Subspace Learning with Partial Information

Alon Gonen   ALONGNN@CS.HUJI.AC.IL
School of Computer Science and Engineering
The Hebrew University
Jerusalem, Israel

Dan Rosenbaum   DANRSM@CS.HUJI.AC.IL
School of Computer Science and Engineering
The Hebrew University
Jerusalem, Israel

Yonina C. Eldar   YONINA@EE.TECHNION.AC.IL
Department of Electrical Engineering
Technion, Israel Institute of Technology
Haifa, Israel

Shai Shalev-Shwartz   SHAIS@CS.HUJI.AC.IL
School of Computer Science and Engineering
The Hebrew University
Jerusalem, Israel

Editor: Kevin Murphy

Abstract

The goal of subspace learning is to find a k-dimensional subspace of $\mathbb{R}^d$, such that the expected squared distance between instance vectors and the subspace is as small as possible. In this paper we study subspace learning in a *partial information* setting, in which the learner can only observe $r \leq d$ attributes from each instance vector. We propose several efficient algorithms for this task, and analyze their sample complexity.

Keywords: principal components analysis, budgeted learning, statistical learning, learning with partial information, learning theory

1. Introduction

Subspace learning is a dimensionality reduction technique in a variety of applications such as face recognition (Yang et al., 2004), image compression (Du and Fowler, 2007), and document classification (Papadimitriou et al., 1998). Recently, there has been growing interest in subspace learning from partially observed data (e.g., (Chen et al., 2013; Wang and Poor, 1998; Chi et al., 2013)). As motivation, consider the scenario of subspace learning from corrupted data. As discussed in Chen et al. (2013), data corruption may cause some (or even most) of the attributes to be missing. Another typical scenario is subspace learning from multiple sources. Many applications (e.g., wireless sensor networks (Chi et al., 2013)) rely on data which is collected from multiple sources (e.g., sensors). When the data dimension is high, it may be impossible or prohibitively expensive to collect every data entry from every source. Note that in the first scenario, we have no control over which

attributes are missing, while in the second one a learner may actively choose which attributes to observe.

The subspace learning problem is formally defined as follows. Let $\mathcal{X}$ be a subset of the Euclidean unit ball in $\mathbb{R}^d$, and let P be some unknown distribution over $\mathcal{X}$. Our goal is to find a rank-k projection matrix $\Pi \in \mathbb{R}^{d \times d}$ such that the expected squared distance, $\mathbb{E}_{x \sim P}[\|x - \Pi x\|_2^2]$, is as small as possible.

When $\mathcal{X}$ is a finite set and P is the uniform distribution over $\mathcal{X}$, the optimal solution to the subspace learning problem is given by the Principal Component Analysis (PCA) algorithm, which returns the projection matrix that corresponds to the k leading eigenvectors of the matrix $\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} xx^\top$. In the more general stochastic optimization setting of subspace learning, $\mathcal{X}$ is not restricted to be a finite set, P is an arbitrary distribution over $\mathcal{X}$, and the information given to the learner has the form of an i.i.d. training sequence $(x_1, \ldots, x_m) \sim P^m$.

In the usual full information setting of subspace learning, the learner has access to all attributes of the sampled vectors. In this paper, we consider subspace learning in a partial information setting, in which only a subset of indices from each vector can be observed. Inspired by the two applications presented above, we study two variants of this problem, which will be named the *passive* setting, and the *active* setting, respectively. In the passive setting, we assume that each attribute is observed with probability $p = r/d$ ($r \le d$). Therefore, the expected number of observed attributes from each vector is r. In the active setting, the learner can choose (possibly at random) r attributes to be revealed. The *sample complexity* of a subspace learning algorithm is defined as the number of samples that are needed by the algorithm in order to find a projection matrix Π with expected squared distance, $\mathbb{E}_{x \sim P}[\|x - \Pi x\|_2^2]$, of at most ϵ more than the optimal expected squared distance. The sample complexity for each of the models is defined as the minimal sample complexity attained by any algorithm. In this paper we propose efficient algorithms for both settings and analyze their sample complexity. We also provide several lower bounds on the sample complexity that can be attained by any algorithm.

1.1 Our Contribution

Our first observation is that subspace learning (in both the passive and active models) using $r = 1$ attributes is impossible. Consider the task of learning a 1-dimensional subspace (a line) in $\mathbb{R}^2$ (see Figure 1). Let $u_1 = (\alpha, \alpha)$, $u_2 = (-\alpha, \alpha)$. Denote by P_1 the uniform distribution over $\{u_1, -u_1\}$, and by P_2 be the uniform distribution over $\{u_2, -u_2\}$. Suppose that the actual distribution P is chosen uniformly at random from $\{P_1, P_2\}$. The task of subspace learning in this case amounts to distinguishing between P_1 and P_2. However, since each single coordinate is distributed uniformly over $\{-\alpha, \alpha\}$, the learner does not obtain any information from a single observation, and hence cannot identify the right subspace.

For the case $2 \le r \le d$ we propose two efficient algorithms, named Partially Observed PCA (Section 2.1) and Matrix Bandit Exponentiated Gradient (Section 3.1), which are designed for the passive and active models, respectively.

Partially Observed PCA (POPCA) is an extension of the PCA algorithm. Denote the population covariance matrix $\mathbb{E}[xx^\top]$ by C. The optimal projection matrix, denoted $\Pi^\star$, is the projection onto the subspace that is spanned by the k leading eigenvectors of C. An ERM (empirical risk minimizer, i.e., an algorithm that minimizes the empirical loss) for the full information setting is given by PCA which returns the projection matrix corresponding to the k leading eigenvectors of
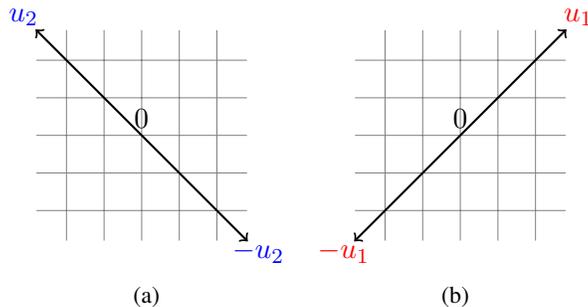
Figure 1: Impossibility of subspace learning using $r = 1$ observed attributes. The distribution of the observed attribute is identical for (a) and for (b), therefore they are indistinguishable.

the matrix $m^{-1} \sum_{i=1}^{m} x_i x_i^\top$ (whose expected value is C). Similarly, the POPCA algorithm uses the random observations in order to construct an estimate $\hat{C}$ of C, and approximates $\Pi^\star$ using the projection matrix onto the k leading eigenvectors of $\hat{C}$. We analyze the sample complexity of this algorithm, showing (see Corollary 3) that it is bounded from above by $(d/r)^2 \frac{k}{\epsilon^2}$.

In the full information setting, the sample complexity is known to be $O(k/\epsilon^2)$ (This bounds coincides with our bound when $r = d$). Hence, the (multiplicative) price of partial information, that is, how many more examples we need in order to compensate for the lack of full information on each individual example, is $O((d/r)^2)$. It is interesting to understand whether a lower price can be achieved. In Theorem 4 we prove that the sample complexity of every algorithm in the passive model is $\Omega\left((d/r)^2 \frac{k}{\epsilon^2}\right)$. The optimality of POPCA in the passive mode is thus established. Another appealing property of POPCA is that, in terms of computational complexity, the challenge of partial information does not incur any additional cost; While the sample complexity grows as r decreases, the runtime per iteration decreases by the same order.

Next, we investigate the active model and ask whether the price of partial information can be reduced due to the ability of the learner to actively choose the observed attributes. Intuitively, one may hope that the learning process would reveal some useful information that can be utilized while choosing which coordinates to observe. Our second algorithm, called Matrix Bandit Exponentiated Gradient (MBEG), exploits the active setting by maintaining a "weight matrix" which is updated with every observation, and induces a non-uniform attribute sampling distribution. For MBEG, we derive an upper bound of $\max\left\{8k \cdot \frac{d+r}{r} \cdot \frac{k}{\epsilon^2} , \frac{d^4 r}{2(d+r)}\right\} \cdot \log(d)$ (see Theorem 7) on the sample complexity. We note that if ϵ is small enough, then the right term in the bound becomes irrelevant, and thus a linear dependence on d/r is obtained (for a detailed comparison, see Section 3.2). The (almost trivial) fact that every subspace learner, even in the active model, must have a sample complexity that grows linearly with d/r is proved in Appendix C. Hence, the dependence of MBEG on d/r is optimal, in the regime where ϵ is small.

The results in the active model immediately lead to the following question: Can we attain a linear price for partial information in the active model, independently of the required accuracy? We discuss possible directions for tackling this question in Section 4.

3

1.2 Related Work

In the full information setting, it has been shown by Shawe-Taylor et al. (2005) and Blanchard et al. (2007) that the optimal sample complexity of subspace learning is at most $O(k/\epsilon^2)$, and this upper bound is achievable by applying PCA on i.i.d. samples according to the distribution P. A similar result is obtained by applying the Stochastic Gradient Descent algorithm (Arora et al., 2013).

Subspace learning in the partial information setting has been studied in Chi et al. (2013), where an algorithm named PETRELS is proposed. However, no formal guarantees are derived for this method. The setting in Mitliagkas et al. (2014) is similar to our passive setting, but they assume that the distribution that generates the instance vectors is Gaussian.

A closely related problem to the task of subspace learning (in the passive setting) is the approximation of the covariance matrix using partially observed attributes. We discusses this relation in Section 2.2.

One possible way to tackle the challenge of subspace learning with partial information is based on the matrix completion method. For example, we may think of the partially observed examples as a data matrix with unobserved entries. Then, one could first fill in the missing entries using a matrix completion technique (e.g., as described in Candès and Recht (2009)), and then apply PCA to the data matrix. We note that this approach may work[1] provided that the average number of observed attributes per example is sufficiently large. More precisely, according to the main result of Candès and Recht (2009), the number of observed attributes per example (column of the data matrix) should scale with the rank of the data matrix (which may be large as d). In contrast, our focus in this paper is on methods that work even when the number of observed attributes per example is much smaller (two attributes per instance suffice).

The active setting resembles the setting of the multi-armed bandit problem (Auer et al., 2002), in which the learner obtains limited feedback at each time, namely, it receives only the reward of the chosen arm. The challenge of learning linear predictors in $\mathbb{R}^d$ with partially observed attributes (e.g., as in Cesa-Bianchi et al. (2011)) may be seen as an extension of this problem. One of the most significant challenges in this work is to adapt the technique used in the vector setting (e.g., those employed by the Exp3 algorithm of Auer et al. (2002)) to the corresponding matrix setting. We already observed one difference: While a single arm suffices in the vector case, subspace learning with $r = 1$ attributes is impossible.

Our MBEG algorithm can be seen as an extension of the Online PCA algorithm of Warmuth and Kuzmin (2008) (see also Nie et al. (2013)) to the partial information setting. Similarly to their approach, MBEG maintains a weight matrix which induces a probability distribution over projection matrices. In MBEG, this weight matrix also induces a distribution over which attributes to observe.

2. The Passive Setting

We begin by investigating the passive setting. We start by describing an algorithm for this case. We then analyze its sample complexity, and discuss its implementation.

2.1 Partially Observed Principal Component Analysis (POPCA)

In this section we describe the POPCA algorithm. We start by reviewing the definition of the loss function and characterizing the minimizer of the loss.

---

1. Under some additional assumptions on the data such as the incoherence assumption made in Candès and Recht (2009)

Denote the set of projection matrices from $\mathbb{R}^d$ onto $\mathbb{R}^k$ by $\mathcal{P}_k^d$. Since $\Pi^2 = \Pi$ for any $\Pi \in \mathcal{P}_k^d$, the loss of a projection matrix $\Pi \in \mathcal{P}_k^d$ can be expressed as

$$L(\Pi) = \mathbb{E}\|x - \Pi x\|_2^2 = \mathbb{E}\left[\|x\|_2^2 - 2x^\top \Pi x + x^\top \Pi^\top \Pi x\right] = \mathbb{E}\left[\|x\|_2^2 - x^\top \Pi x\right], \qquad (1)$$

Define the inner product of matrices A, B by $\langle A, B \rangle = \text{tr}(A^\top B)$. We can further rewrite the loss as

$$L(\Pi) = \mathbb{E}\left[\|x\|_2^2 - \langle \Pi, xx^\top \rangle\right] = \mathbb{E}\left[\|x\|_2^2\right] - \mathbb{E}\left[\langle \Pi, xx^\top \rangle\right] = \mathbb{E}\left[\|x\|_2^2\right] - \langle \Pi, \mathbb{E}[xx^\top]\rangle.$$

Denote the covariance matrix $\mathbb{E}[xx^\top]$ by C. Since $\|x\|_2^2$ does not depend on Π, the goal of subspace learning is equivalent to finding a projection matrix Π such that

$$\langle \Pi, -C \rangle \leq \min_{\Pi' \in \mathcal{P}_k^d} \langle \Pi', -C \rangle + \epsilon. \qquad (2)$$

It is well-known that the rank-k matrix which minimizes the expression $\langle \Pi', -C \rangle$ is the matrix $\sum_{i=1}^k v_i v_i^\top$, where $v_1, \ldots, v_k$ are the leading eigenvectors of C. The PCA approach for subspace learning in the full information setting replaces C with $C(S) = \frac{1}{m} \sum_{i=1}^m x_i x_i^\top$, where $S = (x_1, \ldots, x_m)$ is an i.i.d. training sequence drawn according to the distribution P. Clearly, $\mathbb{E}[C(S)] = C$. In our case, we will construct an unbiased estimate of C based on partially observed examples, as detailed below.

Consider an instance vector $x \sim P$ and let $\hat{x}$ be the observed vector. According to our assumptions, for each $i \in [d]$, the i-th coordinate of $\hat{x}$ satisfies

$$\hat{x}_i = \begin{cases} x_i & \text{w.p. } r/d \\ 0 & \text{w.p. } 1 - r/d. \end{cases} \qquad (3)$$

Denote $p = r/d$. Similarly to Mitliagkas et al. (2014), we form the estimate

$$\hat{C}_{\hat{x}} = \frac{1}{p^2} \hat{x}\hat{x}^\top + \left(\frac{1}{p} - \frac{1}{p^2}\right) \text{diag}(\hat{x}\hat{x}^\top). \qquad (4)$$

Indeed, it is easy to verify that $\hat{C}_{\hat{x}}$ forms an unbiased estimate of C:

$$\mathbb{E}\left[\left(\frac{1}{p^2}\hat{x}\hat{x}^\top\right)_{i,j}\right] = \begin{cases} \frac{1}{p}\mathbb{E}[x_i^2] & i = j \\ \mathbb{E}[x_i x_j] & i \neq j, \end{cases}$$

and

$$\mathbb{E}\left[\left(\frac{1}{p} - \frac{1}{p^2}\right)\text{diag}(\hat{x}\hat{x}^\top)_{i,j}\right] = \begin{cases} \mathbb{E}[x_i^2] - \frac{1}{p}\mathbb{E}[x_i^2] & i = j \\ 0 & i \neq j. \end{cases}$$

Summing the corresponding entries, we see that $\mathbb{E}[(\hat{C}_{\hat{x}})_{i,j}] = \mathbb{E}_{x \sim P}[x_i x_j] = C_{i,j}$. Therefore, $\mathbb{E}[\hat{C}_{\hat{x}}] = C$.

The POPCA algorithm (see Algorithm 1) assumes access to m i.i.d. vectors sampled according to P. For each instance vector, it forms an estimate according to 4. By averaging these m estimates we obtain $\hat{C}$. The returned projection corresponds to the k leading eigenvectors of $\hat{C}$.

5

Algorithm 1 POPCA

> **Input:** $r, k \leq d$
> $\hat{C} = 0 \in \mathbb{R}^{d \times d}$
> **for** $i = 1$ **to** m **do**
>   Let $x_i \sim P$ and let $\hat{x}_i$ be the observed vector
> $\hat{C}_{\hat{x}_i} = \frac{d^2}{r^2} \hat{x}_i \hat{x}_i^\top + \left(\frac{d}{r} - \frac{d^2}{r^2} \right) \mathrm{diag}(\hat{x}_i \hat{x}_i^\top)$
> $\hat{C} = \hat{C} + \frac{1}{m} \hat{C}_{\hat{x}_i}$
> **end for**
> Compute the eigendecomposition $\hat{C} = \sum_{j=1}^d \lambda_j v_j v_j^\top$
> Assuming $\lambda_1 \geq \ldots \geq \lambda_d$, return $\hat{\Pi} = \sum_{j=1}^k v_j v_j^\top$

2.2 Analysis of POPCA

The following lemma relates the success of Algorithm 1 to the quality of the estimation $\hat{C}$ of the covariance matrix C.

Lemma 1 *Suppose that the final estimate $\hat{C}$ of POPCA satisfies*

$$\mathbb{E}[\|C - \hat{C}\|_F] \leq \epsilon/\sqrt{k} \ . \tag{5}$$

Then, the resulting projection matrix $\hat{\Pi}$ satisfies the desired bound

$$\mathbb{E}[\langle \hat{\Pi}, -C \rangle] \leq \min_{\Pi' \in \mathcal{P}_k^d} \langle \Pi', -C \rangle + \epsilon \ .$$

Proof Using the Cauchy-Schwarz inequality, we get

$$
\begin{aligned}
\mathbb{E}[\sup_{\Pi \in \mathcal{P}_k^d} \langle \Pi, -C \rangle - \langle \Pi, -\hat{C} \rangle] &\leq \mathbb{E}[\sup_{\Pi \in \mathcal{P}_k^d} \|\Pi\|_F \|\hat{C} - C\|_F] \\
&\leq \sup_{\Pi \in \mathcal{P}_k^d} \|\Pi\|_F \, \mathbb{E}[\|\hat{C} - C\|_F] \\
&\leq \sqrt{k} \cdot \epsilon/\sqrt{k} \\
&= \epsilon \ .
\end{aligned}
$$

Consequently, since $\hat{\Pi} = \mathrm{argmin}_{\Pi \in \mathcal{P}_k^d} \langle \Pi, -\hat{C} \rangle$, we have

$$
\begin{aligned}
\mathbb{E}[\langle \hat{\Pi}, -C \rangle - \langle \Pi^\star, -C \rangle] &= \mathbb{E}[\langle \hat{\Pi}, -C \rangle - \langle \Pi^\star, -\hat{C} \rangle] \\
&\leq \mathbb{E}[\langle \hat{\Pi}, -\hat{C} \rangle - \langle \Pi^\star, -\hat{C} \rangle] + \epsilon \\
&\leq \epsilon \ ,
\end{aligned}
$$

which completes the proof. ∎

In view of Lemma 1, obtaining upper bound on the sample complexity of POPCA reduces to analyzing the quality of the covariance estimation. A fairly vast body of literature exists on the latter task in the full-information scenario, and several results are known in the case of missing entries. For example, Lounici (2014) considered a setting similar to our passive setting. Corollary 1 in Lounici

(2014) gives a bound on the Frobenius error that depends on the spectral decay of C. We will derive a slightly different (worst-case) bound under the assumption that the instances are bounded in the Euclidean unit ball. A comparison between the two bounds is given Appendix E.

Lemma 2 *Let $\epsilon \in (0, 1)$. If $m \geq \lceil (d/r)^2 \cdot \frac{k}{\epsilon^2} \rceil$, then*

$$\mathbb{E}[\|C - \hat{C}\|_F] \leq \epsilon/\sqrt{k} \ .$$

Proof Using Jensen's inequality, we have

$$\mathbb{E}[\|\hat{C} - C\|_F] = \mathbb{E}[(\|\hat{C} - C\|_F^2)^{1/2}] \leq (\mathbb{E}[\|\hat{C} - C\|_F^2])^{1/2} \ .$$

Since the observations are i.i.d.,

$$\mathbb{E}[\|\hat{C} - C\|_F^2] = \sum_{i,j} \mathrm{Var}[\hat{C}_{i,j}] = \sum_{i,j} \mathrm{Var}\left[(1/m) \sum_{q=1}^{m} \hat{C}_{\hat{x}_q,i,j} \right]$$

$$= \frac{1}{m} \sum_{i,j} \mathrm{Var}[\hat{C}_{\hat{x}_1,i,j}] \leq \frac{1}{m} \sum_{i,j} \mathbb{E}[\hat{C}_{\hat{x}_1,i,j}^2] \ ,$$

where $\hat{C}_{\hat{x}_q,i,j}$ is the $[i, j]$-th entry of $\hat{C}_q$. Denote $\hat{x} = \hat{x}_1$ and $x = x_1$ (subscript indices will now correspond to the entries of these vectors). According to 4,

$$\mathbb{E}[\hat{C}_{\hat{x},i,j}^2 | x] = \begin{cases} p^{-4} \mathbb{E}[\hat{x}_i^2 \hat{x}_j^2 | x] = p^{-2} x_i^2 x_j^2 & i \neq j \\ p^{-2} \mathbb{E}[\hat{x}_i^4 | x] = p^{-1} x_i^4 \leq p^{-2} x_i^4 & i = j \end{cases} .$$

Therefore, since the ℓ_2 norm of the instances is at most 1, we have

$$\sum_{i,j} \mathbb{E}[\hat{C}_{\hat{x}_1,i,j}^2 | x] \leq p^{-2} \sum_{i,j} x_i^2 x_j^2 = p^{-2} \|x\|^4 \leq p^{-2} = \frac{d^2}{r^2} \ ,$$

and $\sum_{i,j} \mathbb{E}[\hat{C}_{\hat{x}_1,i,j}^2] \leq \frac{d^2}{r^2}$. Combining the above bounds, we obtain

$$\mathbb{E}[\|\hat{C} - C\|_F] \leq \frac{1}{\sqrt{m}} \cdot \frac{d}{r} \ .$$

For $m \geq \lceil \frac{d^2}{r^2} \cdot \frac{k}{\epsilon^2} \rceil$, we arrive at the claimed bound. ∎

Let $m_p(d, k, r, \epsilon)$ be the sample complexity of subspace learning in the passive partial information model, namely, how many examples are needed (for the optimal learner) to guarantee that (2) holds. Based on Lemma 2 and Lemma 1, we now conclude the following bound on the sample complexity.

Corollary 3 *Using POPCA (Algorithm 1), we have the following bound on the sample complexity for any integer $r \geq 2$:*

$$m_p(d, k, r, \epsilon) \leq \left\lceil (d/r)^2 \cdot \frac{k}{\epsilon^2} \right\rceil \ .$$

2.3 Optimality of POPCA

In this section we prove the following lower bound on the sample complexity of subspace learning with partial information in the passive model.

Theorem 4 *Assume that $k \leq d/2$ and $\epsilon \in (0, 1/128)$. The sample complexity in the passive model is at least $\Omega\left((d/r)^2 \cdot \frac{k}{\epsilon^2}\right)$. Therefore, we have*

$$m_p(d, k, r, \epsilon) = \Theta\left((d/r)^2 \cdot \frac{k}{\epsilon^2}\right).$$

Note that up to a constant factor, our lower bound coincides with the upper bound obtained by POPCA (Corollary 3). Therefore, Theorem 4 establishes the optimality of POPCA in the passive model.

The proof of Theorem 4 is divided into two parts. First, in Theorem 5 we prove that the sample complexity in the full-information setting is at least $\Omega(k/\epsilon^2)$. Then, we complete the proof of Theorem 4 by showing that the multiplicative price of partial information is at least $\Omega((d/r)^2)$.

Theorem 5 *Assume that $k \leq d/2$ and let $\epsilon \in (0, 1/128)$. The sample complexity of subspace learning with full information is bounded below by*

$$m(d, k, r = d, \epsilon) = \Omega(k/\epsilon^2).$$

We now sketch the proof of Theorem 5. A detailed proof is provided in Appendix A.

Proof (sketch)

The idea is to reduce the problem of coin identification (see Section 5.2 in Anthony and Bartlett (2009)) to that of subspace learning. Assume that $k \leq d/2$ and $\epsilon \in (0, 1/128)$. Let $\mathcal{U} = \{u_j\}_{j=1}^{2k} \subseteq \mathbb{R}^d$ be a set of $2k$ orthonormal vectors. A distribution over $\mathcal{U}$ is defined as follows. First, we draw a sequence $b = (b_1, \ldots, b_k) \in \{-1, 1\}^k$ uniformly at random. We associate the pair $\{u_i, u_{i+k}\}$ with a Bernoulli random variable B_i with parameter $p_i = \frac{1+b_i\alpha}{2}$, where $\alpha = 16\epsilon$. To define a distribution $P := P_b$, we now describe the process of drawing an instance $x \sim P$:

1. An integer $i \in [k]$ is chosen uniformly at random.

2. The i-th coin is flipped (according to p_i). Denote the corresponding random variable by Z.

3. If $Z = 1$, then x is chosen uniformly at random from the set $\{u_i, -u_i\}$. Otherwise ($Z = 0$), x is chosen uniformly at random from the set $\{u_{i+k}, -u_{i+k}\}$.

In Lemma 10 we show that a successful subspace learner must identify the bias of "most" of the coins. Thus, we can reduce k independent tasks of coin identification to the task of subspace learning. A well-known result in statistics (Anthony and Bartlett, 2009)[Lemma 5.1] tells us that $\Omega(1/\alpha^2)$ samples are needed to identify a coin with bias α. Hence, each of the pairs must be observed $\Omega(1/\epsilon^2)$ times. ∎

Proof (of Theorem 4) Consider the construction presented in the proof (sketch) of Theorem 5. We next specify the set $\mathcal{U}$ and prove that the price of partial information in the passive model is $\Omega(d^2/r^2)$. Consequently, this will conclude the proof of Theorem 4.

For every $i \in [k]$, let $u_i = \frac{\sqrt{2}}{2}(e_i + e_{i+k})$, $u_{i+k} = \frac{\sqrt{2}}{2}(-e_i + e_{i+k})$. To specify a distribution P, fix a vector $b = (b_1, \ldots, b_k) \in \{-1, 1\}^k$. Consider now a single interaction between the learner and the environment. A vector x is drawn according to P as described in the proof (sketch) of Theorem 5. Let $i \in [k]$ denote the index of the coin which is associated with x. As we observed in Section 1, each of the coordinates i, $i + k$ is distributed uniformly over $\{-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\}$. Hence, to obtain any information, the learner must observe both of the coordinates i and $i + k$. The probability that both coordinates are observed is at most $O(r^2/d^2)$. Hence, in expectation, (only) a single "meaningful" observation is obtained every $\Omega(d^2/r^2)$ iterations. Therefore, the price of partial information is $\Omega(d^2/r^2)$. ∎

2.4 Implementation of POPCA

As we discussed above, when the data is fully visible, the sample complexity of the PCA algorithm (which computes the k leading eigenvectors of the empirical covariance matrix, $m^{-1} \sum_{i=1}^{m} x_i x_i^{\top}$, and returns the corresponding projection matrix) is $m_f := O(k/\epsilon^2)$. When the number of samples, m_f, is larger than the dimension, a standard implementation of this algorithm costs $O(m_f d^2)$. We next show that POPCA has a similar runtime.

We established above that POPCA requires a training set of size $O((d/r)^2 m_f)$. Consider a single iteration of POPCA. Note that the construction of $\hat{C}_{\hat{x}_i}$ (see 4) costs $O(r^2)$. Therefore, it costs $O(m_f d^2)$ to obtain the average of the estimates, $\hat{C}$. The computation of the SVD of $\hat{C}$ costs $O(d^3)$. Therefore, when $m_f \geq d$, the overall runtime of POPCA is indeed $O(m_f d^2)$.

3. The Active Setting

We now consider the active model of subspace learning with partial information. In particular, we will present and analyze the Matrix Bandit Exponentiated Gradient (MBEG) algorithm.

Before describing MBEG, it should be noticed that POPCA (along with its analysis) can be modified to fit the active model; An active version of POPCA simply selects the r observed attributes uniformly at random (with replacement) and then proceeds similarly to POPCA[2]. It is not hard to verify that the sample complexity of this algorithm is asymptotically equivalent to the sample complexity of POPCA. In particular, the price of partial information is quadratic in d/r.

The main differences between MBEG and POPCA are as follows:

1. MBEG is designed for the active model. It employs a non-uniform sampling method which gives higher priority to "stronger" directions.

2. MBEG is an iterative algorithm which maintains a *weight matrix* that belongs to the convex hull of the set $\mathcal{P}_k^d$ of projection matrices. It can be thought of as a Bandit version of the extension of the Exponentiated Gradient (EG) algorithm to matrices. The EG algorithm, and its extension to matrices are due to Kivinen and Warmuth (1997), and Tsuda et al. (2005), respectively.

---

2. Note that here the attributes are no longer independent and therefore we should replace the weights in the definition of $\hat{C}_{\hat{x}}$.

3.1 Matrix Bandit Exponentiated Gradient (MBEG)

3.1.1 CONVEXIFICATION

In order to be able to apply the Matrix EG algorithm, we first need to formulate our task as a convex optimization problem. Recall that our problem is equivalent to approximately minimizing the objective $\operatorname{argmin}_{\Pi \in \mathcal{P}_k^d} \langle \Pi, -C \rangle$, where $C = \mathbb{E}[xx^\top]$. The objective is linear and thus convex. We will replace the non-convex set $\mathcal{P}_k^d$ with its convex hull, $\mathcal{C}_k^d := \operatorname{conv}(\mathcal{P}_k^d)$. Note that for every W in $\mathcal{C}_k^d$, the gradient is given by $-C$. Therefore, the EG procedure would start with some $W_1 \in \mathcal{C}_k^d$, and, at iteration i, would update according to

$$
\begin{aligned}
&1. \quad U_{i+1} = \exp(\log(W_i) + \eta C) \\
&2. \quad W_{i+1} = \operatorname*{argmin}_{W \in \mathcal{C}_k^d} D_R(W, U_{i+1}) \,,
\end{aligned}
\tag{6}
$$

where $D(R, U) = \operatorname{tr}(W \log W - W \log U - W + U)$ is the Bregman divergence induced by the *quantum entropy* regularizer, $R(W) = \operatorname{tr}(W \log W - W)$. Additional details regarding this regualarizer can be found in Tsuda et al. (2005) and Warmuth and Kuzmin (2008). As in POPCA, the gradient C is unknown but can be estimated. We would like to exploit the active setting, and therefore, we will employ a non-uniform sampling method which relies on the current weight matrix W_t (see Section 3.1.2).

Next, we recall that the output of the algorithm should be a projection matrix, while our algorithm maintains weight matrices, which may not belong to the set $\mathcal{P}_k^d$. Therefore, the final step of the algorithm is to construct an element from $\mathcal{P}_k^d$ that performs "similarly" to the average of the weight matrices maintained during the run of the algorithm. For this purpose, we rely on the following lemma, due to Warmuth and Kuzmin (2008):

Lemma 6 *Every matrix $W \in \mathcal{C}_k^d$ can be decomposed in time $O(d^3)$ into a convex combination of at most d elements from $\mathcal{P}_k^d$.*

For completeness, we recall the decomposition procedure of Warmuth and Kuzmin (2008) in Appendix D. Getting back to our algorithm, let $\hat{W} = \frac{1}{m} \sum_{i=1}^m W_i$ be the average weight matrix, and denote by $\hat{W} = \sum_{j=1}^d \beta_j \Pi_j$ a decomposition of $\hat{W}$ into a convex combination of elements from $\mathcal{P}_k^d$. The final step of MBEG sets the output matrix $\hat{\Pi}$ to be Π_j with probability β_j. This guarantees that the expected performance of $\hat{\Pi}$ is the same as the performance of $\hat{W}$.

3.1.2 PRIORITIZING "STRONGER" DIRECTIONS

In this part we present the non-uniform sampling mechanism which is employed by MBEG for attribute sampling. We first consider the case $r = 2$. Let $W := W_i$ be a weight matrix obtained during the run of MBEG. Recall[3] that $\sum_i W_{i,i} = k$, and for every $i \in [d]$, $W_{i,i} \in [0, 1]$. Therefore, a natural distribution for attribute sampling is to choose each pair $(s, q) \in [d]^2$ with probability $p_{s,q} = \frac{W_{s,s}}{k} \cdot \frac{W_{q,q}}{k}$. Unfortunately, we were not able to obtain a good bound using this sampling technique. Instead, we pick attributes according to

$$
p_{s,q} = (1 - \alpha) \frac{W_{s,s} + W_{q,q}}{2dk} + \frac{\alpha}{d^2} \,,
\tag{7}
$$

---

3. These properties clearly hold for projection matrices, and thus hold for any convex combination of projection matrices.

for some parameter $\alpha \in (0, 1/2)$, which is tuned below. That is, we mix a uniform distribution over $[d]^2$, with a distribution which gives higher sampling probability to pairs for which $W_{s,s}, W_{q,q}$ are high, reflecting a bias toward sampling from "stronger" directions. Mixing with the uniform distribution guarantees that every pair has large enough probability to be sampled, which will later help us ensure that we perform enough "exploration".

Based on this probability distribution over pairs, we define an unbiased estimate of the matrix C by

$$\hat{C} = \frac{1}{2p_{s,q}} x_s x_q (E_{s,q} + E_{q,s}) , \tag{8}$$

where $E_{s,q}$ is the all zeros matrix except 1 in the i, j coordinate.

The extension of MBEG to any (even) $r > 2$ is straightforward. We simply pick $r/2$ independent estimates $\hat{C}_1, \ldots, \hat{C}_{r/2}$, each of which is constructed as in the case $r = 2$, and set $\hat{C} = \frac{2}{r} \sum_{j=1}^{r/2} \hat{C}_j$.

The algorithm is summarized in Algorithm 2. As explained in Tsuda et al. (2005), the projection step w.r.t. the Bregman divergence, $W_{i+1} = \operatorname{argmin}_{W \in \mathcal{C}_k^d} D_R(W, U_{i+1})$ can be performed in time $O(d^3)$. Overall, the running time per iteration is $O(d^3)$.

Algorithm 2 Matrix Bandit Exponentiated Gradient

Input: $r, k < d \ (r \mod 2 = 0)$
$\eta = \sqrt{\frac{r \log(d)}{2m(d+r)}}$
$\alpha = \eta d^2$ (we assume that m is large enough so that $\alpha \leq 1/2$)
$W_1 = \frac{k}{d} I$
for $i = 1$ **to** m **do**
    denote $W = W_i$
    for $j = 1$ **to** $r/2$ **do**
        pick $(s, q) \in [d]^2$ with probability $p_{s,q} = (1 - \alpha) \frac{W_{s,s} + W_{q,q}}{2dk} + \frac{\alpha}{d^2}$
        for $x \sim P$, let (x_s, x_q) be the corresponding attributes
        $\hat{C}_{i,j} = \frac{1}{2p_{s,q}} x_s x_q (E_{s,q} + E_{q,s})$
    end for
    $\hat{C}_i = \frac{2}{r} \sum_{j=1}^{r/2} \hat{C}_{i,j}$
    $U_{i+1} = \exp(\log(W) + \eta \hat{C}_i)$
    $W_{i+1} = \operatorname{argmin}_{W \in \mathcal{C}_k^d} D_R(W, U_{i+1})$
end for
$\hat{W} = \frac{1}{m} \sum_{i=1}^m W_i$
decompose $\hat{W}$ into $\hat{W} = \sum_{j=1}^d \beta_j \Pi_j$ using Algorithm 3
return $\hat{\Pi} = \Pi_j$ with probability β_j

3.1.3 ANALYSIS OF MBEG

Let $m_a(d, k, r, \epsilon)$ be the sample complexity of subspace learning in the active partial information model. We denote the Frobenius norm, the spectral norm, and the trace norm by $\| \cdot \|_F$, $\| \cdot \|_{\mathrm{sp}}$, and $\| \cdot \|_{\mathrm{tr}}$, respectively. Throughout this section we prove the following result.

Theorem 7 *Using MBEG (Algorithm 2), we have the following bound on the sample complexity for any (even) integer $r \geq 2$:*

$$m_a(d, k, r, \epsilon) \leq \max \left\{ 8k \cdot \frac{d+r}{r} \cdot \frac{k}{\epsilon^2} , \frac{2d^4 r}{d+r} \right\} \cdot \log(d) .$$

In order to prove Theorem 7, we next apply the general analysis of Matrix EG (Hazan et al., 2012)[Theorem 13] to our case.

Theorem 8 *Assume that the sequence $\hat{C}_1, \ldots, \hat{C}_m$ obtained during the run of MBEG satisfies $\|\hat{C}_i\|_{\mathrm{sp}} \leq 1/\eta$ for every $i \in [m]$. Then, for every $\Pi^\star \in \mathcal{P}_k^d$,*

$$\sum_{i=1}^m \langle W_i - \Pi^\star, -\hat{C}_i \rangle \leq \frac{k \log(d)}{\eta} + \eta \sum_{i=1}^m \langle W_i, \hat{C}_i^2 \rangle . \tag{9}$$

The right-most term in 9 can be thought as the variance which is associated with the estimation process of MBEG. We now show that in contrast to POPCA, the variance scales only linearly with d/r.

Lemma 9 *For any matrix $W_i \in \mathcal{C}_k^d$ maintained by MBEG at time i, denote the conditional expectation given W_i by $\mathbb{E}_i$. Then,*

$$\mathbb{E}_i \langle W_i, \hat{C}_i^2 \rangle \leq \frac{2k(d+r)}{r} .$$

We now prove the lemma while assuming that $r = 2$. The extension to any $r > 2$ is detailed in Appendix B.

Proof Let $W = W_i, \hat{C} = \hat{C}_i$. Then,

$$\begin{aligned}
\mathbb{E}_i \langle W, \hat{C}^2 \rangle &= \sum_{(s,q) \in [d]^2} p_{s,q}(W_{s,s} + W_{q,q}) \frac{1}{4p_{s,q}^2} x_s^2 x_q^2 \\
&= \sum_{(s,q) \in [d]^2} \frac{(W_{s,s} + W_{q,q})}{4p_{s,q}} x_s^2 x_q^2 .
\end{aligned}$$

Since $\alpha \in (0, 1/2)$, we have

$$\frac{W_{s,s} + W_{q,q}}{p_{s,q}} = \frac{W_{s,s} + W_{q,q}}{(1-\alpha)\frac{W_{s,s}+W_{q,q}}{2dk} + \alpha/d^2} \leq \frac{W_{s,s} + W_{q,q}}{(1-\alpha)\frac{W_{s,s}+W_{q,q}}{2dk}} \leq 4dk .$$

Therefore, since $r = 2$, we obtain

$$\mathbb{E}_i \langle W, \hat{C}^2 \rangle \leq dk \sum_{(s,q) \in [d]^2} x_s^2 x_q^2 \leq dk < \frac{2k(d+r)}{r} ,$$

completing the proof of the lemma. ∎

Proof (of Theorem 7) By definition of $\hat{C}_{i,j}$ we have that $\hat{C}_{i,j}^2 = \hat{C}_{i,j}\hat{C}_{i,j}^\top$ is a diagonal matrix with all elements on the diagonal equal to zero except the (s,s) and (q,q) elements, which are both bounded above by $\frac{x_s^2 x_q^2}{p_{s,q}^2}$. It follows that

$$\|\hat{C}_{i,j}\|_{\mathrm{sp}} \leq \frac{|x_s x_q|}{p_{s,q}} \leq \frac{1}{p_{s,q}} \leq \frac{d^2}{\alpha} = \frac{1}{\eta} .$$

Hence,

$$\|\hat{C}_i\|_{\mathrm{sp}} \leq \frac{2}{r} \sum_{j=1}^{r/2} \|\hat{C}_{i,j}\|_{\mathrm{sp}} \leq \frac{2}{r}\frac{r}{2}\frac{1}{\eta} = \frac{1}{\eta} .$$

Therefore, the conditions of Theorem 8 hold. Taking expectation over 9, we obtain

$$\sum_{i=1}^m \mathbb{E}\left\langle W_i - \Pi^\star, -\hat{C}_i\right\rangle \leq \frac{k\log(d)}{\eta} + \eta\sum_{i=1}^m \mathbb{E}\left\langle W_i, \hat{C}_i^2\right\rangle . \tag{10}$$

Let $\mathbb{E}_i$ denote the conditional expectation given W_i. Then, $\mathbb{E}_i[\hat{C}_i] = C$ and therefore, by the law of total expectation,

$$\mathbb{E}\left\langle W_i - \Pi^\star, -\hat{C}_i\right\rangle = \mathbb{E}\left\langle W_i - \Pi^\star, -C\right\rangle . \tag{11}$$

Combining 11 with Lemma 9, and plugging into 10, we obtain that

$$\mathbb{E}\sum_{i=1}^m \left\langle W_i - \Pi^\star, -C\right\rangle \leq \frac{k\log(d)}{\eta} + \eta m\frac{2k(d+r)}{r} .$$

Dividing by m, denoting $\hat{W} = \frac{1}{m}\sum_{i=1}^m W_i$, substituting $\eta = \sqrt{\frac{r\log(d)}{2(d+r)m}}$, rearranging terms, and observing that $\mathbb{E}[\hat{\Pi}|\hat{W}] = \hat{W}$, we have that

$$\mathbb{E}\left\langle\hat{\Pi} - \Pi^\star, -C\right\rangle = \left\langle\mathbb{E}[\hat{W}] - \Pi^\star, -C\right\rangle \leq \sqrt{\frac{8k^2(d+r)\log(d)}{rm}} .$$

The right-hand side of the above is smaller than ϵ if $m \geq 8k\log(d)\cdot\frac{d+r}{r}\cdot\frac{k}{\epsilon^2}$. Note, however, that we also require that m is large enough so that $\alpha \leq 1/2$. Since $\alpha = \eta d^2$, it follows that m should also satisfy $m \geq \frac{2d^4 r\log(d)}{d+r}$. ∎

3.2 Comparison between the bounds in the passive and the active models

The left term in the bound of MBEG is smaller than the bound of POPCA if $\frac{d^2}{r^2} > 8k\log(d)\cdot\frac{d+r}{r}$. The right term in the bound of MBEG is smaller than the bound of POPCA if

$$\epsilon < \sqrt{\frac{(d+r)k}{2d^2 r^3 \log(d)}} .$$

Hence, if both of the conditions hold, then the bound of MBEG is better than the bound of POPCA. Also, if $\epsilon < \frac{2(d+r)k}{d^2 r}$, then the dependence of the sample complexity of MBEG on d/r is linear.

A reasonable regime in which MBEG enjoys a linear price is when r and k are constants, and ϵ is proportional to $1/d$. Note that in this regime, the bound of POPCA scales with d^4, while the bound of MBEG scales only with d^3 (in the full-information setting, the sample complexity for this case scales with d^2). To summarize, ignoring the dependence on k (and logarithmic factors), MBEG attains the desired linear price when ϵ is 'small'.

4. Discussion

We introduced the problem of subspace learning with partial information, and considered both a passive and active model. Our first observation was that looking at a single coordinate does not give any information. Therefore, our algorithms look at the products of attribute pairs. Using the POPCA algorithm for the passive model, we showed that the sample complexity is tightly characterized by $\Theta((d/r)^2 k/\epsilon^2)$. Hence, the price of partial information in this case is quadratic. For the active model we introduced the MBEG algorithm which exploits the gathered information in order to make a better choice over which attributes to observe. We showed that if the desired accuracy ϵ is small, then MBEG achieves a linear price. Since the expected number of observed attributes from each vector is r, we can not hope for a better dependence on d/r. At this point, a natural question arises: can we attain a linear price of partial information in the active model, independently of the required accuracy? We conclude with an observation regarding MBEG that provides a partial answer to this question.

Assume that ϵ, r and k are constants. Examining the implementation of MBEG, we note that as long as the products, $x_s x_q$, between two consecutive observed attributes is zero, MBEG does not change the sampling distribution (which is initially uniform) nor its current estimation of the covariance matrix. Fix some attribute j and consider the distribution P_j which is concentrated on e_j. The probability that the product between two consecutive observed attributes is not zero is $(1/d)^2$. Since r is constant, only zero products are observed for at least $\Omega(d^2)$ iterations, implying the same bound on the sample complexity.

The implication of this result is that in order to achieve a linear price for any value of ϵ, it is necessary to extract more information from the partial observations (e.g., take into account both the products of attribute pairs and the single attributes). Developing such algorithms or alternatively, tightening the lower bounds, is left for future research.

Acknowledgments

Appendix A. Proof of Theorem 5

In this section we prove Theorem 5.

A.1 The adversarial distribution

Let $\mathcal{U} = \{u_j\}_{j=1}^{2k} \subseteq \mathbb{R}^d$ be a set of $2k$ orthonormal vectors. In the proof sketch of Theorem 5 we described a family $\mathcal{F}$ of distributions over the set $\{u, -u : u \in \mathcal{U}\}$. Our lower bounds will be proved to hold in expectation when choosing $P \in \mathcal{F}$ at random. According to Yao's minimax principle, such a result implies that the lower bound holds for some distribution in $\mathcal{F}$.

A.2 A Successful Subspace Learner Is Also a Successful Coin identifier

In this part we formalize the reduction from coin identification to subspace learning. As we sketched before, a key ingredient of our analysis of the lower bounds is the relation between subspace learning and "coin identification". This relation is formalized next. We first need to introduce some additional notation. To specify the distribution $P \in \mathcal{F}$, fix a vector $(b_1, \ldots, b_k) \in \{-1, 1\}^k$. Let $\hat{\Pi} = \sum_{i=1}^k \hat{u}_i \hat{u}_i^\top \in \mathcal{P}_k^d$ (where $\hat{u}_1, \ldots, \hat{u}_k$ are orthonormal). For every $(i, j) \in [k] \times [2k]$, define $\theta_{i,j} = |\langle \hat{u}_i, u_j \rangle|$ to be the *covariance* between $\hat{u}_i$ and u_j. Next, for each $j \in [2k]$, define $\theta_j^2 = \sum_{i=1}^k \theta_{i,j}^2$. Also, define the set

$$J = \left\{ j \in [k] : b_j = 1 \wedge \theta_j^2 > \theta_{j+k}^2 \right\} \cup \left\{ j \in [k] : b_j = -1 \wedge \theta_j^2 < \theta_{j+k}^2 \right\}. \tag{12}$$

For reasons that will become apparent shortly, we name J the set of identified coins. The following lemma asserts that a successful subspace learner must identify most of the coins.

Lemma 10 *Let $\epsilon \in (0, 1)$. Assume that $L(\hat{\Pi}) - \min_{\Pi \in \mathcal{P}_k^d} L(\Pi) \leq \epsilon$. Let J be defined as in 12. Then, $|J|/k > 1 - \frac{2\epsilon}{\alpha}$.*

Proof Assume w.l.o.g. that $b_1 = \ldots = b_k = 1$. Note that

$$\mathbb{E}[xx^\top] = \frac{1}{k} \sum_{j=1}^k \left(\frac{1+\alpha}{2} u_j u_j^\top + \frac{1-\alpha}{2} u_{j+k} u_{j+k}^\top \right).$$

The optimal projection is obtained by picking the largest eigenvectors of $\mathbb{E}[xx^\top]$. Precisely, the optimal projection matrix is $\Pi^\star = \sum_{i=1}^k u_i u_i^\top$. We assume the loss function as formulated in 1.

The loss of $\hat{\Pi}$ is calculated as follows:

$$
\begin{aligned}
\mathbb{E}[\|x\|_2^2] - L(\hat{\Pi}) &= \langle \sum_{i=1}^{k} \hat{u}_i \hat{u}_i^\top, \ \mathbb{E}[xx^\top] \rangle \\
&= \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{k} \left(\frac{1+\alpha}{2} \langle \hat{u}_i \hat{u}_i^\top, u_j u_j^\top \rangle + \frac{1-\alpha}{2} \langle \hat{u}_i \hat{u}_i^\top, u_{j+k} u_{j+k}^\top \rangle \right) \\
&= \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{k} \left(\frac{(1+\alpha)\theta_{i,j}^2}{2} + \frac{(1-\alpha)\theta_{i,j+k}^2}{2} \right) \\
&= \frac{1}{k} \sum_{j=1}^{k} \left(\frac{(1+\alpha)\theta_j^2}{2} + \frac{(1-\alpha)\theta_{j+k}^2}{2} \right) \\
&= \frac{1}{2k} \sum_{j=1}^{2k} \theta_j^2 + \frac{\alpha}{2k} \sum_{j=1}^{k} (\theta_j^2 - \theta_{j+k}^2) \ .
\end{aligned}
\tag{13}
$$

Since $\{u_j\}_{j=1}^{2k}$ is orthonormal, for each $i \in [k]$ we have $\sum_{j=1}^{2k} \theta_{i,j}^2 \leq 1$. Hence, $\sum_{j=1}^{2k} \theta_j^2 \leq k$, so that the second-to-last term of 13 is at most $\frac{1}{2}$. This value is attained by $\Pi^\star$, and for simplicity, we will assume that is attained by $\hat{\Pi}$ as well. For the same reasons, for each $i \in [k]$, we have $\sum_{j=1}^{k} (\theta_{i,j}^2 - \theta_{i,j+k}^2) \leq 1$. Hence, $\sum_{j=1}^{k} (\theta_j^2 - \theta_{j+k}^2) \leq k$, so that the last term of 13 is at most $\alpha/2$. Once again, this value is attained by $\Pi^\star$. Our last observation is that if the j-th coin is not identified, i.e., $j \notin J$, then $\theta_j^2 - \theta_{j+k}^2 \leq 0$. Combining these observations, we obtain that

$$L(\hat{\Pi}) - L(\Pi^\star) \geq \alpha(1 - |J|/k)/2 \ .$$

The proof is completed by combining the assumption that $L(\hat{\Pi}) - L(\Pi^\star) \leq \epsilon$. ∎

A.2.1 LOWER BOUND ON COIN IDENTIFICATION

We previously informally argued that $\Omega(1/\alpha^2)$ samples are needed to identify a coin with bias $\frac{1\pm\alpha}{2}$. If we have k such independent coins, then $\Omega(k/\alpha^2)$ are needed. Let us formalize this result.

A coin identification problem with parameter α is defined as follows. Consider a binary classification problem with a domain $[k]$ and label set $\{0, 1\}$. The hypothesis class is the set $\mathcal{H} = \{0, 1\}^{[k]}$. The underlying distribution over $[k] \times \{0, 1\}$ is chosen at random using the following mechanism. The marginal distribution over $[k]$ is uniform, and the conditional probability over the label (coin) is determined by $P(y = 1|x = j) = \frac{1+b_j\alpha}{2}$, where each b_j is an independent Rademacher random variable (drawn in advance). We observe that this distribution is identical to the distribution defined (over a shattered set of size k) in (Anthony and Bartlett, 2009, Theorem 5.2).

As usual, an algorithm for coin identification obtains an i.i.d. training sequence S according to P, and has to return a hypothesis $h \in \mathcal{H}$. The generalization error of $h \in \mathcal{H}$, denoted err(h), is defined to be the probability that it misclassifies a new generated point (x, y), i.e., err(h) = $P(h(x) \neq y)$. The next theorem follows from (Anthony and Bartlett, 2009, Theorem 5.2).

Theorem 11 *Let $\mathcal{B}$ be an algorithm for coin identification, and let $\tilde{\epsilon} \in (0, 1/64)$. Consider a coin identification problem with $\alpha = 8\tilde{\epsilon}$. If $m < \frac{k}{8\tilde{\epsilon}^2}$, then there exists a distribution P for which*

$$\mathbb{E}_{S \sim P^m}\, err(\mathcal{B}(S)) - \min_{h \in \mathcal{H}} err(h) > \tilde{\epsilon}\,.$$

A.3 Concluding the Theorem

We are now in position to complete the reduction from the task of coin identification to the task of subspace learning, and consequently conclude Theorem 5.

Let $\mathcal{A}$ be a subspace learner whose sample complexity is $m(d, k, r = d, \epsilon)$. We describe an algorithm $\mathcal{B}$ for coin identification (with parameter α) which uses $\mathcal{A}$ as a subroutine. To this end, we shall provide a (randomized) map between the input of $\mathcal{B}$ to the input of $\mathcal{A}$. These inputs have the form of training sequences, which will be denoted by S_{subspace} and S_{coins}, respectively. For each $j \in [k]$, the pair $(j, 1)$ is associated with u_j or $-u_j$ with equal probability. The pair $(j, 0)$ is associated with u_{j+k} of $-u_{j+k}$ with equal probability. Clearly, the sequence provided to $\mathcal{A}$ is generated according to the distribution discussed in the proof sketch of Theorem 5. Given an accuracy parameter $\tilde{\epsilon} \in (0, 1/64)$ for $\mathcal{B}$, we will require accuracy $\epsilon = \tilde{\epsilon}/2$ from $\mathcal{A}$. To complete the reduction, we will specify how the output of $\mathcal{B}$ is determined using the output of $\mathcal{A}$.

For each $j \in [k]$, the output hypothesis returned by $\mathcal{B}$ is defined by

$$h(j) = \begin{cases} 1 & \theta_j > \theta_{j+k} \\ 0 & \text{otherwise} \end{cases}\,.$$

Denote the set of identified coins by J (as in Lemma 10). Observe that each coin that is not identified, adds α/k to the relative error of $\mathcal{B}$. It follows from Lemma 10 that

$$\text{err}(\mathcal{B}(S_{\text{coins}})) - \min_{h \in \mathcal{H}} \text{err}(h) = \alpha(1 - |J|/k)$$

$$\leq 2\left(L(\mathcal{A}(S_{\text{subspace}})) - \min_{\Pi \in \mathcal{P}_k^d} L(\Pi)\right)\,. \tag{14}$$

If $m \geq m(d, k, r = d, \epsilon)$, then the right-hand side of 14 is at most $2\epsilon = \tilde{\epsilon}$. The proof is completed by applying Theorem 11 (with $\alpha = 8\tilde{\epsilon}$).

Appendix B. Extending Lemma 9 to r > 2

We will now extend the proof of Lemma 9 to any (even) $r > 2$. Fix an iteration i, and denote $\hat{C} = \hat{C}_i, W = W_i$. We have

$$\mathbb{E}_i \langle W, \hat{C}^2 \rangle = \frac{4}{r^2}\left[\sum_{j=1}^{r/2} \mathbb{E}_i \langle W, \hat{C}_j^2 \rangle + \sum_{j \neq t} \mathbb{E}_i \langle W, \hat{C}_j \hat{C}_t \rangle\right]\,.$$

From the case $r = 2$, we already know that the term $\mathbb{E}_i \langle W, \hat{C}_j^2 \rangle$ is at most dk. Fix some pair $j \neq t$. Note that $\hat{C}_j \hat{C}_t \neq 0$ if and only if (at least) one of the indices s_j, q_j is equal to one of the indices

s_t, q_t. Hence,

$$\mathbb{E}_i \langle W, \hat{C}_j \hat{C}_t \rangle \leq 2 \sum_{(s,q) \in [d]^2} p_{s,q}^2 (W_{s,s} + W_{q,q}) \frac{x_s^2 x_q^2}{4 p_{s,q}^2}$$

$$+ 4 \sum_{\substack{(s,q,q') \in [d]^3: \\ q \neq q'}} p_{s,q} p_{s,q'} W_{q,q'} \frac{x_s^2 |x_q| |x_q'|}{4 p_{s,q} p_{s,q'}} \ .$$

The first term can be bounded as follows:

$$2 \sum_{(s,q) \in [d]^2} p_{s,q}^2 (W_{s,s} + W_{q,q}) \frac{x_s^2 x_q^2}{4 p_{s,q}^2} = \frac{2}{4} \sum_{(s,q) \in [d]^2} (W_{s,s} + W_{q,q}) x_s^2 x_q^2$$

$$= \frac{2}{4} \left\langle (W_{s,s} + W_{q,q})_{(s,q) \in [d]^2}, (x_s^2 x_q^2)_{(s,q) \in [d]^2} \right\rangle$$

$$\leq \frac{2}{4} \| (W_{s,s} + W_{q,q})_{(s,q) \in [d]^2} \|_\infty \| (x_s^2 x_q^2)_{(s,q) \in [d]^2} \|_1$$

$$\leq \frac{2}{4} \cdot 2 = 1 \ .$$

The second term can be bounded as:

$$4 \sum_{\substack{(s,q,q') \in [d]^3: \\ q \neq q'}} p_{s,q} p_{s,q'} W_{q,q'} \frac{x_s^2 |x_q| |x_q'|}{4 p_{s,q} p_{s,q'}} \leq 4 \cdot \frac{1}{4} \left(\sum_{s \in [d]} x_s^2 \right) \left(\sum_{(q,q') \in [d]^2} |W_{q,q'}| \cdot |x_q x_{q'}| \right)$$

$$\leq \|W\|_{\mathrm{tr}} \|xx^\top\|_{\mathrm{sp}}$$

$$\leq k \ .$$

Combining the above, we obtain

$$\mathbb{E}_i \langle W, \hat{C}^2 \rangle = \frac{4}{r^2} \left[\frac{r}{2} dk + \frac{r}{2} \left(\frac{r}{2} - 1 \right) (k+1) \right]$$

$$\leq \frac{2dk}{r} + k + 1$$

$$\leq \frac{2dk}{r} + 2k$$

$$= \frac{2k(d+r)}{r} \ .$$

Appendix C. The Price of Partial Information is at Least Linear

Theorem 12 *Let $k = 1$. Then, the sample complexity of subspace learning is bounded below by:*

$$m(d, k = 1, r, \epsilon) \geq \Omega \left(\frac{d}{r \epsilon} \right) \ .$$

Proof Let $\epsilon \in (0, 1/4)$, and let $\mathcal{A}$ be a bandit subspace learner. For each $s \in [d]$, we define a distribution P_s as follows. The zero vector is drawn with probability $1 - c\epsilon$ (where $c > 2$), and the

vector e_s is drawn with probability $c\epsilon$. Note that $\mathbb{E}[xx^\top] = c\epsilon \, e_s e_s^\top$, and thus the optimal projection is given by $\Pi^\star := e_s e_s^\top$.

We next show that a successful subspace learner must identify the distribution. Let P_s be a concrete distribution. Denote by $\hat{\Pi} = \hat{u}\hat{u}^\top$ the output of the learner. Define $\theta_s = u_s^2$. It can be easily seen that if $L(\hat{\Pi}) - L(\Pi^\star) < \epsilon < c\epsilon/2$, then $\theta_s > \theta_q$ for any $q \neq s$. That is, a successful subspace learner must identify the index s.

Next we observe that if the size of the sample is at most $o\left(\frac{d}{r\epsilon}\right)$, then with a non-negligible probability, all the observations made by the learner are equal to zero. It follows that there exists a distribution P_j for which, $\mathbb{E}[L(\hat{\Pi}) - L(\Pi^\star)] > c\epsilon/2 > \epsilon$. $\blacksquare$

Appendix D. Decomposing Elements in $\mathcal{C}_k^d$ into a Convex Combination of Elements From $\mathcal{P}_k^d$

In this part we detail the decomposition procedure mentioned in Lemma 6. For a symmetric $d \times d$ matrix A, we denote its eigenvalues by $\lambda(A)_1 \geq \ldots \geq \lambda(A)_d$. For convenience, we define two subroutines. The procedure $\mathrm{sort}(x, s)$ returns a set of s indices, corresponding to the s largest values of x. Given a vector $x \in \mathbb{R}^d$, the function $\mathrm{diag} : \mathbb{R}^d \to \mathbb{R}^{d \times d}$ returns a diagonal matrix X with $X_{j,j} = x_j$.

Algorithm 3 Decomposition Procedure

Input: $W \in \mathcal{C}_k^d = \mathrm{conv}(\mathcal{P}_k^d)$
perform eigendecomposition $W = U\mathrm{diag}(\lambda(A))U^\top$
$\lambda := k^{-1}(\lambda(A)_1, \ldots, \lambda(A)_d)$
$i := 1$
repeat
    $J := \mathrm{sort}(\lambda, k)$
    $c_i := k^{-1} \sum_{j \in J} e_j$
    $s := \min_{j \in J} \lambda_j$
    $l := \max_{j \in [d] \setminus J} \lambda_j$
    $\beta_i := \min\left\{ sk, \sum_{j=1}^d \lambda_j - lk \right\}$
    $\lambda := \lambda - \beta_i c_i$
    $\Pi_i := U\mathrm{diag}(kc_i)U^\top$
    $i := i + 1$
until $\lambda = (0, \ldots, 0)$
For each $j \in [i-1]$ choose $A = \Pi_j$ with probability β_j

---

It has been proved by Warmuth and Kuzmin (2008) that the loop inside Algorithm 3 repeats at most d times, and decomposes W into a convex combination $\sum \beta_i \Pi_i$ of elements from $\mathcal{P}_k^d$.

Appendix E. Covariance Estimation with Missing Entries

Corollary 1 in Lounici (2014) provides bounds under the assumption that the instances are drawn from a sub-gaussian distribution (see assumption 1 in Lounici (2014)). This assumption is weaker than our boundedness assumption. Translating the results to our notation yields:

Lemma 13 *Given m partial observations, let $\hat{C}$ be the unbiased estimation constructed by POPCA. Let $\lambda > 0$ be a parameter.*

$$\tilde{C} := C(\tilde{\lambda}) = \operatorname*{argmin}_{S \in \mathbb{S}_+^d}\{\|S - \hat{C}\|_F^2 + \lambda\|S\|_1\}\,,$$

where $\mathbb{S}_+^d$ is the set of symmetric positive semi-definite $d \times d$ matrices. Then, for a suitable choice of λ, with probability at least $1 - 1/(2d)$, we have

$$\|\tilde{C} - C\|_F \le \sqrt{\inf_{S \in \mathbb{S}_+^d}\{\|S - C\|_F^2 + c_1\lambda_1^2(C) \cdot \frac{\operatorname{tr}(C)}{\lambda_1(C)} \cdot \frac{d^2}{r^2 m} \cdot rank(S) \cdot \log(2d)\}}\,,$$

where c_1 is a constant and $\lambda_1(C)$ is the leading eigenvalue of C.

Substituting $m = \lceil (d/r)^2 \cdot \frac{k}{\epsilon^2} \rceil$ from Lemma 2 into the above bound, we obtain that the RHS is at most

$$\sqrt{\inf_{S \in \mathbb{S}_+^d}\{\|S - C\|_F^2 + c_1\lambda_1^2(C) \cdot \frac{\operatorname{tr}(C)}{\lambda_1(C)} \cdot \frac{\epsilon^2}{k} \cdot rank(S) \cdot \log(2d))\}}\,.$$

This bound is always worse than our bound (i.e., larger than $\epsilon/\sqrt{k}$).

References

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. cambridge university press, 2009.

Raman Arora, Andrew Cotter, and Nathan Srebro. Stochastic optimization of pca with capped msg. *arXiv preprint arXiv:1307.1674*, 2013.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Gilles Blanchard, Olivier Bousquet, and Laurent Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2-3):259–294, 2007.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

Nicolo Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir. Efficient learning with partially observed attributes. *The Journal of Machine Learning Research*, 12:2857–2878, 2011.

Yudong Chen, Ali Jalali, Sujay Sanghavi, and Constantine Caramanis. Low-rank matrix recovery from errors and erasures. *Information Theory, IEEE Transactions on*, 59(7):4324–4337, 2013.

Yuejie Chi, Y.C. Eldar, and Robert Calderbank. Petrels: Parallel subspace estimation and tracking by recursive least squares from partial observations. *arXiv preprint arXiv:1207.6353*, 2013.

Qian Du and James E Fowler. Hyperspectral image compression using jpeg2000 and principal component analysis. *Geoscience and Remote Sensing Letters, IEEE*, 4(2):201–205, 2007.

Elad Hazan, Satyen Kale, and Shai Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. *arXiv preprint arXiv:1204.0136*, 2012.

Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

Karim Lounici. High-dimensional covariance matrix estimation with missing observations. *Bernoulli*, 20(3):1029–1058, 2014.

Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Streaming pca with many missing entries. *Preprint*, 2014.

Jiazhong Nie, Wojciech Kotłowski, and Manfred K Warmuth. Online pca with optimal regrets. In *Algorithmic Learning Theory*, pages 98–112. Springer, 2013.

Christos H Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168. ACM, 1998.

John Shawe-Taylor, Christopher KI Williams, Nello Cristianini, and Jaz Kandola. On the eigenspectrum of the gram matrix and the generalization error of kernel-pca. *Information Theory, IEEE Transactions on*, 51(7):2510–2522, 2005.

Koji Tsuda, Gunnar Rätsch, and Manfred K Warmuth. Matrix exponentiated gradient updates for on-line learning and bregman projection. In *Journal of Machine Learning Research*, pages 995–1018, 2005.

Xiaodong Wang and H Vincent Poor. Blind multiuser detection: A subspace approach. *Information Theory, IEEE Transactions on*, 44(2):677–690, 1998.

M.K. Warmuth and D. Kuzmin. Randomized online pca algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9:2287–2320, 2008.

Jian Yang, David Zhang, Alejandro F Frangi, and Jing-yu Yang. Two-dimensional pca: a new approach to appearance-based face representation and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(1):131–137, 2004.