Journal of Machine Learning Research 18 (2017) 1-34 Submitted 5/2016; Revised 6/2017; Published 7/26

Recovering PCA and Sparse PCA via Hybrid-(ℓ_1, ℓ_2) Sparse Sampling of Data Elements

Abhisek Kundu ABHISEKKUNDU@GMAIL.COM
Intel Parallel Computing Labs
Intel Tech (I) Pvt Ltd, Devarabeesanhalli, Outer Ring Road
Bangalore, 560103, India

Petros Drineas PDRINEAS@PURDUE.EDU
Computer Science
Purdue University
West Lafayette, IN 47907, USA

Malik Magdon-Ismail MAGDON@CS.RPI.EDU
Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180, USA

Editor: David Wipf

Abstract

This paper addresses how well we can recover a data matrix when only given a few of its elements. We present a randomized algorithm that element-wise sparsifies the data, retaining only a few of its entries. Our new algorithm independently samples the data using probabilities that depend on both squares (ℓ_2 sampling) and absolute values (ℓ_1 sampling) of the entries. We prove that this hybrid algorithm (i) achieves a near-PCA reconstruction of the data, and (ii) recovers sparse principal components of the data, from a sketch formed by a sublinear sample size. Hybrid-(ℓ_1, ℓ_2) inherits the ℓ_2-ability to sample the important elements, as well as the regularization properties of ℓ_1 sampling, and maintains strictly better quality than either ℓ_1 or ℓ_2 on their own. Extensive experimental results on synthetic, image, text, biological, and financial data show that not only are we able to recover PCA and sparse PCA from incomplete data, but we can speed up such computations significantly using our sparse sketch .

Keywords: element-wise sampling, sparse representation, pca, sparse pca, hybrid-(ℓ_1, ℓ_2)

1. Introduction

We address two fundamental data science problems, namely, (i) a near-PCA reconstruction of the data, and (ii) recovering sparse principal components of the data, in a setting where we can use only a few of the data entries (*element-wise matrix sparsification* problem was pioneered by Achlioptas and McSherry 2001, 2007). This is a situation that one is confronted with all too often in machine learning (say, we have a small sample of data points and those data points have missing features). For example, with user-recommendation data, one does not have all the ratings of any given user. Or in a privacy preserving setting, a client may not want to give us all entries in the data matrix. In such a setting, our goal is to show that if the samples that we get are chosen carefully, the top-k PCA and sparse PCA features of the data can be recovered within some provable error bounds. In

fact, we show that solutions of a large class of optimization problems, irrespective of whether they are efficiently solvable (e.g. PCA) or NP-hard (e.g. sparse PCA), can be approximated provably by such element-wise sparse representation of the data.

More formally, the data matrix is $\mathbf{A} \in \mathbb{R}^{m\times n}$ (m data points in n dimensions). Often, real data matrices have low effective rank, so let $\mathbf{A}_k$ be the best rank-k approximation to $\mathbf{A}$ with $\|\mathbf{A}-\mathbf{A}_k\|_2$ being small, where $\|\mathbf{X}\|_2$ is the spectral norm of matrix $\mathbf{X}$. $\mathbf{A}_k$ is obtained by projecting $\mathbf{A}$ onto the subspace spanned by its top-k principal components. In order to approximate this top-k principal subspace, we adopt the following strategy. Select a small number, s, of elements from $\mathbf{A}$ and produce a sparse sketch $\tilde{\mathbf{A}}$; use $\tilde{\mathbf{A}}$ to approximate the top-k principal subspace. Note that both PCA and the sparse PCA problem have the same objective, i.e., to maximize the variance of the data; however, the sparse PCA problem has an additional sparsity constraint on each principal component (and this makes it NP-hard). Sections 2.1 and 2.2 contains the formulations of PCA and sparse PCA, respectively. We give the details of Algorithm 2 to approximate the top-k principal subspace and the corresponding theoretical guarantees in Theorem 3. Theorem 4 shows the quality of approximation for the sparse PCA problem (and this error bound is applicable to a large class of optimization problems as long as the objective function satisfies a generalized notion of Lipschitz continuity to be discussed later). The key quantity that we must control to recover a close approximation to PCA and sparse PCA is how well the sparse sketch approximates the data *in the operator norm*. That is, if $\|\mathbf{A}-\tilde{\mathbf{A}}\|_2$ is small then we can recover PCA and sparse PCA effectively from a small number of samples.

Problem: Element-wise Matrix Sparsification

Given $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\epsilon > 0$, sample s elements to obtain a sparse sketch $\tilde{\mathbf{A}}$ for which

$$\|\mathbf{A}-\tilde{\mathbf{A}}\|_2 \le \epsilon \quad \text{and} \quad \|\tilde{\mathbf{A}}\|_0 \le s. \tag{1}$$

See SECTION 1.1 for notations. Our main result addresses the problem above. In a nutshell, with only partially observed data whose elements have been carefully selected, one can recover an approximation to the top-k principal subspace. An additional benefit is that our approximation to the top-k subspace using iterated matrix multiplication (e.g., power iterations) can benefit computationally from sparsity. To construct $\tilde{\mathbf{A}}$, we use a general randomized approach which independently samples (and rescales) s elements from $\mathbf{A}$ using probability p_{ij} to sample element $\mathbf{A}_{ij}$. We analyze in detail the case $p_{ij} \propto \alpha|\mathbf{A}_{ij}| + (1-\alpha)|\mathbf{A}_{ij}|^2$ to get a bound on $\|\mathbf{A}-\tilde{\mathbf{A}}\|_2$. We now make our discussion precise, starting with our notation.

1.1 Notation

We use bold uppercase (e.g., $\mathbf{X}$) for matrices and bold lowercase (e.g., $\mathbf{x}$) for column vectors. The i-th row of $\mathbf{X}$ is $\mathbf{X}_{(i)}$, and the i-th column of $\mathbf{X}$ is $\mathbf{X}^{(i)}$. Let $[n]$ denote the set $\{1,2,...,n\}$. $\mathbb{E}(X)$ is the expectation of a random variable X; for a matrix, $\mathbb{E}(\mathbf{X})$ denotes the element-wise expectation. For a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, the Frobenius norm $\|\mathbf{X}\|_F$ is $\|\mathbf{X}\|_F^2 = \sum_{i,j=1}^{m,n} \mathbf{X}_{ij}^2$, and the spectral (operator) norm $\|\mathbf{X}\|_2$ is $\|\mathbf{X}\|_2 = \max_{\|\mathbf{y}\|_2=1}\|\mathbf{X}\mathbf{y}\|_2$. We also have the ℓ_1 and ℓ_0 norms: $\|\mathbf{X}\|_1 = \sum_{i,j=1}^{m,n} |\mathbf{X}_{ij}|$ and $\|\mathbf{X}\|_0$ is the number of non-zero entries in $\mathbf{X}$. The k-th largest singular value of $\mathbf{X}$ is $\sigma_k(\mathbf{X})$. For symmetric matrices $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{Y} \succeq \mathbf{X}$ if and only if $\mathbf{Y}-\mathbf{X}$ is positive semi-definite. $\mathbf{I}_n$ is the $n \times n$ identity and $\ln x$ is the natural logarithm of x. We use $\mathbf{e}_i$ to denote standard basis vectors whose dimensions will be clear from the context.

Two popular sampling probabilities are ℓ_1, where $p_{ij} = |\mathbf{A}_{ij}|/\|\mathbf{A}\|_1$ (Achlioptas and McSherry 2001; Achlioptas et al. 2013a), and ℓ_2, where $p_{ij} = \mathbf{A}_{ij}^2/\|\mathbf{A}\|_F^2$ (Achlioptas and McSherry 2001; Drineas and Zouzias 2011). We construct $\tilde{\mathbf{A}}$ as follows: $\tilde{\mathbf{A}}_{ij} = 0$ if (i, j)-th entry is not sampled; sampled elements $\mathbf{A}_{ij}$ are rescaled to $\tilde{\mathbf{A}}_{ij} = \mathbf{A}_{ij}/p_{ij}$ which makes the sketch $\tilde{\mathbf{A}}$ an unbiased estimator of $\mathbf{A}$, so $\mathbb{E}[\tilde{\mathbf{A}}] = \mathbf{A}$. The sketch is *sparse* if the number of sampled elements is sublinear in mn, i.e., $s = o(mn)$. Sampling according to element magnitudes is natural in many applications, for example in a recommendation system users tend to rate a product they like (high positive) or dislike (high negative).

Our main sparsification algorithm (Algorithm 1) receives as input a matrix $\mathbf{A}$ and an accuracy parameter $\epsilon > 0$, and samples s elements from $\mathbf{A}$ in s independent, identically distributed trials with replacement, according to a hybrid-(ℓ_1, ℓ_2) probability distribution specified in Equation (3). The algorithm returns $\tilde{\mathbf{A}} \in \mathbb{R}^{m\times n}$, a sparse and unbiased estimator of $\mathbf{A}$, as a solution to (1).

1.2 Prior Work on Element Sampling

Achlioptas and McSherry (2001, 2007) pioneered the idea of ℓ_2 sampling for element-wise sparsification. However, ℓ_2 sampling on its own is insufficient for provably accurate bounds on $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$. As a matter of fact Achlioptas and McSherry (2001, 2007) observed that "small" entries need to be sampled with probabilities that depend on their absolute values only, thus also introducing the notion of ℓ_1 sampling. The underlying reason for the need of ℓ_1 sampling is the fact that if a small element is sampled and rescaled using ℓ_2 sampling, this would result in a huge entry in $\tilde{\mathbf{A}}$ (because of the rescaling). As a result, the variance of ℓ_2 sampling is quite high, resulting in poor theoretical and experimental behavior. ℓ_1 sampling of small entries rectifies this issue by reducing the variance of the overall approach. Arora et al. (2006) proposed a sparsification algorithm that deterministically keeps large entries, i.e., entries of $\mathbf{A}$ such that $|\mathbf{A}_{ij}| \geq \epsilon/\sqrt{n}$ and randomly rounds the remaining entries using ℓ_1 sampling. Formally, entries of $\mathbf{A}$ that are smaller than $\epsilon\sqrt{n}$ are set to $\text{sign}(\mathbf{A}_{ij})\,\epsilon/\sqrt{n}$ with probability $p_{ij} = \sqrt{n}\,|\mathbf{A}_{ij}|/\epsilon$ and to zero otherwise. They used an ϵ-net argument to show that $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$ was bounded with high probability. Drineas and Zouzias (2011) bypassed the need for ℓ_1 sampling by zeroing-out the small entries of $\mathbf{A}$ (e.g., all entries such that $|\mathbf{A}_{ij}| < \epsilon/2n$ for a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$) and then use ℓ_2 sampling on the remaining entries in order to sparsify the matrix. This simple modification improves Achlioptas and McSherry (2007) and Arora et al. (2006), and comes with an elegant proof using the matrix-Bernstein inequality of Recht (2011). Note that all these approaches need truncation of small entries. Recently, Achlioptas et al. (2013a) showed that ℓ_1 sampling in isolation could be done without any truncation, and argued that (under certain assumptions) ℓ_1 sampling would be better than ℓ_2 sampling, even using the truncation. Their proof is also based on the matrix-valued Bernstein inequality of Recht (2011). Finally, the result that is closest to ours is due to Achlioptas et al. (2013b), where the proposed distribution is

$$p_{ij} = \rho_i \cdot |\mathbf{A}_{ij}|\,/\|\mathbf{A}_{(i)}\|_1 \quad \text{(Bernstein distribution)} \tag{2}$$

where, ρ_i is a distribution over the rows, i.e., $\sum_i \rho_i = 1$. This distribution is derived using matrix-Bernstein inequality for a fixed sampling budget s, and under some assumptions it is shown to be near-optimal in the sense that the error $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$ it produces is within a constant factor of the smallest error produced by an optimal distribution over elements $\mathbf{A}_{ij}$. Similar to this distribution we also use a convex combination of probabilities. However, our proposed hybrid distribution only relies on ℓ_1 and ℓ_2 probabilities, thus requiring considerably less prior information about the data

$\mathbf{A}$ comparing to Bernstein distribution (Achlioptas et al. 2013b), while also being more faithful to how matrices in practice are sampled locally based on element properties, and not based on global properties of a matrix.

1.3 Our Contributions

We introduce an intuitive hybrid approach to element-wise matrix sparsification, by combining ℓ_1 and ℓ_2 sampling. We propose to use sampling probabilities of the form

$$p_{ij} = \alpha \cdot \frac{|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_1} + (1-\alpha)\frac{\mathbf{A}_{ij}^2}{\|\mathbf{A}\|_F^2}, \quad \alpha \in (0,1] \tag{3}$$

for all i, j . We retain the good properties of ℓ_2 sampling that bias us towards data elements in the presence of small noise, while *regularizing* smaller entries using ℓ_1 sampling. We summarize the main contributions below.

• We give a parameterized sampling distribution in the variable $\alpha \in (0,1]$ that controls the balance between ℓ_2 sampling and ℓ_1 regularization. This greater flexibility allows us to achieve better accuracy than both ℓ_1 and ℓ_2. Further, we derive the optimal hybrid-(ℓ_1, ℓ_2) distribution, using Lemma 1 for any given $\mathbf{A}$, by computing the optimal parameter α^* which achieves the desired accuracy with the smallest sample size using our theoretical bound.

Setting $\alpha = 1$ in our bounds we reproduce the result of Achlioptas et al. (2013a) who claim that ℓ_1 is typically better than ℓ_2. Moreover, our result shows that $\alpha^* < 1$ which means that the hybrid approach is better than ℓ_1 (and ℓ_2). Thus, our result generalizes the result of Achlioptas et al. (2013a). Note that the Bernstein distribution of Achlioptas et al. (2013b) (for a given s) is a convex combination of 'intra-row' weights $|\mathbf{A}_{ij}| / \|\mathbf{A}\|_1$, where as, our distribution (for a given ϵ) is a much simpler and intuitive convex combination of ℓ_1 and ℓ_2 probabilities.

• We propose Algorithm 2 to provably recover PCA by constructing a sparse unbiased sketch of (centered) data from a limited number of samples. Moreover, we show how we can effectively approximate the sparse PCA problem (NP-hard) from incomplete data. In fact, we prove how sampling based sketches can approximate the solutions for a large class of optimization problems (irrespective of their computational hardness) from limited number of observed data points.

For the above problems we want to derive probabilities using little or no global information about the data. The Bernstein distribution in Achlioptas et al. (2013b) requires additional information about row norms $\|\mathbf{A}_{(i)}\|_1$, as well as, optimal convex combinations ρ_i, for each row. In contrast, our distribution assumes only one real number α^*. This makes our optimal hybrid distribution more convenient for the above problems in practice.

• Finally, we propose Algorithm 3 to provably implement hybrid sampling using *only one pass* over the data. This is particularly useful in streaming setting where we receive data as a stream of numbers and we have only limited memory to store them. Note that Bernstein sampling of Achlioptas et al. (2013b) can be implemented in streaming setting as well; however, they need information (or estimates) of those global quantities beforehand. In contrast, we need no prior knowledge of parameter α which governs the sampling distribution. Surprisingly, we can set α^* (or its estimate) at a later stage of the sampling process when the stream terminates.

Extensive experimental results on image, text, biological, and financial data show that sketches using our optimal hybrid-(ℓ_1, ℓ_2) sampling achieve better quality of approximation to PCA and sparse PCA problems than either ℓ_1 or ℓ_2 sampling on their own. Moreover, we can speed up such

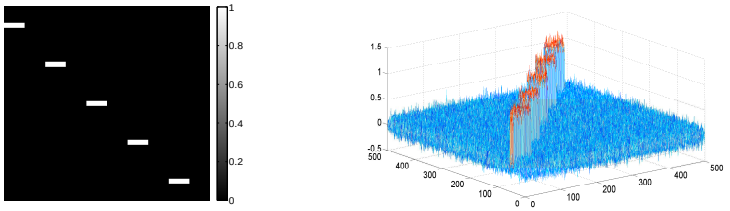


Figure 1: (left) Synthetic 500×500 binary data $\mathbf{D}$; (right) mesh view of noisy data $\mathbf{A}_{0.1}$.

computations significantly using our sparse sketch. Finally, our results are comparable (if not better) than those of Achlioptas et al. (2013b) despite using significantly less information about the data.

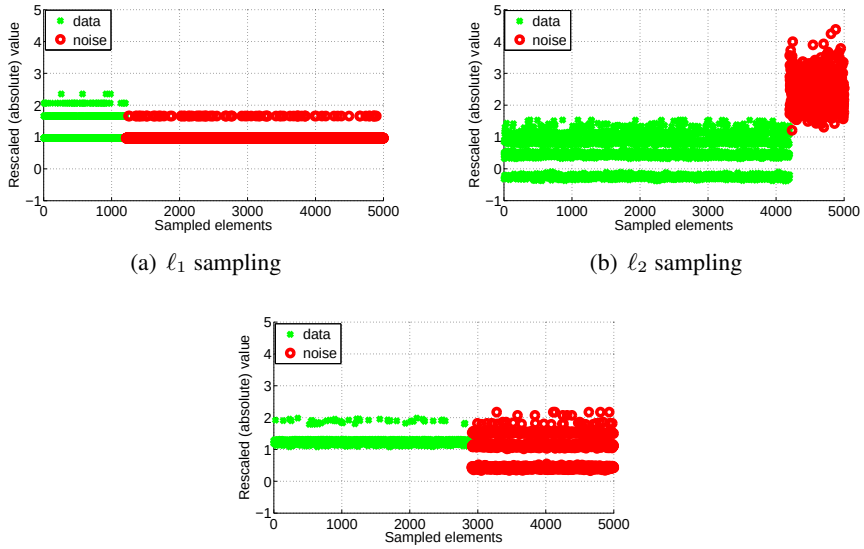
1.4 A Motivating Example for Hybrid Sampling

The main motivation for introducing the idea of hybrid-(ℓ_1, ℓ_2) sampling comes from achieving a tighter bound on s using a simple and intuitive probability distribution on elements of $\mathbf{A}$. For this, we observe certain good properties of both ℓ_1 and ℓ_2 sampling for sparsification of noisy data (in practice, we experience data that are noisy, and it is perhaps impossible to separate "true" data from noise). We illustrate the behavior of ℓ_1 and ℓ_2 sampling on noisy data using the following synthetic example. We construct a 500×500 binary data $\mathbf{D}$ (FIGURE 1), and then perturb it by a random Gaussian matrix $\mathbf{N}$ whose elements $\mathbf{N}_{ij}$ follow Gaussian distribution with mean zero and standard deviation 0.1. We denote this perturbed data matrix by $\mathbf{A}_{0.1}$. First, we note that ℓ_1 and ℓ_2 sampling work *identically* on binary data $\mathbf{D}$. However, FIGURE 2 depicts the change in behavior of ℓ_1 and ℓ_2 sampling to sparsify $\mathbf{A}_{0.1}$. Data elements and noise in $\mathbf{A}_{0.1}$ are the elements with non-zero and zero values in $\mathbf{D}$, respectively. We sample $s = 5000$ indices in i.i.d. trials according to ℓ_1 and ℓ_2 probabilities separately to produce sparse sketch $\tilde{\mathbf{A}}$. FIGURE 2 shows that elements of $\tilde{\mathbf{A}}$, produced by ℓ_1 sampling, have controlled variance but most of them are noise. On the other hand, ℓ_2 sampling is biased towards data elements, although small number of sampled noisy elements create large variance due to rescaling. Such large magnitude noisy elements become outliers in $\tilde{\mathbf{A}}$; consequently, PCA on $\tilde{\mathbf{A}}$ becomes a poor approximation to the true PCA of $\mathbf{A}$. Our hybrid-(ℓ_1, ℓ_2) sampling benefits from this bias of ℓ_2 towards data elements to sample large number of true data. Additionally, the regularization property of ℓ_1 component prevents noisy elements to become outliers in $\tilde{\mathbf{A}}$ helping us to achieve near-PCA reconstruction.

We parameterize our distribution using the variable $\alpha \in (0, 1]$ that controls the balance between ℓ_2 sampling and ℓ_1 regularization. One can view α as a regularization parameter. We derive an expression to compute α^*, the optimal α, corresponding to the smallest sample size that we need to achieve an accuracy ϵ in (1). When $\alpha = 1$ we reproduce the result of Achlioptas et al. (2013a). However, α^* may be smaller than 1, and the bound on sample size, using α^*, is guaranteed to be tighter than that of Achlioptas et al. (2013a) and ℓ_2.

2. Main Results

We present the quality-of-approximation result of our main sampling algorithm (Algorithm 1). We sample some of the non-zero elements of the fixed matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ (and rescale them) to form a sparse matrix $\tilde{\mathbf{A}}$ that is close to $\mathbf{A}$ in spectral norm. It would be useful to interpret the sampling



(a) ℓ_1 sampling

(b) ℓ_2 sampling

(c) Hybrid-(ℓ_1, ℓ_2) sampling

Figure 2: Elements of sparse sketch $\tilde{\mathbf{A}}$ produced from $\mathbf{A}_{0.1}$ by (a) ℓ_1, (b) ℓ_2, and (c) hybrid-(ℓ_1, ℓ_2), $\alpha = 0.7$ sampling. In each plot, x-axis is the number of sampled indices $s = 5000$, including both data and noise. y-axis plots the rescaled absolute values (in $\ln$ scale) of $\tilde{\mathbf{A}}$ corresponding to the sampled indices. ℓ_1 sampling produces elements with controlled variance but it mostly samples noise, whereas ℓ_2 samples a lot of data although producing large variance of rescaled elements. Hybrid-(ℓ_1, ℓ_2) sampling uses ℓ_1 as a regularizer while sampling fairly large number of data that helps to preserve the spectral structure of original data. Data and noisy elements are shown as clusters for better visualization.

of elements from $\mathbf{A}$ as follows. We can express $\mathbf{A}$ as a sum of matrices each having at most one non-zero element.

$$\mathbf{A} = \sum_{i,j=1}^{m,n} \mathbf{A}_{ij} \mathbf{e}_i \mathbf{e}_j^T, \quad \forall (i,j) \in [m] \times [n] \tag{4}$$

We sample (in i.i.d. trials) some of the terms in (4) according to some probability distribution $\{p_{ij}\}_{i,j=1}^{m,n}$ defined over the elements of $\mathbf{A}$ to form a (weighted) partial sum that represents our sparse sketch $\tilde{\mathbf{A}}$ of $\mathbf{A}$. More formally, we define the following sampling operator $\mathcal{S}_\Omega : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ in (5) that extracts some of the terms from (4). Let $\Omega \subset [m] \times [n]$ be a multi-set of sampled indices (i_t, j_t), for $t = 1, ..., s$. Then,

$$\tilde{\mathbf{A}} = \mathcal{S}_\Omega (\mathbf{A}) = \frac{1}{s} \sum_{t=1}^{s} \frac{\mathbf{A}_{i_t j_t}}{p_{i_t j_t}} \mathbf{e}_{i_t} \mathbf{e}_{j_t}^T, \quad (i_t, j_t) \in \Omega \tag{5}$$

Clearly, $\tilde{\mathbf{A}}$ in (5) is a sparse (at most s non-zero elements), unbiased estimator of $\mathbf{A}$. We use the sampling operator in (5) in Algorithm 1 to randomly sample (in i.i.d. trials) s elements of $\mathbf{A}$,

according to p_{ij}'s as in equation (3). In order to bound $\|\mathbf{A}-\tilde{\mathbf{A}}\|_2$ we first express $\mathbf{A}-\tilde{\mathbf{A}}$ as a sum of zero-mean, independent, random matrices, and then use the following matrix-Bernstein inequality in Lemma 1 (due to Recht 2011) that bounds the deviation of spectral norms of such matrices.

Lemma 1 *[Theorem 3.2 of Recht 2011] Let $\mathbf{M}_1, \mathbf{M}_2, ..., \mathbf{M}_s$ be independent, zero-mean random matrices in $\mathbb{R}^{m\times n}$. Suppose $\max_{t\in[s]}\left\{\|\mathbb{E}(\mathbf{M}_t\mathbf{M}_t^T)\|_2, \|\mathbb{E}(\mathbf{M}_t^T\mathbf{M}_t)\|_2\right\} \leq \rho^2$, and $\|\mathbf{M}_t\|_2 \leq \gamma$ for all $t \in [s]$. Then, for any $\epsilon > 0$, $\left\|\frac{1}{s}\sum_{t=1}^{s}\mathbf{M}_t\right\|_2 \leq \epsilon$ holds, subject to a failure probability at most $(m+n)\exp\left(\frac{-s\epsilon^2/2}{\rho^2+\gamma\epsilon/3}\right)$.*

A popular construction of $\mathbf{M}_t \in \mathbb{R}^{m\times n}$, $t \in [s]$, in Lemma 1 is $\mathbf{M}_t = \frac{\mathbf{A}_{i_tj_t}}{p_{i_tj_t}}\mathbf{e}_{i_t}\mathbf{e}_{j_t}^T - \mathbf{A}$ for $p_{i_tj_t} \neq 0$. Clearly, $\frac{1}{s}\sum_{t=1}^{s}\mathbf{M}_t = \tilde{\mathbf{A}} - \mathbf{A}$. We can see that ρ^2 and γ of Lemma 1 are now dependent on p_{ij} (Lemmas 8 and 9), i.e., different choices of p_{ij} lead to different bounds on ρ^2 and γ. Now, upper bounding the failure probability in Lemma 1 by $\delta > 0$, we can express the sample size s as a function of the key quantities ρ^2 and γ, and consequently as a function of p_{ij}:

$$s \geq \frac{2}{\epsilon^2}\cdot(\rho^2 + \gamma\epsilon/3)\cdot\ln((m+n)/\delta).$$

Since our choice of p_{ij} in (3) is a function of distribution parameter α, we essentially parameterize ρ^2 and γ by α, and express s as a function of α. Let us define $f(\alpha)$ as follows.

$$f(\alpha) = \rho^2(\alpha) + \gamma(\alpha)\epsilon\|\mathbf{A}\|_2/3,$$

where

$$\gamma(\alpha) = \max_{\substack{i,j:\\ \mathbf{A}_{ij}\neq 0}}\left\{\frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha)\frac{\|\mathbf{A}\|_1\cdot|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_F^2}}\right\} + \|\mathbf{A}\|_2, \tag{6}$$

$$\rho^2(\alpha) = \max\left\{\max_i\sum_{j=1}^{n}\xi_{ij}(\alpha), \max_j\sum_{i=1}^{m}\xi_{ij}(\alpha)\right\} - \sigma_{min}^2, \tag{7}$$

$$\xi_{ij}(\alpha) = \|\mathbf{A}\|_F^2/\Big(\frac{\alpha\cdot\|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1} + (1-\alpha)\Big), \mathbf{A}_{ij}\neq 0, \tag{8}$$

σ_{min} is the smallest singular value of $\mathbf{A}$. Such parameterization gives us a flexibility to express the sample size as a function of α: $s(\alpha) \propto f(\alpha)$. Naturally, we want to find $\alpha \in (0, 1]$ that results in the smallest $s(\alpha)$. The optimal α may be less than 1, and setting $\alpha = 1$ (in which case equation 3 coincides with ℓ_1 probabilities of Achlioptas et al. 2013a) may not produce the smallest s.

We can see that $\gamma(\alpha)$ prevents $f(\alpha)$ from blowing up (in case of sampling a tiny element of $\mathbf{A}$) by setting α away from 0 (this is a regularization step). Thus, $\rho^2(\alpha)$ becomes the dominating term in $f(\alpha)$ as $\gamma(\alpha)$ is multiplied by a small constant ϵ. We take a closer look at $\xi_{ij}(\alpha)$ as it is the key quantity in $\rho^2(\alpha)$ which is approximately the larger of the largest row sum or the largest column sum of $\xi_{ij}(\alpha)$. While minimizing $f(\alpha)$, α maintains a trade-off between the two terms $\frac{\alpha\cdot\|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1}$ and $(1-\alpha)$ such that the maximum row sum or column sum of $\xi_{ij}(\alpha)$ gets smaller. Along row i or column j, for smaller $|\mathbf{A}_{ij}|$, we typically have $|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1 < \|\mathbf{A}\|_F^2$, and consequently $\alpha \to 1$ reduces $\rho^2(\alpha)$. On the other hand, for larger $|\mathbf{A}_{ij}|$ we have $|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1 > \|\mathbf{A}\|_F^2$ and α away from 1 is preferred to reduce $\rho^2(\alpha)$. Thus α plays an intricate role to reduce the overall sample size

Algorithm 1 Element-wise Matrix Sparsification

1: **Input:** $\mathbf{A} \in \mathbb{R}^{m\times n}$, accuracy $\epsilon > 0$.
2: **Set** s as in eq. (11).
3: **For** $t = 1 \dots s$ (i.i.d. trials with replacement) **randomly sample** pairs of indices $(i_t, j_t) \in [m] \times [n]$ with $\mathbb{P}\left[(i_t, j_t) = (i, j)\right] = p_{ij}$, where p_{ij} are as in (3), using α as in (10).
4: **Output:** $\mathcal{S}_\Omega\left(\mathbf{A}\right) = \frac{1}{s}\sum_{t=1}^{s} \frac{\mathbf{A}_{i_t j_t}}{p_{i_t j_t}} \mathbf{e}_{i_t}\mathbf{e}_{j_t}^T$.

depending on the structure of the data. Note that for $\alpha = 1$ (as in Achlioptas et al. 2013a) we have $\xi_{ij}(\alpha) = |\mathbf{A}_{ij}| \cdot \|\mathbf{A}\|_1$. Therefore, for larger $|\mathbf{A}_{ij}|$ (true data points), maximum row sum or column sum of $\xi_{ij}(\alpha)$ lose the benefit of the delicate role played by $\alpha < 1$ to reduce those quantities, and they become larger than that for $\alpha < 1$. Thus, parameterization and optimization with α gives us a strictly smaller sample size than that of ℓ_1 or ℓ_2 sampling for the problem in (1).

Our main algorithm Algorithm 1 leads us to the following theorem.

Theorem 2 *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and let $\epsilon > 0$ be an accuracy parameter. Let $\mathcal{S}_\Omega$ be the sampling operator defined in (5), and assume that the multi-set Ω is generated using sampling probabilities $\{p_{ij}\}_{i,j=1}^{m,n}$ as in (3). Then, with probability at least $1-\delta$, $\|\mathcal{S}_\Omega(\mathbf{A}) - \mathbf{A}\|_2 \leq \epsilon\|\mathbf{A}\|_2$, if*

$$s \geq \frac{2}{(\epsilon\|\mathbf{A}\|_2)^2} \cdot f(\alpha) \cdot \ln((m+n)/\delta). \tag{9}$$

Furthermore, we can find α^* (optimal α corresponding to the smallest s) by solving (10):

$$\alpha^* \quad = \quad \min_{\alpha\in(0,1]} f(\alpha), \tag{10}$$

and the corresponding optimal sample size is

$$s^* \quad = \quad \frac{2}{(\epsilon\|\mathbf{A}\|_2)^2} \cdot f(\alpha^*) \cdot \ln((m+n)/\delta). \tag{11}$$

For a given matrix $\mathbf{A}$, we can easily compute $\rho^2(\alpha)$ and $\gamma(\alpha)$ for various values of α. Given an accuracy ϵ and failure probability δ, we can compute α^* corresponding to the tightest bound on s. Note that, for $\alpha = 1$ we reproduce the results of Achlioptas et al. (2013a) (which was expressed using various matrix metrics). However, α^* may be smaller than 1, and is guaranteed to produce tighter s comparing to extreme choices of α (e.g. $\alpha = 1$ for ℓ_1). We illustrate this in FIGURE 3. We give a proof of Theorem 2 in APPENDIX A.

2.1 Approximation of PCA via Element Sampling

The top-k principal components of centered data $\mathbf{A} \in \mathbb{R}^{m\times n}$ (m data points in n dimensions), denoted by $\mathbf{V}_k \in \mathbb{R}^{n\times k}$, can be formulated as the solution to the variance maximization problem:

$$\mathbf{V}_k = \underset{\mathbf{V}\in\mathbb{R}^{n\times k}, \mathbf{V}^T\mathbf{V}=\mathbf{I}}{\text{argmax}} \quad \text{trace}(\mathbf{V}^T\mathbf{A}^T\mathbf{A}\mathbf{V}). \tag{12}$$

The maximum variance achievable using $\mathbf{V}_k$, denoted by OPT_k, is the sum of squares of the top-k singular values of $\mathbf{A}$. Note that in (12) the optimal solution is computed using the full data $\mathbf{A}$ (which is typically dense).

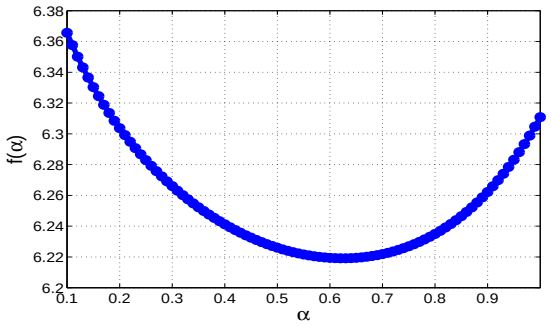


Figure 3: Plot of $f(\alpha)$ in eqn. (10) for data $\mathbf{A}_{0.1}$. We use $\epsilon = 0.05$ and $\delta = 0.1$. x-axis plots α and y-axis is in log scale.

Algorithm 2 Approximation of PCA from Data Samples

1: **Input:** Centered data $\mathbf{A} \in \mathbb{R}^{m \times n}$, sparsity parameter $s > 0$, and rank parameter k.
2: Produce sparse unbiased sketch $\tilde{\mathbf{A}}$ from $\mathbf{A}$, in s i.i.d. trials using Algorithm 1[1].
3: Perform rank truncated SVD on matrix $\tilde{\mathbf{A}}$, i.e., $[\tilde{\mathbf{U}}_k, \tilde{\mathbf{D}}_k, \tilde{\mathbf{V}}_k]$ = SVD($\tilde{\mathbf{A}}$, k).
4: **Output:** $\tilde{\mathbf{V}}_k$ (columns of $\tilde{\mathbf{V}}_k$ are the ordered principal components).

Here we discuss a provable algorithm (Algorithm 2) to approximate PCA by applying element-wise sampling. We produce a sparse unbiased estimator $\tilde{\mathbf{A}}$ of centered data $\mathbf{A}$ by sampling s elements in i.i.d. trials according to our hybrid-(ℓ_1, ℓ_2) distribution in (3). We use this $\tilde{\mathbf{A}}$ instead of $\mathbf{A}$ in (12) to compute the new optimal solution $\tilde{\mathbf{V}}_k$

$$\tilde{\mathbf{V}}_k = \underset{\mathbf{V} \in \mathbb{R}^{n \times k}, \mathbf{V}^T\mathbf{V} = \mathbf{I}}{\text{argmax}} \text{trace}(\mathbf{V}^T \tilde{\mathbf{A}}^T \tilde{\mathbf{A}} \mathbf{V}). \tag{13}$$

The computation of rank-truncated SVD on sparse data requires fewer floating point operations (therefore can be fast), and we consider the right singular vectors $\tilde{\mathbf{V}}_k$ of $\tilde{\mathbf{A}}$ as the approximate principal components of $\mathbf{A}$. Naturally, more samples produce better approximation. However, this reduces sparsity, consequently we may lose the speed advantage. Theorem 3 shows the quality of approximation of principal components produced by Algorithm 2.

Theorem 3 *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a given matrix, and $\tilde{\mathbf{A}}$ be a sparse sketch produced by Algorithm 1. Let $\tilde{\mathbf{V}}_k$ be the PCA's of $\tilde{\mathbf{A}}$ computed in step 3 of Algorithm 2. Then*

$$1) \quad \|\mathbf{A} - \mathbf{A}\tilde{\mathbf{V}}_k \tilde{\mathbf{V}}_k{}^T\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_k\|_F^2 + 4\|\mathbf{A}_k\|_F^2 \cdot \|\mathbf{A} - \tilde{\mathbf{A}}\|_2 / \sigma_k(\mathbf{A}),$$
$$2) \quad \|\mathbf{A}_k - \tilde{\mathbf{A}}_k\|_F \leq \sqrt{8k} \cdot (\|\mathbf{A} - \mathbf{A}_k\|_2 + \|\mathbf{A} - \tilde{\mathbf{A}}\|_2),$$
$$3) \quad \|\mathbf{A} - \tilde{\mathbf{A}}_k\|_F \leq \|\mathbf{A} - \mathbf{A}_k\|_F + \sqrt{8k} \cdot (\|\mathbf{A} - \mathbf{A}_k\|_2 + \|\mathbf{A} - \tilde{\mathbf{A}}\|_2).$$

1. Here we avoid computing $\|\mathbf{A}\|_2$ directly and instead upper bound it by $\|\mathbf{A}\|_F$ in $\gamma(\alpha)$ in Lemma 8. In (10) $\rho^2(\alpha)$ is the dominating term (we ignore $\sigma_{\min}$ as it is typically very small) and $\gamma(\alpha)$ is multiplied by a small constant ϵ; thus in reality such relaxation of $\gamma(\alpha)$ and $\rho^2(\alpha)$ has little effect on α^* (we verified it experimentally). However, we refer to the α that we get using the above as near-optimal.

The first inequality of Theorem 3 bounds the approximation of projected data onto the space spanned by top k approximate principal components. The second and third inequalities measure the quality of $\tilde{\mathbf{A}}_k$ as a surrogate for $\mathbf{A}_k$ and the quality of projection of sparsified data onto approximate principal components, respectively. The proofs of first two inequalities of Theorem 3 follow from Theorem 5 and Theorem 8 of Achlioptas and McSherry (2001), respectively. The last inequality follows from the triangle inequality. The last two inequalities above are particularly useful in cases where $\mathbf{A}$ is inherently low-rank and we choose an appropriate k for which $\|\mathbf{A} - \mathbf{A}_k\|_2$ is small.

2.2 Approximating Sparse PCA from Incomplete Data

PCA constructs a low dimensional subspace of the data such that projection of the data onto this subspace preserves as much information as possible. However a shortcoming of PCA is the interpretation of the principal components (or factors) as they can be linear combinations of *all* the original variables. In many cases the original variables have direct physical significance (e.g. genes in biological applications or assets in financial applications). In such cases it is desirable to have factors which have loadings on only a small number of the original variables. These interpretable factors are sparse principal components (SPCA). To derive sparse principal components, we add a sparsity constraint to the optimization problem (see equation 14): every column of $\mathbf{V}$ should have at most r non-zero entries (r is an input parameter),

$$\mathbf{S}_k = \underset{\mathbf{V}\in\mathbb{R}^{n\times k}, \mathbf{V}^T\mathbf{V}=\mathbf{I}, \|\mathbf{V}^{(i)}\|_0\leq r}{\operatorname{argmax}} \operatorname{trace}(\mathbf{V}^T\mathbf{A}^T\mathbf{A}\mathbf{V}). \tag{14}$$

The sparse PCA problem is not only NP-hard, but also inapproximable (Magdon-Ismail 2017). There are many heuristics for obtaining sparse factors (Cadima and Jolliffe, 1995; Trendafilov et al., 2003; Zou et al., 2006; d'Aspremont et al., 2007, 2008; Moghaddam et al., 2006; Shen and Huang, 2008) including some approximation algorithms with provable guarantees Asteris et al. (2014). The existing research typically addresses the task of getting just the top principal component ($k = 1$) (some exceptions are Ma 2013; Cai et al. 2013; Wang et al. 2014; Lei and Vu 2015). While the sparse PCA problem is hard and interesting, it is *not* the focus of this work.

We address the question: What if we do not know $\mathbf{A}$, but only have a sparse sampling of some of the entries in $\mathbf{A}$ (incomplete data)? The sparse sampling is used to construct a *sketch* of $\mathbf{A}$, denoted $\tilde{\mathbf{A}}$. There is not much else to do but solve the sparse PCA problem with the sketch $\tilde{\mathbf{A}}$ instead of the full data $\mathbf{A}$ to get $\tilde{\mathbf{S}}_k$,

$$\tilde{\mathbf{S}}_k = \underset{\mathbf{V}\in\mathbb{R}^{n\times k}, \mathbf{V}^T\mathbf{V}=\mathbf{I}, \|\mathbf{V}^{(i)}\|_0\leq r}{\operatorname{argmax}} \operatorname{trace}(\mathbf{V}^T\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\mathbf{V}). \tag{15}$$

We study how $\tilde{\mathbf{S}}_k$ performs as an approximation to $\mathbf{S}_k$ with respective to the objective that we are trying to optimize, namely $\operatorname{trace}(\mathbf{S}^T\mathbf{A}^T\mathbf{A}\mathbf{S})$ — the quality of approximation is measured with respect to the true $\mathbf{A}$. We show that the quality of approximation is controlled by how well $\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$ approximates $\mathbf{A}^T\mathbf{A}$ as measured by the spectral norm of the deviation $\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$. This is a general result that does not rely on how one constructs the sketch $\tilde{\mathbf{A}}$.

Lemma 4 (Sparse PCA from a Sketch) *Let $\mathbf{S}_k$ be a solution to the sparse PCA problem that solves (14), and $\tilde{\mathbf{S}}_k$ a solution to the sparse PCA problem for the sketch $\tilde{\mathbf{A}}$ which solves (15). Then,*

$$\textit{trace}(\tilde{\mathbf{S}}_k{}^T\mathbf{A}^T\mathbf{A}\tilde{\mathbf{S}}_k) \geq \textit{trace}(\mathbf{S}_k{}^T\mathbf{A}^T\mathbf{A}\mathbf{S}_k) - 2k\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_2.$$

Lemma 4 says that if we can closely approximate $\mathbf{A}$ with $\tilde{\mathbf{A}}$, then we can compute, from $\tilde{\mathbf{A}}$, sparse components which capture almost as much variance as the optimal sparse components computed from the full data $\mathbf{A}$. We prove that Lemma 4 follows as a corollary of a more general result given in Theorem 10.

In our setting, the sketch $\tilde{\mathbf{A}}$ is computed from a sparse sampling of the data elements in $\mathbf{A}$ (incomplete data). We use probabilities of the form in (3) to determine which elements to sample and how to form the sketch such that the error $\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_2$ is small. We can simplify this quantity in terms of $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$ as follows. Let $\Delta = \mathbf{A} - \tilde{\mathbf{A}}$.

$$\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_2 = \|\mathbf{A}^T\Delta + \Delta^T\mathbf{A} - \Delta^T\Delta\|_2 \leq 2\|\mathbf{A}\|_2\|\Delta\|_2 + \|\Delta\|_2^2. \tag{16}$$

We combine the bound on $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$ for given accuracy ϵ/k from Theorem 2 with Lemma 4 to derive $k\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_2 \leq \epsilon(2 + \epsilon/k)\|\mathbf{A}\|_2^2$ with a sample size $s = k^2 \cdot s^*$ where s^* as in (11). However, we simplify this bound for a better interpretation of s in terms matrix dimensions m and n and stable rank $\tilde{k} = \|\mathbf{A}\|_F^2/\|\mathbf{A}\|_2^2$. Note that $\|\mathbf{A}\|_1 \leq \sqrt{mn}\|\mathbf{A}\|_F$ from Cauchy-Schwartz. Then,

$$\gamma(\alpha)/\|\mathbf{A}\|_2 \leq 1 + \|\mathbf{A}\|_1/(\alpha\|\mathbf{A}\|_2) \leq 1 + \sqrt{mn\tilde{k}}/\alpha = \hat{\gamma}. \tag{17}$$

Also, from $|\mathbf{A}_{ij}| \leq \|\mathbf{A}\|_2$, $\rho^2(\alpha)$ in (7) can be simplified as

$$\xi_{ij}(\alpha) \leq \|\mathbf{A}\|_F^2\Big(\alpha\tilde{k}\|\mathbf{A}\|_2/\|\mathbf{A}\|_1 + (1-\alpha)\Big)^{-1}, \tag{18}$$

$$\rho^2(\alpha)/\|\mathbf{A}\|_2^2 \leq \max\{m,n\}\tilde{k}\Big(\alpha\tilde{k}\|\mathbf{A}\|_2/\|\mathbf{A}\|_1 + (1-\alpha)\Big)^{-1} = \hat{\rho}^2. \tag{19}$$

Using the above relaxations we have the following theorem on sample size complexity.

Theorem 5 (Sampling Complexity for Sparse PCA) *Sample s data-elements from $\mathbf{A} \in \mathbb{R}^{m\times n}$ to form the sparse sketch $\tilde{\mathbf{A}}$ using Algorithm 1. Let $\mathbf{S}_k$ be a solution to the sparse PCA problem that solves (14), and let $\tilde{\mathbf{S}}_k$, which solves (15), be a solution to the sparse PCA problem for the sketch $\tilde{\mathbf{A}}$ formed from the s sampled data elements. Suppose the number of samples s satisfies*

$$s \geq 2k^2\epsilon^{-2}(\hat{\rho}^2 + \epsilon\hat{\gamma}/(3k))\log((m+n)/\delta),$$

($\hat{\rho}^2$ and $\hat{\gamma}$ are dimensionless quantities that depend only on $\mathbf{A}$). Then, with probability at least $1-\delta$

$$\textit{trace}(\tilde{\mathbf{S}}_k{}^T\mathbf{A}^T\mathbf{A}\tilde{\mathbf{S}}_k) \geq \textit{trace}(\mathbf{S}_k{}^T\mathbf{A}^T\mathbf{A}\mathbf{S}_k) - 2\epsilon(2+\epsilon/k)\|\mathbf{A}\|_2^2.$$

The dependence of $\hat{\rho}^2$ and $\hat{\gamma}$ on $\mathbf{A}$ are given in (17) and (19), respectively. Roughly speaking, we can ignore the term with $\hat{\gamma}$ since it is multiplied by ϵ/k, and we have $\hat{\rho}^2 = O(\tilde{k}\max\{m,n\})$. To paraphrase Theorem 5, when the stable rank $\tilde{k}$ is a small constant, with $O(k^2\max\{m,n\})$ samples, one can recover almost as good sparse principal components as with all data (a possible price being a small fraction of the optimal variance, since $OPT_k \geq \|\mathbf{A}\|_2^2$). As far as we know, the only prior work related to the problem we consider here is Lounici (2013) which proposed a specific method to construct sparse PCA from incomplete data. However, we develop a general tool that can be used with any existing sparse PCA heuristic. Moreover, we derive much simpler bounds (Theorems 4 and 5) using matrix concentration inequalities, as opposed to ϵ-net arguments in Lounici (2013).

2.2.1 SPARSE SKETCH USING GREEDY THRESHOLD

We also give an application of Lemma 4 to run sparse PCA after "denoising" the data using a greedy thresholding algorithm that sets the small elements to zero (see Theorem 6). Such denoising is appropriate when the observed matrix has been element-wise perturbed by small noise, and the uncontaminated data matrix is sparse and contains large elements. We show that if an appropriate fraction of the (noisy) data is set to zero, one can still recover sparse principal components. This gives a principled approach to regularizing sparse PCA in the presence of small noise when the data is sparse.

We give the simplest scenario of incomplete data where Lemma 4 gives some reassurance that one can compute good sparse principal components. Suppose the smallest data elements have been set to zero. This can happen, for example, if only the largest elements are measured, or in a noisy setting if the small elements are treated as noise and set to zero. So

$$\tilde{\mathbf{A}}_{ij} = \begin{cases} \mathbf{A}_{ij} & |\mathbf{A}_{ij}| \geq \delta, \\ 0 & |\mathbf{A}_{ij}| < \delta. \end{cases}$$

Recall $\tilde{k} = \|\mathbf{A}\|_F^2/\|\mathbf{A}\|_2^2$ (stable rank of $\mathbf{A}$), and define $\|\mathbf{A}_\delta\|_F^2 = \sum_{|\mathbf{A}_{ij}|<\delta} \mathbf{A}_{ij}^2$. Let $\Delta = \mathbf{A} - \tilde{\mathbf{A}}$. By construction, $\|\Delta\|_F^2 = \|\mathbf{A}_\delta\|_F^2$ and $\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_2 \leq 2\|\mathbf{A}\|_2\|\Delta\|_2 + \|\Delta\|_2^2$ from (16). Suppose the zeroing of elements only loses a fraction of the energy in $\mathbf{A}$, i.e. δ is selected so that $\|\mathbf{A}_\delta\|_F^2 \leq \epsilon^2\|\mathbf{A}\|_F^2/\tilde{k}$; that is an $\epsilon/\tilde{k}$ fraction of the total variance in $\mathbf{A}$ has been lost in the unmeasured (or zero) data. Then $\|\Delta\|_2 \leq \|\Delta\|_F \leq \epsilon\|\mathbf{A}\|_F/\sqrt{\tilde{k}} = \epsilon\|\mathbf{A}\|_2$.

Theorem 6 *Suppose that $\tilde{\mathbf{A}}$ is created from $\mathbf{A}$ by zeroing all elements that are less than δ, and δ is such that the truncated norm satisfies $\|\mathbf{A}_\delta\|_2^2 \leq \epsilon^2\|\mathbf{A}\|_F^2/\tilde{k}$. Then the sparse PCA solution $\tilde{\mathbf{V}}^*$ satisfies*

$$\textit{trace}(\tilde{\mathbf{V}}^{*T}\mathbf{A}\mathbf{A}\tilde{\mathbf{V}}^*) \geq \textit{trace}(\mathbf{V}^{*T}\mathbf{A}\mathbf{A}^T\mathbf{V}^*) - 2k\epsilon\|\mathbf{A}\|_2^2(2+\epsilon).$$

Theorem 6 shows that it is possible to recover sparse PCA after setting small elements to zero. This is appropriate when most of the elements in $\mathbf{A}$ are small noise and a few of the elements in $\mathbf{A}$ contain large data elements. For example if the data consists of sparse $O(\sqrt{nm})$ large elements (of magnitude, say 1) and many $nm - O(\sqrt{nm})$ small elements whose magnitude is $o(1/\sqrt{nm})$ (high signal-to-noise setting), then $\|\mathbf{A}_\delta\|_2^2/\|\mathbf{A}\|_2^2 \to 0$ and with just a sparse sampling of the $O(\sqrt{nm})$ large elements (very incomplete data), we recover near optimal sparse PCA.

Not only do our algorithms preserve the quality of the sparse principal components, but iterative algorithms for sparse PCA, whose running time is proportional to the number of non-zero entries in the input matrix, benefit from the sparsity of $\tilde{\mathbf{A}}$. Our experiments show about five-fold speed gains while producing near-comparable sparse components using less than 10% of the data.

2.3 One-pass Hybrid-(ℓ_1, ℓ_2) Sampling

Here we discuss the implementation of hybrid-(ℓ_1, ℓ_2) sampling in one pass over the input matrix $\mathbf{A}$ using $O(s)$ memory, e.g., a streaming model. Here we propose an algorithm, Algorithm 3, to implement a one-pass version of the hybrid sampling *without a priori knowledge of the regularization parameter* α.

We use SELECT-s algorithm (Algorithm 5 in APPENDIX F) to implement one-pass ℓ_1 and ℓ_2 sampling. We note that steps 2-3 of Algorithm 3 access the elements of $\mathbf{A}$ only once, in parallel,

Algorithm 3 One-pass hybrid sampling

1: **Input:** $\mathbf{A}_{ij}$ for all $(i,j) \in [m] \times [n]$, arbitrarily ordered, and sample size s.
2: Apply SELECT-s algorithm (Algorithm 5) using ℓ_1 probabilities to sample s independent indices (i_{t_1}, j_{t_1}) and corresponding values $\mathbf{A}_{i_{t_1} j_{t_1}}$ to form random multiset S_1 of triples $(i_{t_1}, j_{t_1}, \mathbf{A}_{i_{t_1} j_{t_1}})$, for $t_1 = 1, ..., s$.
3: Apply SELECT-s algorithm using ℓ_2 probabilities to sample s independent indices (i_{t_2}, j_{t_2}) and corresponding values $\mathbf{A}_{i_{t_2} j_{t_2}}$ to form random multiset S_2 of triples $(i_{t_2}, j_{t_2}, \mathbf{A}_{i_{t_2} j_{t_2}})$, for $t_2 = 1, ..., s$.
4: Store $\|\mathbf{A}\|_F^2$ and $\|\mathbf{A}\|_1$ /* these are already computed in SELECT-s algorithms above */
5: /* stream terminates */
6: Set the value of $\alpha \in (0, 1]$ (say, using Algorithm 4).
7: Create empty multiset of triples S.
8: $\mathbf{X} \leftarrow \mathbf{0}_{m \times n}$.
9: **For** $t = 1 \ldots s$
10: Generate a uniform random number $x \in [0, 1]$.
11: if $x \leq \alpha$, $S(t) \leftarrow S_1(t)$; otherwise, $S(t) \leftarrow S_2(t)$.
12: $(i_t, j_t) \leftarrow S(t, 1:2)$.
13: $p \leftarrow \alpha \cdot \frac{|S(t,3)|}{\|\mathbf{A}\|_1} + (1-\alpha) \cdot \frac{|S(t,3)|^2}{\|\mathbf{A}\|_F^2}$
14: $\mathbf{X} \leftarrow \mathbf{X} + \frac{S(t,3)}{p \cdot s} e_{i_t} e_{j_t}^T$.
15: **End**
16: **Output:** random multiset S, and sparse matrix $\mathbf{X}$.

to form independent multisets S_1 and S_2. We store $\|\mathbf{A}\|_F^2$ and $\|\mathbf{A}\|_1$ already computed in steps 2-3. Subsequent steps do not need further access to $\mathbf{A}$. Interestingly, we set α in step 6 when the data stream terminates. Steps 9-15 sample s elements from S_1 and S_2 using the α in step 6, and produces a sparse matrix $\mathbf{X}$ using the sampled entries in multiset S. Theorem 7 proves that Algorithm 3 indeed samples elements of $\mathbf{A}$ according to the hybrid-(ℓ_1, ℓ_2) probabilities in (3).

Theorem 7 *Using the same notations as in Algorithm 3, for $\alpha \in (0, 1]$, $t = 1, ..., s$,*

$$P\left[S(t) = (i, j, \mathbf{A}_{ij})\right] = \alpha \cdot \frac{|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_1} + (1-\alpha) \cdot \frac{\mathbf{A}_{ij}^2}{\|\mathbf{A}\|_F^2}.$$

See proof in APPENDIX E. Note that, Theorem 7 holds for any arbitrary $\alpha \in (0, 1]$ in line 7 of Algorithm 3, i.e., Algorithm 4 is not essential for correctness of Theorem 7. We only need α to be independent of the elements of S_1 and S_2. However, we can use Algorithm 4 to get a good estimate of α^* (SECTION 2.3.1). In this case, we need additional independent samples from $\mathbf{A}$ to 'learn' the parameter α^*.

2.3.1 ESTIMATE OF α^* FROM SAMPLES

Here we discuss Algorithm 4, to estimate α^* from samples of $\mathbf{A}$ and few other quantities, such as, $\|\mathbf{A}\|_F$, $\|\mathbf{A}\|_1$, and $\mathbf{A}_{min}$, where $\mathbf{A}_{min}$ is defined as

$$\mathbf{A}_{min} = \min_{\substack{i,j: \\ \mathbf{A}_{ij} \neq 0}} |\mathbf{A}_{ij}|.$$

Algorithm 4 Estimating α^* from Samples

1: **Input:** Ω, the set of triples $\{(i,j,\mathbf{A}_{ij})\}$ where each $\mathbf{A}_{ij}$ is sampled with probability p_{ij} (say $p_{ij} = |\mathbf{A}_{ij}|/\|\mathbf{A}\|_1$), accuracy ε, $\mathbf{A}_{min}$, $\|\mathbf{A}\|_F^2$, $\|\mathbf{A}\|_1$, and matrix dimensions m and n.
2: Compute $\tilde{\gamma}(\alpha)$ as in equation (20) and $\tilde{\xi}_{ij}(\alpha)$ as in equation (21), for a given α.
3: **Output:** $\tilde{\alpha}$ in equation (23).

Let Ω be a set of sampled triples $\{(i,j,\mathbf{A}_{ij})\}$ where each $\mathbf{A}_{ij}$ is sampled with probability p_{ij} (e.g., $p_{ij} = |\mathbf{A}_{ij}|/\|\mathbf{A}\|_1$). Let us define, for a fixed α,

$$\tilde{\gamma}(\alpha) = \max_{\substack{i,j:\\ \mathbf{A}_{ij}\neq 0}} \left\{ \frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha)\frac{\|\mathbf{A}\|_1 \cdot |\mathbf{A}_{ij}|}{\|\mathbf{A}\|_F^2}} \right\} + \|\mathbf{A}\|_F = \frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha)\frac{\|\mathbf{A}\|_1}{\|\mathbf{A}\|_F^2}\cdot \mathbf{A}_{min}} + \|\mathbf{A}\|_F. \quad (20)$$

Note that, $\tilde{\gamma}(\alpha)$ can be computed using only one pass over the data. Further, we define the following random variable (parameterized by α)

$$\tilde{\xi}_{ij}(\alpha) \;=\; \frac{\|\mathbf{A}\|_F^2}{\frac{\alpha\cdot\|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1} + (1-\alpha)} \cdot \delta_{ij}, \quad (21)$$

where δ_{ij} is an indicator function defined as $\delta_{ij} := \mathbb{I}((i,j,\mathbf{A}_{ij}) \in \Omega)$. Note that, $\delta_{ij} = 1$ w.p. p_{ij}, and 0 otherwise. For a given column j, we define the following random variable $\tilde{Y}_j(\alpha)$:

$$\tilde{Y}_j(\alpha) = \sum_{i=1}^{m} \tilde{\xi}_{ij}(\alpha).$$

Similarly, for a given row i, we define $\tilde{Y}_i(\alpha) = \sum_{j=1}^{n} \tilde{\xi}_{ij}(\alpha)$. Finally, we define

$$\tilde{\rho}^2(\alpha) = \max\{\max_j \tilde{Y}_j(\alpha), \max_i \tilde{Y}_i(\alpha)\}. \quad (22)$$

Using the above quantities we solve the optimization problem in (23) and use the solution as an estimate for α^*.

$$\tilde{\alpha} : \min_{\alpha\in(0,1]} \quad \tilde{\rho}^2(\alpha) + \tilde{\gamma}(\alpha)\varepsilon/3. \quad (23)$$

Note that, $\tilde{\gamma}(\alpha)$ in equation (20) closely approximates $\gamma(\alpha)$ in equation (6), where the error of approximation is $\|\mathbf{A}\|_F - \|\mathbf{A}\|_2$. This error becomes small in equation (23) as $\tilde{\gamma}(\alpha)$ is multiplied by a small quantity $\varepsilon/3$. Next we analyze how well $\tilde{\xi}_{ij}(\alpha)$ approximates $\xi_{ij}(\alpha)$ using a simplified data model. For this we assume $|\mathbf{A}_{ij}| \in \{L, \epsilon, 0\}$, where $L \gg \epsilon > 0$, and let s_L and s_ϵ be the number of elements with magnitudes L and ϵ, respectively. Also, let the minimum singular value $\sigma_{\min} \approx 0$. For $(i,j,\mathbf{A}_{ij}) \in \Omega$ and $|\mathbf{A}_{ij}| = \epsilon$ (which is unlikely), $|\mathbf{A}_{ij}| \cdot \|\mathbf{A}\|_1 \ll \|\mathbf{A}\|_F^2$; consequently $\tilde{\xi}_{ij}(\alpha)$ is small and does not contribute much to $\tilde{Y}_j(\alpha)$ (or $\tilde{Y}_i(\alpha)$). On the other hand, for $(i,j,\mathbf{A}_{ij}) \in \Omega$ and $|\mathbf{A}_{ij}| = L$ (which is very likely), $|\mathbf{A}_{ij}| \cdot \|\mathbf{A}\|_1 \gg \|\mathbf{A}\|_F^2$, and consequently $\tilde{\xi}_{ij}(\alpha)$ contributes significantly to $\tilde{Y}_j(\alpha)$ (or $\tilde{Y}_i(\alpha)$). So, if we sample s_L number of such large elements $\tilde{\rho}^2(\alpha)$ is close to $\rho^2(\alpha)$, and $\tilde{\alpha} \approx \alpha^*$. Finally, note that Algorithm 4 can be implemented using only one-pass over the data. Experimental results on real data validate the quality of Algorithm 4.

3. Experiments

We perform various element-wise sampling on synthetic and real data to show how well the sparse sketches preserve the structure of the original data, in spectral norm. Also, we show results on the quality and computation time of the principal components and sparse principal components derived from sparse sketches.

3.1 Algorithms for Sparse Sketches

Algorithm 1 is our prototypical algorithm to produce (unbiased) sparse sketches $\tilde{\mathbf{A}}$ from a given matrix via various sampling methods. We construct our sparse sketch using our optimal hybrid-(ℓ_1, ℓ_2) probabilities, and compare its quality with other sketches produced via ℓ_1 and ℓ_2 sampling. We borrow same definitions from (Achlioptas et al. 2013a) for comparing our results with ℓ_1 sampling.

$$\text{nd}(\mathbf{A}) := \frac{\|\mathbf{A}\|_1^2}{\|\mathbf{A}\|_F^2}, \quad \text{rs}_0(\mathbf{A}) := \frac{\max_i \|\mathbf{A}_{(i)}\|_0}{\|\mathbf{A}\|_0/m}, \quad \text{rs}_1(\mathbf{A}) := \frac{\max_i \|\mathbf{A}_{(i)}\|_1}{\|\mathbf{A}\|_1/m},$$

where nd is numeric density and rs is row density skewness. Achlioptas et al. (2013a) argues that ℓ_1 outperforms ℓ_2 sampling when $\text{rs}_0 > \text{rs}_1$. We compute the theoretical optimal mixing parameter α^* by solving (10) for various datasets [2], and compare this α^* with the theoretical condition derived by Achlioptas et al. (2013a). Further, we verify the accuracy of α^* by measuring $\mathcal{E} = \|\mathbf{A} - \tilde{\mathbf{A}}\|_2 / \|\mathbf{A}\|_2$ for distributions corresponding to various α, for a given sample size s [3]. We expect $\mathcal{E}$ to be the smallest for $\alpha \approx \alpha^*$ rather than $\alpha = 1$ (ℓ_1) or $\alpha \approx 0$ (ℓ_2).

3.2 Algorithms for PCA from Sparse Sketches

We first compare three algorithms for computing PCA of the centered data $\mathbf{A}$. Let the actual PCA of the original data be $\mathcal{A}$. We use Algorithm 2 to compute approximate PCA from random samples via our optimal hybrid-(ℓ_1, ℓ_2) sampling. Let us denote this approximate PCA by $\mathcal{H}$. Also, we compute PCA of a Gaussian random projection of the original data to compare its quality with $\mathcal{H}$. Let $\mathbf{A}_G = \mathbf{G}\mathbf{A} \in \mathbb{R}^{r \times n}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the original data, and $\mathbf{G}$ is a $r \times m$ standard Gaussian matrix. Let the PCA of this random projection $\mathbf{A}_G$ be $\mathcal{G}$. The quality of various PCA is determined by measuring how much variance of the data they can capture (as in equation 12). For this, let σ_k, σ_h, and σ_g denote the variance preserved by $\mathcal{A}$, $\mathcal{H}$, and $\mathcal{G}$, respectively. Note that, $\mathcal{A}$ and $\mathcal{G}$ require access to the full data, while $\mathcal{H}$ is computed from only few samples of $\mathbf{A}$. Also, we compare the computation time (in milliseconds) t_a, t_h, and t_G for $\mathcal{A}$, $\mathcal{H}$, and $\mathcal{G}$, respectively.

Finally, we compare the quality of our optimal hybrid sampling with ℓ_1 sampling and Bernstein sampling (Achlioptas et al. 2013b) by quantifying the variance, σ_{ℓ_1} and σ_b, preserved by the corresponding sparse sketches.

3.3 Algorithms for Approximate Sparse PCA from Sketches

We show the experimental results for sparse PCA from a sketch using several real data matrices. As we mentioned, sparse PCA is NP-Hard, and so we use heuristics. These heuristics are discussed next, followed by the data, the experimental design and finally the results.

2. we find α^* from the plot of $f(\alpha)$ for $\alpha \in [0.1, 1]$.

3. here we use $\mathbf{A}$, s, and α as input to Algorithm 1 to form $\tilde{\mathbf{A}}$ by s i.i.d samples from p_{ij} in (3) for given α.

Let $\mathcal{G}$ (ground truth) denote the algorithm which computes the principal components (which may not be sparse) of the full data matrix $\mathbf{A}$; the optimal variance is OPT_k. We consider six heuristics for getting sparse principal components.

$\mathcal{G}_{\text{max},r}$	The r largest-magnitude entries in each principal component generated by $\mathcal{G}$.
$\mathcal{G}_{\text{sp},r}$	r-sparse components using the *Spasm* toolbox of Sjstrand et al. (2012) with $\mathbf{A}$.
$\mathcal{H}_{\text{max},r}$	The r largest entries of the principal components for the (ℓ_1, ℓ_2)-sampled sketch $\tilde{\mathbf{A}}$.
$\mathcal{H}_{\text{sp},r}$	r-sparse components using *Spasm* with the (ℓ_1, ℓ_2)-sampled sketch $\tilde{\mathbf{A}}$.
$\mathcal{U}_{\text{max},r}$	The r largest entries of the principal components for the *uniformly* sampled sketch $\tilde{\mathbf{A}}$.
$\mathcal{U}_{\text{sp},r}$	r-sparse components using *Spasm* with the uniformly sampled sketch $\tilde{\mathbf{A}}$.

Output of an algorithm $\mathcal{Z}$ is sparse principal components $\mathbf{V}$, and we consider the metric $f(\mathcal{Z}) = \text{trace}(\mathbf{V}^T\mathbf{A}^T\mathbf{A}\mathbf{V})$, where $\mathbf{A}$ is the original centered data. We consider the following statistics.

$\frac{f(\mathcal{G}_{\text{max},r})}{f(\mathcal{G}_{\text{sp},r})}$	Relative loss of greedy thresholding versus *Spasm*, illustrating the value of a good sparse PCA algorithm. Our sketch based algorithms *do not* address this loss.
$\frac{f(\mathcal{H}_{\text{max/sp},r})}{f(\mathcal{G}_{\text{max/sp},r})}$	Relative loss of using the (ℓ_1, ℓ_2)-sketch $\tilde{\mathbf{A}}$ instead of complete data $\mathbf{A}$. A ratio close to 1 is desired.
$\frac{f(\mathcal{U}_{\text{max/sp},r})}{f(\mathcal{G}_{\text{max/sp},r})}$	Relative loss of using the uniform sketch $\tilde{\mathbf{A}}$ instead of complete data $\mathbf{A}$. A benchmark to highlight the value of a good sketch.

We also report the computation time for the algorithms. We show results to confirm that sparse PCA algorithms using the (ℓ_1, ℓ_2)-sketch are nearly comparable to those same algorithms on the complete data; and, gain in computation time from sparse sketch is proportional to the sparsity.

Also, we evaluate the quality of Algorithm 4 to estimate α^* from a small number of samples from various data.

3.4 Other Sketching Probabilities

We consider p_{ij} proportional to the sum of row and column *leverage scores* of $\mathbf{A}$ (Chen et al. 2014) to produce sparse sketch $\tilde{\mathbf{A}}$ using Algorithm 1. Let $\mathbf{A}$ be an $m \times n$ matrix of rank ϱ, and its SVD is given by $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ contains ϱ orthonormal left and right singular vectors, respectively, and $\mathbf{\Sigma}$ is the diagonal matrix of singular values. We define leverage scores as follows (as in Mahoney and Drineas 2009)

$$\begin{aligned} \text{(row leverage)} \quad \mu_i &= \|\mathbf{U}_{(i)}\|_2^2, \quad \forall i \in [m] \\ \text{(column leverage)} \quad \nu_j &= \|\mathbf{V}_{(j)}\|_2^2, \quad \forall j \in [n] \end{aligned}$$

and element-wise leverage score (similar to Chen et al. 2014)

$$p_{lev} = \frac{1}{2}\frac{\mu_i + \nu_j}{(m+n)\varrho} + \frac{1}{2mn}, \quad i \in [m], j \in [n].$$

Note that p_{lev} is a probability distribution on the elements of $\mathbf{A}$. At a high level, the leverage scores of an element (i, j) is proportional to the squared norms of the i-th row of the left singular matrix and the j-th row of the right singular matrix. The constant term in p_{lev} prevents the rescaled values from blowing up in case of sampling an element with tiny leverage score. Such leverage score sampling is distinct from uniform sampling only for *low-rank* matrices or *low-rank approximations* to matrices. So, we consider sparsification of low-rank and rank-truncated data, and compare the quality of sparse sketches produced from such low-rank matrices using our optimal hybrid sampling with that of using p_{lev}.

	m	n	$\|\mathbf{A}\|_0$	nd	$\text{rs}_0 > \text{rs}_1$	α^*
$\mathbf{A}_{0.1}$	500	500	2.5e+5	9.2e+4	no	.63
TTC1	139	15170	37831	12204	yes	1
TTC2	138	11859	29334	9299	yes	1
TTC3	125	15485	47304	14201	yes	1
TTC4	125	14392	35018	10252	yes	1
Digit	2313	256	5.9e+5	5.1e+5	no	.20
Stock	1218	7956	5.5e+6	6.5e+3	no	.74
Gene	107	22215	2.4e+6	2.25e+6	yes	.99

Table 1: α^* for various $m \times n$ data sets ($\epsilon = .05$ is the desired relative-error accuracy). $\text{rs}_0 > \text{rs}_1$ implies ℓ_1 is better than ℓ_2 (Achlioptas et al. 2013a), and we reproduce this ($\alpha^* = 1$). But, for $\text{rs}_0 \leq \text{rs}_1$, $\alpha^* < 1$ and our hybrid sampling is strictly better than both ℓ_1 and ℓ_2.

3.5 Description of Data

Synthetic Data: $\mathbf{A}_{0.1}$ as described in SECTION 1.4 (FIGURE 1).

TechTC Datasets (TTC): (Gabrilovich and Markovitch 2004) These datasets are bag-of-words features for document-term data describing two topics (ids). We choose four such datasets. Rows represent documents and columns are the words. We preprocessed the data by removing all the words of length four or smaller, and then divide each row by its Frobenius norm.

Digit Data: (Hull 1994) A data set of three handwritten digits: six (664 samples), nine (644 samples), and one (1005 samples). Pixels are treated as features, and pixel values are normalized in [-1,1]. Each 16×16 digit image is encoded to form a row in the data matrix (2313 rows and 256 columns).

Stock Data (S&P): We use a temporal data containing 1218 stocks (rows) collected between 1983 and 2011. This temporal data set has with 7056 snapshots (columns) of prices.

Gene Expression Data: We use GSE10072 gene expression data for lung cancer from NCBI Gene Expression Omnibus database. There are 107 samples (58 lung tumor cases and 49 normal lung controls) forming the rows of the matrix, with 22,215 probes (features) from GPL96 platform annotation table.

3.6 Results

All the random results are based on mean of several independent trials (small variance observed).

3.6.1 QUALITY OF SPARSE SKETCH

Table 1 summarizes α^* for various data sets. Achlioptas et al. (2013a) argued that, for $\text{rs}_0 > \text{rs}_1$, ℓ_1 sampling is better than ℓ_2 (even with truncation). Our results on α^* in Table 1 reproduce this condition ($\alpha^* = 1$ implies ℓ_1). Moreover, our method can derive the right blend of ℓ_1 and ℓ_2 sampling even when the above condition fails. In this sense, we generalize the results of Achlioptas et al. (2013a).

	$s/(k(m+n))$	$\hat{\alpha}$
$\mathbf{A}_{0.1}$, $k=5$	2	0.7
	3	0.7
Digit, $k=3$	3	0.3
	5	0.3
Stock, $k=1$	2	0.7
	3	0.7

Table 2: $\hat{\alpha}$ corresponding to the minimum relative error $\|\mathbf{A}-\tilde{\mathbf{A}}\|_2/\|\mathbf{A}\|_2$, for a fixed budget of samples size s. $\hat{\alpha} \approx \alpha^*$ in Table 1, indicating that our optimal hybrid sampling produces strictly better sparse sketch than both ℓ_1 and ℓ_2. $k \approx \|\mathbf{A}\|_F^2/\|\mathbf{A}\|_2^2$ (stable rank).

Also, we note the value of α [4] for the minimum relative error $\frac{\|\mathbf{A}-\tilde{\mathbf{A}}\|_2}{\|\mathbf{A}\|_2}$, for a fixed sample size s in Table 2. Table 2 shows that the quality of the sparse sketch produced by our optimal hybrid sampling is strictly better than that of both ℓ_1 and ℓ_2.

• *Comparison with ℓ_2-with-truncation:* We also compare our optimal hybrid sampling with ℓ_2 sampling with truncation. We use two predetermined truncation parameters, $\varepsilon = 0.1$ and $\varepsilon = 0.01$. First, ℓ_2 sampling without truncation turns out to produce the worst quality sparse sketches for all datasets. For real datasets, hybrid-(ℓ_1, ℓ_2) sampling using α^* outperforms ℓ_2 with truncation $\varepsilon = 0.1$ and $\varepsilon = 0.01$. For $\mathbf{A}_{0.1}$, ℓ_2 with $\varepsilon = 0.01$ appears to produce $\tilde{\mathbf{A}}$ that is as bad as ℓ_2 without truncation. However, ℓ_2 with $\varepsilon = 0.1$ shows better quality than optimal hybrid sampling (for $\mathbf{A}_{0.1}$ only) as this ε filters out the noisy elements. We should point out that we know a good ε for $\mathbf{A}_{0.1}$ as we control the noise in this case. However, in reality, we have no control over the noise, and choosing the right ε with no prior knowledge of the data, is an improbable outcome.

3.6.2 QUALITY OF PCA FROM SPARSE SKETCHES

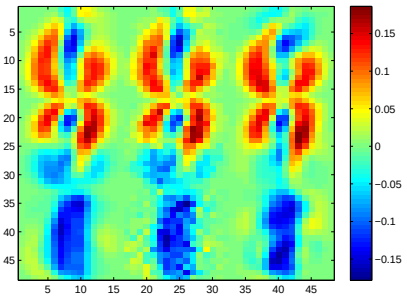
We investigate the quality of Algorithm 2 for Digit data and $\mathbf{A}_{0.1}$. We set $r = 30k$ for the random projection matrix $\mathbf{A}_G$ to achieve a comparable runtime of $\mathcal{G}$ with $\mathcal{H}$ (k is 3 and 5 for Digit and $\mathbf{A}_{0.1}$, respectively). FIGURE 4 shows that $\mathcal{H}$ has better quality than $\mathcal{G}$ for Digit data. Also, Table 3 lists the gain in computation time for Algorithm 2 due to sparsification (using hybrid sampling with α^*). Finally, our optimal hybrid sampling outperforms ℓ_1 and Bernstein sampling (Achlioptas et al. 2013b) by preserving more variance of data via PCA (Table 4).

3.6.3 QUALITY OF SPARSE PCA ALGORITHMS

We report results for primarily the top principal component ($k = 1$) which is the case most considered in the literature. When $k > 1$, our results do not qualitatively change. We note the optimal mixing parameter α^* using Algorithm 1 for various datasets in Table 5.

Handwritten Digits. We sample approximately 7% of the elements from the centered data using (ℓ_1, ℓ_2)-sampling, as well as uniform sampling. The quality of solution for small r is shown in Table 5, including the running time τ. For this data, $f(\mathcal{G}_{\text{max},r})/f(\mathcal{G}_{\text{sp},r}) \approx 0.23$ ($r = 10$), so it is important to use a good sparse PCA algorithm. We see from Table 5 that the (ℓ_1, ℓ_2)-sketch

4. We vary α from 0.1 to 1 with step size 0.1.



Hybrid-(ℓ_1, ℓ_2) σ_h/σ_k	Random Projection σ_g/σ_k
97.94%	96.89%

Figure 4: [Digit data] Approximation quality of PCA (Algorithm 2) from samples of data. (Left) Visualization of principal components as 16×16 image. Principal components are ordered from the top row to the bottom. First column of PCA's are exact ($\mathcal{A}$). Second column of PCA's ($\mathcal{H}$) are computed on sparse sketch with 7% non-zeros of all the elements via hybrid sampling using optimal α. Third column of PCA's ($\mathcal{G}$) are computed on $\mathbf{A}_G$. (Right) Variance preserved by PCA algorithms. Hybrid sampling based PCA, despite using less information, is better than PCA of random projection.

	Sparsified Digit	Sparsified $\mathbf{A}_{0.1}$
# non-zeros	7%	6%
$t_a/t_h/t_G$	151/30/36	73/18/36

Table 3: Computational gain of Algorithm 2 comparing to exact PCA. We report the computation time of MATLAB function 'svds($\mathbf{A}$,k)' for actual data $\mathbf{A}$ (t_a), sparsified data $\tilde{\mathbf{A}}$ (t_h), and random projection data $\mathbf{A}_G$ (t_G). We use only 7% and 6% of all the elements of Digit data and $\mathbf{A}_{0.1}$, respectively, to construct respective sparse sketches. $k = 3$ for Digit and $k = 5$ for $\mathbf{A}_{0.1}$.

significantly outperforms the uniform sketch. A more extensive comparison of recovered variance is given in FIGURE 6(a). We also observe a speed-up of a factor of about 6 for the (ℓ_1, ℓ_2)-sketch. We point out that the uniform sketch is reasonable for the digits data because most data elements are close to either $+1$ or -1, since the pixels are either black or white.

We show a visualization of the principal components in FIGURE 5. We observe that the sparse components from the (ℓ_1, ℓ_2)-sketch are almost identical to that of from the complete data.

TechTC Data. We sample approximately 5% of the elements from the centered data using our (ℓ_1, ℓ_2)-sampling, as well as uniform sampling. For this data, $f(\mathcal{G}_{\max,r})/f(\mathcal{G}_{\text{sp},r}) \approx 0.84$ ($r = 10$). We observe a very significant quality difference between the (ℓ_1, ℓ_2)-sketch and uniform sketch. A more extensive comparison of recovered variance is given in FIGURE 6(b). We also observe a speed-up of a factor of about 6 for the (ℓ_1, ℓ_2)-sketch. Unlike the digits data which is uniformly near ± 1, the text data is "spikey" and now it is important to sample with a bias toward larger elements, which is why the uniform-sketch performs very poorly.

	ℓ_1 σ_{ℓ_1}/σ_k	Bernstein σ_b/σ_k	Hybrid-(ℓ_1,ℓ_2) σ_h/σ_k
Digit ($\alpha^* = 0.42$)	98.47 %	98.46 %	**98.51**%
Stock ($\alpha^* = 0.1$)	85.09 %	95.61 %	**97.07**%

Table 4: Variance preserved by PCA of various sparse sketches. Our optimal hybrid sampling outperforms other sampling methods including Bernstein sampling (Achlioptas et al. 2013b). We sample 9% non-zeros from digit data and 2% non-zeros from stock data.

Centered Data	α^*	r	$\frac{f(\mathcal{H}_{\text{max/ sp},r})}{f(\mathcal{G}_{\text{max/ sp},r})}$	$\frac{\tau(\mathcal{G})}{\tau(\mathcal{H})}$	$\frac{f(\mathcal{U}_{\text{max/ sp},r})}{f(\mathcal{G}_{\text{max/ sp},r})}$	$\frac{\tau(\mathcal{G})}{\tau(\mathcal{U})}$
Digit	.42	40	0.99/ 0.90	6.21	1.01/ 0.70	5.33
TTC1	1	40	0.94/ 0.99	5.70	0.41/ 0.38	5.96
Stock	.10	40	1.00/ 1.00	3.72	0.66/ 0.66	4.76
Gene	.82	40	0.82/ 0.88	3.61	0.65/ 0.15	2.53

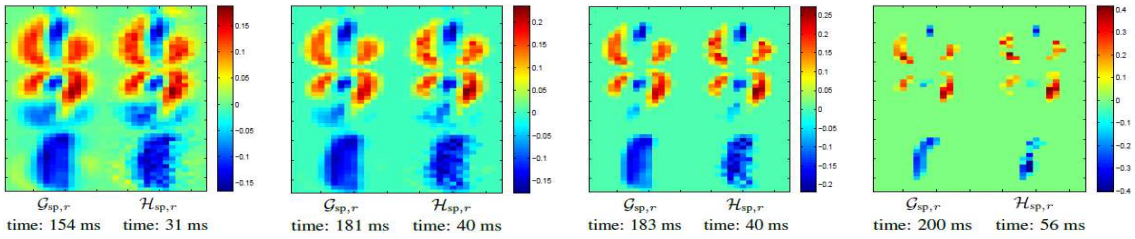
Table 5: Comparison of sparse principal components from the (ℓ_1,ℓ_2)-sketch and uniform sketch. Recall, $\mathcal{G}$ is the ground truth.

As a final comparison, we look at the actual sparse top component with sparsity parameter $r = 10$. The topic IDs in the TechTC data are 10567="*US: Indiana: Evansville*" and 11346="*US: Florida*". The top-10 features (words) in the full PCA on the complete data are shown in Table 6. In Table 7 we show which words appear in the top sparse principal component with sparsity $r = 10$ using various sparse PCA algorithms. We observe that the sparse PCA from the (ℓ_1,ℓ_2)-sketch with only 5% of the data sampled matches quite closely with the same sparse PCA algorithm using the complete data ($\mathcal{G}_{\text{max/sp},r}$ matches $\mathcal{H}_{\text{max/sp},r}$).

Stock Data. We sample about 2% of the non-zero elements from the centered data using the (ℓ_1,ℓ_2)-sampling, as well as uniform sampling. For this data, $f(\mathcal{G}_{\text{max},r})/f(\mathcal{G}_{\text{sp},r}) \approx 0.96$ ($r = 10$). We observe a very significant quality difference between the (ℓ_1,ℓ_2)-sketch and uniform sketch. A more extensive comparison of recovered variance is given in FIGURE 6(c). We also observe a speed-up of a factor of about 4 for the (ℓ_1,ℓ_2)-sketch. Similar to TechTC data this data set is also "spikey", so biased sampling toward larger elements significantly outperforms the uniform-sketch.

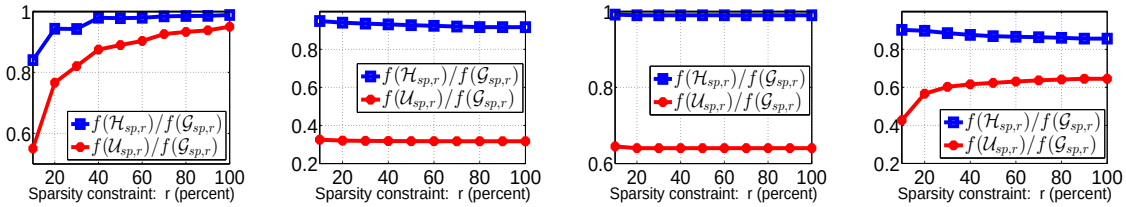
We now look at the actual sparse top component with sparsity parameter $r = 10$. The top-10 features (stocks) in the full PCA on the complete data are shown in Table 8. In Table 9 we show which stocks appear in the top sparse principal component using various sparse PCA algorithms. We observe that the sparse PCA from the (ℓ_1,ℓ_2)-sketch with only 2% of the non-zero elements sampled matches quite closely with the same sparse PCA algorithm using the complete data ($\mathcal{G}_{\text{max/sp},r}$ matches $\mathcal{H}_{\text{max/sp},r}$).

Gene Expression Data. We sample about 9% of the elements from the centered data using the (ℓ_1,ℓ_2)-sampling, as well as uniform sampling. For this data, $f(\mathcal{G}_{\text{max},r})/f(\mathcal{G}_{\text{sp},r}) \approx 0.05$ ($r = 10$) which means a good sparse PCA algorithm is imperative. We observe a very significant quality difference between the (ℓ_1,ℓ_2)-sketch and uniform sketch. A more extensive comparison of re-



(a) $r = 100\%$ (b) $r = 50\%$ (c) $r = 30\%$ (d) $r = 10\%$

Figure 5: [Digits] Visualization of top-3 sparse principal components. In each figure, left panel shows $\mathcal{G}_{\text{sp},r}$ and right panel shows $\mathcal{H}_{\text{sp},r}$. r is the maximum number of non-zeros allowed.



(a) Digit (b) TechTC (c) Stock (d) Gene

Figure 6: Quality of sparse PCA for (ℓ_1, ℓ_2)-sketch and uniform sketch over an extensive range for the sparsity constraint r. The variance preserved by the uniform sketch is significantly lower, highlighting the importance of a good sketch.

covered variance is given in FIGURE 6(d). We also observe a speed-up of a factor of about 4 for the (ℓ_1, ℓ_2)-sketch. Similar to TechTC data this data set is also "spikey", and consequently biased sampling toward larger elements significantly outperforms the uniform-sketch.

Also, we look at the actual sparse top component with sparsity parameter $r = 10$. The top-10 features (probes) in the full PCA on the complete data are shown in Table 10. In Table 11 we show which probes appear in the top sparse principal component with sparsity $r = 10$ using various sparse PCA algorithms. We observe that the sparse PCA from the (ℓ_1, ℓ_2)-sketch with only 9% of the elements sampled matches reasonably with the same sparse PCA algorithm using the complete data ($\mathcal{G}_{\text{max/sp},r}$ matches $\mathcal{H}_{\text{max/sp},r}$).

Finally, we validate the genes corresponding to the top probes in the context of lung cancer. Table 12 lists the top twelve gene symbols in Table 10. Note that a gene can occur multiple times in principal component as genes can be associated with different probes. Genes like SFTPC, AGER, WIF1, and FABP4 are down-regulated in lung cancer, while SPP1 is up-regulated (see the functional gene grouping: `www.sabiosciences.com/rt_pcr_product/HTML/PAHS-134Z.html`). Co-expression analysis on the set of eight genes for $\mathcal{H}_{\text{max},r}$ and $\mathcal{H}_{\text{sp},r}$ using the tool ToppFun (`toppgene.cchmc.org/`) shows that all eight genes appear in a list of selected probes characterizing non-small-cell lung carcinoma (NSCLC) in (Hou et al., 2010, Table

ID	Top 10 in $\mathcal{G}_{\max,r}$	ID	Other words
1	**evansville**	11	service
2	**florida**	12	small
3	**south**	13	frame
4	**miami**	14	tours
5	**indiana**	15	faver
6	**information**	16	transaction
7	**beach**	17	needs
8	**lauderdale**	18	commercial
9	**estate**	19	bullet
10	**spacer**	20	inlets
		21	producer

Table 6: [TechTC] Top ten words in top principal component of the complete data (the other words are discovered by some of the sparse PCA algorithms).

$\mathcal{G}_{\max,r}$	$\mathcal{H}_{\max,r}$	$\mathcal{U}_{\max,r}$	$\mathcal{G}_{\text{sp},r}$	$\mathcal{H}_{\text{sp},r}$	$\mathcal{U}_{\text{sp},r}$
1	1	6	1	1	6
2	2	14	2	2	14
3	3	15	3	3	15
4	4	16	4	4	16
5	5	17	5	5	17
6	7	7	6	7	7
7	6	18	7	8	18
8	8	19	8	6	19
9	11	20	9	12	20
10	12	21	13	11	21

Table 7: [TechTC] Relative ordering of the words (w.r.t. $\mathcal{G}_{\max,r}$) in the top sparse principal component with sparsity parameter $r = 10$.

ID	Top 10 in $\mathcal{G}_{\max,r}$	ID	Other stocks
1	**T.2**	11	HET.
2	**AIG**	12	ONE.1
3	**C**	13	MA
4	**UIS**	14	XOM
5	**NRTLQ**	15	PHA.1
6	**S.1**	16	CL
7	**GOOG**	17	WY
8	**MTLQQ**		
9	**ROK**		
10	**EK**		

Table 8: [Stock data] Top ten stocks in top principal component of the complete data (the other stocks are discovered by some of the sparse PCA algorithms).

$\mathcal{G}_{\max,r}$	$\mathcal{H}_{\max,r}$	$\mathcal{U}_{\max,r}$	$\mathcal{G}_{\text{sp},r}$	$\mathcal{H}_{\text{sp},r}$	$\mathcal{U}_{\text{sp},r}$
1	1	2	1	1	2
2	2	11	2	2	11
3	3	12	3	3	12
4	4	13	4	4	13
5	5	14	5	5	14
6	6	3	6	7	3
7	7	15	7	6	15
8	9	9	8	8	9
9	8	16	9	9	16
10	11	17	10	11	17

Table 9: [Stock data] Relative ordering of the stocks (w.r.t. $\mathcal{G}_{\max,r}$) in the top sparse principal component with sparsity parameter $r = 10$.

S1). Further, AGER and FAM107A appear in the *top five* highly discriminative genes in (Hou et al., 2010, Table S3). Additionally, AGER, FCN3, SPP1, and ADH1B appear among the 162 most differentiating genes across two subtypes of NSCLC and normal lung cancer in (Dracheva et al., 2007, Supplemental Table 1). Such findings show that our method can identify, from incomplete data, important genes for complex diseases like cancer. Also, notice that our sampling-based method is able to identify additional important genes, such as, FCN3 and FAM107A in top ten genes.

• *Quality of Other Sketches:* We briefly report on other options for sketching $\mathbf{A}$. We consider suboptimal α (not α^* from (10) to construct a suboptimal hybrid distribution, and use this in proto-Algorithm 1 to construct a sparse sketch. FIGURE 7 reveals that a good sketch using the α^* is important.

ID	**Top 10 in $\mathcal{G}_{\mathbf{max},r}$**	ID	Other probes
1	**210081_at**	11	205866_at
2	**214387_x_at**	12	209074_s_at
3	**211735_x_at**	13	205311_at
4	**209875_s_at**	14	216379_x_at
5	**205982_x_at**	15	203571_s_at
6	**215454_x_at**	16	205174_s_at
7	**209613_s_at**	17	204846_at
8	**210096_at**	18	209116_x_at
9	**204712_at**	19	202834_at
10	**203980_at**	20	209425_at
		21	215356_at
		22	221805_at
		23	209942_x_at
		24	218450_at
		25	202508_s_at

Table 10: [Gene data] Top ten probes in top principal component of the complete data (the other probes are discovered by some of the sparse PCA algorithms).

$\mathcal{G}_{\text{max},r}$	$\mathcal{H}_{\text{max},r}$	$\mathcal{U}_{\text{max},r}$	$\mathcal{G}_{\text{sp},r}$	$\mathcal{H}_{\text{sp},r}$	$\mathcal{U}_{\text{sp},r}$
1	4	13	1	4	13
2	1	14	2	1	16
3	11	3	3	2	15
4	2	15	4	11	19
5	3	5	5	3	20
6	8	16	6	8	21
7	7	6	7	7	22
8	9	17	8	9	23
9	5	4	9	5	24
10	12	18	10	12	25

Table 11: [Gene data] Relative ordering of the probes (w.r.t. $\mathcal{G}_{\text{max},r}$) in the top sparse principal component with sparsity parameter $r = 10$.

$\mathcal{G}_{\text{max},r}$	ν	$\mathcal{H}_{\text{max},r}$	ν	$\mathcal{H}_{\text{sp},r}$	ν
SFTPC	4	SFTPC	3	SFTPC	3
AGER	1	SPP1	1	SPP1	1
SPP 1	1	AGER	1	AGER	1
ADH1B	1	FCN3	1	FCN3	1
CYP4B1	1	CYP4B1	1	CYP4B1	1
WIF1	1	ADH1B	1	ADH1B	1
FABP4	1	WIF1	1	WIF1	1
		FAM107A	1	FAM107A	1

Table 12: [Gene data] Gene symbols corresponding to top probes in Table 11. One gene can be associated with multiple probes. Here ν is the frequency of occurrence of a gene in top ten probes of their respective principal component.

3.6.4 ESTIMATE OF α^* FROM SAMPLES

Table 13 shows the quality of estimated α^* computed by Algorithm 4 using a small number of observed data elements. Note that most of the elements of Digit data have magnitude close to 1; consequently we need more samples to estimate α^* accurately.

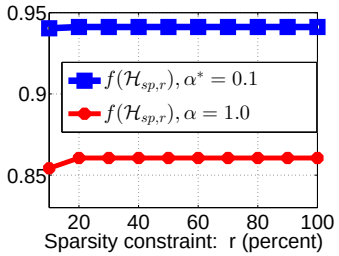


Figure 7: [Stock data] Quality of sketch using *suboptimal* α to illustrate the importance of the optimal mixing parameter α^*.

	Non-zeros sampled	Estimated α^*
Digit, $\alpha^* = 0.42$	6 %	0.80
	19 %	0.67
Stock, $\alpha^* = 0.1$	2 %	0.1
	10 %	0.1

Table 13: Estimated α^* using Algorithm 4 from small number of non-zeros of Digit and Stock data. Most of elements of Digit data are close to either $+1$ or -1, thus requiring more samples to accurately estimate α^*.

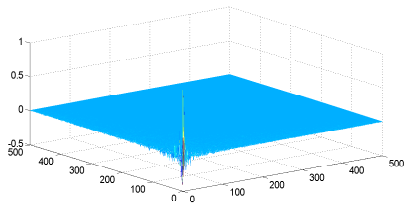
3.6.5 OTHER SKETCHING PROBABILITIES

We compare the quality of sparse sketches produced from low-rank and rank-truncated synthetic and real data using our optimal hybrid sampling with p_{lev}.

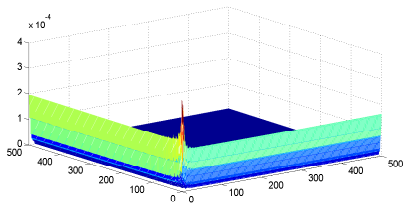
• *Power-law Matrix:* First, we construct a 500×500 low-rank *power-law* matrix, similar to Chen et al. (2014), as follows: $\mathbf{A}_{pow} = \mathbf{D}\mathbf{X}\mathbf{Y}^T\mathbf{D}$, where, matrices $\mathbf{X}$ and $\mathbf{Y}$ are 500×5 i.i.d. Gaussian $\mathcal{N}(0,1)$ and $\mathbf{D}$ is a diagonal matrix with power-law decay, $\mathbf{D}_{ii} = i^{-\gamma}$, $1 \leq i \leq 500$. The parameter γ controls the 'incoherence' of the matrix (larger γ makes the data more 'spiky').

Further, we construct low-rank approximation by projecting a data set onto a low dimensional subspace. We notice that the datasets projected onto the space spanned by top few principal components preserve the linear structure of the data. For example, Digit data show good separation of digits when projected onto the top three principal components. For TechTC and Gene data the top two respective principal components are good enough to form a low-dimensional subspace where the datasets show reasonable separation of two classes of samples. For the stock data we use top three PCAs because the stable rank is close to 2.

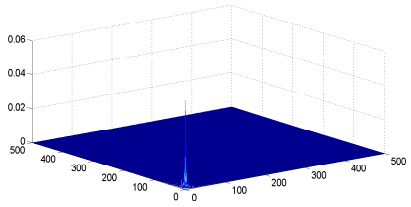
Quality of Sparse Sketch: We measure $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2 / \|\mathbf{A}\|_2$ ($\mathbf{A}$ is the low-rank data) for the above-mentioned two sampling methods using various sample size. Table 14 lists the quality of sparse sketches produced from $\mathbf{A}_{pow}$ via the two sampling methods. Similarly, for all other datasets our optimal hybrid sampling (with α^*) outperforms the leverage score sampling (Table 15) for (low-rank) matrix sparsification.



(a) Low-rank data $\mathbf{A}_{pow}$



(b) Element-wise leverage scores for $\mathbf{A}_{pow}$



(c) Optimal hybrid distribution for $\mathbf{A}_{pow}$

Figure 8: Comparing optimal hybrid-(ℓ_1, ℓ_2) distribution with leverage scores p_{lev} for data $\mathbf{A}_{pow}$ for $\gamma = 1.0$. (a) Structure of $\mathbf{A}_{pow}$, (b) distribution p_{lev}, (c) optimal hybrid-(ℓ_1, ℓ_2) distribution. Our optimal hybrid distribution is more aligned with the structure of the data, requiring much smaller sample size to achieve a given accuracy of sparsification. This is supported by Table 14.

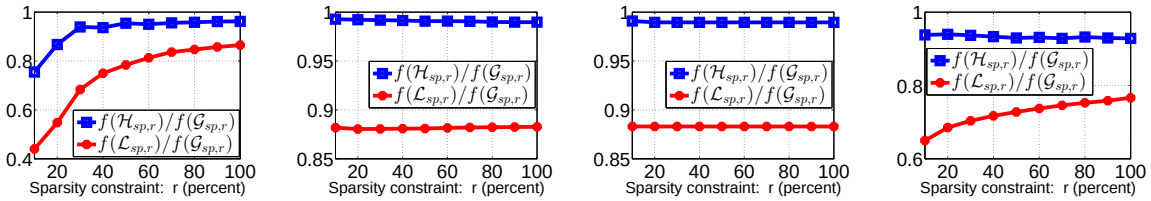
	$\frac{s}{k(m+n)}$	hybrid-(ℓ_1, ℓ_2)	p_{lev}
$\gamma = 0.5$	3	42%	58%
	5	31%	43%
$\gamma = 0.8$	3	15%	43%
	5	12%	40%
$\gamma = 1.0$	3	8%	42%
	5	6%	39%

Table 14: Sparsification quality $\|\mathbf{A}_{pow} - \tilde{\mathbf{A}}_{pow}\|_2 / \|\mathbf{A}_{pow}\|_2$ for low-rank 'power-law' matrix $\mathbf{A}_{pow}$ ($k = 5$). We compare the quality of hybrid-(ℓ_1, ℓ_2) sampling and leverage score sampling for two sample size. We note (mean) α^* of hybrid-(ℓ_1, ℓ_2) distribution for data $\mathbf{A}_{pow}$ using $\epsilon = 0.05, \delta = 0.1$. For $\gamma = 0.5, 0.8, 1.0$, we have mean $\alpha^* = 0.11, 0.72, 0.8$, respectively, with very small variance.

We notice that with increasing γ leverage scores get more aligned with the structure of the data $\mathbf{A}_{pow}$ resulting in gradually improving approximation quality, for the same sample size. Larger γ produces more variance in data elements. ℓ_2 component of our hybrid distribution bias us towards the larger data elements, while ℓ_1 works as a regularizer to maintain the variance of the sampled

	$\frac{s}{k(m+n)}$	Hybrid-(ℓ_1, ℓ_2)	p_{lev}
Digit, $k = 3$	3	44%	61%
	5	34%	47%
$\mathbf{A}_{0.1}$, $k = 5$	3	25%	80%
	5	21%	62%

Table 15: Sparsification of rank-truncated data. We compare the quality of $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2 / \|\mathbf{A}\|_2$ for hybrid-(ℓ_1, ℓ_2) sampling and leverage score sampling using two different sample size.



(a) Digit (rank 3) (b) TechTC (rank 2) (c) Stock (rank 3) (d) Gene (rank 2)

Figure 9: [Low-rank data] Quality of sparse PCA of low-rank data for optimal (ℓ_1, ℓ_2)-sketch and leverage score sketch over an extensive range for the sparsity constraint r. The quality of the optimal hybrid sketch is considerably better highlighting the importance of a good sketch.

(and rescaled) elements. With increasing γ we need more regularization to counter the problem of rescaling. Interestingly, our optimal parameter α^* adapts itself with this changing structure of data, e.g. for $\gamma = 0.5, 0.8, 1.0$, we have $\alpha^* = 0.11, 0.72, 0.8$, respectively. This shows the benefit of our parameterized hybrid distribution to achieve a superior approximation quality. FIGURE 8 shows the structure of the data $\mathbf{A}_{pow}$ for $\gamma = 1.0$ along with the optimal hybrid-(ℓ_1, ℓ_2) distribution and leverage score distribution p_{lev}. The figure suggests our optimal hybrid distribution is better aligned with the structure of the data, resulting in a superior sparse sketch of the data. Similar results are observed for other datasets as well.

Quality of Sparse PCA: Let $\mathcal{L}_{\text{sp},r}$ be the r-sparse components using *Spasm* for the leverage score sampled sketch $\tilde{\mathbf{A}}$. FIGURE 9 shows that leverage score sampling is not as effective as the optimal hybrid (ℓ_1, ℓ_2)-sampling for sparse PCA of low-rank data.

3.7 Conclusion

It is possible to use a sparse sketch (incomplete data) to recover nearly as good principal components and sparse principal components as one would have gotten with the complete data. We mention that, while $\mathcal{G}_{\text{max}}$ which uses the largest weights in the unconstrained PCA does not perform well with respect to the variance, it does identify good features. A simple enhancement to $\mathcal{G}_{\text{max}}$ is to recalibrate the sparse component after identifying the features - this is an unconstrained PCA problem on just

the columns of the data matrix corresponding to the features. This method of recalibrating can be used to improve any sparse PCA algorithm.

Our algorithms are simple and efficient, and many interesting avenues for further research remain. Can the sampling complexity for the top-k sparse PCA be reduced from $O(k^2)$ to $O(k)$. We suspect that this should be possible by getting a better bound on $\sum_{i=1}^{k} \sigma_i(\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}})$; we used the crude bound $k\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_2$. We also presented a general surrogate optimization bound which may be of interest in other applications. In particular, it is pointed out in Magdon-Ismail and Boutsidis (2016) that though PCA optimizes variance, a more natural way to look at PCA is as the linear projection of the data that minimizes the *information loss*. Magdon-Ismail and Boutsidis (2016) gives efficient algorithms to find sparse linear dimension reduction that minimizes information loss – the information loss of sparse PCA can be considerably higher than optimal. To minimize information loss, the objective to maximize is $f(\mathbf{V}) = \text{trace}(\mathbf{A}^T\mathbf{A}\mathbf{V}(\mathbf{A}\mathbf{V})^\dagger\mathbf{A})$. It would be interesting to see whether one can recover sparse low-information-loss linear projectors from incomplete data.

Overall, the experimental results demonstrate the quality of the algorithms presented here, indicating the superiority of our approach to other extreme choices of element-wise sampling methods, such as, ℓ_1 and ℓ_2 sampling. Also, we demonstrate the theoretical and practical usefulness of hybrid-(ℓ_1, ℓ_2) sampling for fundamental data analysis tasks such as recovering PCA and sparse PCA from sketches. Finally, our sampling scheme outperforms element-wise leverage scores for the sparsification of various *low-rank* synthetic and real data.

Acknowledgments

PD was supported by NSF IIS-1661760. AK was partially supported by NSF IIS-1661760. MMI was supported in part by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (the ARL Network Science CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

Appendix A. Proof of Theorem 2

In this section we provide a proof of Theorem 2 following the proof outline of Drineas and Zouzias (2011); Achlioptas et al. (2013a). We use Lemma 1 as our main tool. For all $t \in [s]$ we define the matrix $\mathbf{M}_t \in \mathbb{R}^{m\times n}$ as follows:

$$\mathbf{M}_t = \frac{\mathbf{A}_{i_t j_t}}{p_{i_t j_t}} \mathbf{e}_{i_t} \mathbf{e}_{j_t}^T - \mathbf{A}.$$

It now follows that $\frac{1}{s}\sum_{t=1}^{s} \mathbf{M}_t = S_\Omega(\mathbf{A}) - \mathbf{A}$. We can bound $\|\mathbf{M}_t\|_2$ for all $t \in [s]$. We define the following quantity:

$$\lambda = \frac{\|\mathbf{A}\|_1 \cdot |\mathbf{A}_{ij}|}{\|\mathbf{A}\|_F^2}, \text{ for } \mathbf{A}_{ij} \neq 0 \tag{24}$$

Lemma 8 *Using our notation, and using probabilities of the form (3), for all $t \in [s]$,*

$$\|\mathbf{M}_t\|_2 \leq \max_{\substack{i,j:\\ \mathbf{A}_{ij}\neq 0}} \frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha)\lambda} + \|\mathbf{A}\|_2.$$

Proof Using probabilities of the form (3), and because $\mathbf{A}_{ij} = 0$ is never sampled,

$$\begin{aligned} \|\mathbf{M}_t\|_2 &= \|(\mathbf{A}_{i_t j_t}/p_{i_t j_t})\mathbf{e}_{i_t}\mathbf{e}_{j_t}^T - \mathbf{A}\|_2 \\ &\leq \max_{\substack{i,j:\\ \mathbf{A}_{ij}\neq 0}} \left\{ \left(\frac{\alpha}{\|\mathbf{A}\|_1} + \frac{(1-\alpha)\cdot|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_F^2} \right)^{-1} \right\} + \|\mathbf{A}\|_2, \end{aligned}$$

Using (24), we obtain the bound. ■

Next we bound the spectral norm of the expectation of $\mathbf{M}_t\mathbf{M}_t^T$.

Lemma 9 *Using our notation, and using probabilities of the form (3), for all $t \in [s]$,*

$$\|\mathbb{E}(\mathbf{M}_t\mathbf{M}_t^T)\|_2 \leq \|\mathbf{A}\|_F^2\beta_1 - \sigma^2,$$

$$\beta_1 = \max_i \sum_{j=1}^{n} \left(\frac{\alpha\cdot\|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1} + (1-\alpha) \right)^{-1},$$

for $\mathbf{A}_{ij} \neq 0$.

Proof Recall that $\mathbf{A} = \sum_{i,j=1}^{m,n} \mathbf{A}_{ij}\mathbf{e}_i\mathbf{e}_j^T$ and $\mathbf{M}_t = \frac{\mathbf{A}_{i_t j_t}}{p_{i_t j_t}}\mathbf{e}_{i_t}\mathbf{e}_{j_t}^T - \mathbf{A}$. We derive

$$\mathbb{E}[\mathbf{M}_t\mathbf{M}_t^T] = \sum_{i,j=1}^{m,n} (\mathbf{A}_{ij}^2/p_{ij})\mathbf{e}_i\mathbf{e}_i^T - \mathbf{A}\mathbf{A}^T.$$

Sampling according to probabilities of (3), and because $\mathbf{A}_{ij} = 0$ is never sampled, we get,

$$\begin{aligned} \sum_{i,j=1}^{m,n} \frac{\mathbf{A}_{ij}^2}{p_{ij}} &= \|\mathbf{A}\|_F^2 \sum_{i,j=1}^{m,n} \left(\frac{\alpha\cdot\|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1} + (1-\alpha) \right)^{-1}, \\ &\leq \|\mathbf{A}\|_F^2 \sum_{i=1}^{m} \max_i \sum_{j=1}^{n} \left(\frac{\alpha\cdot\|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1} + (1-\alpha) \right)^{-1}. \end{aligned}$$

for $\mathbf{A}_{ij} \neq 0$. Thus,

$$\mathbb{E}[\mathbf{M}_t \mathbf{M}_t^T] \quad \preceq \quad \|\mathbf{A}\|_F^2 \beta_1 \sum_{i=1}^{m} \mathbf{e}_i \mathbf{e}_i^T - \mathbf{A}\mathbf{A}^T = \|\mathbf{A}\|_F^2 \beta_1 \mathbf{I}_m - \mathbf{A}\mathbf{A}^T.$$

Note that, $\|\mathbf{A}\|_F^2 \beta_1 \mathbf{I}_m$ is diagonal with all entries non-negative, and $\mathbf{A}\mathbf{A}^T$ is postive semi-definite. Therefore,

$$\|\mathbb{E}[\mathbf{M}_t \mathbf{M}_t^T]\|_2 \leq \|\mathbf{A}\|_F^2 \beta_1 - \sigma^2.$$

■

Similarly, we can obtain

$$\left\|\mathbb{E}[\mathbf{M}_t^T \mathbf{M}_t]\right\|_2 \leq \|\mathbf{A}\|_F^2 \beta_2 - \sigma^2,$$

where, for $\mathbf{A}_{ij} \neq 0$,

$$\beta_2 = \max_j \sum_{i=1}^{m} \Big(\frac{\alpha \cdot \|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}| \cdot \|\mathbf{A}\|_1} + (1-\alpha) \Big)^{-1}.$$

We can now apply Lemma 1 with

$$\rho^2(\alpha) = \|\mathbf{A}\|_F^2 \max\{\beta_1, \beta_2\} - \sigma^2,$$

and

$$\gamma(\alpha) = \frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha)\lambda} + \|\mathbf{A}\|_2,$$

to conclude that $\|\mathcal{S}_\Omega(\mathbf{A}) - \mathbf{A}\|_2 \leq \varepsilon$ holds subject to a failure probability at most

$$(m+n) \exp\left(\frac{-s\varepsilon^2/2}{\rho^2(\alpha) + \gamma(\alpha)\varepsilon/3} \right).$$

Bounding the failure probability by δ, and setting $\varepsilon = \epsilon \cdot \|\mathbf{A}\|_2$, we complete the proof.

Appendix B. Proof of Lemma 4

Lemma 4 will be a corollary of a more general result, for a class of optimization problems involving a Lipschitz-like objective function over an arbitrary (not necessarily convex) domain. Let $f(\mathbf{V}, \mathbf{X})$ be a function that is defined for a matrix variable $\mathbf{V}$ and a matrix parameter $\mathbf{X}$. The optimization variable $\mathbf{V}$ is in some feasible set $\mathcal{S}$ which is arbitrary. The parameter $\mathbf{X}$ is also arbitrary. We assume that f is locally Lipschitz in $\mathbf{X}$, that is

$$|f(\mathbf{V}, \mathbf{X}) - f(\mathbf{V}, \tilde{\mathbf{X}})| \leq \gamma(\mathbf{X}) \|\mathbf{X} - \tilde{\mathbf{X}}\|_2, \qquad \forall \mathbf{V} \in \mathcal{S}.$$

(Note we allow the "Lipschitz constant" to depend on the fixed matrix $\mathbf{X}$ but not the variables $\mathbf{V}, \tilde{\mathbf{X}}$; this is more general than a globally Lipshitz objective) The next lemma is the key tool we need to prove Lemma 4 and it may be of independent interest in other optimization settings. We are interested in maximizing $f(\mathbf{V}, \mathbf{X})$ with respect to $\mathbf{V}$ to obtain $\mathbf{V}^*$. But, we only have an approximation $\tilde{\mathbf{X}}$ for $\mathbf{X}$, and so we maximize $f(\mathbf{V}, \tilde{\mathbf{X}})$ to obtain $\tilde{\mathbf{V}}^*$, which will be a suboptimal solution with respect to $\mathbf{X}$. We wish to bound $f(\mathbf{V}^*, \mathbf{X}) - f(\tilde{\mathbf{V}}^*, \mathbf{X})$ which quantifies how suboptimal $\tilde{\mathbf{V}}^*$ is with respect to $\mathbf{X}$.

Lemma 10 (Surrogate optimization bound) *Let $f(\mathbf{V}, \mathbf{X})$ be γ-locally Lipschitz w.r.t. $\mathbf{X}$ over the domain $\mathbf{V} \in \mathcal{S}$. Define* $\mathbf{V}^* = \arg\max_{\mathbf{V}\in\mathcal{S}} f(\mathbf{V}, \mathbf{X})$; $\tilde{\mathbf{V}}^* = \arg\max_{\mathbf{V}\in\mathcal{S}} f(\mathbf{V}, \tilde{\mathbf{X}})$. *Then,*

$$f(\mathbf{V}^*, \mathbf{X}) - f(\tilde{\mathbf{V}}^*, \mathbf{X}) \leq 2\gamma(\mathbf{X})\|\mathbf{X} - \tilde{\mathbf{X}}\|_2.$$

In the lemma, the function f and the domain $\mathcal{S}$ are arbitrary. In our setting, $\mathbf{X} \in \mathbb{R}^{n\times n}$, the domain $\mathcal{S} = \{\mathbf{V} \in \mathbb{R}^{n\times k}; \mathbf{V}^T\mathbf{V} = \mathbf{I}_k; \|\mathbf{V}^{(j)}\|_0 \leq r\}$, and $f(\mathbf{V}, \mathbf{X}) = \text{trace}(\mathbf{V}^T\mathbf{X}\mathbf{V})$. We first show that f is Lipschitz w.r.t. $\mathbf{X}$ with $\gamma = k$ (a constant independent of $\mathbf{X}$). Let the representation of $\mathbf{V}$ by its columns be $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$. Then,

$$|\text{trace}(\mathbf{V}^T\mathbf{X}\mathbf{V}) - \text{trace}(\mathbf{V}^T\tilde{\mathbf{X}}\mathbf{V})| = |\text{trace}((\mathbf{X} - \tilde{\mathbf{X}})\mathbf{V}\mathbf{V}^T)| \leq \sum_{i=1}^{k} \sigma_i(\mathbf{X} - \tilde{\mathbf{X}}) \leq k\|\mathbf{X} - \tilde{\mathbf{X}}\|_2,$$

where, $\sigma_i(\mathbf{A})$ is the i-th largest singular value of $\mathbf{A}$ (we used Von-neumann's trace inequality and the fact that $\mathbf{V}\mathbf{V}^T$ is a k-dimensional projection). Now, by Lemma 10, $\text{trace}(\mathbf{V}^{*T}\mathbf{X}\mathbf{V}^*) - \text{trace}(\tilde{\mathbf{V}}^{*T}\mathbf{X}\tilde{\mathbf{V}}^*) \leq 2k\|\mathbf{X} - \tilde{\mathbf{X}}\|_2$. Lemma 4 follows by setting $\mathbf{X} = \mathbf{A}^T\mathbf{A}$ and $\tilde{\mathbf{X}} = \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$.

Appendix C. Proof of Lemma 10

In this section we provide a proof of Lemma 10. First, we need the following lemma.

Lemma 11 *Let f and g be functions on a domain $\mathcal{S}$. Then,*

$$\sup_{x\in\mathcal{S}} f(x) - \sup_{y\in\mathcal{S}} g(y) \leq \sup_{x\in\mathcal{S}}(f(x) - g(x)).$$

Proof

$$\sup_{x\in\mathcal{S}}(f(x) - g(x)) \geq f(x) - g(x) \geq f(x) - \sup_{y\in\mathcal{S}} g(y), \quad \forall x \in \mathcal{S}.$$

Since RHS holds for all x, $\sup_{x\in\mathcal{S}}(f(x) - g(x))$ is an upper bound for $f(x) - \sup_{y\in\mathcal{S}} g(\mathbf{U})$, and hence

$$\sup_{x\in\mathcal{S}}(f(x) - g(x)) \quad \geq \quad \sup_{x\in\mathcal{S}}\left(f(x) - \sup_{y\in\mathcal{S}} g(y)\right) = \sup_{x\in\mathcal{S}} f(x) - \sup_{y\in\mathcal{S}} g(y).$$

■

Proof (Lemma 10) Suppose that $\max_{\mathbf{V}\in\mathcal{S}} f(\mathbf{V}, \mathbf{X})$ is attained at $\mathbf{V}^*$ and $\max_{\mathbf{V}\in\mathcal{S}} f(\mathbf{V}, \tilde{\mathbf{X}})$ is attained at $\tilde{\mathbf{V}}^*$, and define $\epsilon = f(\mathbf{V}^*, \mathbf{X}) - f(\tilde{\mathbf{V}}^*, \mathbf{X})$. We have that

$$\begin{aligned}
\epsilon &= f(\mathbf{V}^*, \mathbf{X}) - f(\tilde{\mathbf{V}}^*, \tilde{\mathbf{X}}) + f(\tilde{\mathbf{V}}^*, \tilde{\mathbf{X}}) - f(\tilde{\mathbf{V}}^*, \mathbf{X}) \\
&= \max_{\mathbf{V}} f(\mathbf{V}, \mathbf{X}) - \max_{\mathbf{U}} f(\mathbf{U}, \tilde{\mathbf{X}}) + f(\tilde{\mathbf{V}}^*, \tilde{\mathbf{X}}) - f(\tilde{\mathbf{V}}^*, \mathbf{X}) \\
&\leq \max_{\mathbf{V}} \Big(f(\mathbf{V}, \mathbf{X}) - f(\mathbf{V}, \tilde{\mathbf{X}})\Big) + f(\tilde{\mathbf{V}}^*, \tilde{\mathbf{X}}) - f(\tilde{\mathbf{V}}^*, \mathbf{X}),
\end{aligned}$$

where the last step follows from Lemma 11. Therefore,

$$\begin{aligned}
|\epsilon| &\leq \max_{\mathbf{V}} \Big|f(\mathbf{V}, \mathbf{X}) - f(\mathbf{V}, \tilde{\mathbf{X}})\Big| + |f(\tilde{\mathbf{V}}^*, \tilde{\mathbf{X}}) - f(\tilde{\mathbf{V}}^*, \mathbf{X})| \\
&\leq \max_{\mathbf{V}} \gamma(\mathbf{X})\|\mathbf{X} - \tilde{\mathbf{X}}\|_2 + \gamma(\mathbf{X})\|\mathbf{X} - \tilde{\mathbf{X}}\|_2 = 2\gamma(\mathbf{X})\|\mathbf{X} - \tilde{\mathbf{X}}\|_2.
\end{aligned}$$

(We used the Lipschitz condition in the second step.) ■

Appendix D. An alternative proof of Lemma 4

Let $\varepsilon = \text{trace}(\mathbf{V}^T\mathbf{X}\mathbf{V}) - \text{trace}(\tilde{\mathbf{V}}^T\mathbf{X}\tilde{\mathbf{V}})$.

$$\begin{aligned}
\varepsilon &= \text{trace}(\mathbf{V}^T\mathbf{X}\mathbf{V}) - \text{trace}(\mathbf{V}^T\tilde{\mathbf{X}}\mathbf{V}) + \text{trace}(\mathbf{V}^T\tilde{\mathbf{X}}\mathbf{V}) - \text{trace}(\tilde{\mathbf{V}}^T\mathbf{X}\tilde{\mathbf{V}}) \\
&\leq k\|\mathbf{X} - \tilde{\mathbf{X}}\|_2 + \text{trace}(\mathbf{V}^T\tilde{\mathbf{X}}\mathbf{V}) - \text{trace}(\tilde{\mathbf{V}}^T\mathbf{X}\tilde{\mathbf{V}}) \\
&\leq k\|\mathbf{X} - \tilde{\mathbf{X}}\|_2 + \text{trace}(\tilde{\mathbf{V}}^T\tilde{\mathbf{X}}\tilde{\mathbf{V}}) - \text{trace}(\tilde{\mathbf{V}}^T\mathbf{X}\tilde{\mathbf{V}}) \\
&\leq 2k\|\mathbf{X} - \tilde{\mathbf{X}}\|_2.
\end{aligned}$$

Setting $\mathbf{X} = \mathbf{A}^T\mathbf{A}$ and $\tilde{\mathbf{X}} = \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$ we derive the result.

Appendix E. Proof of Theorem 7

Proof Here we use the notations in Algorithm 3. Let, $p_1 = \frac{|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_1}$, and $p_2 = \frac{\mathbf{A}_{ij}^2}{\|\mathbf{A}\|_F^2}$.
Note that t-th elements of S_1 and S_2 are sampled independently with ℓ_1 and ℓ_2 probabilities, respectively. We consider the following disjoint events:

$$\begin{aligned}
&\mathcal{E}_1 : S_1(t) = (i, j, \mathbf{A}_{ij}) \wedge S_2(t) \neq (i, j, \mathbf{A}_{ij}) \\
&\mathcal{E}_2 : S_1(t) \neq (i, j, \mathbf{A}_{ij}) \wedge S_2(t) = (i, j, \mathbf{A}_{ij}) \\
&\mathcal{E}_3 : S_1(t) = (i, j, \mathbf{A}_{ij}) \wedge S_2(t) = (i, j, \mathbf{A}_{ij}) \\
&\mathcal{E}_4 : S_1(t) \neq (i, j, \mathbf{A}_{ij}) \wedge S_2(t) \neq (i, j, \mathbf{A}_{ij})
\end{aligned}$$

Let us denote the events $x_1 : x \leq \alpha$ and $x_2 : x > \alpha$. Clearly, $P[x_1] = \alpha, P[x_2] = 1 - \alpha$.
The elements $S_1(t)$ and $S_2(t)$ are sampled independently, we have

$$\begin{aligned}
P[\mathcal{E}_1] &= P[S_1(t) = (i, j, \mathbf{A}_{ij})]P[S_2(t) \neq (i, j, \mathbf{A}_{ij})] = p_1(1 - p_2). \\
P[\mathcal{E}_2] &= (1 - p_1)p_2, \quad P[\mathcal{E}_3] = p_1p_2, \quad P[\mathcal{E}_4] = (1 - p_1)(1 - p_2).
\end{aligned}$$

We note that α is independent of elements of S_1 and S_2. Therefore, events x_1 and x_2 are independent of the events $\mathcal{E}_j$, $j = 1, 2, 3, 4$. Thus,

$$\begin{aligned}
P[S(t) = (i, j, \mathbf{A}_{ij})] &= P[(\mathcal{E}_1 \wedge x_1) \vee (\mathcal{E}_2 \wedge x_2) \vee \mathcal{E}_3] \\
&= P[\mathcal{E}_1 \wedge x_1] + P[\mathcal{E}_2 \wedge x_2] + P[\mathcal{E}_3] \\
&= P[\mathcal{E}_1]P[x_1] + P[\mathcal{E}_2]P[x_2] + P[\mathcal{E}_3] \\
&= p_1(1 - p_2)\alpha + (1 - p_1)p_2(1 - \alpha) + p_1p_2 \\
&= \alpha \cdot p_1 + (1 - \alpha) \cdot p_2
\end{aligned}$$

■

Algorithm 5 SELECT-s: One-pass SELECT Algorithm to Sample s Elements

Input: $\{a_1, ..., a_N\}$, $a_i \geq 0$, read in one pass sequentially over the data; number of samples s.
$I[1, ..., s] \leftarrow 0$; $V[1, ..., s] \leftarrow 0$ /* arrays to hold s indices and values */
$p_i = 0$.
/* Stream begins */
For $i = 1, ..., N$
    $p_i = p_i + a_i$.
    (in parallel) generate s independent uniform random numbers $r_j \in [0, 1]$, $j = 1, ..., s$.
    (in parallel) $I[j] = i$ and $V[j] = a_i$, if $r_j \leq a_i/p_i$.
End
Output: random indices I and corresponding values V.

Appendix F. SELECT-s Algorithm

Here we extend the SELECT algorithm appeared on p. 137 of Drineas et al. (2006) to select s elements in parallel. For one-pass ℓ_1 and ℓ_2 sampling on elements of $\mathbf{A}$ we call SELECT-s algorithm with inputs $\{|\mathbf{A}_{ij}|\}$ and $\{\mathbf{A}_{ij}^2\}$, respectively, for all i, j.

References

D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *Symposium on the Theory of Computing*, pages 611–618, 2001.

D. Achlioptas and F. McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, page 54(2):9, 2007.

D. Achlioptas, Z. Karnin, and E. Liberty. Matrix entry-wise sampling : Simple is best. In `https://pdfs.semanticscholar.org/aa64/b8fb3382e42f90ee93a1dd0c78f13833963f.pdf`, 2013a.

D. Achlioptas, Z. Karnin, and E. Liberty. Near-optimal entrywise sampling for data matrices. In *Neural Information Processing Systems*, pages 1565–1573, 2013b.

S. Arora, E. Hazan, and S. Kale. A Fast Random Sampling Algorithm for Sparsifying Matrices. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 272–279, vol 4110. Springer, 2006.

M. Asteris, D. Papailiopoulos, and A. Dimakis. Non-negative sparse PCA with provable guarantees. In *International Conference on Machine Learning*, 2014.

J. Cadima and I. Jolliffe. Loadings and correlations in the interpretation of principal components. *Applied Statistics*, 22:203–214, 1995.

T. T. Cai, Z. Ma, and Y. Wu. Sparse pca: Optimal rates and adaptive estimation. *The Annals of Statistics*, 41(6):3074–3110, 2013.

Y. Chen, S. Bhojanapalli, S. Sanghavi, and R. Ward. Coherent Matrix Completion. *International Conference on Machine Learning*, pages 674–682, 2014.

A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

A. d'Aspremont, F. Bach, and L. E. Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, June 2008.

T. Dracheva, R. Philip, W. Xiao, AG Gee, and et al. Distinguishing Lung Tumours From Normal Lung Based on a Small Set of Genes. In *Lung Cancer*, pages 157–164, 55(2), 2007.

P. Drineas and A. Zouzias. A note on element-wise matrix sparsification via a matrix-valued Bernstein inequality. In *Information Processing Letters*, pages 385–389, 111(8), 2011.

P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices I: approximating matrix multiplication. In *SIAM Journal on Computing*, pages 132–157, 36(1), 2006.

E. Gabrilovich and S. Markovitch. Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5. In *International Conference on Machine Learning*, 2004.

J. Hou, J. Aerts, B. den Hamer, and et al. Gene Expression-Based Classification of Non-Small Cell Lung Carcinomas and Survival Prediction. In *PLoS One*, page 5(4):e10312, 2010.

J. J. Hull. A database for handwritten text recognition research. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 550–554, 16(5), 1994.

J. Lei and V. Q. Vu. Sparsistency and agnostic inference in sparse pca. *The Annals of Statistics*, 43 (1):299–322, 2015.

K. Lounici. Sparse principal component analysis with missing observations. *High Dimensional Probability VI: The Banff Volume*, pages 327–356, 2013.

Z. Ma. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41 (2):772–801, 2013.

M. Magdon-Ismail. NP-hardness and inapproximability of sparse PCA. *Information Processing Letters*, 126:35–38, 2017. Earlier version: arXiv:1502.05675, 2015.

M. Magdon-Ismail and C. Boutsidis. Optimal sparse linear encoders and sparse pca. In *Neural Information Processing Systems*, 2016.

M.W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. In *National Academy of Sciences*, pages 697–702, 106 (3), 2009.

B. Moghaddam, Y. Weiss, and S. Avidan. Generalized spectral bounds for sparse LDA. In *International Conference on Machine Learning*, 2006.

B. Recht. A simpler approach to matrix completion. In *The Journal of Machine Learning Research*, pages 3413–3430, 12, 2011.

H. Shen and Z. J. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99:1015–1034, July 2008.

K. Sjstrand, L.H. Clemmensen, R. Larsen, and B. Ersbll. Spasm: A matlab toolbox for sparse statistical modeling. In *Journal of Statistical Software (Accepted for publication)*, 2012.

N. Trendafilov, I. T. Jolliffe, and M. Uddin. A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, 12:531–547, 2003.

Z. Wang, H. Lu, and H. Liu. Nonconvex statistical optimization: Minimax-optimal sparse pca in polynomial time. *arXiv preprint arXiv:1408.5352*, 2014.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational & Graphical Statistics*, 15(2):265–286, 2006.