Memory and Computation Efficient PCA via Very Sparse Random Projections

Farhad Pourkamali-Anaraki FARHAD.POURKAMALI@COLORADO.EDU
Shannon M. Hughes SHANNON.HUGHES@COLORADO.EDU
Department of Electrical, Computer, and Energy Engineering, University of Colorado at Boulder, CO, 80309, USA

Abstract

Algorithms that can efficiently recover principal components in very high-dimensional, streaming, and/or distributed data settings have become an important topic in the literature. In this paper, we propose an approach to principal component estimation that utilizes projections onto very sparse random vectors with Bernoulli-generated nonzero entries. Indeed, our approach is simultaneously efficient in memory/storage space, efficient in computation, and produces accurate PC estimates, while also allowing for rigorous theoretical performance analysis. Moreover, one can tune the sparsity of the random vectors deliberately to achieve a desired point on the trade-offs between memory, computation, and accuracy. We rigorously characterize these trade-offs and provide statistical performance guarantees. In addition to these very sparse random vectors, our analysis also applies to more general random projections. We present experimental results demonstrating that this approach allows for simultaneously achieving a substantial reduction of the computational complexity and memory/storage space, with little loss in accuracy, particularly for very high-dimensional data.

1. Introduction

Principal component analysis (PCA) is a fundamental tool in unsupervised learning and data analysis that finds the low-dimensional linear subspace that minimizes the mean-squared error between the original data and the data projected onto the subspace. The principal components (PCs) can be obtained by a singular value decomposition (SVD) of the data matrix or eigendecomposition of the data's covariance matrix. PCA is frequently used for dimensionality reduction, feature extraction, and as a pre-processing step for learning and recognition tasks such as classification.

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

There is a wealth of existing literature that develops computationally efficient approaches to computing these PCs. However, the overwhelming majority of this literature assumes ready access to the stored full data samples.

However, this full data access is not always possible in modern data settings. Modern data acquisition capabilities have increased massively in recent years, which can lead to a wealth of rapidly changing high-dimensional data. Hence, in very large database environments, it may not be feasible or practical to access all the data in storage (Muthukrishnan, 2005).

Moreover, in applications such as sensor networks, distributed databases, and surveillance, data is typically distributed over many sensors. Accessing all the data at once requires tremendous communication costs between the sensors and a central processing unit. Algorithms that don't require access to all the data can help reduce this communication cost (Balcan et al., 2013). A third case is streaming data, where one must acquire and store the data in real time to have full access, which may not be feasible.

One promising strategy to address these issues in a computationally efficient way, which also allows for rigorous theoretical analysis, is to use very sparse random projections. Random projections provide informative lower-dimensional representations of high-dimensional data, thereby saving memory and computation. They are widely used in many applications, including databases and data stream processing (Li et al., 2006; Indyk, 2006) and compressive sensing (Donoho, 2006).

Initial attempts have been made to perform PCA using only the information embedded in random projections. Unfortunately, however, theoretical guarantees have generally only been given for random vectors with i.i.d. entries drawn from the Gaussian distribution. This common choice is convenient in terms of theoretical analysis, but undesirable in practice. Such dense random vectors require relatively high storage space, and high computation because of the large amount of floating point arithmetic needed to compute each projection.

In this paper, we instead aim to recover PCs from very

sparse random projections with Bernoulli entries. These sparse random projections can be implemented using simple database operations. For example, this type of random projection can be obtained by simply adding two small subsets of the entries of a data sample and then subtracting the results. They thus require little computation or data access. For distributed data, this type of sparse Bernoulli projection could be obtained via localized aggregation in the network requiring minimal communication (assuming all sensors can communicate with one another). (If a network topology must be respected, the sparse random projections could presumably be adjusted accordingly, but we have not yet analyzed this case.) In short, very sparse random projections are or could potentially be extremely practical for a variety of situations.

Our theoretical analysis begins by assuming a probabilistic generative model for the data, related to the spiked covariance model. Under this model, we show that PCs computed from very sparse random projections are close estimators of the true underlying PCs. Moreover, one can adjust the sparsity of the random projections as desired to greatly reduce memory and computation (at the cost of some accuracy). We give rigorous theoretical analysis of the resulting tradeoffs between memory, computation, and accuracy as we vary sparsity, showing that efficiency in memory and computation may be gained with little sacrifice in accuracy. In fact, our analysis will also apply more generally to any random projections with i.i.d. zero mean entries and bounded second-, fourth-, sixth- and eighth-order moments, although we focus on the sparse-Bernoulli case.

In Section 2, we present a brief review of related work. The model assumptions and notation are in Section 3. We present an overview of the main contributions in Section 4. In Section 5, the main results are stated with some discussion of their consequences. Proofs are reserved to the supplementary material. Finally, we present experimental results demonstrating the performance and efficiency of our approach compared with prior work in Section 6.

2. Related Work

Algorithms that can efficiently recover PCs from a collection of *full* data samples have been an important topic in the literature for decades. A comprehensive survey of these algorithms can be found in (Halko et al., 2011b; Gilbert et al., 2012) and the references therein. This includes several lines of work. The first involves techniques that are based on dimensionality reduction, sketching, and sub-sampling for low-rank matrix approximation such as (Halko et al., 2011a). In these methods, the computational complexity is typically reduced by performing SVD on the smaller matrix obtained by sketching or subsampling. However, these methods require accessible storage of all the data samples. This may not be practical for modern data processing applications where data samples are too vast or generated too quickly to be stored accessibly.

The second line of work involves online algorithms specifically tailored to have extremely low-memory complexity such as (Arora et al., 2012) and the references therein. Typically, these algorithms assume that the data is streaming by, that real-time PC estimates are needed, and they obtain these by solving a stochastic optimization problem, in which each arriving data sample is used to update the PCs in an iterative procedure. As a couple recent examples of this line of work, (Mitliagkas et al., 2013) show that a blockwise stochastic variant of the power method can recover PCs in this low-memory setting from $O(p \log p)$ samples, although the computational cost is not examined. Meanwhile, (Arora et al., 2013) bound the generalization error of PCs learned with their algorithm to new data samples and also analyze its computational cost.

Our problem lies somewhere between the above two lines of work. We don't assume that memory/data access is not a concern, but at the same time, we also don't assume the extremely restrictive setting where one-sample-at-a-time real-time PC updates are required. Instead, we aim to reduce both memory and computation simultaneously for PCA across a broad class of big data settings, e.g. for enormous databases where loading into local memory may be difficult or costly, for streaming data when PC estimates do not have to be real-time, or for distributed data. We also aim to provide tunable tradeoffs for the amount of accuracy that will be sacrificed for each given reduction in memory/computation, in order to aid in choosing a desired balance point between these.

To do this, we recover PCs from random projections. There have been some related prior attempts to extract PCs from random projections of data (Fowler, 2009; Qi and Hughes, 2012). In both, the problem of recovering PCs from random projections has been considered only for dense Gaussian random projections. However, dense vectors are undesirable for practical applications since they require relatively high storage space and computation (including lots of floating point arithmetic) as noted in the introduction. Our work will make use of sparse random vectors with Bernoulli entries which will be more efficiently implementable in a large database environment.

Chen et al. (2013) have estimated the covariance matrix of data from general sub-Gaussian random projections to reduce memory use. However, convergence guarantees are given only for the case of infinite data samples, making it hard to realistically use these results in memory/computation vs. accuracy tradeoffs, and computational cost is not examined. We will address both these issues.

As a final note, we observe that our work also can be

viewed as an example of emerging ideas in computational statistics (see (Chandrasekaran and Jordan, 2013)) in which tradeoffs between computational complexity, dataset size, and estimation accuracy are explicitly characterized, so that a user may choose to reduce computation in very high-dimensional data settings with knowledge of the risk to the accuracy of the result.

3. Problem Formulation and Notation

In this paper, we focus on a statistical model for the data that is applicable to various scenarios. Assume that our original data in $\mathbb{R}^p$ are centered at $\overline{\mathbf{x}} \in \mathbb{R}^p$ and $\{\mathbf{v}_i\}_{i=1}^d \in \mathbb{R}^p$ are the d orthonormal PCs. We consider the following probabilistic generative model for the data samples, $\mathbf{x}_i = \overline{\mathbf{x}} + \sum_{j=1}^d w_{ij}\sigma_j \mathbf{v}_j + \mathbf{z}_i$, $i=1,\ldots,n$, where $\{\mathbf{w}_i\}_{i=1}^n$ and $\{\mathbf{z}_i\}_{i=1}^n$ are drawn i.i.d. from $\mathcal{N}(\mathbf{0}, \mathbf{I}_{d\times d})$ and $\mathcal{N}(\mathbf{0}, \frac{\epsilon^2}{p}\mathbf{I}_{p\times p})$, respectively. Also, $\{\sigma_i\}_{i=1}^d$ are scalar constants reflecting the energy of the data in each principal direction such that $\sigma_1 > \sigma_2 > \ldots > \sigma_d > 0$. The additive noise term $\mathbf{z}_i$ allows for some error in our assumptions. Note that the underlying covariance matrix of the data is $\mathbf{C}_{true} \triangleq \sum_{j=1}^d \sigma_j^2 \mathbf{v}_j \mathbf{v}_j^T$, and the signal-to-noise ratio is SNR$=\frac{h}{\epsilon^2}$, where $h \triangleq \sum_{j=1}^d \sigma_j^2$. In fact, this model is related to the spiked covariance model (Johnstone, 2001) in which the data's covariance matrix is assumed to be a low-rank perturbation of the identity matrix.

We then introduce a very general class of random projections. Assume that matrices $\{\mathbf{R}_i\}_{i=1}^n \in \mathbb{R}^{p\times m}$, $m<p$, are formed by drawing each of their i.i.d. entries from a distribution whose mean μ_1 is assumed to be zero and whose k^{th} order moments, μ_k, are assumed finite for $k = 2, 4, 6, 8$. In particular, we will be interested in a popular class of sparse random projections, but our analysis will apply to any distribution satisfying these assumptions.

Each random projection $\mathbf{y}_i \in \mathbb{R}^m$ is then obtained by taking inner products of the data sample $\mathbf{x}_i \in \mathbb{R}^p$ with the random vectors comprising the columns of $\mathbf{R}_i$, i.e. $\mathbf{y}_i = \mathbf{R}_i^T \mathbf{x}_i$. The main goal of this paper is to provide theoretical guarantees for estimating the center and PCs of $\{\mathbf{x}_i\}_{i=1}^n$ from these random projections.

4. Our Contributions

In this paper, we introduce two estimators for the center and underlying covariance matrix of data $\{\mathbf{x}_i\}_{i=1}^n$ from sparse random projections $\{\mathbf{y}_i = \mathbf{R}_i^T \mathbf{x}_i\}_{i=1}^n$. In typical PCA, the center is estimated using the empirical center $\overline{\mathbf{x}}_{emp} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$. PCs are then obtained by eigendecomposition of the empirical covariance matrix $\mathbf{C}_{emp} = \frac{1}{n}\sum_{i=1}^n (\mathbf{x}_i - \overline{\mathbf{x}}_{emp})(\mathbf{x}_i - \overline{\mathbf{x}}_{emp})^T$, that typically comes close to the true covariance matrix (Vershynin, 2012).

Similar to typical PCA, we show that the empirical center and empirical covariance matrix of the new data samples $\{\mathbf{R}_i \mathbf{y}_i\}_{i=1}^n$ (scaled by a known factor) result in accurate estimates of the original center $\overline{\mathbf{x}}$, and the true underlying covariance matrix $\mathbf{C}_{true}$. (Note that $\mathbf{R}_i \mathbf{y}_i$ approximately represents a projection in $\mathbb{R}^p$ of $\mathbf{x}_i$ onto the column space of $\mathbf{R}_i$, but we have eliminated a computationally expensive matrix inverse here.) We will provide rigorous theoretical analysis for the performance of these estimators in terms of parameters such as the measurement ratio m/p, number of samples n, SNR, and moments μ_k.

Our approach is quite general and we believe it can eventually be applicable to various data processing applications in which the data is very high-dimensional, streaming, or distributed. Particularly for the case of distributed data, we may need to adjust the set-up to ensure the random projections respect network topology, but we believe it could be done following the strategies in (Wang et al., 2012a;b).

We will be particularly interested in applying our general distribution results to the case of very sparse measurement matrices. Achlioptas (2001) first showed that, in the classic Johnson–Lindenstrauss result on pairwise distance preservation, the dense Gaussian projection matrices can be replaced with sparse projection matrices, where each entry is distributed on $\{-1, 0, 1\}$ with probabilities $\{\frac{1}{6}, \frac{2}{3}, \frac{1}{6}\}$, achieving a three-fold speedup in processing time. Li et al. (2006) then drew each entry from $\{-1, 0, 1\}$ with probabilities $\{\frac{1}{2s}, 1 - \frac{1}{s}, \frac{1}{2s}\}$, achieving a more significant s-fold speedup in processing time. In this paper, we refer to this second distribution as a *sparse-Bernoulli* distribution with sparsity parameter s. Sparse random projections have been applied in many other applications to substantially reduce computational complexity and memory requirements (Omidiran and Wainwright, 2010; Zhang et al., 2012).

Motivated by the success of these methods, we propose to recover PCs from sparse random projections of the data, in which each entry of $\{\mathbf{R}_i\}_{i=1}^n$ is drawn i.i.d. from the sparse-Bernoulli distribution. In this case, each column of $\{\mathbf{R}_i\}_{i=1}^n$ has $\frac{p}{s}$ nonzero entries, on average. This choice has the following properties simultaneously:

- The computation cost for obtaining each projection is $O(\frac{mp}{s})$ and thus the cost to acquire/access/hold in memory the data needed for the algorithm is $O(\frac{mpn}{s})$. Specifically, we are interested in choosing m and s so that the *compression factor* $\gamma \triangleq \frac{m}{s} < 1$. In this case, our framework requires significantly less computation cost and storage space. First, the computation cost to acquire/access each data sample is $O(\gamma p)$, $\gamma<1$, in contrast to the cost for acquiring each original data sample $O(p)$. This results in a substantial cost reduction for the sensing process, e.g. for streaming data. Second, once acquired, observe that the projected data samples $\{\mathbf{R}_i \mathbf{y}_i\}_{i=1}^n \in \mathbb{R}^p$ will be sparse, having at most $O(\gamma p)$ nonzero entries each. This results in a significant reduction, $O(\gamma pn)$ as opposed to

$O(pn)$, in memory/storage requirements and/or communication cost, e.g. transferring distributed data to a central processing unit.

- Given the sparse data matrix formed by $\{\mathbf{R}_i\mathbf{y}_i\}_{i=1}^n$, one can make use of efficient algorithms for performing (partial) SVD on very large sparse matrices, such as the Lanczos algorithm (Golub and Van Loan, 2012) and svds in MATLAB. In general, for a $p \times n$ matrix, the computational cost of SVD is $O(p^2n)$. However, for large sparse matrices such as ours, the cost can be reduced to $O(\gamma p^2 n)$ (Lin and Gunopulos, 2003).

In the remainder of this paper, we will characterize the accuracy of the estimated center and PCs in terms of m, p, n, SNR, moments of the distribution (which for sparse-Bernoulli will scale with s), etc. As we will see, under certain conditions on the PCs, we may choose γ as low as $\gamma \propto \frac{1}{p}$ for constant accuracy. Hence, assuming $n = O(p)$ samples, the memory/storage requirements for our approach can scale with p in contrast to p^2 for standard algorithms that store the full data, and a similar factor of p savings in computation can be achieved compared with regular SVD. Less aggressive savings will also be available for other PC types.

5. Main Results

We present the main results of our work in this section, with all proofs delayed to the supplemental material. Interestingly, we will see that the shape of the distribution for each entry of $\{\mathbf{R}_i\}_{i=1}^n$ plays an important role in our results. The kurtosis, defined as $\kappa \triangleq \frac{\mu_4}{\mu_2^2} - 3$, is a measure of peakedness and heaviness of tail for a distribution. It can also be thought of as a measure of non-Gaussianity, since the kurtosis of the Gaussian distribution is zero. It turns out that the distribution's kurtosis is a key factor in determining PC estimation accuracy. For sparse-Bernoulli, the kurtosis increases with increasing sparsity parameter s.

5.1. Mean and Variance of Center Estimator

Theorem 1. *Assume that $\{\mathbf{R}_i\}_{i=1}^n$, $\{\mathbf{x}_i\}_{i=1}^n$, $\{\mathbf{y}_i\}_{i=1}^n$, m, n, and μ_2 are as defined in Section 3, and define the n-sample center estimator $\widehat{\overline{\mathbf{x}}}_n = \frac{1}{m\mu_2}\frac{1}{n}\sum_{i=1}^n \mathbf{R}_i\mathbf{y}_i$. Then, the mean of the estimator $\widehat{\overline{\mathbf{x}}}_n$ is the true center of the original data $\overline{\mathbf{x}}$, i.e. $\mathbb{E}[\widehat{\overline{\mathbf{x}}}_n] = \overline{\mathbf{x}}$, for all n, including the base case $n=1$. Furthermore, as $n \to \infty$, the estimator $\widehat{\overline{\mathbf{x}}}_n$ converges to the true center: $\lim_{n\to\infty} \widehat{\overline{\mathbf{x}}}_n = \overline{\mathbf{x}}$.*

We see that the empirical center of $\{\mathbf{R}_i\mathbf{y}_i\}_{i=1}^n$ is a (scaled) unbiased estimator for the true center $\overline{\mathbf{x}}$. Note that this theorem does not depend on the number of projections m or sparsity parameter s, and thus does not depend on γ, as a sufficiently high number of samples will compensate for unfavorable values of these parameters. We further note that, when $n \to \infty$, there is no difference between the Gaussian, very sparse, or other choices of random projections. This is consistent with the observation that random projection matrices consisting of i.i.d. entries must only be zero mean to preserve pairwise distances in the Johnson-Lindenstrauss theorem (Li et al., 2006).

Theorem 2. *Assume that $\{\mathbf{R}_i\}_{i=1}^n$, $\{\mathbf{x}_i\}_{i=1}^n$, $\{\mathbf{y}_i\}_{i=1}^n$, m, n, p, μ_2, h, and SNR are as defined in Section 3, and kurtosis κ is as defined above. Then, the variance of the unbiased center estimator $\widehat{\overline{\mathbf{x}}}_n = \frac{1}{m\mu_2}\frac{1}{n}\sum_{i=1}^n \mathbf{R}_i\mathbf{y}_i$ is*

$$Var\left(\widehat{\overline{\mathbf{x}}}_n\right) = \frac{1}{n\frac{m}{p}}\left(h\left(1+\frac{1}{SNR}\right)\left(1+\frac{m}{p}+\frac{\kappa+1}{p}\right) + \left(1+\frac{\kappa+1}{p}\right)\|\overline{\mathbf{x}}\|_2^2\right). \quad (5.1)$$

We see that as the number of samples n and measurement ratio m/p increase, the variance of this estimator decreases at rate $\frac{1}{n}$ and close to $\frac{1}{m/p}$. Interestingly, the power of the signal, i.e. $h = \sum_{j=1}^d \sigma_j^2$, works against the accuracy of the estimator. The intuition for this is that, for the center estimation problem, it is desirable to have all the data samples close to the center, which happens for small h. For sparse random projections, we observe that the kurtosis is $\kappa = s - 3$ and thus $\frac{\kappa+1}{p} \approx \frac{s}{p}$. Hence, variance scales with increasing sparsity, although sufficient data samples n are enough to combat this effect. Indeed, when $s > p$, the variance increases heavily since many of the random vectors are zero, and thus the corresponding projections cannot capture any information about the original data. Overall, this result shows an explicit tradeoff between reducing n or increasing s to reduce memory/computation and the variance of the resulting estimator. Finally, given this mean and variance, probabilistic error bounds can be immediately obtained via Chebyshev, Bernstein, etc. inequalities.

5.2. Mean and Variance of Covariance Estimator

Theorem 3. *Assume that $\{\mathbf{R}_i\}_{i=1}^n$, $\{\mathbf{x}_i\}_{i=1}^n$, $\{\mathbf{y}_i\}_{i=1}^n$, m, n, p, μ_2, h, ϵ, and $\mathbf{C}_{true}$ are as defined in Section 3, and κ is the kurtosis. Moreover, assume that $\{\mathbf{x}_i\}_{i=1}^n$ are centered at $\overline{\mathbf{x}} = \mathbf{0}$. Define the n-sample covariance estimator $\widehat{\mathbf{C}}_n = \frac{1}{(m^2+m)\mu_2^2}\frac{1}{n}\sum_{i=1}^n \mathbf{R}_i\mathbf{y}_i\mathbf{y}_i^T\mathbf{R}_i^T$. Then, for all n, the mean of this estimator is: $\mathbb{E}[\widehat{\mathbf{C}}_n] = \widehat{\mathbf{C}}_{true} + \mathbf{E}$, where $\widehat{\mathbf{C}}_{true} \triangleq \mathbf{C}_{true} + \alpha\mathbf{I}_{p\times p}$, $\alpha \triangleq \frac{h}{m+1} + (\frac{\kappa}{p(m+1)} + \frac{(m+p+1)}{p(m+1)})\epsilon^2$, and $\mathbf{E} \triangleq \frac{\kappa}{m+1}\sum_{j=1}^d \sigma_j^2 diag(\mathbf{v}_j\mathbf{v}_j^T)$, where $diag(\mathbf{A})$ denotes the matrix formed by zeroing all but the diagonal entries of $\mathbf{A}$. Furthermore, let $\mathbf{C}_\infty \triangleq \widehat{\mathbf{C}}_{true} + \mathbf{E}$. Then, as $n \to \infty$, the estimator $\widehat{\mathbf{C}}_n$ converges to $\mathbf{C}_\infty$: $\lim_{n\to\infty} \widehat{\mathbf{C}}_n = \mathbf{C}_\infty$.*

We observe that the limit of the estimator $\widehat{\mathbf{C}}_n$ has two components. The first, $\widehat{\mathbf{C}}_{true}$, has the same eigenvectors with slightly perturbed eigenvalues (α tends to be very small in high dimensions) and the other, $\mathbf{E}$, is an error perturbation term. Both α and $\mathbf{E}$ scale with the kurtosis, reflecting the necessary tradeoff between increasing sparsity (decreasing memory/computation) and maintaining accuracy.

We first consider a simple example to gain some intuition for this theorem. A set of data samples $\{\mathbf{x}_i\}_{i=1}^{3000} \in \mathbb{R}^{1000}$

are generated from one PC. We also generate the measurement matrices $\{\mathbf{R}_i\}_{i=1}^{3000} \in \mathbb{R}^{1000\times 200}$ ($m/p = 0.2$) with i.i.d. entries both for the Gaussian distribution and the sparse-Bernoulli distribution for various values of the sparsity parameter s. In Fig. 5.1, we view two dimensions (the original PC's and one other) of the data $\{\mathbf{x}_i\}_{i=1}^{3000}$ and the scaled projected data $\frac{1}{\sqrt{(m^2+m)\mu_2^2}}\{\mathbf{R}_i\mathbf{y}_i\}_{i=1}^{3000}$, represented by blue dots and red circles respectively. We see that the projected data samples are scattered somewhat into other directions for all four cases. However, the amount of scattered energy for the Gaussian and sparse-Bernoulli for s=3 is quite small. This can be easily verified from the fact that the amount of perturbation depends on the *kurtosis*, and for both cases the kurtosis is κ=0. As we increase the parameter s, the kurtosis κ=$s-3$ gets larger, and this is consistent with the observation that the projected data samples get more scattered into other directions. We also note the similarity of our findings to (Li et al., 2006)'s result that the variance of the pairwise distances in Johnson–Lindenstrauss depends on the kurtosis of the distribution being used for random projections. Despite the perturbation, in all cases, the PC can be recovered accurately. Note also that scaling the projected data points by $1/\sqrt{(m^2+m)\,\mu_2^2}$ preserves the energy in the direction of the PC (i.e. the eigenvalue).

In Theorem 3, we see that $\mathbf{C}_{true}$ and $\widehat{\mathbf{C}}_{true}$ have the same set of eigenvectors with the eigenvalues of $\mathbf{C}_{true}$ increased by α=$h\{\frac{1}{m+1}+(\frac{\kappa}{p(m+1)}+\frac{1}{p}+\frac{1}{m+1})\frac{1}{\text{SNR}}\}$. Thus, α is a decreasing function of p, m/p and SNR, and in particular goes to 0 as $p\to\infty$ for constant projection ratio m/p. This is illustrated in Fig. 5.2. Thus, surprisingly, in the high-dimensional regime, the amount of perturbation of eigenvalues becomes increasingly negligible even for small measurement ratios.

Now, let's examine the error matrix $\mathbf{E}$. We observe that $\mathbf{E}$ can be viewed as representing a bias of the estimated PCs towards the nearest canonical basis vectors; it stems from anisotropy in the distribution for $\mathbf{R}_i$ when this is non-Gaussian (note $\kappa=0$, and thus $\mathbf{E}=\mathbf{0}$, for the Gaussian case). In later sections, we will use the 2-norm of $\mathbf{E}$, $\|\mathbf{E}\|_2$, to bound the angle between the estimated and true PCs. Indeed, we find, for constant $\delta \triangleq \frac{\|\mathbf{E}\|_2}{h}$, the same angular PC estimation error is achieved. We now study $\|\mathbf{E}\|_2$, leading to useful observations, for several types of PCs. (An expanded discussion with full derivations is included in the supplementary materials.)

(1) Smooth PCs: It has frequently been observed that sparse-Bernoulli random projections are most effective on vectors that are "smooth" (Ailon and Chazelle, 2009), meaning that their maximum entry is of size $O(\frac{1}{\sqrt{p}})$. Large images, videos, and other natural signals with distributed energy are obvious examples of this type. (Other signals are often preconditioned to be smooth via multiplication with a Hadamard conditioning matrix.) We may easily observe then that $\|\mathbf{E}\|_2 \le \frac{\kappa}{m+1}\mu_{max}^2 h$, or $\delta \le \frac{\kappa}{m+1}\mu_{max}^2$, where μ_{max} is the mutual coherence (Elad, 2007) between the PCs and the canonical basis, and we note $\frac{\kappa}{m+1} \le \frac{1}{\gamma}$. As we will see in Section 5.3, we will want to keep δ small enough to guarantee a certain fixed angular error θ_0. In fact, this can be satisfied by requiring $\gamma \ge C(\theta_0)\mu_{max}^2$, where $C(\theta_0)$ is a constant depending on the error θ_0. Hence, for smooth PCs, we need only have $\gamma \propto \frac{1}{p}$, reducing memory and computation by a rather remarkable factor of p.

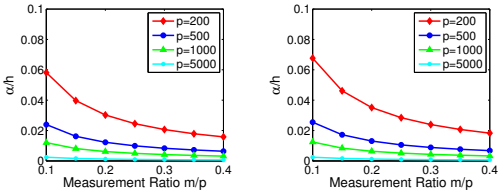


Figure 5.2. Variation of the parameter $\frac{\alpha}{h}$ for (a) $\kappa=0$ and (b) $\kappa=200$, varying p and measurement ratio m/p, and fixed SNR $=5$.

(2) All Sparse PCs: In the case of all sparse PCs, we may write $\mathbf{E}$ as $\mathbf{E}=\frac{\kappa}{m+1}\mathbf{C}_{true}+\mathbf{E}_0$ where $\|\mathbf{E}_0\|_2 \le \frac{\kappa}{m+1}\sqrt{1-\mu_{min}^4}\,h$ and $\mu_{min} \triangleq \min_{1\le i\le d}\max_{1\le j\le p}|\langle \mathbf{v}_i, \mathbf{e}_j\rangle|$ represents the closeness of the PCs to the canonical basis $\{\mathbf{e}_j\}_{j=1}^p$. Thus, unlike for other sparse-Bernoulli applications, we find that sparse PCs can still be recovered very well here, although the eigenvalues may be heavily scaled by the known factor $1+\frac{\kappa}{m+1}$. Doing this, and taking $\mathbf{E}_0$ as the resulting error term, we can let $\gamma \propto \sqrt{1-\mu_{min}^4}$ to maintain constant δ.

(3) Neither Sparse nor Smooth PCs: In this case, we can still apply the analysis for case (1), just with a larger μ_{max}^2 and less aggressive memory/computation savings.

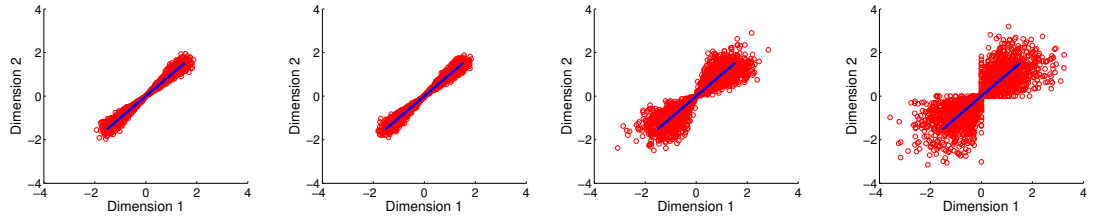
(4) Mixture of PC Types: In this case, we may split $\mathbf{E}$ into two error matrices, associated with each of the sparse and non-sparse PCs. Recovery of the d-dimensional PC subspace still performs well here. However, if the eigenvalues $\{\sigma_j^2\}_{j=1}^d$ do not decay sufficiently fast, scaling of the eigenvalues for the sparse PCs may reorder the individual components. Please see the supplementary material for further discussions and simulations.

Theorem 4. *Assume that* $\{\mathbf{R}_i\}_{i=1}^n$, $\{\mathbf{x}_i\}_{i=1}^n$, $\{\mathbf{y}_i\}_{i=1}^n$, m, n, p, μ_k, h, *and SNR are as defined in Section 3. Consider the covariance matrix estimator* $\widehat{\mathbf{C}}_n = \frac{1}{n}\sum_{i=1}^n \frac{1}{m(m+1)\mu_2^2}\mathbf{R}_i\mathbf{y}_i\mathbf{y}_i^T\mathbf{R}_i^T$. *Then, the deviation of our* n*-sample estimator from its mean value is upper bounded:*

$$\mathbb{E}\left[\left\|\widehat{\mathbf{C}}_n - \mathbf{C}_\infty\right\|_F^2\right] \le \frac{1}{n}\left(\tau_1-\tau_2\right)h^2 \qquad (5.2)$$

where $\tau_1 \triangleq \xi\left\{\left(1+\frac{1}{SNR}\right)^2 + 2\left(\frac{\tilde{h}}{h^2}+\frac{2}{p}\frac{1}{SNR}+\frac{1}{p}\left(\frac{1}{SNR}\right)^2\right)\right\}$,

$\tau_2 \triangleq \frac{\tilde{h}}{h^2}\left(\frac{p-1}{p}+\frac{1}{p}\left(1+\frac{\kappa}{m+1}\right)^2\right)+\beta\left(2+\beta p+2\frac{\kappa}{m+1}\right) \ge 0$,



(a) Gaussian (b) $s = 3$ (c) $s = 20$ (d) $s = 50$

Figure 5.1. Accurate recovery of the PC under random projections using both Gaussian and sparse random projection matrices for various values of s. In each figure, there are n=3000 data samples uniformly distributed on a line in $\mathbb{R}^{1000}$. $\{\mathbf{R}_i\}_{i=1}^n \in \mathbb{R}^{p\times m}$, m/p=0.2, are generated with i.i.d. entries drawn from (a) $\mathcal{N}(0,1)$ and (b,c,d) the sparse-Bernoulli distribution for s=3, 20, 50. In each figure, we view two dimensions (the original PC's and one other) of the data $\{\mathbf{x}_i\}_{i=1}^n$ (blue dots) and the scaled projected data $1/\sqrt{(m^2+m)\,\mu_2^2}\{\mathbf{R}_i\mathbf{R}_i^T\mathbf{x}_i\}_{i=1}^n$ (red circles). We observe that, in all cases, the projected data samples are symmetrically distributed around the PC, and the inner product magnitude between the PC estimated from the projected data and the true PC is at least 0.998.

$\widetilde{h} \triangleq \sum_{j=1}^d \sigma_j^4$, $\beta = \frac{\alpha}{h}$ *and* $\xi = \max(\xi_1, \xi_2)$, *where*

$$\xi_1 \leq \frac{\mu_8/\mu_2^4}{m^3} + \frac{\mu_6/\mu_2^3}{m^2}\left(4 + \frac{2}{m/p}\right) + \frac{(\mu_4/\mu_2^2)^2}{m^2}\left(3 + \frac{1}{m/p}\right) + \frac{\mu_4/\mu_2^2}{m}\left(6 + \frac{6}{m/p} + \frac{1}{(m/p)^2}\right) + \left(1 + \frac{1}{m/p}\right)^2 + \frac{3}{p\,(m/p)^2}$$

and

$$\xi_2 \leq \frac{\mu_6/\mu_2^3}{m^2}\left(\frac{6}{pm/p}\right) + \frac{(\mu_4/\mu_2^2)^2}{m^2}\left(1 + \frac{5}{pm/p}\right) + \frac{\mu_4/\mu_2^2}{m}\left(2 + \frac{26}{pm/p} + \frac{2}{m/p} + \frac{13}{p\,(m/p)^2}\right) + \left(1 + \frac{1}{m/p}\right)^2 + \frac{10}{pm/p} + \frac{7}{p^2\,(m/p)^2} + \frac{13}{p\,(m/p)^2} + \frac{2}{p\,(m/p)^3}.$$

Note that ξ has various terms that scale with $\frac{1}{p}$, $\frac{1}{m/p}$, and the higher order moments μ_8/μ_2^4, μ_6/μ_2^3, and μ_4/μ_2^2.

We see that as the number of data samples n increases, the variance decreases at rate $\frac{1}{n}$, converging quickly to the limit. Moreover, the variance of our estimator is a decreasing function of the measurement ratio m/p and SNR. We further note that the parameter ξ gives us important information about the effect of the tails of the distribution on the convergence rate of the covariance estimator. More precisely, for sparse random projections, we see that $\frac{\mu_8/\mu_2^4}{m^3} = (\frac{s}{m})^3 = \frac{1}{\gamma^3}$, $\frac{\mu_6/\mu_2^3}{m^2} = \frac{(\mu_4/\mu_2^2)^2}{m^2} = \frac{1}{\gamma^2}$, and $\frac{\mu_4/\mu_2^2}{m} = \frac{1}{\gamma}$. Hence, for a fixed number of data samples, decreasing the compression factor γ leads to an increase of the variance and a loss in accuracy, as we will see in Section 6. This is as we would expect since there is an inherent tradeoff between saving computation and memory and the accuracy. However, characterizing this tradeoff allows γ to be chosen in an informed way for large datasets.

5.3. Memory, Computation and PC Accuracy Tradeoffs

We now use the covariance matrix estimator results to bound the error of its eigenvalues and eigenvectors, using related results from matrix perturbation theory.

First, note that using the variance of our estimator (Eq. 5.2) in the Chebyshev inequality yields $\left\|\widehat{\mathbf{C}}_n - \mathbf{C}_\infty\right\|_F \leq \varepsilon$, with probability at least $1 - \frac{1}{n\varepsilon^2}(\tau_1 - \tau_2)h^2$. Hence,

$$\left\|\widehat{\mathbf{C}}_n - \widehat{\mathbf{C}}_{true}\right\|_2 \leq \left\|\widehat{\mathbf{C}}_n - \mathbf{C}_\infty\right\|_2 + \left\|\mathbf{C}_\infty - \widehat{\mathbf{C}}_{true}\right\|_2 \leq \left\|\widehat{\mathbf{C}}_n - \mathbf{C}_\infty\right\|_F + \|\mathbf{E}\|_2 \leq \|\mathbf{E}\|_2 + \varepsilon \tag{5.3}$$

with probability at least $1 - \frac{1}{n\varepsilon^2}(\tau_1 - \tau_2)\,h^2$. In fact, Eq. 5.3 can be used to characterize tradeoffs between memory, computation, and PC estimation accuracy (as an angle between estimated subspaces) in terms of our parameters n, m/p, etc. For simplicity in what follows and to help keep the intuition clear, we focus on the case where the number of samples $n \to \infty$ and $\varepsilon \to 0$ in Eq. 5.3 above. However, it is trivial to adjust these results to the case of finite n by including a nonzero ε in the derivations that follow.

For illustrative purposes, we start by analyzing the case of a single PC and use the following Lemma. In the following, $\lambda(\mathbf{A})$ and $\lambda_i(\mathbf{A})$ denote the set of all eigenvalues and the i^{th} eigenvalue of $\mathbf{A}$, respectively.

Lemma 5. *(Hogben, 2006; Davis and Kahan, 1970) Suppose* $\mathbf{A}$ *is a real symmetric matrix and* $\widetilde{\mathbf{A}} = \mathbf{A} + \mathbf{E}$ *is the perturbed matrix. Assume that* $(\widetilde{\lambda}, \widetilde{\mathbf{v}})$ *is an exact eigenpair of* $\widetilde{\mathbf{A}}$ *where* $\|\widetilde{\mathbf{v}}\|_2 = 1$. *Then*

(a) $\left|\widetilde{\lambda} - \lambda\right| \leq \|\mathbf{E}\|_2$ *for some eigenvalue* λ *of* $\mathbf{A}$.

(b) Let λ *be the closest eigenvalue of* $\mathbf{A}$ *to* $\widetilde{\lambda}$ *and* $\mathbf{v}$ *be its associated eigenvector with* $\|\mathbf{v}\|_2 = 1$, *and let* $\eta = \min_{\lambda_0 \in \lambda(\mathbf{A}), \lambda_0 \neq \lambda} \left|\widetilde{\lambda} - \lambda_0\right|$. *If* $\eta > 0$, *then*

$$\sin \angle(\widetilde{\mathbf{v}}, \mathbf{v}) \leq \frac{\|\mathbf{E}\|_2}{\eta} \tag{5.4}$$

where $\angle(\widetilde{\mathbf{v}}, \mathbf{v})$ *denotes the canonical angle between the two eigenvectors.*

We will use this Lemma to bound the angle between the PC estimate from $\widehat{\mathbf{C}}_n$ and the true PC in the single PC

case. Since $\mathbf{C}_{true}$ has only one eigenpair $(\sigma^2, \mathbf{v})$ with nonzero eigenvalue, $\widehat{\mathbf{C}}_{true}$ has an eigenpair $(\sigma^2+\alpha, \mathbf{v})$ and $\lambda_i(\widehat{\mathbf{C}}_{true})=\alpha$, $i=2,\ldots,p$. From Lemma 5, we see that the largest eigenvalue of $\widehat{\mathbf{C}}_n$ satisfies $\left|\lambda_1(\widehat{\mathbf{C}}_n)-(\sigma^2+\alpha)\right| \leq \|\mathbf{E}\|_2=\delta\sigma^2$. We find the parameter η:

$$\begin{aligned}\eta &= \min_{i=2,\ldots,p}\left|\lambda_1\left(\widehat{\mathbf{C}}_n\right)-\lambda_i\left(\widehat{\mathbf{C}}_{true}\right)\right| = \left|\lambda_1\left(\widehat{\mathbf{C}}_n\right)-\alpha\right| \\ &\geq \sigma^2 - \|\mathbf{E}\|_2 = (1-\delta)\,\sigma^2. \end{aligned} \tag{5.5}$$

We then get the following tradeoff between the accuracy of the estimated eigenvector and the parameters of our model:

$$\sin\angle(\widetilde{\mathbf{v}}, \mathbf{v}) \leq \frac{\delta}{1-\delta}. \tag{5.6}$$

This equation allows us to characterize the statistical tradeoff between the sparsity parameter s and the accuracy of the estimated PC. Observe that this is the same $\delta=\frac{\|\mathbf{E}\|_2}{h}$ that we discussed in Section 5.2. To ensure fixed maximum angular error for PC estimation, i.e. $\sin\angle(\widetilde{\mathbf{v}}, \mathbf{v}) \leq \sin\theta_0$, we should choose γ such that $\delta \leq \frac{\sin\theta_0}{1+\sin\theta_0}$. For smooth PCs, we may satisfy this by choosing $\gamma \geq C(\theta_0)\mu_{max}^2$ for $C(\theta_0) \triangleq \frac{1+\sin\theta_0}{\sin\theta_0}$, which gives $\gamma \geq O(\frac{1}{p})$. Hence, the memory/storage requirements of our method can scale with p in contrast to standard algorithms that scale with p^2, while the computational complexity of SVD can scale with p^2 as opposed to p^3. Although the smooth case is of special interest, less aggressive, but still substantial, savings are also available for other PC types.

For the general case of d PCs, we consider the eigendecomposition of the perturbed matrix $\widehat{\mathbf{C}}_n$ and $\widehat{\mathbf{C}}_{true}$:

$$\widehat{\mathbf{C}}_{true} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{bmatrix}$$

$$\widehat{\mathbf{C}}_n = \begin{bmatrix} \widetilde{\mathbf{V}}_1 & \widetilde{\mathbf{V}}_2 \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{S}}_1 & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{S}}_2 \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{V}}_1^T \\ \widetilde{\mathbf{V}}_2^T \end{bmatrix}.$$

The distance between each perturbed eigenvalue and the corresponding original eigenvalue depends on the amount of perturbation. We now have that $\left|\lambda_j(\widehat{\mathbf{C}}_n) - \lambda_j(\widehat{\mathbf{C}}_{true})\right| \leq \|\mathbf{E}\|_2=\delta h$ for all $j=1,\ldots,d$.

Moreover, it is possible to quantify the rotation of eigenvectors using the notion of *canonical angle matrix* defined in (Davis and Kahan, 1970). Note that $\mathbf{V}_1, \widetilde{\mathbf{V}}_1 \in \mathbb{R}^{p\times d}$ are the first (true and estimated) PCs. The canonical angles between them are defined as $\theta_i=\arccos\rho_i$, where $\{\rho_i\}_{i=1}^d$ are the singular values of $(\widetilde{\mathbf{V}}_1^T\widetilde{\mathbf{V}}_1)^{-1/2}\widetilde{\mathbf{V}}_1^T\mathbf{V}_1(\mathbf{V}_1^T\mathbf{V}_1)^{-1/2}$, in our case, just $\widetilde{\mathbf{V}}_1^T\mathbf{V}_1$. The canonical angle matrix is then defined as $\Theta(\widetilde{\mathbf{V}}_1, \mathbf{V}_1)=\mathrm{diag}(\theta_1,\ldots,\theta_d)$. Based on the results given in (Davis and Kahan, 1970; Gilbert et al., 2012):

$$\left\|\sin\Theta(\widetilde{\mathbf{V}}_1, \mathbf{V}_1)\right\|_2 \leq \frac{\|\mathbf{E}\|_2}{\eta}$$

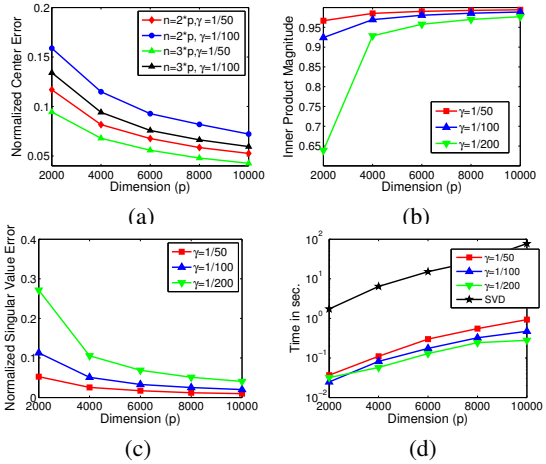


Figure 6.1. Results for synthetic data: (a) normalized estimation error for the center for varying n and γ, (b) magnitude of the inner product between the estimated and true PC for varying γ, (c) normalized estimation error for σ for varying γ, and (d) computation time to perform the SVD for the original vs. randomly projected data for varying γ.

where $\eta \triangleq \min_{1\leq i\leq d, 1\leq j\leq p-d}\left|(\mathbf{S}_1)_{ii} - \left(\widetilde{\mathbf{S}_2}\right)_{jj}\right| > 0$. Using the same logic as in 5.5, we find $\eta \geq \sigma_d^2 - \delta h$. Hence, choosing s, m, etc. such that δ satisfies $\delta < \frac{\sigma_d^2}{h}$, the maximum canonical angle between $\widetilde{\mathbf{V}}_1$ and $\mathbf{V}_1$ satisfies

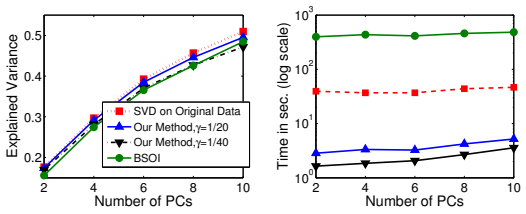
$$\sin\theta_i \leq \frac{\delta}{\frac{\sigma_d^2}{h}-\delta}, \quad i=1,\ldots,d. \tag{5.7}$$

This is the same form we saw in Eq. 5.6. Hence, for smooth PCs, we may again choose $\gamma \propto \frac{1}{p}$.

6. Experimental Results

In this section, we examine the tradeoffs between memory, computation, and accuracy for the sparse random projections approach on both synthetic and real-world datasets. First, we synthetically generate samples $\{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^p$ distributed along one PC with $\sigma=20$. Each entry of the center and PC is drawn from the uniform distribution on $[0, 20)$ and $[0, 1)$, respectively. The PC is then normalized to have unit ℓ_2-norm. We consider a relatively noisy situation with SNR=1. We then estimate the center of the original data from the sparse random projections, where $m/p=0.2$, for varying n and compression factors γ. Our results are averaged over 10 independent trials. Fig. 6.1(a) shows the accuracy for the estimated center, where the error is the distance between the estimated and the true center normalized by the true center's norm. As expected, when n or dimension p increase, the compression factor γ can be tuned to achieve a substantial reduction of storage space while obtaining accurate estimates. This is desirable for high-dimensional data stream processing.

We then fix $n=2p$, and plot the inner product magnitude between the estimated and true PC in Fig. 6.1(b) and the



(a) (b)

Figure 6.2. Results for the MNIST dataset. Our proposed approach is compared with two methods: (1) performing MATLAB's svds on the full original data, (2) BSOI (Mitliagkas et al., 2013). Plot of (a) performance accuracy based on the explained variance and (b) computation time for performing SVD. We see that our approach performs as well as SVD on the original data and outperforms BSOI with significantly less computation time.

computation time in Fig. 6.1(d) for varying γ. We observe that, despite saving nearly two orders of magnitude in computation time and also in memory (note $\gamma = \frac{1}{50}, \frac{1}{100}, \frac{1}{200}$) compared to PCA on the full data, the PC is well-estimated. Moreover, the approach remains increasingly effective for higher dimensions, which is of crucial importance for modern data processing applications. We further note that, as the dimension increases, we can decrease the compression factor γ while still achieving a desired performance. For example, $\gamma{=}\frac{1}{100}$ for $p{=}4{\times}10^3$ and $\gamma{=}\frac{1}{200}$ for $p{=}10^4$ have almost the same accuracy. This is consistent with the observation $\gamma{\propto}\frac{1}{p}$ from before.

We also plot the estimation error for the singular value σ in Fig. 6.1(c). The error is the distance between the singular value obtained by performing SVD on $\{\mathbf{R}_i\mathbf{y}_i\}_{i=1}^n$ and on the original data $\{\mathbf{x}_i\}_{i=1}^n$, normalized by the latter value.

Finally, we consider the MNIST dataset to see a real-world application outside the spiked covariance model. This dataset contains 70,000 samples of handwritten digits, which we have resized to $40{\times}40$ pixels. Hence, we have 70,000 samples in $\mathbb{R}^{1600}$. To evaluate the performance of our method, we use the *explained variance* described in (Mitliagkas et al., 2013). Given estimates of d PCs $\widetilde{\mathbf{V}}{\in}\mathbb{R}^{p\times d}$ and the data matrix $\mathbf{X}$, the fraction of explained variance is defined as $tr(\widetilde{\mathbf{V}}^T\mathbf{X}\mathbf{X}^T\widetilde{\mathbf{V}})/tr(\mathbf{X}\mathbf{X}^T)$. We compare the performance of our approach with (1) performing SVD (using MATLAB svds) on the original data that are fully acquired and stored, and as a useful point of comparison, with (2) the online algorithm Block-Stochastic Orthogonal Iteration (BSOI) (Mitliagkas et al., 2013), where the data samples are fully acquired but not stored. We show the results in Fig. 6.2 for the measurement ratio $m/p{=}0.1$.

In terms of accuracy, our approach performs about as well as SVD on the original data, and has slightly better performance compared to BSOI. The sparse random projections result in a significant reduction of computational complexity, with one order and two orders of magnitude speedup compared to the original SVD and BSOI, respectively. In terms of memory requirements, 340 MB is needed to store the original data. However, the required memory for our framework is 44 MB for $\gamma{=}\frac{1}{20}$ and 24 MB for $\gamma{=}\frac{1}{40}$. The projected data thus can easily reside in the main memory.

Moreover, we have compared our method with the fast randomized SVD algorithm in (Halko et al., 2011a). The estimation accuracy of this method is very close to SVD on the original data, and the computation time is about 1.2 seconds, which is slightly less than the computation time of our method. This is as we would expect, since fast randomized SVD is designed specifically for low-computational complexity. However, (Halko et al., 2011a) is a full data method, meaning that it is assumed that the full data is available for computation and does not require time or cost to access. Our approach performs approximately as well in similar computation time while also allowing a reduction in memory (or data access or data communication costs) by a factor of γ, in this case $\frac{1}{20}$ and $\frac{1}{40}$. This can be a significant advantage in the case where data is stored in a large database system or distributed network.

This example indicates that our approach results in a significant simultaneous reduction of memory and/or computational cost with little loss in accuracy.

7. Conclusions

We have presented a memory- and computation-efficient approach for estimation of PCs via very sparse random projections. This approach simultaneously reduces substantially the required memory and computation for PC estimation, while still providing high accuracy. More importantly, it allows us to rigorously analyze each of memory, computation, and accuracy in terms of the sparsity of the projection, for various PC models. Thus, we have been able to give provable tradeoffs between memory, computation, and accuracy. Furthermore, a user of this approach could even use the sparsity of the projections to tune to any desired point on this three-way tradeoff. We believe that this approach could be valuable for various important modern data processing applications such as massive databases, distributed networks, and high-dimensional data stream processing, although we have not focused on the specific details of these in favor of more theoretical analysis. Indeed, we observe that our approach performs well in initial practical simulations, e.g. for the MNIST dataset, with large reduction of both memory and computation without sacrificing accuracy.

Acknowledgements: This material is based upon work supported by the National Science Foundation under Grant CCF-1117775.

References

D. Achlioptas. Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 274–281, 2001. 4

N. Ailon and B. Chazelle. The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, 39:302–322, 2009. 5.2

R. Arora, A. Cotter, K. Livescu, and N. Srebro. Stochastic optimization for PCA and PLS. In *50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 861–868, 2012. 2

R. Arora, A. Cotter, and N. Srebro. Stochastic optimization of PCA with capped MSG. In *NIPS*, pages 1815–1823, 2013. 2

M. Balcan, S. Ehrlich, and Y. Liang. Distributed k-means and k-median clustering on general topologies. In *NIPS*, pages 1995–2003, 2013. 1

V. Chandrasekaran and M. Jordan. Computational and statistical tradeoffs via convex relaxation. *Proc. of the National Academy of Sciences*, 110:E1181–E1190, 2013. 2

Y. Chen, Y. Chi, and A. Goldsmith. Exact and stable covariance estimation from quadratic sampling via convex programming. *arXiv preprint arXiv:1310.0807*, 2013. 2

C. Davis and W. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM J. on Numerical Analysis*, 7: 1–46, 1970. 5, 5.3

D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52:1289–1306, 2006. 1

M. Elad. Optimized projections for compressed sensing. *IEEE Trans. SP*, 55:5695–5702, 2007. 5.2

J. Fowler. Compressive-projection principal component analysis. *IEEE Trans. on Image Process.*, pages 2230–2242, 2009. 2

A. Gilbert, J. Park, and M. Wakin. Sketched SVD: Recovering spectral features from compressive measurements. *arXiv preprint arXiv:1211.0361*, 2012. 2, 5.3

G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012. 4

N. Halko, P. Martinsson, Y. Shkolnisky, and M. Tygert. An algorithm for the principal component analysis of large data sets. *SIAM Journal on Scientific Computing*, 33(5): 2580–2594, 2011a. 2, 6

N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011b. 2

L. Hogben. *Handbook of linear algebra*. CRC Press, 2006. 5

P. Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM (JACM)*, 53(3):307–323, 2006. 1

I. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of Statistics*, 29(2):295–327, 2001. 3

P. Li, T. Hastie, and K. Church. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 287–296, 2006. 1, 4, 5.1, 5.2

J. Lin and D. Gunopulos. Dimensionality reduction by random projection and latent semantic indexing. In *proceedings of the Text Mining Workshop, at the 3rd SIAM International Conference on Data Mining*, 2003. 4

I. Mitliagkas, C. Caramanis, and P. Jain. Memory limited, Streaming PCA. In *NIPS*, 2013. 2, 6.2, 6

S. Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005. 1

D. Omidiran and M. Wainwright. High-dimensional variable selection with sparse random projections: measurement sparsity and statistical efficiency. *The Journal of Machine Learning Research*, 99:2361–2386, 2010. 4

H. Qi and S. Hughes. Invariance of principal components under low-dimensional random projection of the data. In *ICIP*, pages 937–940, 2012. 2

R. Vershynin. How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25(3):655–686, 2012. 4

M. Wang, W. Xu, E. Mallada, and A. Tang. Sparse recovery with graph constraints: Fundamental limits and measurement construction. In *IEEE Proceedings INFOCOM*, pages 1871–1879, 2012a. 4

M. Wang, W. Xu, E. Mallada, and A. Tang. Sparse recovery with graph constraints. *CoRR*, abs/1207.2829, 2012b. 4

K. Zhang, L. Zhang, and M. Yang. Real-time compressive tracking. In *Computer Vision–ECCV*, pages 864–877. Springer, 2012. 4