One-Pass Additive-Error Subset Selection for ℓ_p Subspace Approximation

Amit Deshpande
Microsoft Research, Bengaluru, India

Rameshwar Pratap
Indian Institute of Technology, Mandi, H.P., India

Abstract

We consider the problem of subset selection for ℓ_p subspace approximation, that is, to efficiently find a *small* subset of data points such that solving the problem optimally for this subset gives a good approximation to solving the problem optimally for the original input. Previously known subset selection algorithms based on volume sampling and adaptive sampling [16], for the general case of $p \in [1, \infty)$, require multiple passes over the data. In this paper, we give a one-pass subset selection with an additive approximation guarantee for ℓ_p subspace approximation, for any $p \in [1, \infty)$. Earlier subset selection algorithms that give a one-pass multiplicative $(1 + \epsilon)$ approximation work under the special cases. Cohen et al. [11] gives a one-pass subset section that offers multiplicative $(1 + \epsilon)$ approximation guarantee for the special case of ℓ_2 subspace approximation. Mahabadi et al. [31] gives a one-pass *noisy* subset selection with $(1+\epsilon)$ approximation guarantee for ℓ_p subspace approximation when $p \in \{1, 2\}$. Our subset selection algorithm gives a weaker, additive approximation guarantee, but it works for any $p \in [1, \infty)$.

2012 ACM Subject Classification Theory of computation → Streaming models; Mathematics of computing → Dimensionality reduction; Computing methodologies → Dimensionality reduction and manifold learning; Theory of computation → Sketching and sampling

Keywords and phrases Subspace approximation, streaming algorithms, low-rank approximation, adaptive sampling, volume sampling, subset selection

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.51

Category Track A: Algorithms, Complexity and Games

Acknowledgements We would like to thank anonymous reviewers for their careful reading of our manuscript and their insightful comments and suggestions.

1 Introduction

In *subset selection* problems, the objective is to pick a small subset of the given data such that solving a problem optimally on this subset gives a good approximation to solving it optimally on the entire data. Many coreset constructions in computational geometry and clustering [22], sampling-based algorithms for large matrices [24], algorithms for submodular optimization and active learning [37] essentially perform subset selection. The main advantage of subset selection lies in its interpretability, for example, in gene expression analysis, we would like to find a representative subset of genes from gene expression data rather than just fitting a subspace to the data [20, 33, 36, 32, 29]. In several machine learning applications such as document classification, face recognition etc., it is desirable to go beyond dimension reduction alone, and pick a subset of representative items or features [28, 33]. Subset selection has been well studied for many fundamental problems such as k-means clustering [2, 14], low-rank approximation [24, 17, 15, 28] and regression [13], to name a few. In low-rank and subspace approximation, the subset selection approach leads to more interpretable solutions than using SVD or random projections-based results. Therefore, subset selection has been a separate and well-studied problem even within the low-rank approximation and subspace approximation literature [28, 12].

In the following, we formally state the ℓ_p subspace approximation problem for $p \in [1, \infty)$.

ℓ_p subspace approximation: In this problem, given a dataset $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ of n points in $\mathbb{R}^d$, a positive integer $1 \leq k \leq d$ and a real number $p \in [1, \infty)$, the objective is to find a linear subspace V in $\mathbb{R}^d$ of dimension at most k that minimizes the sum of p-th powers of the Euclidean distances of all the points to the subspace V, that is, to minimize

$$\mathrm{err}_p(\mathcal{X}, V) := \sum_{i=1}^{n} d(x_i, V)^p. \tag{1}$$

Throughout this paper, we use V^* to denote the optimal subspace for ℓ_p subspace approximation. The optimal solutions are different for different values of p but we do not include that in the notation to keep the presentation simple, as our results hold for any $p \in [1, \infty)$.

Before stating our results, we first explain what a *small* subset and a *good* approximation means in the context of subset selection for ℓ_p subspace approximation.

For ℓ_p subspace approximation, we consider n and d to be large, $k \ll n, d$, and p to be a small constant. Thus, a *small* subset of $\mathcal{X}$ desired in subset selection has size independent of n and d, and is bounded by $\mathrm{poly}(k/\epsilon)$, where ϵ is a parameter that controls the approximation guarantee (as explained later). Note that the trivial solution $V = 0$ gives $\mathrm{err}_p(\mathcal{X}, V) = \sum_{i=1}^{n} \|x_i\|^p$. Using the standard terminology from previous work [24, 15, 16], an additive approximation guarantee means outputting V such that $\mathrm{err}_p(\mathcal{X}, V) \leq \mathrm{err}_p(\mathcal{X}, V^*) + \epsilon \sum_{i=1}^{n} \|x_i\|^p$, whereas a multiplicative approximation guarantee means $\mathrm{err}_p(\mathcal{X}, V) \leq (1 + \epsilon)\ \mathrm{err}_p(\mathcal{X}, V^*)$. Most subset selection algorithms for ℓ_p subspace approximation select a $\mathrm{poly}(k/\epsilon)$-sized subset of $\mathcal{X}$ such that its span contains a subspace V of dimension at most k that is close enough to V^* to obtain the above approximation guarantees.

Our objective in this paper is to propose an efficient, one-pass sampling algorithm that performs subset selection for ℓ_p subspace approximation for $p \in [1, \infty)$ defined as above. We note that the problem of one-pass subset selection for ℓ_p subspace approximation has been studied for special values of p, for example, Cohen et al. [11] gives one-pass subset selection for $p = 2$, Mahabadi et al. [31] suggest one-pass *noisy* subset selection for $p = \{1, 2\}$. To the best of our knowledge this problem has not been studied in generality for $p \in [1, \infty)$. In this work, we consider studying this problem. We state our results as follows.

1.1 Our results

Our main technical contribution is a one-pass MCMC-based sampling algorithm that can approximately simulate multiple rounds of adaptive sampling. As a direct application of the above, we get the following results for the ℓ_p subspace approximation problem: For $p \in [1, \infty)$, our algorithm makes only one pass over the given data and outputs a subset of $\mathrm{poly}(k/\epsilon)^p$ points whose span contains a k dimensional subspace with an additive approximation guarantee for ℓ_p subspace approximation. This generalizes the well-known squared-length sampling algorithm of Frieze et al. [24] that gives additive approximation guarantee for ℓ_2 subspace approximation (or low-rank approximation under the Frobenium norm). Even though stronger multiplicative $(1+\epsilon)$ approximation algorithms for ℓ_p subspace approximation are known in the previous work, either they cannot do subset selection, or they are not one-pass, or they do not work for all $p \in [1, \infty)$.

Organization of the paper. In Section 2, we compare and contrast our result with the state-of-the-art algorithms, and explain the key technical challenges, and workarounds. In Section 3, we state our MCMC based subset selection algorithm for subset selection for ℓ_p

subspace approximation. In Section 4, we give theoretical bounds on the sample size and approximation guarantee. Finally, in Section 5, we conclude our discussion and state some potential open questions of the paper.

2 Related work

In this section, we discuss related work on sampling and sketching algorithms for ℓ_p subspace approximation, and do a thorough comparison of our results with the state of the art.

2.1 Sampling-based ℓ_p subspace approximation

Frieze et al. [24] show that selecting a subset of $O(k/\epsilon)$ data points as an *i.i.d.* sample from $x_1, x_2, \ldots, x_n$ picked by squared-length sampling, i.e., x_i is picked with probability proportional to $\|x_i\|_2^2$, gives an additive approximation for ℓ_2 subspace approximation (also known as low-rank approximation under the Frobenius norm). Squared-length sampling can be implemented in one pass over $\mathcal{X}$ using reservoir sampling [35, 21]. It is known how to improve the additive approximation guarantee to a multiplicative approximation by combining two generalizations of squared-length sampling, namely, adaptive sampling and volume sampling [15, 16] but it requires $O(k \log k)$ passes over the data. In adaptive sampling, we pick points with probability proportional to the distance from the span of previously picked points, and in volume sampling, we pick a subset of points with probability proportional to the squared volume of the parallelepiped formed by them. Volume sampling a subset of size k can itself be simulated with an approximation factor $k!$ in k rounds of adaptive sampling [15]. For $p = 2$, it is also known that picking a subset of $O(k/\epsilon)$ points by volume sampling gives a bi-criteria $(1+\epsilon)$ approximation for ℓ_2 subspace approximation [28]. For general $p \in [1, \infty)$, it is known that subset selection based on adaptive sampling and volume sampling can be generalized to get a $(1+\epsilon)$ multiplicative approximation for ℓ_p subspace approximation, for any $p \in [1, \infty)$, where the subset is of size $O\left((k/\epsilon)^p\right)$ and it is picked in $O(k \log k)$ passes over the data [16]. The main bottleneck for implementing this in one pass is the inability to simulate multiple rounds of adaptive sampling in a single pass.

The only known workarounds to get one-pass subset selection for ℓ_p subspace approximation are known for the special cases $p = 1$ and $p = 2$. Cohen et al. [11] give a one-pass subset selection algorithm with a multiplicative $(1+\epsilon)$ approximation guarantee for ℓ_2 subspace approximation based on ridge leverage score sampling. Their one-pass implementation crucially uses deterministic matrix sketching [25] to approximate the SVD and ridge leverage scores, and works only for $p = 2$, to the best of our knowledge. Braverman et al. [6] give online algorithms for ℓ_2 subspace approximation (or low-rank approximation) via subset selection but their subset size $O(\frac{k}{\epsilon} \log n \log \kappa)$ is not independent on n and depends logarithmically on the number of points n and the condition number κ. Recent work by Mahabadi et al. [31] gives a one-pass algorithm with a multiplicative $(1+\epsilon)$ approximation guarantee for ℓ_p subspace approximation. However, their algorithm works only in the special cases $p \in \{1, 2\}$ and it outputs a subset of noisy data points instead of the actual data points.

A different objective for ℓ_p subspace approximation has also been studied in literature [5, 9], namely, minimizing the entry-wise ℓ_p-norm low-rank approximation error. To state it formally, given an input matrix $A \in \mathbb{R}^{n \times d}$ and a real number $p \in [0, \infty)$, their objective is to find a matrix B of rank at most k that minimizes $\sum_{i,j} |A_{i,j} - B_{i,j}|^p$.

2.2 Sketching-based ℓ_p subspace approximation

Sketching-based algorithms compute a sketch of a given data in a single pass, using which one can compute an approximately optimal solution to a given problem on the original data. The problem of ℓ_p subspace approximation has been well-studied in previous work on sketching algorithms. However, a limitation of these results is that they do not directly perform subset selection. We mention a few notable results as follows: For $p = 2$, extending deterministic matrix sketching of Liberty [30], Ghashami et al. [27, 26] give a deterministic one-pass sketching algorithm that gives a multiplicative $(1 + \epsilon)$ approximation guarantee for ℓ_2 subspace approximation (or low-rank approximation under the Frobenius norm). Cormode et al. [19] extend the above deterministic sketching idea to $p \neq 2$ and give a poly(k) approximation for entry-wise ℓ_1-norm low-rank approximation and an additive $\epsilon \ \|b\|_\infty$ approximation for ℓ_∞ regression. There is another line of work based on sketching algorithms using random projection. Random projection gives a multiplicative $(1+\epsilon)$ approximation for ℓ_2 subspace approximation in running time $O(\text{nnz}(X) \cdot \text{poly}(k/\epsilon))$, subsequently improved to a running time of $O(\text{nnz}(X) + (n + d) \cdot \text{poly}(k/\epsilon))$ by Clarkson and Woodruff [10]. Feldman et al. [23] also give a one-pass algorithm for multiplicative $(1 + \epsilon)$ approximation for ℓ_p subspace approximation, for $p \in [1, 2]$. However, these results do not provide a one-pass subset selection.

2.3 Comparison with other MCMC-based sampling results

Theorem 4 of Anari et al. [1] gives a MCMC based sampling algorithm to approximate volume sampling distribution. However, their algorithm requires a greedy algorithm to pick the initial subset that requires k passes over the input.

The MCMC sampling has also been explored in the context of k-means clustering. The D^2-sampling proposed by Arthur and Vassilvitskii [2] adaptively samples k points – one point in each passes over the input, and the sampled points give $O(\log k)$ approximation to the optimal clustering solution. The results due to [4, 3] suggest generating MCMC sampling distribution by taking only one pass over the input that closely approximates the underlying D^2 sampling distribution, and offer close to the optimal clustering solution. Building on these MCMC based sampling techniques, Pratap et al. [34] gives one pass subset section for *spherical* k-means clustering [18].

3 MCMC sampling algorithm

In this section, we state our MCMC based sampling algorithm for subset selection for ℓ_p subspace approximation. We first recall the adaptive sampling algorithm[15, 16] for ℓ_p subspace approximation.

Adaptive sampling [15, 16] *w.r.t.* a subset $S \subseteq \mathcal{X}$ is defined as picking points from $\mathcal{X}$ such that the probability of picking any point $x \in \mathcal{X}$ is proportional to $d(x, \text{span}\,(S))^p$. We denote this probability by

$$p_S(x) = \frac{d(x, \text{span}\,(S))^p}{\text{err}_p(\mathcal{X}, S)}, \quad \text{for } x \in \mathcal{X}. \tag{2}$$

For any subset S whose $\text{err}_p(\mathcal{X}, S)$ is not too small, we show that adaptive sampling *w.r.t.* S can be approximately simulated by an MCMC sampling algorithm that only has access to *i.i.d.* samples of points $x \in \mathcal{X}$ picked from the following easier distribution:

$$q(x) = \frac{d(x, \text{span}\left(\tilde{S}\right))^p}{2\ \text{err}_p(\mathcal{X}, \tilde{S})} + \frac{1}{2\,|\mathcal{X}|}, \tag{3}$$

for some initial subset $\tilde{S}$. We give the above definition of $q(x)$ using an arbitrary initial or *pivot* subset $\tilde{S}$ because it will be useful in our analysis of multiple rounds of adaptive sampling. However, our final algorithm uses a fixed subset $\tilde{S} = \emptyset$ such that

$$q(x) = \frac{\|x\|_2^p}{2\sum_{x\in\mathcal{X}} \|x\|_2^p} + \frac{1}{2\,|\mathcal{X}|}. \tag{4}$$

Note that sampling from this easier distribution, namely, picking $x \in \mathcal{X}$ with probability $q(x)$ (mentioned in Equation (4)), can be done in only one pass over $\mathcal{X}$ using weighted reservoir sampling [8]. Weighted reservoir sampling keeps a reservoir of finite items, and for every new item, calculates its relative weight to randomly decide if the item should be added to the reservoir. If the new item is selected, then one of the existing items from the reservoir is picked uniformly and replaced with the new item. Further, given any non-negative weights w_x, for each point $x \in \mathcal{X}$, weighted reservoir sampling can pick an *i.i.d.* sample of points, where x is picked with probability proportional to its weight w_x. Note that this does not require the knowledge of $\sum_{x\in\mathcal{X}} w_x$. Thus, we can run two reservoir sampling algorithms in parallel to maintain two samples, one that picks points with probability proportional to $||x||_2^p$, and another that picks points with uniform probability. Our actual sampling with probability proportional $q(x) = \frac{\|x\|_2^p}{2\sum_{x\in\mathcal{X}}\|x\|_2^p} + \frac{1}{2|\mathcal{X}|}$ picks from one of the two reservoirs with probability $1/2$ each. Therefore, our MCMC algorithm uses a single pass of $\mathcal{X}$ to pick a small sample of *i.i.d.* random points from the probability distribution $q(\cdot)$, in advance. Note that $q(\cdot)$ is an easier and fixed distribution compared to $p_S(\cdot)$. The latter one depends on S and could change over multiple rounds of adaptive sampling.

Let $x \in \mathcal{X}$ be a random point sampled with probability $q(x)$. Consider a random walk whose single step is defined as follows: sample another point $y \in \mathcal{X}$ independently with probability $q(y)$ and sample a real number r *u.a.r.* from the interval $(0, 1)$, and if

$$\frac{d(y, \text{span}\,(S))^p\ q(x)}{d(x, \text{span}\,(S))^p\ q(y)} = \frac{p_S(y)\ q(x)}{p_S(x)\ q(y)} > r,$$

then move from x to y, else, stay at x. Essentially, this does rejection sampling using a simpler distribution $q(\cdot)$. Observe that the stationary distribution of the above random walk is the adaptive sampling distribution $p_S(\cdot)$. We use $\tilde{P}_m^{(1)}(\cdot \mid S)$ to denote the resulting distribution on $\mathcal{X}$ after m steps of the above random walk. Note that m steps of the above random walk can be simulated by sampling m *i.i.d.* points from the distribution $q(\cdot)$ in advance, and representing them implicitly as m-dimensional points.

Lemma 1 below shows that for any subsets $\tilde{S} \subseteq S \subseteq \mathcal{X}$ (where $\tilde{S}$ is the initial subset, and S is the current subset), either $\text{err}_p(\mathcal{X}, S)$ is small compared to $\text{err}_p(\mathcal{X}, \tilde{S})$, or our MCMC sampling distribution closely approximates the adaptive sampling distribution $p_S(\cdot)$ in total variation distance. Proof of Lemma 1 relies on Corollary 1 of Cai [7] that gives an upper bound on the TV distance between these two distributions in terms of: 1) length of the Markov chain, and 2) upper bound on the ratio between these two distributions for any input point.

▶ **Lemma 1.** *Let $\epsilon_1, \epsilon_2 \in (0, 1)$ and $\tilde{S} \subseteq S \subseteq \mathcal{X}$. Let $P^{(1)}(\cdot \mid S)$ denote the distribution over an i.i.d. sample of t points picked from adaptive sampling w.r.t. S, and let $\tilde{P}_m^{(1)}(\cdot \mid \tilde{S})$ denote the distribution over t points picked by t independent random walks of length m each in our one-pass adaptive sampling algorithm; see step 3(a). Then for $m \geq 1 + \frac{2}{\epsilon_1}\log\frac{1}{\epsilon_2}$, either $\text{err}_p(\mathcal{X}, S) \leq \epsilon_1\ \text{err}_p(\mathcal{X}, \tilde{S})$ or $\left\| P^{(1)}(\cdot \mid S) - \tilde{P}_m^{(1)}(\cdot \mid S) \right\|_{TV} \leq \epsilon_2 t$.*

One-pass (approximate MCMC) adaptive sampling algorithm:
Input: a discrete subset $\mathcal{X} \subseteq \mathbb{R}^d$ and integer parameters $t, l, m \in \mathbf{Z}_{\geq 0}$.
Ouput: a subset $S \subseteq \mathcal{X}$.

1. Pick an *i.i.d.* sample $\mathcal{Y}$ of size $|\mathcal{Y}| = ltm$ from $\mathcal{X}$, without replacement, where the probability of picking $x \in \mathcal{X}$ is

$$q(x) = \frac{d(x, \operatorname{span}(\tilde{S}))^p}{2 \operatorname{err}_p(\mathcal{X}, \tilde{S})} + \frac{1}{2|\mathcal{X}|}.$$

   We use the *pivot* subset $\tilde{S} = \emptyset$ so the corresponding distribution is

$$q(x) = \frac{1}{2} \frac{\|x\|_2^p}{\sum_{x \in \mathcal{X}} \|x\|_2^p} + \frac{1}{2|\mathcal{X}|}.$$

   %% This can be implemented in one pass over $\mathcal{X}$ using weighted reservoir sampling [8]. Weighted reservoir sampling is a weighted version of the classical reservoir sampling where the probability of inclusion of an item in the sample is proportional to the weight associated with the item.
2. Initialize $S \leftarrow \emptyset$.
3. For $i = 1, 2, \ldots, l$ do:
   a. Pick an *i.i.d.* sample A_i of size $|A_i| = t$ from $\mathcal{X}$ as follows. Each point in A_i is sampled by taking m steps of the following random walk starting from a point x picked with probability $q(x)$. In each step of the random walk, we pick another point y from $\mathcal{X}$ with probability $q(y)$ and pick a real number r uniformly at random from the interval $(0, 1)$. If $\frac{d(y, \operatorname{span}(S))^p \; q(x)}{d(x, \operatorname{span}(S))^p \; q(y)} > r$ then move to y, else, stay at the current point.
   %% Note that we add only the final point obtained after the m-step random walk in the subset S.
   %% We note that the steps 1-3 of the algorithm can be simulated by taking only one pass over the input as discussed below. Suppose we have a single-pass Algorithm A for sampling from a particular distribution, we can design another Algorithm B that runs in parallel to Algorithm A and post-processes its sample. In our setting, once we know how to get an *i.i.d.* sample of points, where point x is picked with probability $q(x)$, we can run another parallel thread that simulates a random walk whose each step requires a point picked with probability $q(x)$ and performs Step 3.
   b. $S \leftarrow S \cup A_i$.
4. Output S.

Proof. First, consider the $l = 1, t = 1$ case of the one-pass adaptive sampling algorithm described above. In this case, the procedure outputs only one element of $\mathcal{X}$. This random element is picked by m steps of the following random walk starting from an x picked with probability $q(x)$. In each step, we pick another point y with probability $q(y)$ and sample a real

One-pass MCMC ℓ_p subspace approximation algorithm:
Input: a discrete subset $\mathcal{X} \subseteq \mathbb{R}^d$, an integer parameter $k \in \mathbf{Z}_{\geq 0}$ and an error parameter $\delta \in \mathbb{R}_{\geq 0}$.
Output: a subset $\mathcal{S} \subseteq \mathcal{X}$ of $\tilde{O}\left((k/\epsilon)^{p+1}\right)$ points.
1. Repeat the following $O(k \log \frac{1}{\epsilon})$ times in parallel and pick the best sample, $\mathcal{S}$ that minimizes $\sum_{x \in \mathcal{X}} d(x, \text{span}\,(\mathcal{S}))^p$.
   a. Call **One-pass (approximate MCMC) adaptive sampling algorithm** with $t = \tilde{O}((k/\epsilon)^{p+1})$, $l = k$ and $m = 1 + \frac{2}{\epsilon_1} \log \frac{1}{\epsilon_2}$.
2. Output $\mathcal{S}$.

number r *u.a.r.* from the interval $(0, 1)$, and if $p_S(y)q(x)/p_S(x)q(y) > r$, then we move from x to y, else, we stay at x. Observe that the stationary distribution of the above random walk is the adaptive sampling distribution *w.r.t.* S given by $p_S(x) = d(x, \text{span}\,(S))^p / \text{err}_p(\mathcal{X}, S)$. Using Corollary 1 of [7], the total variation distance after m steps of the random walk is bounded by

$$\left(1 - \frac{1}{\gamma}\right)^{m-1} \leq e^{-(m-1)/\gamma} \leq \epsilon_2, \text{ where } \gamma = \max_{x \in \mathcal{X}} \frac{p_S(x)}{q(x)}.$$

The above bound is at most ϵ_2 if we choose to run the random walk for $m \geq 1 + \gamma \log \frac{1}{\epsilon_2}$ steps. Now suppose $\text{err}_p(\mathcal{X}, S) > \epsilon_1 \text{ err}_p(\mathcal{X}, \tilde{S})$. Then, for any $x \in \mathcal{X}$

$$\begin{aligned}\frac{p_S(x)}{q(x)} &= \frac{\frac{d(x, \text{span}\,(S))^p}{\text{err}_p(\mathcal{X}, S)}}{\frac{1}{2}\frac{d(x, \text{span}\,(\tilde{S}))^p}{\text{err}_p(\mathcal{X}, \tilde{S})} + \frac{1}{2|\mathcal{X}|}} \\ &\leq \frac{2\, d(x, \text{span}\,(S))^p \text{ err}_p(\mathcal{X}, \tilde{S})}{d(x, \text{span}\,(\tilde{S}))^p \text{ err}_p(\mathcal{X}, S)} \leq \frac{2}{\epsilon_1},\end{aligned}$$

using $d(x, \text{span}\,(S))^p \leq d(x, \text{span}\,(\tilde{S}))^p$ because $\tilde{S} \subseteq S$, and the above assumption $\text{err}_p(\mathcal{X}, S) > \epsilon_1 \text{ err}_p(\mathcal{X}, \tilde{S})$. Therefore, $m > \frac{2}{\epsilon_1} \log \frac{1}{\epsilon_2}$ ensures that m steps of the random walk gives a distribution within total variation distance ϵ_2 from the adaptive sampling distribution for picking a single point.

Note that for $t > 1$ both the adaptive sampling and the MCMC sampling procedure pick an *i.i.d.* sample of t points, so the total variation distance is additive in t, which means

$$\left\| P^{(1)}(\cdot \mid S) - \tilde{P}_m^{(1)}(\cdot \mid S) \right\|_{TV} \leq \epsilon_2 t,$$

assuming $\text{err}_p(\mathcal{X}, S) > \epsilon_1 \text{ err}_p(\mathcal{X}, \tilde{S})$. This completes a proof of the lemma. ◀

4 ℓ_p subspace approximation

In this section, we give our result for one pass subset selection for ℓ_p subspace approximation. We first show (in Lemma 2) that the true adaptive sampling can be well approximated by one pass (approximate) MCMC based sampling algorithm. Building on this result, in Proposition 3 and Theorem 4, we show bounds on the number of steps taken by the Markov chain, and on the sample size that gives an additive approximation for the ℓ_p subspace approximation. Our MCMC-based sampling ensures that our problem statement's single-pass subset selection criteria are satisfied.

First, let's set up the notation required to analyze the true adaptive sampling as well as our one-pass (approximate MCMC) adaptive sampling algorithm. For any fixed subset $S \subseteq \mathcal{X}$, we define

$$\mathrm{err}_p(\mathcal{X}, S) = \sum_{x \in \mathcal{X}} d(x, \mathrm{span}\,(S))^p, \tag{5}$$

$$P^{(1)}(T|S) = \prod_{x \in T} \frac{d(x, \mathrm{span}\,(S))^p}{\mathrm{err}_p(\mathcal{X}, S)}, \tag{6}$$

$$\text{for any subset } T \text{ of size } t,$$

$$\mathop{\mathrm{E}}_{T}\left[\mathrm{err}_p(\mathcal{X}, S \cup T)\right] = \sum_{T\,:\,|T|=t} P^{(1)}(T \mid S)\ \mathrm{err}_p(\mathcal{X}, S \cup T). \tag{7}$$

Given a subset $S \subseteq \mathcal{X}$, $P^{(1)}(T \mid S)$ denotes the probability of picking a subset $T \subseteq \mathcal{X}$ of *i.i.d.* t points by adaptive sampling *w.r.t.* S. We use $P^{(l)}(T_{1:l}|S)$ to denote the probability of picking a subset $T_{1:l} = B_1 \cup B_2 \cup \ldots \cup B_l \subseteq \mathcal{X}$ of tl points by l iterative rounds of adaptive sampling, where in the first round we sample a subset B_1 consisting of *i.i.d.* t points *w.r.t.* S, in the second round we sample a subset B_2 consisting of *i.i.d.* t points *w.r.t.* $S \cup B_1$, and so on to pick $T_{1:l} = B_1 \cup B_2 \cup \ldots \cup B_l$ over l iterations. Similarly, in the context of adaptive sampling, we use $T_{2:l}$ to denote $B_2 \cup \ldots \cup B_l$. We abuse the notation $\mathop{\mathrm{E}}_{T_{1:l} \mid S}[\cdot]$ to denote the expectation over $T_{1:l}$ picked in l iterative rounds of adaptive sampling starting from S.

Given a *pivot* subset $\tilde{S} \subseteq \mathcal{X}$ and another subset $S \subseteq \mathcal{X}$ such that $\tilde{S} \subseteq S$, consider the following MCMC sampling with parameters l, t, m that picks l subsets $A_1, A_2, \ldots, A_l$ of t points each, where m denotes the number of steps of a random walk used to pick these points. This sampling can be implemented in a single pass over $\mathcal{X}$, for any l, t, m, and any given subsets $\tilde{S} \subseteq S$. For $T_{1:l} = A_1 \cup A_2 \cup \ldots \cup A_l$. We use $\tilde{P}^{(l)}_m(T_{1:l} \mid S)$ to denote the probability of picking $T_{1:l}$ as the output of the following MCMC sampling procedure. Similarly, in the context of MCMC sampling, we use $T_{2:l}$ to denote $A_2 \cup \ldots \cup A_l$. We abuse the notation $\mathop{\tilde{\mathrm{E}}}_{T_{1:l} \mid S}[\cdot]$ to denote the expectation over $T_{1:l}$ picked using the MCMC sampling procedure starting from S with a pivot subset $\tilde{S} \subseteq S$.

We require the following additional notation in our analysis of the above MCMC sampling. We use $\tilde{P}^{(1)}_m(T \mid S)$ to denote the resulting distribution over subsets T of size t, when we use the above sampling procedure with $l = 1$. We define the following expressions:

$$\mathrm{ind}_p(\mathcal{X}, S) = \mathbb{1}\left(\mathrm{err}_p(\mathcal{X}, S) \leq \epsilon_1\ \mathrm{err}_p(\mathcal{X}, \tilde{S})\right), \tag{8}$$

$$\mathop{\tilde{\mathrm{E}}}_{T}\left[\mathrm{err}_p(\mathcal{X}, S \cup T)\right] = \sum_{T\,:\,|T|=t} \tilde{P}^{(1)}_m(T \mid S)\ \mathrm{err}_p(\mathcal{X}, S \cup T), \tag{9}$$

$$\mathop{\tilde{\mathrm{E}}}_{T}\left[\mathrm{ind}_p(\mathcal{X}, S \cup T)\right] = \sum_{T\,:\,|T|=t} \tilde{P}^{(1)}_m(T \mid S)\ \mathrm{ind}_p(\mathcal{X}, S \cup T). \tag{10}$$

The expression $\mathrm{ind}_p(\mathcal{X}, S)$ (in Equation (8)) denotes an indicator random variable that takes value 1 if error *w.r.t.* subset S is smaller than ϵ_1 times error *w.r.t.* subset $\tilde{S}$, and 0 otherwise. The expression $\mathop{\tilde{\mathrm{E}}}_{T}\left[\mathrm{err}_p(\mathcal{X}, S \cup T)\right]$ (in Equation (9)) denotes the expected error over the subset T picked using the MCMC sampling procedure starting from the set S such that the initial subset $\tilde{S} \subseteq S$.

Now Lemma 2 analyzes the effect of starting with an initial subset S_0 and using the same S_0 as a pivot subset for doing the MCMC sampling for l subsequent iterations of adaptive sampling, where we pick t *i.i.d.* points in each iteration using t independent random walks

of m steps. Lemma 2 shows that the expected error for subspace approximation after doing the l iterations of adaptive sampling is not too far from the expected error for subspace approximation after replacing the l iterations with MCMC sampling.

▶ **Lemma 2.** *For any subset $S_0 \subseteq \mathcal{X}$, any $\epsilon_1, \epsilon_2 \in (0,1)$ and any positive integers t, l, m with $m \geq 1 + \frac{2}{\epsilon_1} \log \frac{1}{\epsilon_2}$,*

$$\tilde{\mathrm{E}}_{T_{1:l} \mid S_0} \left[\mathrm{err}_p(\mathcal{X}, S_0 \cup T_{1:l})\right] \leq \mathrm{E}_{T_{1:l} \mid S_0} \left[\mathrm{err}_p(\mathcal{X}, S_0 \cup T_{1:l})\right] + (\epsilon_1 + \epsilon_2 tl)\, \mathrm{err}_p(\mathcal{X}, S_0).$$

Proof. We show a slightly stronger inequality than the one given above, i.e., for any S_0 such that $\tilde{S} \subseteq S_0$,

$$\tilde{\mathrm{E}}_{T_{1:l} \mid S_0} \left[\mathrm{err}_p(\mathcal{X}, S_0 \cup T_{1:l})\right] \leq \mathrm{E}_{T_{1:l} \mid S_0} \left[\mathrm{err}_p(\mathcal{X}, S_0 \cup T_{1:l})\right] + \left(\epsilon_1 \tilde{\mathrm{E}}_{T_{1:l} \mid S_0} \left[\mathrm{ind}_p(\mathcal{X}, S_0 \cup T_{1:l})\right] + \epsilon_2 tl\right) \mathrm{err}_p(\mathcal{X}, \tilde{S}).$$

The special case $S_0 = \tilde{S}$ gives the lemma. We prove the above-mentioned stronger statement by induction on l. For $l = 0$, the above inequality holds trivially. Now assuming induction hypothesis, the above holds true for $l-1$ iterations (instead of l) starting with any subset $S_1 = S_0 \cup A \subseteq \mathcal{X}$ because $\tilde{S} \subseteq S_0 \subseteq S_1$.

$$\begin{aligned}
&\tilde{\mathrm{E}}_{T_{1:l} \mid S_0} \left[\mathrm{err}_p(\mathcal{X}, S_0 \cup T_{1:l})\right] \\
&= \tilde{\mathrm{E}}_{S_1 \mid S_0} \left[\tilde{\mathrm{E}}_{T_{2:l} \mid S_1} \left[\mathrm{err}_p(\mathcal{X}, S_1 \cup T_{2:l})\right]\right] \\
&= \sum_{S_1 \,:\, \mathrm{ind}_p(\mathcal{X}, S_1) = 1} \tilde{P}_m^{(1)}(S_1 \mid S_0) \tilde{\mathrm{E}}_{T_{2:l} \mid S_1} \left[\mathrm{err}_p(\mathcal{X}, S_1 \cup T_{2:l})\right] \\
&\quad + \sum_{S_1 \,:\, \mathrm{ind}_p(\mathcal{X}, S_1) = 0} \tilde{P}_m^{(1)}(S_1 \mid S_0) \tilde{\mathrm{E}}_{T_{2:l} \mid S_1} \left[\mathrm{err}_p(\mathcal{X}, S_1 \cup T_{2:l})\right]. \qquad (11)
\end{aligned}$$

If $\mathrm{ind}_p(\mathcal{X}, S_1) = 1$ then $\mathrm{err}_p(\mathcal{X}, S_1 \cup T_{2:l}) \leq \mathrm{err}_p(\mathcal{X}, S_1) \leq \epsilon_1\, \mathrm{err}_p(\mathcal{X}, S_0)$, so the first part of the above sum can be bounded as follows.

$$\begin{aligned}
&\sum_{S_1 \,:\, \mathrm{ind}_p(\mathcal{X}, S_1) = 1} \tilde{P}_m^{(1)}(S_1 \mid S_0) \tilde{\mathrm{E}}_{T_{2:l} \mid S_1} \left[\mathrm{err}_p(\mathcal{X}, S_1 \cup T_{2:l})\right] \\
&\leq \epsilon_1\, \mathrm{err}_p(\mathcal{X}, S_0) \cdot \sum_{S_1 \,:\, \mathrm{ind}_p(\mathcal{X}, S_1) = 1} \tilde{P}_m^{(1)}(S_1 \mid S_0) \tilde{\mathrm{E}}_{T_{2:l} \mid S_1} \left[\mathrm{ind}_p(\mathcal{X}, S_1 \cup T_{2:l})\right]. \qquad (12)
\end{aligned}$$

We give an upper bound on the second part as follows.

$$\begin{aligned}
&\sum_{S_1 \,:\, \mathrm{ind}_p(\mathcal{X}, S_1) = 0} \tilde{P}_m^{(1)}(S_1 \mid S_0) \tilde{\mathrm{E}}_{T_{2:l} \mid S_1} \left[\mathrm{err}_p(\mathcal{X}, S_1 \cup T_{2:l})\right] \\
&= \sum_{S_1 \,:\, \mathrm{ind}_p(\mathcal{X}, S_1) = 0} \tilde{P}_m^{(1)}(S_1 \mid S_0) \tilde{\mathrm{E}}_{T_{2:l} \mid S_1} \left[\mathrm{err}_p(\mathcal{X}, S_1 \cup T_{2:l})\right]. \\
&\leq \sum_{S_1 \,:\, \mathrm{ind}_p(\mathcal{X}, S_1) = 0} \tilde{P}_m^{(1)}(S_1 \mid S_0) \cdot \\
&\quad \left(\mathrm{E}_{T_{2:l} \mid S_1} \left[\mathrm{err}_p(\mathcal{X}, S_1 \cup T_{2:l})\right] + (\epsilon_1 \tilde{\mathrm{E}}_{T_{2:l} \mid S_1} \left[\mathrm{ind}_p(\mathcal{X}, S_1 \cup T_{2:l})\right] + \epsilon_2 t(l-1))\, \mathrm{err}_p(\mathcal{X}, \tilde{S})\right). \qquad (13) \\
&\qquad \text{(by applying the induction hypothesis to } (l-1) \text{ iterations starting from } S_1.) \\
&\leq \sum_{S_1 \,:\, \mathrm{ind}_p(\mathcal{X}, S_1) = 0} P^{(1)}(S_1 \mid S_0) \mathrm{E}_{T_{2:l} \mid S_1} \left[\mathrm{err}_p(\mathcal{X}, S_1 \cup T_{2:l})\right]
\end{aligned}$$

$$
\begin{aligned}
&+ \epsilon_1 \operatorname{err}_p(\mathcal{X}, \tilde{S}) \cdot \sum_{S_1 \,:\, \operatorname{ind}_p(\mathcal{X}, S_1)=0} \tilde{P}_m^{(1)}(S_1 \mid S_0) \mathop{\tilde{\mathrm{E}}}_{T_{2:l} \mid S_1} [\operatorname{ind}_p(\mathcal{X}, S_1 \cup T_{2:l})] \\
&+ \epsilon_2 t(l-1) \operatorname{err}_p(\mathcal{X}, \tilde{S}) \sum_{S_1 \,:\, \operatorname{ind}_p(\mathcal{X}, S_1)=0} \tilde{P}_m^{(1)}(S_1 \mid S_0) \\
&+ \sum_{S_1 \,:\, \operatorname{ind}_p(\mathcal{X}, S_1)=0} \left| \tilde{P}_m^{(1)}(S_1 \mid S_0) - P^{(1)}(S_1 \mid S_0) \right| \cdot \mathop{\mathrm{E}}_{T_{2:l} \mid S_1} [\operatorname{err}_p(\mathcal{X}, S_1 \cup T_{2:l})] .
\end{aligned}
$$

$$
\left(\text{by adding and subtracting the term } \sum_{S_1 \,:\, \operatorname{ind}_p(\mathcal{X}, S_1)=0} P^{(1)}(S_1 \mid S_0) \mathop{\mathrm{E}}_{T_{2:l} \mid S_1} [\operatorname{err}_p(\mathcal{X}, S_1 \cup T_{2:l})] \text{ in Eq. (13).} \right)
$$

$$
\begin{aligned}
&\leq \sum_{S_1} P^{(1)}(S_1 \mid S_0) \mathop{\mathrm{E}}_{T_{2:l} \mid S_1} [\operatorname{err}_p(\mathcal{X}, S_1 \cup T_{2:l})] \\
&\quad + \epsilon_1 \operatorname{err}_p(\mathcal{X}, \tilde{S}) \sum_{S_1 \,:\, \operatorname{ind}_p(\mathcal{X}, S_1)=0} \tilde{P}_m^{(1)}(S_1 \mid S_0) \cdot \mathop{\tilde{\mathrm{E}}}_{T_{2:l} \mid S_1} [\operatorname{ind}_p(\mathcal{X}, S_1 \cup T_{2:l})] \\
&\quad + \epsilon_2 t(l-1) \operatorname{err}_p(\mathcal{X}, \tilde{S}) + \sum_{S_1 \,:\, \operatorname{ind}_p(\mathcal{X}, S_1)=0} \left| \tilde{P}_m^{(1)}(S_1 \mid S_0) - P^{(1)}(S_1 \mid S_0) \right| \cdot \operatorname{err}_p(\mathcal{X}, \tilde{S}).
\end{aligned}
$$

$$
\left(\text{by upper bounding the probability expression } \sum_{S_1 \,:\, \operatorname{ind}_p(\mathcal{X}, S_1)=0} \tilde{P}_m^{(1)}(S_1 \mid S_0) \text{ by 1.} \right)
$$

$$
\begin{aligned}
&\leq \mathop{\mathrm{E}}_{T_{1:l} \mid S_0} [\operatorname{err}_p(\mathcal{X}, S_0 \cup T_{1:l})] \\
&\quad + \epsilon_1 \operatorname{err}_p(\mathcal{X}, \tilde{S}) \sum_{S_1 \,:\, \operatorname{ind}_p(\mathcal{X}, S_1)=0} \tilde{P}_m^{(1)}(S_1 \mid S_0) \cdot \mathop{\tilde{\mathrm{E}}}_{T_{2:l} \mid S_1} [\operatorname{ind}_p(\mathcal{X}, S_1 \cup T_{2:l})] \\
&\quad + \epsilon_2 t(l-1) \operatorname{err}_p(\mathcal{X}, \tilde{S}) + \left\| \tilde{P}^{(1)}(\cdot \mid S_0) - P^{(1)}(\cdot \mid S_0) \right\|_{TV} \operatorname{err}_p(\mathcal{X}, \tilde{S}).
\end{aligned}
$$

$$
\left(\text{as } \mathop{\mathrm{E}}_{T_{1:l} \mid S_0} [\operatorname{err}_p(\mathcal{X}, S_0 \cup T_{1:l})] = \sum_{S_1} P^{(1)}(S_1 \mid S_0) \mathop{\mathrm{E}}_{T_{2:l} \mid S_1} [\operatorname{err}_p(\mathcal{X}, S_1 \cup T_{2:l})] \text{ by Eq. (7).} \right)
$$

$$
\begin{aligned}
&\leq \mathop{\mathrm{E}}_{T_{1:l} \mid S_0} [\operatorname{err}_p(\mathcal{X}, S_0 \cup T_{1:l})] \\
&\quad + \epsilon_1 \operatorname{err}_p(\mathcal{X}, \tilde{S}) \sum_{S_1 \,:\, \operatorname{ind}_p(\mathcal{X}, S_1)=0} \tilde{P}_m^{(1)}(S_1 \mid S_0) \cdot \mathop{\tilde{\mathrm{E}}}_{T_{2:l} \mid S_1} [\operatorname{ind}_p(\mathcal{X}, S_1 \cup T_{2:l})] \\
&\quad + \epsilon_2 t(l-1) \operatorname{err}_p(\mathcal{X}, \tilde{S}) + \epsilon_2 t \operatorname{err}_p(\mathcal{X}, \tilde{S}). \qquad (14)
\end{aligned}
$$

Finally, Equation (14) holds using Lemma 1 about the total variation distance between $P^{(1)}$ and $\tilde{P}^{(1)}$ distributions. Plugging the bounds (12) and (14) into (11), we get

$$
\begin{aligned}
&\mathop{\tilde{\mathrm{E}}}_{T_{1:l} \mid S_0} [\operatorname{err}_p(\mathcal{X}, S_0 \cup T_{1:l})] \\
&\leq \mathop{\mathrm{E}}_{T_{1:l} \mid S_0} [\operatorname{err}_p(\mathcal{X}, S_0 \cup T_{1:l})] + \epsilon_1 \operatorname{err}_p(\mathcal{X}, \tilde{S}) \sum_{S_1} \tilde{P}_m^{(1)}(S_1 \mid S_0) \cdot \mathop{\tilde{\mathrm{E}}}_{T_{2:l} \mid S_1} [\operatorname{ind}_p(\mathcal{X}, S_1 \cup T_{2:l})] \\
&\quad + \epsilon_2 t(l-1) \operatorname{err}_p(\mathcal{X}, \tilde{S}) + \epsilon_2 t \operatorname{err}_p(\mathcal{X}, \tilde{S}). \\
&= \mathop{\mathrm{E}}_{T_{1:l} \mid S_0} [\operatorname{err}_p(\mathcal{X}, S_0 \cup T_{1:l})] + \left(\epsilon_1 \mathop{\tilde{\mathrm{E}}}_{T_{1:l} \mid S_0} [\operatorname{ind}_p(\mathcal{X}, S_0 \cup T_{1:l})] + \epsilon_2 t l \right) \operatorname{err}_p(\mathcal{X}, \tilde{S}). \\
&\leq \mathop{\mathrm{E}}_{T_{1:l} \mid S_0} [\operatorname{err}_p(\mathcal{X}, S_0 \cup T_{1:l})] + (\epsilon_1 + \epsilon_2 t l) \operatorname{err}_p(\mathcal{X}, \tilde{S}),
\end{aligned}
$$

which completes the proof of Lemma 2. ◀

Theorem 5 from [16] shows that in $l = k$ rounds of adaptive sampling, where in each round we pick $t = \tilde{O}\left((k/\epsilon)^{p+1}\right)$ points and take their union, gives an additive approximation guarantee for ℓ_p subspace approximation with probability at least $1/2k$. Repeating it multiple times and taking the best can boost the probability further. We restate the main part of this theorem below.

▶ **Proposition 3** (Theorem 5, [16]). *Let k be any positive integer, let $\epsilon \in (0,1)$ and $S_0 = \emptyset$. Let $l = k$ and $t = \tilde{O}\left((k/\epsilon)^{p+1}\right)$. If $S_l = S_0 \cup T_{1:l}$ is obtained by starting from S_0 and doing adaptive sampling according to the p-th power of distances in l iterations, and in each iteration we add t points from $\mathcal{X}$, then we have $|S_l| = tl = \tilde{O}(k \cdot (k/\epsilon)^{p+1})$ such that*

$$\operatorname{err}_p(\mathcal{X}, S_0 \cup T_{1:l})^{1/p} \leq \operatorname{err}_p(\mathcal{X}, V^*)^{1/p} + \epsilon \operatorname{err}_p(\mathcal{X}, \emptyset)^{1/p},$$

with probability at least $1/2k$, and where V^* minimizes $\operatorname{err}_p(\mathcal{X}, V)$ over all linear subspaces V of dimension at most k. If we repeat this $O(k \log \frac{1}{\epsilon})$ times then the probability of success can be boosted to $1 - \epsilon$.

Combining Lemma 2 and Proposition 3 we get the following Theorem.

▶ **Theorem 4.** *For any positive integer k, any $p \in [1, \infty)$, and any $\delta \in \mathbb{R}_{\geq 0}$, starting from $S_0 = \emptyset$ and setting the following parameters in one-pass MCMC ℓ_p subspace approximation algorithm (see Section 3)*

$$\begin{aligned}
\epsilon &= \delta/4, \\
\epsilon_1 &= \delta^p/2^{p+1}, \\
\epsilon_2 &= \delta^p/2^{p+1}tl, \\
m &= 1 + \frac{2}{\delta^p} \log \frac{k}{\delta^p}, \\
t &= \tilde{O}((k/\epsilon)^{p+1}), \\
l &= k,
\end{aligned}$$

we get a subset $\mathcal{S}$ of size $\tilde{O}(k \cdot (k/\delta)^{p+1})$ with an additive approximation guarantee on its expected error as $\operatorname{err}_p(\mathcal{X}, V^*)^{1/p} + \delta \operatorname{err}_p(\mathcal{X}, \emptyset)^{1/p}$. Further, the running time of the algorithm is $nd + k \cdot \tilde{O}\left(\left(\frac{k}{\delta}\right)^{p+1}\right)$.

Proof. From Lemma 2 we know that

$$\tilde{\mathrm{E}}_{T_{1:l} \mid \emptyset} \left[\operatorname{err}_p(\mathcal{X}, T_{1:l})\right] \leq \mathrm{E}_{T_{1:l} \mid \emptyset} \left[\operatorname{err}_p(\mathcal{X}, T_{1:l})\right] + (\epsilon_1 + \epsilon_2 tl) \operatorname{err}_p(\mathcal{X}, \emptyset).$$

Thus, for $p \in [1, \infty)$ we have

$$\begin{aligned}
\tilde{\mathrm{E}}_{T_{1:l} \mid \emptyset} \left[\operatorname{err}_p(\mathcal{X}, T_{1:l})\right]^{1/p} &\leq \mathrm{E}_{T_{1:l} \mid \emptyset} \left[\operatorname{err}_p(\mathcal{X}, T_{1:l})\right]^{1/p} + (\epsilon_1 + \epsilon_2 tl)^{1/p} \operatorname{err}_p(\mathcal{X}, \emptyset)^{1/p}. \\
&\leq (1 - \epsilon) \left(\operatorname{err}_p(\mathcal{X}, V^*)^{1/p} + \epsilon \operatorname{err}_p(\mathcal{X}, \emptyset)^{1/p}\right) + \epsilon \operatorname{err}_p(\mathcal{X}, \emptyset)^{1/p} \\
&\qquad + (\epsilon_1 + \epsilon_2 tl)^{1/p} \operatorname{err}_p(\mathcal{X}, \emptyset)^{1/p}. \\
&\qquad \text{(using Proposition 3.)} \\
&\leq \operatorname{err}_p(\mathcal{X}, V^*)^{1/p} + \left(2\epsilon + (\epsilon_1 + \epsilon_2 tl)^{1/p}\right) \operatorname{err}_p(\mathcal{X}, \emptyset)^{1/p}. \\
&\leq \operatorname{err}_p(\mathcal{X}, V^*)^{1/p} + \delta \operatorname{err}_p(\mathcal{X}, \emptyset)^{1/p},
\end{aligned}$$

using $\epsilon = \delta/4$, $\epsilon_1 = \delta^p/2^{p+1}$ and $\epsilon_2 = \delta^p/2^{p+1}tl$.

We now give a bound on the running time of our algorithm. We require nd time to generate the probability distribution $q(x)$, for $x \in \mathcal{X}$. Further, the running time of MCMC sampling step is $t \cdot m \cdot l = k \cdot \tilde{O}\left(\left(\frac{k}{\delta}\right)^{p+1}\right)$. Therefore, the overall running time of the algorithm is $nd + k \cdot \tilde{O}\left(\left(\frac{k}{\delta}\right)^{p+1}\right)$. ◀

5 Conclusion and open questions

In this work, we give an efficient one-pass MCMC algorithm that does subset selection with additive approximation guarantee for ℓ_p subspace approximation, for any $p \in [1, \infty)$. Previously this was only known for the special case of $p = 2$ [11]. For general case $p \in [1, \infty)$, adaptive sampling algorithm due to [16] requires taking multiple passes over the input. Coming up with a one-pass subset selection algorithm that offers stronger multiplicative guarantees for $p \in [1, \infty)$ remains an interesting open problem.

References

1 Nima Anari, Shayan Oveis Gharan, and Alireza Rezaei. Monte carlo markov chain algorithms for sampling strongly rayleigh distributions and determinantal point processes. In *29th Annual Conference on Learning Theory (COLT)*, volume 49, pages 103–115. PMLR, 2016. URL: `http://proceedings.mlr.press/v49/anari16.html`.

2 David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. In Nikhil Bansal, Kirk Pruhs, and Clifford Stein, editors, *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 1027–1035. SIAM, 2007. URL: `http://dl.acm.org/citation.cfm?id=1283383.1283494`.

3 Olivier Bachem, Mario Lucic, S. Hamed Hassani, and Andreas Krause. Approximate k-means++ in sublinear time. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 1459–1467. AAAI Press, 2016.

4 Olivier Bachem, Mario Lucic, Seyed Hamed Hassani, and Andreas Krause. Fast and provably good seedings for k-means. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 55–63, 2016. URL: `https://proceedings.neurips.cc/paper/2016/hash/d67d8ab4f4c10bf22aa353e27879133c-Abstract.html`.

5 Frank Ban, Vijay Bhattiprolu, Karl Bringmann, Pavel Kolev, Euiwoong Lee, and David P. Woodruff. A PTAS for p-low rank approximation. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 747–766. SIAM, 2019. `doi:10.1137/1.9781611975482.47`.

6 Vladimir Braverman, Petros Drineas, Cameron Musco, Christopher Musco, Jalaj Upadhyay, David P. Woodruff, and Samson Zhou. Near optimal linear algebra in the online and sliding window models. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 517–528. IEEE, 2020. `doi:10.1109/FOCS46700.2020.00055`.

7 Haiyan Cai. Exact bound for the convergence of metropolis chains. *Stochastic Analysis and Applications*, 18(1):63–71, 2000. `doi:10.1080/07362990008809654`.

8 M. T. CHAO. A general purpose unequal probability sampling plan. *Biometrika*, 69(3):653–656, December 1982. `doi:10.1093/biomet/69.3.653`.

9 Flavio Chierichetti, Sreenivas Gollapudi, Ravi Kumar, Silvio Lattanzi, Rina Panigrahy, and David P. Woodruff. Algorithms for ℓ_p low-rank approximation. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 806–814. PMLR, 06–11 August 2017. URL: `https://proceedings.mlr.press/v70/chierichetti17a.html`.

10 Kenneth L. Clarkson and David P. Woodruff. Low-rank approximation and regression in input sparsity time. *J. ACM*, 63(6), January 2017. doi:10.1145/3019134.

11 Michael B. Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1758–1777. SIAM, 2017. doi:10.1137/1.9781611974782.115.

12 Chen Dan, Hong Wang, Hongyang Zhang, Yuchen Zhou, and Pradeep K Ravikumar. Optimal analysis of subset-selection based l_p low-rank approximation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/80a8155eb153025ea1d513d0b2c4b675-Paper.pdf.

13 Michal Derezinski and Manfred K. Warmuth. Unbiased estimates for linear regression via volume sampling. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3084–3093, 2017. URL: https://proceedings.neurips.cc/paper/2017/hash/54e36c5ff5f6a1802925ca009f3ebb68-Abstract.html.

14 Amit Deshpande, Praneeth Kacham, and Rameshwar Pratap. Robust k-means++. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence, UAI 2020, virtual online, August 3-6, 2020*, volume 124 of *Proceedings of Machine Learning Research*, pages 799–808. AUAI Press, 2020. URL: http://proceedings.mlr.press/v124/deshpande20a.html.

15 Amit Deshpande, Luis Rademacher, Santosh S. Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. *Theory Comput.*, 2(12):225–247, 2006. doi:10.4086/toc.2006.v002a012.

16 Amit Deshpande and Kasturi Varadarajan. Sampling-based dimension reduction for subspace approximation. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 641–650, New York, NY, USA, 2007. Association for Computing Machinery. doi:10.1145/1250790.1250884.

17 Amit Deshpande and Santosh S. Vempala. Adaptive sampling and fast low-rank matrix approximation. In Josep Díaz, Klaus Jansen, José D. P. Rolim, and Uri Zwick, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2006 and 10th International Workshop on Randomization and Computation, RANDOM 2006, Barcelona, Spain, August 28-30 2006, Proceedings*, volume 4110 of *Lecture Notes in Computer Science*, pages 292–303. Springer, 2006. doi:10.1007/11830924_28.

18 Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Mach. Learn.*, 42(1/2):143–175, 2001. doi:10.1023/A:1007612920971.

19 Charlie Dickens, Graham Cormode, and David Woodruff. Leveraging well-conditioned bases: Streaming and distributed summaries in Minkowski p-norms. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1243–1251. PMLR, 10–15 July 2018. URL: https://proceedings.mlr.press/v80/dickens18a.html.

20 Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM J. Matrix Anal. Appl.*, 30(2):844–881, 2008. doi:10.1137/07070471X.

21 Pavlos S. Efraimidis and Paul (Pavlos) Spirakis. Weighted random sampling. In *Encyclopedia of Algorithms*, pages 2365–2367. Springer, 2016. doi:10.1007/978-1-4939-2864-4_478.

22 Dan Feldman. Introduction to core-sets: an updated survey, 2020. arXiv:2011.09384.

23 Dan Feldman, Morteza Monemizadeh, Christian Sohler, and David P. Woodruff. Coresets and sketches for high dimensional subspace approximation problems. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 630–649. SIAM, 2010. doi:10.1137/1.9781611973075.53.

24 Alan M. Frieze, Ravi Kannan, and Santosh S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, 2004. doi:10.1145/1039488.1039494.

25 Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent directions : Simple and deterministic matrix sketching. *CoRR*, abs/1501.01711, 2015. arXiv:1501.01711.

26 Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM J. Comput.*, 45(5):1762–1792, 2016. doi:10.1137/15M1009718.

27 Mina Ghashami and Jeff M. Phillips. Relative errors for deterministic low-rank matrix approximations. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 707–717, USA, 2014. Society for Industrial and Applied Mathematics.

28 Venkatesan Guruswami and Ali Kemal Sinop. Optimal column-based low-rank matrix reconstruction. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1207–1214. SIAM, 2012. doi:10.1137/1.9781611973099.95.

29 Yasutoshi Ida, Sekitoshi Kanai, Yasuhiro Fujiwara, Tomoharu Iwata, Koh Takeuchi, and Hisashi Kashima. Fast deterministic CUR matrix decomposition with accuracy assurance. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4594–4603. PMLR, 13–18 July 2020. URL: http://proceedings.mlr.press/v119/ida20a.html.

30 Edo Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 581–588, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2487575.2487623.

31 Sepideh Mahabadi, Ilya P. Razenshteyn, David P. Woodruff, and Samson Zhou. Non-adaptive adaptive sampling on turnstile streams. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1251–1264. ACM, 2020. doi:10.1145/3357713.3384331.

32 Michael W Mahoney. Randomized algorithms for matrices and data. *arXiv preprint*, 2011. arXiv:1104.5557.

33 Michael W. Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proc. Natl. Acad. Sci. USA*, 106(3):697–702, 2009. doi:10.1073/pnas.0803205106.

34 Rameshwar Pratap, Anup Anand Deshmukh, Pratheeksha Nair, and Tarun Dutt. A faster sampling algorithm for spherical k-means. In Jun Zhu and Ichiro Takeuchi, editors, *Proceedings of The 10th Asian Conference on Machine Learning, ACML 2018, Beijing, China, November 14-16, 2018*, volume 95 of *Proceedings of Machine Learning Research*, pages 343–358. PMLR, 2018. URL: http://proceedings.mlr.press/v95/pratap18a.html.

35 Jeffrey Scott Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985. doi:10.1145/3147.3165.

36 Shusen Wang and Zhihua Zhang. Improving CUR matrix decomposition and the nyström approximation via adaptive sampling. *J. Mach. Learn. Res.*, 14(1):2729–2769, 2013. URL: http://dl.acm.org/citation.cfm?id=2567748.

37 Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1954–1963, 2015.