The last two examples rely heavily on the commutativity of the summands as well as the infinite divisibility of the normal and Poisson distributions. As a consequence, it may appear that the logarithms only appear in very special contexts. In fact, many (but not all!) examples that arise in practice do require the logarithms that appear in the matrix Bernstein inequality. It is a subject of ongoing research to obtain a simple criterion for deciding when the logarithms belong.

6.2 Example: Matrix Approximation by Random Sampling

In applied mathematics, we often need to approximate a complicated target object by a more structured object. In some situations, we can solve this problem using a beautiful probabilistic approach called *empirical approximation*. The basic idea is to construct a "simple" random object whose expectation equals the target. We obtain the approximation by averaging several independent copies of the simple random object. As the number of terms in this average increases, the approximation becomes more complex, but it represents the target more faithfully. The challenge is to quantify this tradeoff.

In particular, we often encounter problems where we need to approximate a matrix by a more structured matrix. For example, we may wish to find a sparse matrix that is close to a given matrix, or we may need to construct a low-rank matrix that is close to a given matrix. Empirical approximation provides a mechanism for obtaining these approximations. The matrix Bernstein inequality offers a natural tool for assessing the quality of the randomized approximation.

This section develops a general framework for empirical approximation of matrices. Subsequent sections explain how this technique applies to specific examples from the fields of randomized linear algebra and machine learning.

6.2.1 Setup

Let $\boldsymbol{B}$ be a target matrix that we hope to approximate by a more structured matrix. To that end, let us represent the target as a sum of "simple" matrices:

$$\boldsymbol{B} = \sum_{i=1}^{N} \boldsymbol{B}_i. \tag{6.2.1}$$

The idea is to identify summands with desirable properties that we want our approximation to inherit. The examples in this chapter depend on decompositions of the form (6.2.1).

Along with the decomposition (6.2.1), we need a set of sampling probabilities:

$$\sum_{i=1}^{N} p_i = 1 \quad \text{and} \quad p_i > 0 \quad \text{for } i = 1, \dots, N. \tag{6.2.2}$$

We want to ascribe larger probabilities to "more important" summands. Quantifying what "important" means is the most difficult aspect of randomized matrix approximation. Choosing the right sampling distribution for a specific problem requires insight and ingenuity.

Given the data (6.2.1) and (6.2.2), we may construct a "simple" random matrix $\boldsymbol{R}$ by sampling:

$$\boldsymbol{R} = p_i^{-1} \boldsymbol{B}_i \quad \text{with probability } p_i. \tag{6.2.3}$$

This construction ensures that $\boldsymbol{R}$ is an unbiased estimator of the target: $\mathbb{E}\,\boldsymbol{R} = \boldsymbol{B}$. Even so, the random matrix $\boldsymbol{R}$ offers a poor approximation of the target $\boldsymbol{B}$ because it has a lot more structure.

To improve the quality of the approximation, we average n independent copies of the random matrix $\boldsymbol{R}$. We obtain an estimator of the form

$$\bar{\boldsymbol{R}}_n = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{R}_k \quad \text{where each } \boldsymbol{R}_k \text{ is an independent copy of } \boldsymbol{R}.$$

By linearity of expectation, this estimator is also unbiased: $\mathbb{E}\,\bar{\boldsymbol{R}}_n = \boldsymbol{B}$. The approximation $\bar{\boldsymbol{R}}_n$ remains structured when the number n of terms in the approximation is small as compared with the number N of terms in the decomposition (6.2.1).

Our goal is to quantify the approximation error as a function of the complexity n of the approximation:

$$\mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{B}\| \le \operatorname{error}(n).$$

There is a tension between the total number n of terms in the approximation and the error $\operatorname{error}(n)$ the approximation incurs. In applications, it is essential to achieve the right balance.

6.2.2 Error Estimate for Matrix Sampling Estimators

We can obtain an error estimate for the approximation scheme described in Section 6.2.1 as an immediate corollary of the matrix Bernstein inequality, Theorem 6.1.1.

Corollary 6.2.1 (Matrix Approximation by Random Sampling)**.** *Let $\boldsymbol{B}$ be a fixed $d_1 \times d_2$ matrix. Construct a $d_1 \times d_2$ random matrix $\boldsymbol{R}$ that satisfies*

$$\mathbb{E}\,\boldsymbol{R} = \boldsymbol{B} \quad \text{and} \quad \|\boldsymbol{R}\| \le L.$$

Compute the per-sample second moment:

$$m_2(\boldsymbol{R}) = \max\left\{\|\mathbb{E}(\boldsymbol{R}\boldsymbol{R}^*)\|,\ \|\mathbb{E}(\boldsymbol{R}^*\boldsymbol{R})\|\right\}. \tag{6.2.4}$$

Form the matrix sampling estimator

$$\bar{\boldsymbol{R}}_n = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{R}_k \quad \text{where each } \boldsymbol{R}_k \text{ is an independent copy of } \boldsymbol{R}.$$

Then the estimator satisfies

$$\mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{B}\| \le \sqrt{\frac{2m_2(\boldsymbol{R})\log(d_1+d_2)}{n}} + \frac{2L\log(d_1+d_2)}{3n}. \tag{6.2.5}$$

Furthermore, for all $t \ge 0$,

$$\mathbb{P}\left\{\|\bar{\boldsymbol{R}}_n - \boldsymbol{B}\| \ge t\right\} \le (d_1+d_2)\exp\left(\frac{-nt^2/2}{m_2(\boldsymbol{R}) + 2Lt/3}\right). \tag{6.2.6}$$

Proof. Since $\boldsymbol{R}$ is an unbiased estimator of the target matrix $\boldsymbol{B}$, we can write

$$\boldsymbol{Z} = \bar{\boldsymbol{R}}_n - \boldsymbol{B} = \frac{1}{n}\sum_{k=1}^{n}(\boldsymbol{R}_k - \mathbb{E}\,\boldsymbol{R}) = \sum_{k=1}^{n}\boldsymbol{S}_k.$$

We have defined the summands $\boldsymbol{S}_k = n^{-1}(\boldsymbol{R}_k - \mathbb{E}\,\boldsymbol{R})$. These random matrices form an independent and identically distributed family, and each $\boldsymbol{S}_k$ has mean zero.

Now, each of the summands is subject to an upper bound:

$$\|\boldsymbol{S}_k\| \le \frac{1}{n}\left(\|\boldsymbol{R}_k\| + \|\mathbb{E}\,\boldsymbol{R}\|\right) \le \frac{1}{n}\left(\|\boldsymbol{R}_k\| + \mathbb{E}\,\|\boldsymbol{R}\|\right) \le \frac{2L}{n}.$$

The first relation is the triangle inequality; the second is Jensen's inequality. The last estimate follows from our assumption that $\|\boldsymbol{R}\| \le L$.

To control the matrix variance statistic $v(\boldsymbol{Z})$, first note that

$$v(\boldsymbol{Z}) = \max\left\{\left\|\sum_{k=1}^{n}\mathbb{E}\left(\boldsymbol{S}_k\boldsymbol{S}_k^*\right)\right\|, \left\|\sum_{k=1}^{n}\mathbb{E}\left(\boldsymbol{S}_k^*\boldsymbol{S}_k\right)\right\|\right\} = n\cdot\max\left\{\|\mathbb{E}(\boldsymbol{S}_1\boldsymbol{S}_1^*)\|,\ \|\mathbb{E}(\boldsymbol{S}_1^*\boldsymbol{S}_1)\|\right\}.$$

The first identity follows from the expression (6.1.2) for the matrix variance statistic, and the second holds because the summands $\boldsymbol{S}_k$ are identically distributed. We may calculate that

$$\begin{aligned}\boldsymbol{0} \preccurlyeq \mathbb{E}(\boldsymbol{S}_1\boldsymbol{S}_1^*) &= n^{-2}\,\mathbb{E}\left[(\boldsymbol{R}-\mathbb{E}\,\boldsymbol{R})(\boldsymbol{R}-\mathbb{E}\,\boldsymbol{R})^*\right]\\ &= n^{-2}\left[\mathbb{E}(\boldsymbol{R}\boldsymbol{R}^*) - (\mathbb{E}\,\boldsymbol{R})(\mathbb{E}\,\boldsymbol{R})^*\right] \preccurlyeq n^{-2}\,\mathbb{E}(\boldsymbol{R}\boldsymbol{R}^*).\end{aligned}$$

The first relation holds because the expectation of the random positive-semidefinite matrix $\boldsymbol{S}_1\boldsymbol{S}_1^*$ is positive semidefinite. The first identity follows from the definition of $\boldsymbol{S}_1$ and the fact that $\boldsymbol{R}_1$ has the same distribution as $\boldsymbol{R}$. The second identity is a direct calculation. The last relation holds because $(\mathbb{E}\,\boldsymbol{R})(\mathbb{E}\,\boldsymbol{R})^*$ is positive semidefinite. As a consequence,

$$\|\mathbb{E}(\boldsymbol{S}_1\boldsymbol{S}_1^*)\| \le \frac{1}{n^2}\,\|\mathbb{E}(\boldsymbol{R}\boldsymbol{R}^*)\|.$$

Likewise,

$$\|\mathbb{E}(\boldsymbol{S}_1^*\boldsymbol{S}_1)\| \le \frac{1}{n^2}\,\|\mathbb{E}(\boldsymbol{R}^*\boldsymbol{R})\|.$$

In summary,

$$v(\boldsymbol{Z}) \le \frac{1}{n}\max\left\{\|\mathbb{E}(\boldsymbol{R}\boldsymbol{R}^*)\|,\ \|\mathbb{E}(\boldsymbol{R}^*\boldsymbol{R})\|\right\} = \frac{m_2(\boldsymbol{R})}{n}.$$

The last line follows from the definition (6.2.4) of $m_2(\boldsymbol{R})$.

We are prepared to apply the matrix Bernstein inequality, Theorem 6.1.1, to the random matrix $\boldsymbol{Z} = \sum_k \boldsymbol{S}_k$. This operation results in the statement of the corollary. □

6.2.3 Discussion

One of the most common applications of the matrix Bernstein inequality is to analyze empirical matrix approximations. As a consequence, Corollary 6.2.1 is one of the most useful forms of the matrix Bernstein inequality. Let us discuss some of the important aspects of this result.

Understanding the Bound on the Approximation Error

First, let us examine how many samples n suffice to bring the approximation error bound in Corollary 6.2.1 below a specified positive tolerance ε. Examining inequality (7.3.5), we find that

$$n \ge \frac{2m_2(\boldsymbol{R})\log(d_1+d_2)}{\varepsilon^2} + \frac{2L\log(d_1+d_2)}{3\varepsilon} \quad\text{implies}\quad \mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{B}\| \le 2\varepsilon. \tag{6.2.7}$$

Roughly, the number n of samples should be on the scale of the per-sample second moment $m_2(\boldsymbol{R})$ and the uniform upper bound L.

The bound (6.2.7) also reveals an unfortunate aspect of empirical matrix approximation. To make the tolerance ε small, the number n of samples must increase proportional with ε^{-2}. In other words, it takes many samples to achieve a highly accurate approximation. We cannot avoid this phenomenon, which ultimately is a consequence of the central limit theorem.

On a more positive note, it is quite valuable that the error bounds (7.3.5) and (7.3.6) involve the spectral norm. This type of estimate simultaneously controls the error in every linear function of the approximation:

$$\|\bar{\boldsymbol{R}}_n - \boldsymbol{B}\| \le \varepsilon \quad \text{implies} \quad \left|\operatorname{tr}(\bar{\boldsymbol{R}}_n \boldsymbol{C}) - \operatorname{tr}(\boldsymbol{B}\boldsymbol{C})\right| \le \varepsilon \quad \text{when } \|\boldsymbol{C}\|_{S_1} \le 1.$$

The Schatten 1-norm $\|\cdot\|_{S_1}$ is defined in (2.1.29). These bounds also control the error in each singular value $\sigma_j(\bar{\boldsymbol{R}}_n)$ of the approximation:

$$\|\bar{\boldsymbol{R}}_n - \boldsymbol{B}\| \le \varepsilon \quad \text{implies} \quad \left|\sigma_j(\bar{\boldsymbol{R}}_n) - \sigma_j(\boldsymbol{B})\right| \le \varepsilon \quad \text{for each } j = 1, 2, 3, \dots, \min\{d_1, d_2\}.$$

When there is a gap between two singular values of $\boldsymbol{B}$, we can also obtain bounds for the discrepancy between the associated singular vectors of $\bar{\boldsymbol{R}}_n$ and $\boldsymbol{B}$ using perturbation theory.

To construct a good sampling estimator $\boldsymbol{R}$, we ought to control both $m_2(\boldsymbol{R})$ and L. In practice, this demands considerable creativity. This observation hints at the possibility of achieving a bias–variance tradeoff when approximating $\boldsymbol{B}$. To do so, we can drop all of the "unimportant" terms in the representation (6.2.1), i.e., those whose sampling probabilities are small. Then we construct a random approximation $\boldsymbol{R}$ only for the "important" terms that remain. Properly executed, this process may decrease both the per-sample second moment $m_2(\boldsymbol{R})$ and the upper bound L. The idea is analogous with shrinkage in statistical estimation.

A General Sampling Model

Corollary 6.2.1 extends beyond the sampling model based on the finite expansion (6.2.1). Indeed, we can consider a more general decomposition of the target matrix $\boldsymbol{B}$:

$$\boldsymbol{B} = \int_\Omega \boldsymbol{B}(\omega)\, \mathrm{d}\mu(\omega)$$

where μ is a probability measure on a sample space Ω. As before, the idea is to represent the target matrix $\boldsymbol{B}$ as an average of "simple" matrices $\boldsymbol{B}(\omega)$. The main difference is that the family of simple matrices may now be infinite. In this setting, we construct the random approximation $\boldsymbol{R}$ so that

$$\mathbb{P}\{\boldsymbol{R} \in E\} = \mu\{\omega : \boldsymbol{B}(\omega) \in E\} \quad \text{for } E \subset \mathbb{M}^{d_1 \times d_2}$$

In particular, it follows that

$$\mathbb{E}\,\boldsymbol{R} = \boldsymbol{B} \quad \text{and} \quad \|\boldsymbol{R}\| \le \sup_{\omega \in \Omega} \|\boldsymbol{B}(\omega)\|.$$

As we will discuss, this abstraction is important for applications in machine learning.

Suboptimality of Sampling Estimators

Another fundamental point about sampling estimators is that they are usually suboptimal. In other words, the matrix sampling estimator may incur an error substantially worse than the error in the best structured approximation of the target matrix.

To see why, let us consider a simple form of low-rank approximation by random sampling. The method here does not have practical value, but it highlights the reason that sampling estimators usually do not achieve ideal results. Suppose that $\boldsymbol{B}$ has singular value decomposition

$$\boldsymbol{B} = \sum_{i=1}^{N} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^* \quad \text{where} \quad \sum_{i=1}^{N} \sigma_i = 1 \quad \text{and} \quad N = \min\{d_1, d_2\}.$$

Given the SVD, we can construct a random rank-one approximation $\boldsymbol{R}$ of the form

$$\boldsymbol{R} = \boldsymbol{u}_i \boldsymbol{v}_i^* \quad \text{with probability} \quad \sigma_i.$$

Per Corollary 6.2.1, the error in the associated sampling estimator $\bar{\boldsymbol{R}}_n$ of $\boldsymbol{B}$ satisfies

$$\|\bar{\boldsymbol{R}}_n - \boldsymbol{B}\| \le \sqrt{\frac{2\log(d_1+d_2)}{n}} + \frac{2\log(d_1+d_2)}{n}$$

On the other hand, a best rank-n approximation of $\boldsymbol{B}$ takes the form $\boldsymbol{B}_n = \sum_{j=1}^{n} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^*$, and it incurs error

$$\|\boldsymbol{B}_n - \boldsymbol{B}\| = \sigma_{n+1} \le \frac{1}{n+1}.$$

The second relation is Markov's inequality, which provides an accurate estimate only when the singular values $\sigma_1, \dots, \sigma_{n+1}$ are comparable. In that case, the sampling estimator arrives within a logarithmic factor of the optimal error. But there are many matrices whose singular values decay quickly, so that $\sigma_{n+1} \ll (n+1)^{-1}$. In the latter situation, the error in the sampling estimator is much worse than the optimal error.

Warning: Frobenius-Norm Bounds

We often encounter papers that develop Frobenius-norm error bounds for matrix approximations, perhaps because the analysis is more elementary. But one must recognize that Frobenius-norm error bounds are not acceptable in most cases of practical interest:

Frobenius-norm error bounds are typically vacuous.

In particular, this phenomenon occurs in data analysis whenever we try to approximate a matrix that contains white or pink noise.

To illustrate this point, let us consider the ubiquitous problem of approximating a low-rank matrix corrupted by additive white Gaussian noise:

$$\boldsymbol{B} = \boldsymbol{x}\boldsymbol{x}^* + \alpha \boldsymbol{E} \in \mathbb{M}_d. \quad \text{where} \quad \|\boldsymbol{x}\|^2 = 1. \tag{6.2.8}$$

The desired approximation of the matrix $\boldsymbol{B}$ is the rank-one matrix $\boldsymbol{B}_{\text{opt}} = \boldsymbol{x}\boldsymbol{x}^*$. For modeling purposes, we assume that $\boldsymbol{E}$ has independent $\text{NORMAL}(0, d^{-1})$ entries. As a consequence,

$$\|\boldsymbol{E}\| \approx 2 \quad \text{and} \quad \|\boldsymbol{E}\|_{\text{F}} \approx \sqrt{d}.$$

Now, the spectral-norm error in the desired approximation satisfies

$$\|\boldsymbol{B}_{\text{opt}} - \boldsymbol{B}\| = \alpha \|\boldsymbol{E}\| \approx 2\alpha.$$

On the other hand, the Frobenius-norm error in the desired approximation satisfies

$$\|\boldsymbol{B}_{\text{opt}} - \boldsymbol{B}\|_{\text{F}} = \alpha \|\boldsymbol{E}\|_{\text{F}} \approx \alpha\sqrt{d}.$$

We see that the Frobenius-norm error can be quite large, even when we find the required approximation.

Here is another way to look at the same fact. Suppose we construct an approximation $\widehat{\boldsymbol{B}}$ of the matrix $\boldsymbol{B}$ from (6.2.8) whose Frobenius-norm error is comparable with the optimal error:

$$\|\widehat{\boldsymbol{B}} - \boldsymbol{B}\|_{\text{F}} \leq \varepsilon\sqrt{d}.$$

There is no reason for the approximation $\widehat{\boldsymbol{B}}$ to have any relationship with the desired approximation $\boldsymbol{B}_{\text{opt}}$. For example, the approximation $\widehat{\boldsymbol{B}} = \alpha\boldsymbol{E}$ satisfies this error bound with $\varepsilon = d^{-1/2}$ even though $\widehat{\boldsymbol{B}}$ consists only of noise.

6.3 Application: Randomized Sparsification of a Matrix

Many tasks in data analysis involve large, dense matrices that contain a lot of redundant information. For example, an experiment that tabulates many variables about a large number of subjects typically results in a low-rank data matrix because subjects are often similar with each other. Many questions that we pose about these data matrices can be addressed by spectral computations. In particular, factor analysis involves a singular value decomposition.

When the data matrix is approximately low rank, it has fewer degrees of freedom than its ambient dimension. Therefore, we can construct a simpler approximation that still captures most of the information in the matrix. One method for finding this approximation is to replace the dense target matrix by a sparse matrix that is close in spectral-norm distance. An elegant way to identify this sparse proxy is to randomly select a small number of entries from the original matrix to retain. This is a type of empirical approximation.

Sparsification has several potential advantages. First, it is considerably less expensive to store a sparse matrix than a dense matrix. Second, many algorithms for spectral computation operate more efficiently on sparse matrices.

In this section, we examine a very recent approach to randomized sparsification due to Kundu & Drineas [KD14]. The analysis is an immediate consequence of Corollary 6.2.1. See the notes at the end of the chapter for history and references.

6.3.1 Problem Formulation & Randomized Algorithm

Let $\boldsymbol{B}$ be a fixed $d_1 \times d_2$ complex matrix. The sparsification problem requires us to find a sparse matrix $\widehat{B}$ that has small distance from $\boldsymbol{B}$ with respect to the spectral norm. We can achieve this goal using an empirical approximation strategy.

First, let us express the target matrix as a sum of its entries:

$$\boldsymbol{B} = \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} b_{ij}\, \mathbf{E}_{ij}.$$

Introduce sampling probabilities

$$p_{ij} = \frac{1}{2}\left[\frac{|b_{ij}|^2}{\|\boldsymbol{B}\|_{\mathrm{F}}^2} + \frac{|b_{ij}|}{\|\boldsymbol{B}\|_{\ell_1}}\right] \quad \text{for } i = 1,\dots,d_1 \text{ and } j = 1,\dots,d_2. \tag{6.3.1}$$

The Frobenius norm is defined in (2.1.2), and the entrywise ℓ_1 norm is defined in (2.1.30). It is easy to check that the numbers p_{ij} form a probability distribution. Let us emphasize that the non-obvious form of the distribution (6.3.1) represents a decade of research.

Now, we introduce a $d_1 \times d_2$ random matrix $\boldsymbol{R}$ that has exactly one nonzero entry:

$$\boldsymbol{R} = \frac{1}{p_{ij}} \cdot b_{ij}\mathbf{E}_{ij} \quad \text{with probability } p_{ij}.$$

We use the convention that $0/0 = 0$ so that we do not need to treat zero entries separately. It is immediate that

$$\mathbb{E}\,\boldsymbol{R} = \sum_{i=1}^{d_1}\sum_{j=1}^{d_2} \frac{1}{p_{ij}} \cdot b_{ij}\mathbf{E}_{ij} \cdot p_{ij} = \sum_{i=1}^{d_1}\sum_{j=1}^{d_2} b_{ij}\mathbf{E}_{ij} = \boldsymbol{B}.$$

Therefore, $\boldsymbol{R}$ is an unbiased estimate of $\boldsymbol{B}$.

Although the expectation of $\boldsymbol{R}$ is correct, its variance is quite high. Indeed, $\boldsymbol{R}$ has only one nonzero entry, while $\boldsymbol{B}$ typically has many nonzero entries. To reduce the variance, we combine several independent copies of the simple estimator:

$$\bar{\boldsymbol{R}}_n = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{R}_k \quad \text{where each } \boldsymbol{R}_k \text{ is an independent copy of } \boldsymbol{R}.$$

By linearity of expectation, $\mathbb{E}\,\bar{\boldsymbol{R}}_n = \boldsymbol{B}$. Therefore, the matrix $\bar{\boldsymbol{R}}_n$ has at most n nonzero entries, and its also provides an unbiased estimate of the target. The challenge is to quantify the error $\|\bar{\boldsymbol{R}}_n - \boldsymbol{B}\|$ as a function of the sparsity level n.

6.3.2 Performance of Randomized Sparsification

The randomized sparsification method is clearly a type of empirical approximation, so we can use Corollary 6.2.1 to perform the analysis. We will establish the following error bound.

$$\mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{B}\| \le \sqrt{\frac{4\,\|\boldsymbol{B}\|_{\mathrm{F}}^2 \cdot \max\{d_1,d_2\}\log(d_1+d_2)}{n}} + \frac{4\,\|\boldsymbol{B}\|_{\ell_1}\log(d_1+d_2)}{3n}. \tag{6.3.2}$$

The short proof of (6.3.2) appears below in Section 6.3.3.

Let us explain the content of the estimate (6.3.2). First, the bound (2.1.31) allows us to replace the ℓ_1 norm by the Frobenius norm:

$$\|\boldsymbol{B}\|_{\ell_1} \le \sqrt{d_1 d_2} \cdot \|\boldsymbol{B}\|_{\mathrm{F}} \le \max\{d_1,\, d_2\} \cdot \|\boldsymbol{B}\|_{\mathrm{F}}.$$

Placing the error (6.3.2) on a relative scale, we see that

$$\frac{\mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{B}\|}{\|\boldsymbol{B}\|} \le \frac{\|\boldsymbol{B}\|_{\mathrm{F}}}{\|\boldsymbol{B}\|} \cdot \left[\sqrt{\frac{4\max\{d_1,\, d_2\}\log(d_1+d_2)}{n}} + \frac{4\max\{d_1,\, d_2\}\log(d_1+d_2)}{3n}\right]$$

The stable rank srank($\boldsymbol{B}$), defined in (2.1.25), emerges naturally as a quantity of interest.

Now, suppose that the sparsity level n satisfies

$$n \geq \varepsilon^{-2} \cdot \operatorname{srank}(\boldsymbol{B}) \cdot \max\{d_1,\, d_2\} \log(d_1 + d_2)$$

where the tolerance $\varepsilon \in (0, 1]$. We determine that

$$\frac{\mathbb{E}\,\|\tilde{\boldsymbol{R}}_n - \boldsymbol{B}\|}{\|\boldsymbol{B}\|} \leq 2\varepsilon + \frac{4}{3} \cdot \frac{\varepsilon^2}{\sqrt{\operatorname{srank}(\boldsymbol{B})}}.$$

Since the stable rank always exceeds one and we have assumed that $\varepsilon \leq 1$, this estimate implies that

$$\frac{\mathbb{E}\,\|\tilde{\boldsymbol{R}}_n - \boldsymbol{B}\|}{\|\boldsymbol{B}\|} \leq 4\varepsilon.$$

We discover that it is possible to replace the matrix $\boldsymbol{B}$ by a matrix with at most n nonzero entries while achieving a small relative error in the spectral norm. When $\operatorname{srank}(\boldsymbol{B}) \ll \min\{d_1,\, d_2\}$, we can achieve a dramatic reduction in the number of nonzero entries needed to carry the spectral information in the matrix $\boldsymbol{B}$.

6.3.3 Analysis of Randomized Sparsification

Let us proceed with the analysis of randomized sparsification. To apply Corollary 6.2.1, we need to obtain bounds for the per-sample variance $m_2(\boldsymbol{R})$ and the uniform upper bound L. The key to both calculations is to obtain appropriate *lower* bounds on the sampling probabilities p_{ij}. Indeed,

$$p_{ij} \geq \frac{1}{2} \cdot \frac{|b_{ij}|}{\|\boldsymbol{B}\|_{\ell_1}} \quad \text{and} \quad p_{ij} \geq \frac{1}{2} \cdot \frac{|b_{ij}|^2}{\|\boldsymbol{B}\|_{\mathrm{F}}^2}. \tag{6.3.3}$$

Each estimate follows by neglecting one term in (6.3.3).

First, we turn to the uniform bound on the random matrix $\boldsymbol{R}$. We have

$$\|\boldsymbol{R}\| \leq \max_{ij} \|p_{ij}^{-1} b_{ij} \mathbf{E}_{ij}\| = \max_{ij} \frac{1}{p_{ij}} \cdot |b_{ij}| \leq 2\,\|\boldsymbol{B}\|_{\ell_1}.$$

The last inequality depends on the first bound in (6.3.3). Therefore, we may take $L = 2\,\|\boldsymbol{B}\|_{\ell_1}$.

Second, we turn to the computation of the per-sample second moment $m_2(\boldsymbol{R})$. We have

$$\begin{aligned}
\mathbb{E}(\boldsymbol{R}\boldsymbol{R}^*) &= \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \frac{1}{p_{ij}^2} \cdot (b_{ij}\mathbf{E}_{ij})(b_{ij}\mathbf{E}_{ij})^* p_{ij} \\
&= \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \frac{|b_{ij}|^2}{p_{ij}} \cdot \mathbf{E}_{ii} \\
&\preccurlyeq 2\,\|\boldsymbol{B}\|_{\mathrm{F}}^2 \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \mathbf{E}_{ii} = 2d_2\,\|\boldsymbol{B}\|_{\mathrm{F}}^2 \cdot \mathbf{I}_{d_1}.
\end{aligned}$$

The semidefinite inequality holds because each matrix $|b_{ij}|^2 \mathbf{E}_{ii}$ is positive semidefinite and because of the second bound in (6.3.3). Similarly,

$$\mathbb{E}(\boldsymbol{R}^*\boldsymbol{R}) \preccurlyeq 2d_1\,\|\boldsymbol{B}\|_{\mathrm{F}}^2 \cdot \mathbf{I}_{d_2}.$$

In summary,

$$m_2(\boldsymbol{R}) = \max\left\{ \|\mathbb{E}(\boldsymbol{R}\boldsymbol{R}^*)\|, \|\mathbb{E}(\boldsymbol{R}^*\boldsymbol{R})\| \right\} \le 2\max\{d_1, d_2\}.$$

This is the required estimate for the per-sample second moment.

Finally, to reach the advertised error bound (6.3.2), we invoke Corollary 6.2.1 with the parameters $L = \|\boldsymbol{B}\|_{\ell_1}$ and $m_2(\boldsymbol{R}) \le 2\max\{d_1, d_2\}$.

6.4 Application: Randomized Matrix Multiplication

Numerical linear algebra (NLA) is a well-established and important part of computer science. Some of the basic problems in this area include multiplying matrices, solving linear systems, computing eigenvalues and eigenvectors, and solving linear least-squares problems. Historically, the NLA community has focused on developing highly accurate deterministic methods that require as few floating-point operations as possible. Unfortunately, contemporary applications can strain standard NLA methods because problems have continued to become larger. Furthermore, on modern computer architectures, computational costs depend heavily on communication and other resources that the standard algorithms do not manage very well.

In response to these challenges, researchers have started to develop randomized algorithms for core problems in NLA. In contrast to the classical algorithms, these new methods make random choices during execution to achieve computational efficiencies. These randomized algorithms can also be useful for large problems or for modern computer architectures. On the other hand, randomized methods can fail with some probability, and in some cases they are less accurate than their classical competitors.

Matrix concentration inequalities are one of the key tools used to design and analyze randomized algorithms for NLA problems. In this section, we will describe a randomized method for matrix multiplication developed by Magen & Zouzias [MZ11, Zou13]. We will analyze this algorithm using Corollary 6.2.1. Turn to the notes at the end of the chapter for more information about the history.

6.4.1 Problem Formulation & Randomized Algorithm

One of the basic tasks in numerical linear algebra is to multiply two matrices with compatible dimensions. Suppose that $\boldsymbol{B}$ is a $d_1 \times N$ complex matrix and that $\boldsymbol{C}$ is an $N \times d_2$ complex matrix, and we wish to compute the product $\boldsymbol{B}\boldsymbol{C}$. The straightforward algorithm forms the product entry by entry:

$$(\boldsymbol{B}\boldsymbol{C})_{ik} = \sum_{j=1}^{N} b_{ij} c_{jk} \quad \text{for each } i = 1, \dots, d_1 \text{ and } k = 1, \dots, d_2. \tag{6.4.1}$$

This approach takes $\mathrm{O}(N \cdot d_1 d_2)$ arithmetic operations. There are algorithms, such as Strassen's divide-and-conquer method, that can reduce the cost, but these approaches are not considered practical for most applications.

Suppose that the inner dimension N is substantially larger than the outer dimensions d_1 and d_2. In this setting, both matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ are rank-deficient, so the columns of $\boldsymbol{B}$ contain a lot of linear dependencies, as do the rows of $\boldsymbol{C}$. As a consequence, a random sample of columns from $\boldsymbol{B}$ (or rows from $\boldsymbol{C}$) can be used as a proxy for the full matrix. Formally, the key to this approach

is to view the matrix product as a sum of outer products:

$$\boldsymbol{BC} = \sum_{j=1}^{N} \boldsymbol{b}_{:j}\boldsymbol{c}_{j:}. \tag{6.4.2}$$

As usual, $\boldsymbol{b}_{:j}$ denotes the jth column of $\boldsymbol{B}$, while $\boldsymbol{c}_{j:}$ denotes the jth row of $\boldsymbol{C}$. We can approximate this sum using the empirical method.

To develop an algorithm, the first step is to construct a simple random matrix $\boldsymbol{R}$ that provides an unbiased estimate for the matrix product. To that end, we pick a random index and form a rank-one matrix from the associated columns of $\boldsymbol{B}$ and row of $\boldsymbol{C}$. More precisely, define

$$p_j = \frac{\|\boldsymbol{b}_{:j}\|^2 + \|\boldsymbol{c}_{j:}\|^2}{\|\boldsymbol{B}\|_{\mathrm{F}}^2 + \|\boldsymbol{C}\|_{\mathrm{F}}^2} \quad \text{for } j = 1, 2, 3, \dots, N. \tag{6.4.3}$$

The Frobenius norm is defined in (2.1.2). Using the properties of the norms, we can easily check that $(p_1, p_2, p_3, \dots, p_N)$ forms a bonafide probability distribution. The cost of computing these probabilities is at most $O(N \cdot (d_1 + d_2))$ arithmetic operations, which is much smaller than the cost of forming the product $\boldsymbol{BC}$ when d_1 and d_2 are large.

We now define a $d_1 \times d_2$ random matrix $\boldsymbol{R}$ by the expression

$$\boldsymbol{R} = \frac{1}{p_j} \cdot \boldsymbol{b}_{:j}\boldsymbol{c}_{j:} \quad \text{with probability } p_j.$$

We use the convention that $0/0 = 0$ so we do not have to treat zero rows and columns separately. It is straightforward to compute the expectation of $\boldsymbol{R}$:

$$\mathbb{E}\,\boldsymbol{R} = \sum_{j=1}^{N} \frac{1}{p_j} \cdot \boldsymbol{b}_{:j}\boldsymbol{c}_{j:} \cdot p_j = \sum_{j=1}^{N} \boldsymbol{b}_{:j}\boldsymbol{c}_{j:} = \boldsymbol{BC}.$$

As required, $\boldsymbol{R}$ is an unbiased estimator for the product $\boldsymbol{BC}$.

Although the expectation of $\boldsymbol{R}$ is correct, its variance is quite high. Indeed, $\boldsymbol{R}$ has rank one, while the rank of $\boldsymbol{BC}$ is usually larger! To reduce the variance, we combine several independent copies of the simple estimator:

$$\bar{\boldsymbol{R}}_n = \frac{1}{n} \sum_{k=1}^{n} \boldsymbol{R}_k \quad \text{where} \quad \text{each } \boldsymbol{R}_k \text{ is an independent copy of } \boldsymbol{R}. \tag{6.4.4}$$

By linearity of expectation, $\mathbb{E}\,\bar{\boldsymbol{R}}_n = \boldsymbol{BC}$, so we imagine that $\bar{\boldsymbol{R}}_n$ approximates the product well.

To see whether this heuristic holds true, we need to understand how the error $\mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{BC}\|$ depends on the number n of samples. It costs $O(n \cdot d_1 d_2)$ floating-point operations to determine all the entries of $\bar{\boldsymbol{R}}_n$. Therefore, when the number n of samples is much smaller than the inner dimension N of the matrices, we can achieve significant economies over the naïve matrix multiplication algorithm.

In fact, it requires no computation beyond sampling the row/column indices to express $\bar{\boldsymbol{R}}_n$ in the form (6.4.4). This approach gives an inexpensive way to represent the product approximately.

6.4.2 Performance of Randomized Matrix Multiplication

To simplify our presentation, we will assume that both matrices have been scaled so that their spectral norms are equal to one:

$$\|\boldsymbol{B}\| = \|\boldsymbol{C}\| = 1.$$

It is relatively inexpensive to compute the spectral norm of a matrix accurately, so this preprocessing step is reasonable.

Let $\mathsf{asr} = \frac{1}{2}(\mathsf{srank}(\boldsymbol{B}) + \mathsf{srank}(\boldsymbol{C}))$ be the average stable rank of the two factors; see (2.1.25) for the definition of the stable rank. In §6.4.3, we will prove that

$$\mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{BC}\| \le \sqrt{\frac{4 \cdot \mathsf{asr} \cdot \log(d_1 + d_2)}{n}} + \frac{2 \cdot \mathsf{asr} \cdot \log(d_1 + d_2)}{3n}. \tag{6.4.5}$$

To appreciate what this estimate means, suppose that the number n of samples satisfies

$$n \ge \varepsilon^{-2} \cdot \mathsf{asr} \cdot \log(d_1 + d_2)$$

where ε is a positive tolerance. Then we obtain a relative error bound for the randomized matrix multiplication method

$$\frac{\mathbb{E}\,\|\bar{\boldsymbol{R}}_n - \boldsymbol{BC}\|}{\|\boldsymbol{B}\|\,\|\boldsymbol{C}\|} \le 2\varepsilon + \frac{2}{3}\varepsilon^2.$$

This expression depends on the normalization of $\boldsymbol{B}$ and $\boldsymbol{C}$. The computational cost of forming the approximation is

$$O(\varepsilon^{-2} \cdot \mathsf{asr} \cdot d_1 d_2 \log(d_1 + d_2)) \quad \text{arithmetic operations.}$$

In other words, when the average stable rank asr is substantially smaller than the inner dimension N of the two matrices $\boldsymbol{B}$ and $\boldsymbol{C}$, the random estimate $\bar{\boldsymbol{R}}_n$ for the product $\boldsymbol{BC}$ achieves a small error relative to the scale of the factors.

6.4.3 Analysis of Randomized Matrix Multiplication

The randomized matrix multiplication method is just a specific example of empirical approximation, and the error bound (6.4.5) is an immediate consequence of Corollary 6.2.1.

To pursue this approach, we need to establish a uniform bound on the norm of the estimator $\boldsymbol{R}$ for the product. Observe that

$$\|\boldsymbol{R}\| \le \max_j \|p_j^{-1} \boldsymbol{b}_{:j} \boldsymbol{c}_{j:}\| = \max_j \frac{\|\boldsymbol{b}_{:j}\|\,\|\boldsymbol{c}_{j:}\|}{p_j}.$$

To obtain a bound, recall the value (6.4.3) of the probability p_j, and invoke the inequality between geometric and arithmetic means:

$$\|\boldsymbol{R}\| \le \big(\|\boldsymbol{B}\|_{\mathrm{F}}^2 + \|\boldsymbol{C}\|_{\mathrm{F}}^2\big) \cdot \max_j \frac{\|\boldsymbol{b}_{:j}\|\,\|\boldsymbol{c}_{j:}\|}{\|\boldsymbol{b}_{:j}\|^2 + \|\boldsymbol{c}_{j:}\|^2} \le \frac{1}{2}\big(\|\boldsymbol{B}\|_{\mathrm{F}}^2 + \|\boldsymbol{C}\|_{\mathrm{F}}^2\big).$$

Since the matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ have unit spectral norm, we can express this inequality in terms of the average stable rank:

$$\|\boldsymbol{R}\| \le \frac{1}{2}\big(\mathsf{srank}(\boldsymbol{B}) + \mathsf{srank}(\boldsymbol{C})\big) = \mathsf{asr}.$$

This is the exactly kind of bound that we need.

Next, we need an estimate for the per-sample second moment $m_2(\boldsymbol{R})$. By direct calculation,

$$\begin{aligned}\mathbb{E}(\boldsymbol{R}\boldsymbol{R}^*) &= \sum_{j=1}^{N} \frac{1}{p_j^2} \cdot (\boldsymbol{b}_{:j}\boldsymbol{c}_{j:})(\boldsymbol{b}_{:j}\boldsymbol{c}_{j:})^* \cdot p_j \\ &= \left(\|\boldsymbol{B}\|_{\mathrm{F}}^2 + \|\boldsymbol{C}\|_{\mathrm{F}}^2\right) \cdot \sum_{j=1}^{n} \frac{\|\boldsymbol{c}_{j:}\|^2}{\|\boldsymbol{b}_{:j}\|^2 + \|\boldsymbol{c}_{j:}\|^2} \cdot \boldsymbol{b}_{:j}\boldsymbol{b}_{:j}^* \\ &\preccurlyeq \left(\|\boldsymbol{B}\|_{\mathrm{F}}^2 + \|\boldsymbol{C}\|_{\mathrm{F}}^2\right) \cdot \boldsymbol{B}\boldsymbol{B}^*.\end{aligned}$$

The semidefinite relation holds because each fraction lies between zero and one, and each matrix $\boldsymbol{b}_{:j}\boldsymbol{b}_{:j}^*$ is positive semidefinite. Therefore, increasing the fraction to one only increases in the matrix in the semidefinite order. Similarly,

$$\mathbb{E}(\boldsymbol{R}\boldsymbol{R}^*) \preccurlyeq \left(\|\boldsymbol{B}\|_{\mathrm{F}}^2 + \|\boldsymbol{C}\|_{\mathrm{F}}^2\right) \cdot \boldsymbol{C}^*\boldsymbol{C}.$$

In summary,

$$\begin{aligned}m_2(\boldsymbol{R}) &= \max\left\{\|\mathbb{E}(\boldsymbol{R}\boldsymbol{R}^*)\|, \|\mathbb{E}(\boldsymbol{R}^*\boldsymbol{R})\|\right. \\ &\le \left(\|\boldsymbol{B}\|_{\mathrm{F}}^2 + \|\boldsymbol{C}\|_{\mathrm{F}}^2\right) \cdot \max\left\{\|\boldsymbol{B}\boldsymbol{B}^*\|, \|\boldsymbol{C}^*\boldsymbol{C}\|\right\} \\ &= \left(\|\boldsymbol{B}\|_{\mathrm{F}}^2 + \|\boldsymbol{C}\|_{\mathrm{F}}^2\right) \\ &= 2 \cdot \mathsf{asr}.\end{aligned}$$

The penultimate line depends on the identity (2.1.24) and our assumption that both matrices $\boldsymbol{B}$ and $\boldsymbol{C}$ have norm one.

Finally, to reach the stated estimate (6.4.5), we apply Corollary 6.2.1 with the parameters $L = \mathsf{asr}$ and $m_2(\boldsymbol{R}) \le 2 \cdot \mathsf{asr}$.

6.5 Application: Random Features

As a final application of empirical matrix approximation, let us discuss a contemporary idea from machine learning called *random features*. Although this technique may appear more sophisticated than randomized sparsification or randomized matrix multiplication, it depends on exactly the same principles. Random feature maps were proposed by Ali Rahimi and Ben Recht [RR07]. The analysis in this section is due to David Lopez-Paz et al. [LPSS+14].

6.5.1 Kernel Matrices

Let $\mathscr{X}$ be a set. We think about the elements of the set $\mathscr{X}$ as (potential) observations that we would like to use to perform learning and inference tasks. Let us introduce a bounded measure Φ of similarity between pairs of points in the set:

$$\Phi : \mathscr{X} \times \mathscr{X} \to [-1, +1].$$

The similarity measure Φ is often called a *kernel*. We assume that the kernel returns the value $+1$ when its arguments are identical, and it returns smaller values when its arguments are dissimilar. We also assume that the kernel is symmetric; that is, $\Phi(\boldsymbol{x}, \boldsymbol{y}) = \Phi(\boldsymbol{y}, \boldsymbol{x})$ for all arguments $\boldsymbol{x}, \boldsymbol{y} \in \mathscr{X}$.