johnson-Lindenstrauss transform.

One can think of random projection in the most basic form as preserving approximately the lengths of n vectors. Of course, from such a statement one can easily deduce a version that preserves an Euclidean metric (that is, pairwise distances) by considering all pairwise differences and then applying the basic form that preserves lengths. Similarly, we can deduce a version that preserves pairwise angles.

Part of the algorithmic power of random projection comes from the fact that it is given simply by a linear map and it is *data oblivious*, that is, it can be chosen randomly with high probability of success *without looking at the input points*.

The basic idea for a proof using Gaussian matrices is that if R is a k-by-n matrix with standard Gaussian entries and x is an n-dimensional vector, then the entries of Rx are Gaussian and $\|Rx\|^2$ is highly concentrated around its expected value (as it has a χ^2 distribution with k degrees of freedom, after a suitable scaling). More precisely, we have the following standard concentration result:

Lemma 3 (as exposed in Santosh's book)**.** *Let each entry of a $k \times n$ matrix R be chosen independently from $N(0,1)$. Let $v = \frac{1}{\sqrt{k}} Ru$ for $u \in \mathbb{R}^n$. Then for any $\epsilon > 0$,*

1. $\mathbb{E}(\|v\|^2) = \|u\|^2$,
2. $\mathbb{P}(|\|v\|^2 - \|u\|^2| \geq \epsilon \|u\|^2) < 2e^{-(\epsilon^2 - \epsilon^3)k/4}$.

Proof. Intuition: sum of bounded independent random variables is highly concentrated around its mean, as in Chernoff's inequality.

Let $X_j = R_j u/\|u\|$. Then X_j is distributed as a standard Gaussian, as it is a one dimensional marginal of R_j, a Gaussian vector. Let $X = \sum X_j^2 = k\|v\|^2/\|u\|^2$. X is a sum of k independent squared standard Gaussians (a χ^2 with k degrees of freedom). Then, for $\lambda > 0$,

$$\begin{aligned}\Pr(\|v\|^2 \geq (1+\epsilon)\|u\|^2) &= \Pr(X \geq (1+\epsilon)k) \\ &= \Pr(e^{\alpha X} \geq e^{(1+\epsilon)\alpha k}) \\ &\leq \frac{\mathbb{E}(e^{\alpha X})}{e^{(1+\epsilon)\alpha k}} \\ &= \left(\frac{\mathbb{E}(e^{\alpha X_1^2})}{e^{(1+\epsilon)\alpha}}\right)^k\end{aligned}$$

Using the explicit density of the Gaussian distribution, one can easily show (for $\lambda < 1/2$)

$$\mathbb{E}(e^{\alpha X_1^2}) = \frac{1}{\sqrt{1-2\alpha}}$$

to get

$$\Pr(X \geq (1+\epsilon)k) \leq \left(\frac{e^{-2\alpha(1+\epsilon)}}{1-2\alpha}\right)^{k/2}.$$

The optimal choice of α is $\epsilon/2(1+\epsilon)$, and this implies

$$\Pr(X \geq (1+\epsilon)k) \leq ((1+\epsilon)e^{-\epsilon})^{k/2} < e^{-(\epsilon^2-\epsilon^3)k/4}$$

where the last inequality is obtained from a Taylor expansion of $\log(1+\epsilon)$. The inequality for the lower tail is proven similarly. □

We can use this and the union bound to prove Theorem 2:

Proof (of Theorem 2). For $x \in X$, let $f(x) = k^{1/2}Rx$ for R as in Lemma 3 and k to be fixed later. As the embedding map is linear, we just need to preserve approximately the lengths of the at most n^2 vectors of the form $x - y$ for $x, y \in X$. By Lemma 3 and for $\epsilon \leq 1/2$, the probability that $\|f(x) - f(y)\|^2 \notin [(1-\epsilon),(1+\epsilon)]\|x-y\|^2$ for any single pair x, y is at most $2e^{-\epsilon^2 k/8}$. By the union bound, the probability that f is a $(1+\epsilon)$-embedding is at least $1 - 2n^2 e^{-\epsilon^2 k/8}$. This is positive for $k = O(\epsilon^{-2}\log n)$. □

As an application, we will see an algorithm by [Sarlos] for fast approximate matrix multiplication. To compute the product of two matrices $A \in \mathbb{R}^{m\times d}$ and $B \in \mathbb{R}^{d\times p}$, the algorithm randomly projects the rows of A and the columns of B. The idea is that the matrix product involves mp inner products of $m+p$ vectors in $\mathbb{R}^d$, and random projection preserves inner products:

Corollary 4. *For any $\epsilon < 1/2$ and for any $V \subseteq \mathbb{R}^d$, $|V| = n$ there exists a linear map $R : \mathbb{R}^n \to \mathbb{R}^k$ with $k = O(\epsilon^{-2}\log n)$, such that for all $u, v \in V$ we have*

$$u \cdot v - \epsilon\|u\|\|v\| \leq Ru \cdot Rv \leq u \cdot v + \epsilon\|u\|\|v\|.$$

Moreover, a random Gaussian matrix scaled by $1/\sqrt{k}$ is such R with high probability.

Exercise 2. *Prove Corollary 4.*

Theorem 5 (Sarlos)**.** *Let $A \in \mathbb{R}^{m\times d}$, $B \in \mathbb{R}^{d\times p}$. Let $R : \mathbb{R}^d \to \mathbb{R}^k$ be a random Gaussian matrix scaled by $1/\sqrt{k}$. Then, with high probability for $k = O(\epsilon^{-2}\log(m+p))$ we have*

$$\|AB - AR^T RB\|_F \leq \epsilon \|A\|_F \|B\|_F$$

Proof. By Corollary 4,

$$\begin{aligned}\|AB - AR^T RB\|_F^2 &= \sum_{i,j} (A_i \cdot B^j - RA_i \cdot RB^j)^2 \\ &\leq \epsilon^2 \sum_{i,j} \|A_i\|^2 \|B^j\|^2 \\ &= \epsilon \|A\|_F^2 \|B\|_F^2.\end{aligned}$$

□

The generation of the projected matrices takes time $O((md+dp)\epsilon^{-2}\log(m+p))$ and space $O((m+p)\epsilon^{-2}\log(m+p))$. Multiplying the projected matrices would take time $O(mp\epsilon^{-2}\log(m+p)$. For fixed ϵ, the whole algorithm takes time that is nearly linear (up to logarithmic factors) in the size of the input plus the size of the output.

3.1 Fast random projection (Ailon-Chazelle)

The projection step is a bottleneck for many algorithms (e.g. approximate nearest neighbor search) using random projection: $\Omega(dk)$ per vector. One could make the projection matrix sparser (Achlioptas), but it still needs a constant fraction of non-zero entries. Intuitively, it needs many non-zero entries to preserve the length of a sparse vector (e.g. canonical vectors). Another idea would be to make k, the target dimension smaller. But Alon showed a lower bound of $k = \Omega(\frac{\log n}{\epsilon^2 \log(1/\epsilon)})$.

Idea for a fast Johnson-Lindenstrauss transform (FJLT): "Pre-process" by pre-multiplying vectors by a matrix H that turns sparse vectors into dense vectors and structured so that matrix-vector multiplication is fast. Then, project using a sparse Gaussian matrix. An example of a pre-processing matrix H considered in the literature, the discrete Fourier transform (DFT). In the case of Ailon-Chazelle, they use the Hadamard-Walsh transform, a sort of generalized Fourier transform, but which is actually quite easy to understand.

Intuitively, why does a DFT turn sparse into dense? Heisenberg uncertainty principle: A signal and its spectrum cannot be both concentrated. In other words, a vector and its Fourier transform cannot both be sparse.

More precisely, the FJLT is given by a matrix Φ, which is the composition of 3 linear maps:

$$\Phi = PHD$$

where

1. P is a $k \times d$ matrix, each entry is 0 with probability $1 - q$ and $N(0, 1/q)$ with probability q, where

$$q = \min\{\Theta(\frac{\log^2 n}{d}), 1\}.$$

2. H is the $d \times d$ Hadamard-Walsh transform (for simplicity we assume that d is a power of 2).

3. D is a $d \times d$ diagonal matrix with independent random $\{-1, 1\}$ entries.

The H-W transform $H = H_t/\sqrt{d}$ for $d = 2^t$ is a $d \times d$ matrix given recursively by

$$H_t = \begin{pmatrix} H_{t-1} & H_{t-1} \\ H_{t-1} & -H_{t-1}. \end{pmatrix} \qquad H_1 = \begin{pmatrix} 1 \end{pmatrix}$$

The transform H is just a change of basis: It is an orthonormal matrix. The entries of H_t are -1, 1.

Note that H turns canonical vectors (sparsest) into dense vectors. On the other hand, if x is some row of H, then Hx is a canonical vector (sparsest). But the actual transform pre-multiplies by D, so Dx has a very low probability of being equal to a row of H. Informally, dense vectors that become sparse through HD are rare.

Naive matrix-vector multiplication with H_t would have cost $O(d^2)$, but the recursive structure of H_t gives an $O(d \log d)$ algorithm (in the spirit of FFT, but actually simpler): For $x = (y, z)$ with $y, z \in \mathbb{R}^{2^{t-1}}$, compute $w = H_t x$ by recursively computing $w_1 = H_{t-1} y$ and $w_2 = H_{t-1} z$ and setting $w = (w_1 + w_2, w_1 - w_2)$.

Theorem 6 (Ailon-Chazelle)**.** *Given a fixed set of n points in $\mathbb{R}^d$ and $\epsilon < 1$, draw a matrix Φ from FJLT. With probability at least $2/3$, the following two events occur:*

1. *For any $x \in X$;*

$$(1 - \epsilon)k\|x\| \leq \|\Phi x\| \leq (1 + \epsilon)k\|x\|.$$

2. *The mapping $\Phi : \mathbb{R}^d \to \mathbb{R}^k$ requires*

$$O(d \log d + \min\{d\epsilon^{-2} \log n, \epsilon^{-2} \log^3 n\}).$$

Proof. Ailon-Chazelle's CACM article. □

Exercise 3. *In this exercise you will investigate the use of random projection to preserve the areas of triangles. Let S be a set of n points in $\mathbb{R}^d$.*

1. *Show that for any ϵ there are 3 points in $\mathbb{R}^n$ and a $(1+\epsilon)$-embedding of them such that the area (and therefore the heights) can be distorted by the embedding by an arbitrarily large factor.*
2. *Show that a $(1+\epsilon)$-embedding of an isosceles right angle triangle preserves the heights to within a $(1+O(\epsilon))$ factor.*
3. *Conclude that one can embed S in $\mathbb{R}^m$ for $m = O(\epsilon^{-2}\log n)$ so that all distances are preserved to within a factor of $1+\epsilon$ and all areas of triangles determined by 3 points in S are preserved to within a factor of $(1+\epsilon)^2$. (Hint: for every triangle T, add an isosceles right angle triangle to be $(1+\epsilon)$-embedded whose height is the same as one height of T)*

4 Matrix decompositions (L3)

4.1 Best subspace fitting, matrix approximation and the singular value decomposition (SVD)

This section is partly based on the upcoming book on spectral algorithms by Ravi Kannan and Santosh Vempala.

We will now discuss the first basic problem mentioned in the introduction: to find a k-dimensional subspace that is close to a given set of points in the sense of minimizing the sum of squared distances.

A theoretical framework as well as an algorithm to understand this problem is given by the singular value decomposition that we will review now. To this end we will think of the input points as rows of an $m \times n$ matrix that we will denote A. As mentioned in the introduction, the geometric problem is equivalent to the problem of finding the best rank-k approximation to A in the Frobenius norm (discussed below).

Let A be an $m \times n$ matrix with entries a_{ij}.

For $k = 1$, we want to find a line through the origin L such that $\sum d(A_i, L)^2$ is minimized. We can think of this line as being spanned by a unit vector v and then by the Pythagorean theorem the problem is equivalent to

$$\max_{v \in S^{n-1}} \|Av\|^2. \tag{1}$$

We have:

Proposition 7. *Any top unit eigenvector of A^TA is a solution of* (1)

Proof. A^TA is a symmetric positive semi-definite matrix and has an orthonormal basis of eigenvectors $v_1, \ldots, v_n$, and corresponding non-negative eigenvalues $\lambda_1, \ldots, \lambda_n$, which we can assume are sorted in decreasing order. The objective at v can be written in terms of this basis as $v = \sum \alpha_i v_i$ and then $\|Av\|^2 = \sum \alpha_i^2 \lambda_i$, which is maximized by setting $\alpha_i = 0$ whenever λ_i is not maximal. □

While A does not necessarily have eigenvalues and eigenvectors (as it may not even be a square matrix), it has a generalized notion, singular values and singular vectors: If $u \in \mathbb{R}^m$, $v \in \mathbb{R}^n$ and $\sigma > 0$ satisfy $Av = \sigma u$ and $u^T A = \sigma v^T$, then we say that u is a left singular vector, v is a right singular vector, and σ is a singular value of A. It is easy to see that the right singular vectors are the eigenvectors of $A^T A$, the left singular vectors are the eigenvectors of AA^T, and the singular values squared are the eigenvalues of AA^T and $A^T A$.

In this new language, a solution to (1) is a top right singular vector of A. We will now see that the subspace approximation problem for general k can be solved in terms of singular vectors, as well as showing that the singular vectors and singular values can be used to express any matrix in a generalized form of eigendecomposition, the singular value decomposition (SVD).

Theorem 8 (existence of SVD and Eckart-Young)**.** *Let* $V_k = \operatorname{span}(v_1, \ldots, v_k)$ *where*

$$\begin{aligned} v_1 &\in \operatorname{argmax}_{x \in S^{n-1}} \|Ax\| \\ v_2 &\in \operatorname{argmax}_{x \in S^{n-1}, x \perp V_1} \|Ax\| \\ &\vdots \\ v_k &\in \operatorname{argmax}_{x \in S^{n-1}, x \perp V_{k-1}} \|Ax\| \end{aligned}$$

Then V_k *is optimal for*

$$\min_{V \text{ subspace, } \dim(V) = k} \sum_{i=1}^{m} d(A_i, V)^2.$$

Moreover, $v_1, \ldots, v_n$ *are left singular vectors with singular values* $\sigma_i = \|Av_i\|$*. Finally,* $A = \sum \sigma_i u_i v_i^T$*.*

Proof. Idea: induction in k.

For $k = 1$, by definition.

Suppose V_k' is an optimal k-dimensional subspace. Let $\{w_1, \ldots, w_k\}$ be a basis of V_k' such that w_k is orthogonal to V_{k-1}. Optimality of V_{k-1} implies

$$\|Aw_1\|^2 + \cdots + \|Aw_{k-1}\|^2 + \|Aw_k\|^2 \leq \|Av_1\|^2 + \cdots + \|Av_{k-1}\|^2 + \|Aw_k\|^2.$$

Optimality of v_k implies

$$\|Aw_k\|^2 \leq \|Av_k\|^2.$$

and this completes the induction.

The fact that the v_is are right singular vectors is an easy generalization of Proposition 7.

$A = \sum \sigma_i u_i v_i^T$ follows by evaluating both sides of this identity on the orthonormal basis (v_i). □

There are efficient algorithms to compute the SVD. A basic way is the power method: Observe that $U\Sigma\Sigma^T U^T$ is an eigendecomposition of AA^T. So if we

compute an eigendecomposition of AA^T we can get U and Σ. Then $U^T A = \Sigma V$, from which we can get V. Let $B = AA^T$, a symmetric positive semidefinite matrix with eigenvectors (u_i) and eigenvalues $\lambda_i = \sigma_i^2$. To find an eigendecomposition of B, consider the following procedure (the power method) to find a top eigenvector: Pick a random unit vector in $\mathbb{R}^m$, say x_0 and compute the sequence $x_t = Bx_{t-1}/\|Bx_{t-1}\|$. If $\lambda_1 > \lambda_2$ and $x_0 \cdot u_1 \neq 0$, then the sequence (x_t) converges to a multiple of u_1, a top eigenvector of B. To see this, write x_0 in the basis (u_i), that is, $x_0 = \sum_i \alpha_i u_i$, and then we have

$$\begin{aligned} B^t x_0 &= \sum \alpha_i \lambda_i^t u_i \\ &= \alpha_1 \lambda_1^t \left(u_1 + \frac{\alpha_2}{\alpha_1}(\frac{\lambda_2}{\lambda_1})^k + \cdots + \frac{\alpha_m}{\alpha_1}(\frac{\lambda_m}{\lambda_1})^k \right) \end{aligned}$$

where the expression in parenthesis converges to u_1 and x_t converges to a vector $\tilde{u}_1$, a multiple of u_1. An approximate top eigenvalue is given by $\tilde{\lambda}_1 = \|B\tilde{u}_1\|$. Let $B \leftarrow B - \tilde{\lambda}_1 \tilde{u}_1 \tilde{u}_1^T$ and repeat to find subsequent eigenvalues and eigenvectors.

5 Fast randomized numerical linear algebra

We already saw an algorithm for fast matrix multiplication based on random projection. We will now discuss low rank matrix approximation, which is the linear algebraic analog of subspace approximation.

5.1 Randomized low rank matrix approximation (L4)

Some drawbacks of classical methods that this line of work tries to address:

- Dependence on eigenvalue gaps: We want algorithms that work for all inputs.
- Avoid random access to memory: We want algorithms that work with only a few sequential passes over the input.

5.1.1 Sampling rows independently

The idea of sampling rows for matrix approximation was introduced in a seminal paper by Frieze, Kannan and Vempala. Sampling-based approximation is of course an old statistical trick, so it makes sense that in the span of a random sample of rows of a matrix one may find something close to the best subspace approximation. But uniform sampling is easily seen to be a bad idea: For a matrix that has a single non-zero entry one would need a sample of essentially the same size as the matrix to "see" the interesting entry. So FKV introduced the idea of sampling rows independently with repetition with probability proportional to their squared length:

Theorem 9 (FKV). *Let $A \in \mathbb{R}^{m\times n}$. Let $k \leq l$. Let $S = (s_1, \ldots, s_t)$ be a random sample of t rows of A chosen independently from the following distribution: row i is picked with probability*

$$P_i = \frac{\|A_i\|^2}{\|A\|_F^2}.$$

Then

$$\mathbb{E}(\|A - \tilde{A}\|_F^2) \leq \|A - A_k\|_F^2 + \frac{k}{t}\|A\|_F^2.$$

where $\tilde{A} = \pi_{S,k}(A)$, the best rank-k approximation of A with rows in span (A_S)

Proof. Let $V = \text{span}\,(v_1, \ldots, v_k)$. The best rank-$k$ approximation is given by $A_k = \pi_V(A) = \sum_{i=1}^k Av_iv_i^T$. Given that $\sum_{j=1}^n (u_i)_j A_j = u_i^T A = \sigma_i v_i$, the following is a natural replacement of v_i given our sampling distribution: For j a random row according to squared length, the random variable $X_i = (u_i)_j A_j/P_j$, which has expectation $\sigma_i v_i$. But we are picking t rows (to decrease the variance, say), so our actual random estimate of v_i is

$$y_i = \frac{1}{\sigma_i t}\sum_{j=1}^t (u_i)_j A_j/P_j.$$

A simple perturbation of A_k to get a rank-k approximation to A with rows in span (S) is then $F = \sum_{i=1}^k Av_iy_i^T$. Clearly, the best rank-k approximation with rows in span (A_S) has error satisfying $\|A - \tilde{A}\|_F^2 \leq \|A - F\|_F^2$. To bound this last expression, we write it in a suitable basis, (u_j):

$$\begin{aligned}
\|A - F\|_F^2 &= \sum_{j=1}^n \|u_j^T A - u_j^T \sum_{i=1}^k Av_iy_i^T\|^2 \\
&= \sum_{j=k+1}^n \|\sigma_j v_j\|^2 + \sum_{j=1}^k \|\sigma_j v_j - \sigma_j y_j^T\|^2 \\
&= \|A - A_k\|_F^2 + \sum_{j=1}^k \|\sigma_j v_j - \sigma_j y_j\|^2
\end{aligned}$$

The last term is the sum of second moments of the distance of our approximation $\sigma_j y_j$ to its expected value. This variance is $\mathbb{E}(\|X_j - \sigma_j v_j\|^2)/t$ as we will see now. For any random vector X with mean μ we have $\mathbb{E}(\|X - \mu\|^2) = \mathbb{E}(X^TX) - \mu^T\mu$. For t independent copies $X(1), \ldots, X(t)$ of X we have, similarly,

$$\mathbb{E}(\|\frac{1}{t}\sum_{i=1}^t X(i) - \mu\|^2) = \frac{1}{t}(\mathbb{E}(X^TX) - \mu^T\mu)$$

This implies

$$\mathbb{E}(\|X_j - \sigma_j v_j\|^2) = \sum_{i=1}^{m} \frac{(u_j)_i^2 \|A_i\|^2}{P_i} - \sigma_j^2 = \|A\|_F^2 - \sigma_j^2$$

and

$$\sum_{j=1}^{k} \|\sigma_j v_j - \sigma_j y_j\|^2 = \frac{1}{t} \sum_{j=1}^{k} (\|A\|_F^2 - \sigma_j^2) \leq \frac{k}{t} \|A\|_F^2.$$

This completes the proof. □

For an algorithm, one can first set $t = k/\epsilon$ to control the additive error (it becomes $\epsilon \|A\|_F^2$) and then use Markov's inequality (on the non-negative random variable $\|A - \tilde{A}\|_F^2) - \|A - A_k\|_F^2$) to get a guarantee with constant probability instead of on average. Then one can compute $\tilde{A}$ by running any (truncated) SVD algorithm on $\pi_S(A)$, which is an m by k/ϵ matrix, in additional time $O(mk^2/\epsilon^2)$. The sampling and projection steps take time $O(kmn/\epsilon)$.

Strengths: The algorithm to pick the rows needs only two passes over the data.

Weaknesses: The additive error could be large, when $\|A\|_F^2$ is much larger than $\|A - A_k\|_F^2$ (which can happen when the spectrum decays slowly). It would be better to have an error relative to $\|A - A_k\|_F^2$ (within a multiplicative factor).

Observation: FKV's result *does* gives a factor 2 approximation when $k = t = 1$. That is, one row sampled according to squared length gives a one-dimensional subspace with an error that is no worse than twice the best possible error.

Exercise 4. *Prove the observation.*

5.1.2 Adaptive sampling

Can one improve the error of FKV substantially by picking rows adaptively? Here "adaptively" means that we pick rows in rounds, say t rounds of s rows, where the rows of a certain round are picked with probability proportional to the squared length in the residual or error matrix E from the previous rounds: After r rounds where we have picked rows $S \subset [m]$, $E = A - \pi_S(A)$. Then, using induction on the number of rounds and a slight generalization of FKV (for the inductive step) one can prove that the error decreases exponentially as a function of the number of rounds:

Theorem 10. *Let $S = S_1 \cup \cdots \cup S_t$ be a random sample of rows of an $m \times n$ matrix A where, for $j = 1, \ldots, t$, each set S_j is a sample of s rows of A chosen independently from the following distribution: row i is picked with probability*

$$P_i^{(j)} = \frac{\|E_j^{(i)}\|^2}{\|E_j\|_F^2}$$

where $E_1 = A$, $E_j = A - \pi_{S_1 \cup \cdots \cup S_{j-1}}(A)$. *Then for* $s \geq k/\epsilon$, $\operatorname{span}(A_S)$ *contains the rows of a matrix* $\tilde{A}_k$ *of rank at most* k *such that*

$$\mathbb{E}_S(\|A - \tilde{A}_k\|_F^2) \leq \frac{1}{1-\epsilon}\|A - A_k\|_F^2 + \epsilon^t \|A\|_F^2 \ .$$

The following theorem gives the inductive step:

Theorem 11. *Let* $A \in \mathbb{R}^{m \times n}$, *and* $V \subseteq \mathbb{R}^n$ *be a vector subspace. Let* $E = A - \pi_V(A)$ *and let* S *be a random sample of* s *rows of* A *from a distribution* D *such that row* i *is chosen with probability*

$$P_i = \frac{\|E^{(i)}\|^2}{\|E\|_F^2} \ . \tag{2}$$

Then, for any nonnegative integer k,

$$\mathbb{E}_S(\|A - \pi_{V+\operatorname{span}(S),k}(A)\|_F^2) \leq \|A - \pi_k(A)\|_F^2 + \frac{k}{s}\|E\|_F^2 \ .$$

5.1.3 Volume sampling and relative error approximation

Definition 12. *Given* $A \in \mathbb{R}^{m \times n}$, *volume sampling is defined as picking a* k*-subset* S *of* $[m]$ *with probability proportional to*

$$\det\left(A_S A_S^T\right) = \left(k! \cdot \operatorname{vol}\operatorname{conv}\{\bar{0}\} \cup \{a_i : i \in S\}\right)^2 ,$$

where a_i *denotes the* i*-th row of* A, $A_S \in \mathbb{R}^{k \times n}$ *denotes the row-submatrix of* A *given by rows with indices* $i \in S$, *and* $\operatorname{conv} \cdot$ *denotes the convex hull.*

The application of volume sampling to low-rank approximation and, more importantly, to the *subset selection* problem, is given by the following theorem shown in [**?**]. It says that picking a subset of k rows according to volume sampling and projecting all the rows of A onto their span gives a $(k+1)$-approximation to the nearest rank-k matrix to A.

Theorem 13. *[**?**] Given any* $A \in \mathbb{R}^{m \times n}$,

$$\mathsf{E}\left[\|A - \pi_S(A)\|_F^2\right] \leq (k+1)\|A - A_k\|_F^2 ,$$

when S *is picked according to volume sampling,* $\pi_S(A) \in \mathbb{R}^{m \times n}$ *denotes the matrix obtained by projecting all the rows of* A *onto* $\operatorname{span}(a_i : i \in S)$, *and* A_k *is the matrix of rank* k *closest to* A *under the Frobenius norm.*

Exercise 5. *How to compute the characteristic polynomial of a matrix efficiently? Recall that the characteristic polynomial of an* $n \times n$ *matrix is given by* $p_A(x) = \det(xI - A)$. *Design an algorithm and analyze its running time, based on the following idea: The coefficients of the characteristic polynomial are elementary symmetric polynomials of the eigenvalues. On the other hand,* $trA^k = \sum \lambda_i^k$ *(where* (λ_i) *are the eigenvalues of* A*), which can be computed efficiently for* $k = 1, \ldots, n$. *It is not hard to see that one can go from these sums of powers to the symmetric polynomials efficiently (this is know as "Newton's identities").*

5.2 Subset selection (L5)

Now we discuss the second basic problem discussed in the introduction: Given m points in $\mathbb{R}^n$ and $k \leq l = \min\{m, n\}$, find a k-dimensional subspace spanned by k of the input points so that the sum of squared distances is small. In linear-algebraic terms it is also know as interpolative decomposition: We want to find matrices X, R such that $\|A - XR\|_F$ is small, where the rows of $R \in \mathbb{R}^{k \times n}$ are k rows of A and $X \in \mathbb{R}^{m \times k}$.

This problem could in principle be much harder than subspace approximation, as it has a combinatorial flavor: Select k rows.

We can think of the interpolative decomposition as a refinement of subspace approximation: In subspace approximation we want to find a k-dimensional subspace V such that the error of projecting every row of the given matrix onto V is small in the Frobenius norm:

$$\min_{V:\dim(V)=k} \sum_i d(A_i, V)^2.$$

Eckart-Young implies that a solution to this problem is the span of the top k right singular vectors: $V = \operatorname{span}(u_1, \ldots, u_k)$. Let $U_k = (u_1, \ldots, u_k)$, then the matrix of projected columns is $B = U_k X$, where $X = U_k^T A$ (so $B = U_k U_k^T A$). In the case of the subset selection problem, we want the subspace V to be spanned by k *actual* rows of A (rather than arbitrary vectors, as in PCA). More precisely, the problem is the following: Given A and $k \leq l$, find a subset S of k rows of A such that $\|A - \pi_S(A)\|_F$ is small, where $\pi_S(A)$ denotes matrix having as rows the projections of the rows of A onto the span of the rows in S.

In the literature there are sometimes additional requirements in the subset selection problem. Say, A_S has large smallest singular value (rank revealing QR, RRQR []), and entries in X are not too large (strong RRQR []).

Definition 14. *Given $A \in \mathbb{R}^{m \times n}$, volume sampling is defined as picking a k-subset S of $[m]$ with probability proportional to*

$$\det\left(A_S A_S^T\right) = \left(k! \cdot \operatorname{vol} \operatorname{conv}\left\{\bar{0}\right\} \cup \left\{a_i : i \in S\right\}\right)^2,$$

where a_i denotes the i-th row of A, $A_S \in \mathbb{R}^{k \times n}$ denotes the row-submatrix of A given by rows with indices $i \in S$, and $\operatorname{conv} \cdot$ denotes the convex hull.

The application of volume sampling to low-rank approximation and, more importantly, to the *row-subset selection* problem, is given by the following theorem shown in [**?**]. It says that picking a subset of k rows according to volume sampling and projecting all the rows of A onto their span gives a $(k+1)$-approximation to the nearest rank-k matrix to A.

Theorem 15. *[**?**] Given any $A \in \mathbb{R}^{m \times n}$,*

$$\mathsf{E}\left[\|A - \pi_S(A)\|_F^2\right] \leq (k+1)\|A - A_k\|_F^2,$$

when S is picked according to volume sampling, $\pi_S(A) \in \mathbb{R}^{m \times n}$ denotes the matrix obtained by projecting all the rows of A onto $\operatorname{span}(a_i : i \in S)$, and A_k is the matrix of rank k closest to A under the Frobenius norm.

Theorem 15 gives only an existence result for row-subset selection and we also know a matching lower bound that says this is the best we can possibly do.

Theorem 16. *[?] For any $\epsilon > 0$, there exists a matrix $A \in \mathbb{R}^{(k+1)\times k}$ such that picking any k-subset S of its rows gives*

$$\|A - \pi_S(A)\|_F \geq (1-\epsilon)\sqrt{k+1}\,\|A - A_k\|_F .$$

A delicate analysis of the Johnson-Lindenstrauss transform by Magen and Zouzias [?] shows that one can preserve the volumes of all k-subsets of rows to within a factor of $1+\epsilon$ with a target dimension of $d = O(k^2 \log(m)/\epsilon^2)$. Here is a restatement of Theorem 1 of [?] using $O(\epsilon/k)$ instead of ϵ in their original statement.

Theorem 17. *[?] For any $A \in \mathbb{R}^{m\times n}$, $1 \leq k \leq n$ and $0 < \epsilon \leq 1/2$, there is*

$$d = O\left(\frac{k^2 \log m}{\epsilon^2}\right),$$

and there is a mapping $f : \mathbb{R}^n \to \mathbb{R}^d$ such that

$$\det\left(A_S A_S^T\right) \leq \det\left(\tilde{A}_S \tilde{A}_S^T\right) \leq (1+\epsilon)\det\left(A_S A_S^T\right),$$

for all $S \subseteq [m]$ such that $|S| \leq k$, where $\tilde{A} \in \mathbb{R}^{m\times d}$ has its i-th row as $f(a_i)$. Moreover, with constant probability, multiplication with a random n by d matrix with i.i.d. Gaussian entries (suitably scaled) is such a mapping, so computing $\tilde{A}$ takes time $O(mnd)$.

Theorem 18 (fast volume sampling)**.** *Using random projection for dimensionality reduction, the polynomial time algorithm for volume sampling mentioned in Theorem 19 (i.e., Algorithm 1 with Algorithm 2 as its subroutine), gives $(1+\epsilon)$-approximate volume sampling, using*

$$O\left(mn\log m \cdot \frac{k^2}{\epsilon^2} + m\log^\omega m \cdot \frac{k^{2\omega+1}}{\epsilon^{2\omega}}\log(k\epsilon^{-1}\log m)\right).$$

arithmetic operations.

Theorem 19 (polynomial-time volume sampling)**.** *The randomized algorithm given by the combination of the algorithm outlined in Algorithm 1 with Algorithm 2 as its subroutine, when given a matrix $A \in \mathbb{R}^{m\times n}$ and an integer $1 \leq k \leq \mathrm{rk}(A)$, outputs a random k-subset of the rows of A according to volume sampling, using $O(kmn^\omega \log n)$ arithmetic operations.*

The basic idea of the algorithm is as follows: instead of picking a k-subset, pick an ordered k-tuple of rows according to volume sampling (i.e., volume sampling suitably extended to all k-tuples such that for any fixed k-subset, all its $k!$ permutations are all equally likely). We observe that the marginal distribution of the first coordinate of such a random tuple can be expressed in

terms of coefficients of the characteristic polynomials of AA^T and $B_iB_i^T$, where $B_i \in \mathbb{R}^{m\times n}$ is the matrix obtained by projecting each row of A orthogonal to the i-th row a_i. Using this interpretation, it is easy to sample the first index of the k-tuple with the right marginal probability. Now we project the rows of A orthogonal to the chosen row and repeat to pick the next row, until we have picked k of them.

Algorithm 1. Outline of our volume sampling algorithm

Input: a matrix $A \in \mathbb{R}^{m\times n}$ and $1 \le k \le \mathrm{rk}(A)$.
Output: a subset S of k rows of A picked with probability proportional to $\det\left(A_S A_S^T\right)$.

1. Initialize $S \leftarrow \emptyset$ and $B \leftarrow A$. For $t = 1$ to k do:
   (a) For $i = 1$ to m compute:
   $$p_i = \|b_i\|^2 \cdot |c_{m-k+t}(C_i C_i^T)|,$$
   where $C_i = B - \pi_{\{i\}}(B)$ is a matrix obtained by projecting each row of B orthogonal to b_i.
   (b) Pick i with probability proportional to p_i. Let $S \leftarrow S \cup \{i\}$ and $B \leftarrow C_i$.
2. Output S.

Algorithm 2. First subroutine for marginal probabilities

Input: $B \in \mathbb{R}^{m\times n}$.
Output: $p_1, p_2, \ldots, p_m$.

For $i = 1$ to m do:

1. Compute the matrix $C_i^T C_i \in \mathbb{R}^{n\times n}$ by the following formula
$$C_i^T C_i = B^T B - \frac{B^T B b_i b_i^T}{\|b_i\|^2} - \frac{b_i b_i^T B^T B}{\|b_i\|^2} + \frac{b_i b_i^T B^T B b_i b_i^T}{\|b_i\|^4}.$$
2. Compute the characteristic polynomial of $C_i^T C_i$ and output
$$p_i = \|b_i\|^2 \cdot |c_{n-k+t}(C_i^T C_i)|.$$

open problems: tensor decomposition, rank, etc restricted invertibility, Kadison-Singer a la Vershynin