Journal of Machine Learning Research 17 (2016) 1-49 Submitted 6/15; Revised 8/16; Published 9/16

Online PCA with Optimal Regret*

Jiazhong Nie NIEJIAZHONG@CSE.UCSC.EDU
Department of Computer Science, University of California, Santa Cruz

Wojciech Kotłowski WKOTLOWSKI@CS.PUT.POZNAN.PL
Institute of Computing Science, Poznań University of Technology, Poland

Manfred K. Warmuth MANFRED@CSE.UCSC.EDU
Department of Computer Science, University of California, Santa Cruz

Editor: Nathan Srebro

Abstract

We investigate the online version of Principle Component Analysis (PCA), where in each trial t the learning algorithm chooses a k-dimensional subspace, and upon receiving the next instance vector $\boldsymbol{x}_t$, suffers the "compression loss", which is the squared Euclidean distance between this instance and its projection into the chosen subspace. When viewed in the right parameterization, this compression loss is linear, i.e. it can be rewritten as $\mathrm{tr}(\boldsymbol{W}_t \boldsymbol{x}_t \boldsymbol{x}_t^\top)$, where $\boldsymbol{W}_t$ is the parameter of the algorithm and the outer product $\boldsymbol{x}_t \boldsymbol{x}_t^\top$ (with $\|\boldsymbol{x}_t\| \leq 1$) is the instance matrix. In this paper generalize PCA to arbitrary positive definite instance matrices $\boldsymbol{X}_t$ with the linear loss $\mathrm{tr}(\boldsymbol{W}_t \boldsymbol{X}_t)$.

We evaluate online algorithms in terms of their worst-case regret, which is a bound on the additional total loss of the online algorithm on all instances matrices over the compression loss of the best k-dimensional subspace (chosen in hindsight). We focus on two popular online algorithms for generalized PCA: the Gradient Descent (GD) and Matrix Exponentiated Gradient (MEG) algorithms. We show that if the regret is expressed as a function of the number of trials, then both algorithms are optimal to within a constant factor on worst-case sequences of positive definite instances matrices with trace norm at most one (which subsumes the original PCA problem with outer products). This is surprising because MEG is believed be suboptimal in this case. We also show that when considering regret bounds as a function of a loss budget, then MEG remains optimal and strictly outperforms GD when the instance matrices are trace norm bounded.

Next, we consider online PCA when the adversary is allowed to present the algorithm with positive semidefinite instance matrices whose largest eigenvalue is bounded (rather than their trace which is the sum of their eigenvalues). Again we can show that MEG is optimal and strictly better than GD in this setting.

Keywords: online learning, regret bounds, expert setting, k-sets, PCA, Gradient Descent, Matrix Exponentiated Gradient algorithm

1. Introduction

In Principal Component Analysis (PCA), the data points $\boldsymbol{x}_t \in \mathbb{R}^n$ are projected / compressed onto a k-dimensional subspace. Such a subspace can be represented by its projection matrix $\boldsymbol{P}$ which is a symmetric matrix in $\mathbb{R}^{n \times n}$ with k eigenvalues equal 1 and $n - k$

*. A preliminary version of this paper appeared in the 24th International Conference on Algorithmic Learning Theory (2013) (Nie et al., 2013).

eigenvalues equal 0. The goal of *uncentered PCA* is to find the rank k projection matrix that minimizes the total *compression loss* $\sum_t \|\boldsymbol{P}\boldsymbol{x}_t - \boldsymbol{x}_t\|^2$, i.e. the sum of the squared Euclidean distances between the original and the projected data points.In *centered PCA* the goal is to minimize $\sum_t \|\boldsymbol{P}(\boldsymbol{x}_t - \boldsymbol{\mu}) - (\boldsymbol{x}_t - \boldsymbol{\mu})\|^2$ where $\boldsymbol{P}$ is a projection matrix of rank k and $\boldsymbol{m} \in \mathbb{R}^n$ is a second mean parameter. For the sake of simplicity we focus on the optimal algorithms for uncentered PCA. However we believe that our results will essentially carry over to the centered case as was already partially done in Warmuth and Kuzmin (2008). Surprisingly, this loss can be written as a linear loss (Warmuth and Kuzmin, 2008):

$$\sum_t \|\boldsymbol{P}\boldsymbol{x}_t - \boldsymbol{x}_t\|^2 \;=\; \sum_t \|(\boldsymbol{P} - \boldsymbol{I})\boldsymbol{x}_t\|^2 \;=\; \sum_t \boldsymbol{x}_t^\top (\boldsymbol{I} - \boldsymbol{P})^{\cancel{2}} \boldsymbol{x}_t \;=\; \operatorname{tr}\big((\boldsymbol{I} - \boldsymbol{P}) \underbrace{\sum_t \boldsymbol{x}_t \boldsymbol{x}_t^\top}_{\boldsymbol{C}}\big),$$

where in the 3rd equality we used the fact that $\boldsymbol{I} - \boldsymbol{P}$ is a projection matrix and therefore $(\boldsymbol{I} - \boldsymbol{P})^2 = \boldsymbol{I} - \boldsymbol{P}$. The final expression of the compression loss is linear in the projection matrix $\boldsymbol{P} - \boldsymbol{I}$ as well as the covariance matrix $\boldsymbol{C} = \sum_t \boldsymbol{x}_t \boldsymbol{x}_t^\top$. The projection matrix $\boldsymbol{P} - \boldsymbol{I}$ is a sum of $n - k$ outer products: $\boldsymbol{P} - \boldsymbol{I} = \sum_{i=1}^{n-k} \boldsymbol{u}_i \boldsymbol{u}_i^\top$, where the $\boldsymbol{u}_i$ are unit length and orthogonal. The crucial point to note here is that the compression loss is *linear* in the projection matrix $\boldsymbol{P} - \boldsymbol{I}$ but not in the direction vectors $\boldsymbol{u}_i$.

The batch version of uncentered PCA is equivalent to finding the eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ belonging to the k largest eigenvalues of the covariance matrix $\boldsymbol{C}$: if $\boldsymbol{P} = \sum_{i=1}^{k} \boldsymbol{u}_i \boldsymbol{u}_i^\top$ is the k dimensional projection matrix formed from these k eigenvectors, then $\boldsymbol{I} - \boldsymbol{P}$ is the complimentary $n-k$ dimensional projection matrix minimizing the linear loss $\operatorname{tr}((\boldsymbol{I}-\boldsymbol{P})\boldsymbol{C})$.

In this paper we consider the online version of uncentered PCA (Warmuth and Kuzmin, 2008), where in each trial $t = 1, \ldots, T$, the algorithm chooses (based on the previously observed points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}$) a subspace of dimension k described by its projection matrix $\boldsymbol{P}_t$ of rank k. Then a next point $\boldsymbol{x}_t$ (or instance matrix $\boldsymbol{x}_t \boldsymbol{x}_t^\top$) is revealed and the algorithm suffers the *compression loss*:

$$\|\boldsymbol{x}_t - \boldsymbol{P}_t \boldsymbol{x}_t\|^2 \;=\; \operatorname{tr}\big((\boldsymbol{I} - \boldsymbol{P}_t)\, \boldsymbol{x}_t \boldsymbol{x}_t^\top\big). \tag{1.1}$$

The goal here is to obtain an online algorithm whose cumulative loss over trials $t = 1, \ldots, T$ is close to the cumulative loss of the best rank k projection matrix chosen in hindsight after seeing all T instances. The maximum difference between the cumulative loss of the algorithm and the best off-line comparator is called the (worst-case) *regret*. This regret naturally scales with the maximum square L_2-norm of the data points $\boldsymbol{x}_t$. For the sake of simplicity we assume that all points have L_2-norm bounded by one, i.e. $\|\boldsymbol{x}_t\| \leq 1$ for all t. In the paper we find the optimal algorithm for online PCA (and some generalizations), where optimal here means that the upper bounds we prove for the regret of the algorithm is at most a constant factor larger than the lower bound we can prove for the learning problem.

There are two main families of algorithms in online learning, which differ in how the parameter vector/matrix is updated: the Gradient Descent (GD) family (Cesa-Bianchi et al., 1996; Kivinen and Warmuth, 1997; Zinkevich, 2003) and the Exponentiated Gradient (EG) family (Kivinen and Warmuth, 1997). The updated parameters of both families of algorithms are solutions to certain minimization problems which trade off a divergence to

the last parameter against the loss on the current instance. The GD family uses the squared Euclidean distance divergence in the trade-off, whereas the Exponentiated Gradient (EG) family is motivated by the relative entropy divergence (Kivinen and Warmuth, 1997). The first family leads to *additive updates* of the parameter vector/matrix. When there are no constraints on the parameter space, then the parameter vector/matrix of the GD family is a linear combination of the instances. However when there are constraints, then after the update the parameter is projected onto the constraints (by a Bregman projection with respect to the squared Euclidean distance). The second family leads to *multiplicative update* algorithms. For that family, the components of the parameter are non-negative and if the parameter space consists of probability vectors, then the non-negativity is already enforced by the relative entropy divergence and less projections are needed.

What is the best parameter space for uncentered PCA? The compression loss (1.1) is linear in the projection matrix matrix $\boldsymbol{I} - \boldsymbol{P}_t$ which is of rank $n-k$. An online algorithm has uncertainty over the best projection matrix. Therefore the parameter matrix $\boldsymbol{W}_t$ of the algorithm is a mixture of such matrices (Warmuth and Kuzmin, 2008) which must be a positive semi-definite matrix of trace $n-k$ whose eigenvalues are capped at 1. The algorithm chooses its projection matrix $\boldsymbol{I}-\boldsymbol{P}_t$ by sampling from this mixture $\boldsymbol{W}_t$, i.e. $\mathbb{E}[\boldsymbol{I}-\boldsymbol{P}_t] = \boldsymbol{W}_t$. The loss of the algorithm is $\operatorname{tr}((\boldsymbol{I} - \boldsymbol{P}_t)\, \boldsymbol{x}_t \boldsymbol{x}_t^\top)$ and its expected loss $\operatorname{tr}(\boldsymbol{W}_t\, \boldsymbol{x}_t \boldsymbol{x}_t^\top)$.

In Warmuth and Kuzmin (2008), a matrix version of the multiplicative update was applied to PCA, whose regret bound is logarithmic in the dimension n. This algorithm uses the quantum relative entropy in its motivation and is called the *Matrix Exponentiated Gradient* (MEG) algorithm (Tsuda et al., 2005). It does a matrix version of a multiplicative update and then projects onto the "trace equal $n-k$" and the "capping" constraints (Here the projections are with respect to the quantum relative entropy).

For the PCA problem, the (expected) loss of the algorithm at trial t is $\operatorname{tr}(\boldsymbol{W}_t \boldsymbol{x}_t \boldsymbol{x}_t^\top)$. Consider the generalization to the loss $\operatorname{tr}(\boldsymbol{W}_t \boldsymbol{X}_t)$ where now $\boldsymbol{X}_t$ is any positive semi-definite symmetric instance matrix and the parameter $\boldsymbol{W}_t$ is still a convex combination of rank $n-k$ dimensional projection matrices, i.e. $\boldsymbol{W}_t = \boldsymbol{E}[\boldsymbol{I} - \boldsymbol{P}_t]$ where $\boldsymbol{P}_t$ is the rank k projection matrix chosen by the algorithm at trial t. The linear loss $\operatorname{tr}(\boldsymbol{E}[\boldsymbol{I}-\boldsymbol{P}_t]\boldsymbol{X}_t)$ still has a meaning in terms of a compression loss: For any decomposition of $\boldsymbol{X}_t$ into a linear combination of outer products, i.e. $\boldsymbol{X}_t = \sum_q \lambda_q \boldsymbol{z}_q \boldsymbol{z}_q^\top$ (where the λ_i may be positive or negative and the $\boldsymbol{z}_q \in \boldsymbol{R}^n$ don't have to be orthogonal) we have

$$\operatorname{tr}((\boldsymbol{I} - \boldsymbol{P}_t)\boldsymbol{X}_t) = \sum_q \lambda_q \operatorname{tr}((\boldsymbol{I} - \boldsymbol{P}_t)\boldsymbol{z}_q \boldsymbol{z}_q^\top) = \sum_q \lambda_q \, \|\boldsymbol{z}_q - \boldsymbol{P}_t \boldsymbol{z}_q\|^2. \tag{1.2}$$

In this paper we analyze our algorithm for two classes of positive definite instance matrices. Recall that in the vanilla PCA problem the instance matrices are the outer products, i.e. $\boldsymbol{X}_t = \boldsymbol{x}_t \boldsymbol{x}_t^\top$, where $\|\boldsymbol{x}_t\| \leq 1$. Such instance matrices have a "sparse spectrum" in the sense that they have at most one non-zero eigenvalue. Our first class consists of the convex hull of outer products of length at most one or equivalently all positive semidefinite matrices of trace norm at most one. We call this class *L_1-bounded* instance matrices. The most important fact to remember is that the case of L_1-bounded instances contains vanilla PCA with outer product instances as a special case.

Beginning with some of the early work on linear regression (Kivinen and Warmuth, 1997), it is known that multiplicative updates are especially useful when the non-negative

instance vectors are allowed to be "dense", i.e. their maximum component is bounded by say one but it could contain many components of size up to one. In the matrix context this means that the symmetric positive semi-definite instance matrices $\boldsymbol{X}_t$ have maximum eigenvalue (or spectral norm) at most one and are thus "spectrally dense". We call this second class L_∞-*bounded* instance matrices.

We will show that MEG is optimal for L_∞-bounded instance matrices and GD is sub-optimal in this case. However for L_1-bounded instances one might suspect that MEG is not able to fully exploit the spectral sparsity. For example, in the case of linear regression GD is known to have the advantage when the instance vectors[1] have bounded L_2 norm (Kivinen and Warmuth, 1997) and consistently with that, when GD is used for PCA with L_1-bounded instance matrices, then its regret is bounded by a term that is *independent* of the dimension of the instances. The advantage of GD in the spectrally sparse case is also supported by a general survey of Mirror Descent algorithms (to which GD and MEG belong) for the case when the gradient vectors of the convex loss functions (which may have negative components) lie in certain symmetric norm balls (Srebro et al., 2011). Again when the gradient vectors of the losses are sparse, then GD has the advantage.

Surprisingly, the situation is quite different for PCA: We show that MEG achieves the same regret bound as GD for online PCA with L_1-bounded instances matrices (despite the spectral sparseness) and the regret bounds for both algorithms are within a constant factor of a new lower bound proved in this paper that holds for any algorithm for PCA with L_1-bounded instance matrices. This surprising performance of MEG seems to come from the fact that gradients $\boldsymbol{X}_t$ of the linear loss $\mathrm{tr}(\boldsymbol{W}_t\boldsymbol{X}_t)$ of our generalized online PCA problem are restricted to be non-negative. Therefore our results are qualitatively different from the cases studied in Srebro et al. (2011) where the gradients of the loss functions are within a $p-$norm ball, i.e. symmetric around zero.

Actually, there are two kinds of regret bounds in the literature: bounds expressed as a function of the time horizon T and bounds that depend on an *upper bound* on the loss of the best comparator (which we call a *loss budget* following Abernethy et al. (2008)). In typical applications for PCA, there exists a low dimensional subspace which captures most of the variance in the data and the compression loss is small. Therefore, guarding against the worst-case loss that grows with the number of trials T is overly pessimistic. We can show that when considering regret bounds as a function of a loss budget, MEG is optimal and strictly better than GD by a factor of $\sqrt{k}$. This suggests that the multiplicative updates algorithm is the best choice for prediction problems in which the parameters are mixtures of projection matrices and the gradients of the losses are non-negative. Note that in this paper we call an algorithm *optimal* for a particular problem if we can prove an upper bound on its worst-case regret that is within a constant factor of the lower bound for the problem (which must holds for any algorithm).

1.1 Related Work and Our Contribution:

The comparison of the GD and MEG algorithms has an extensive history (see, e.g. Kivinen and Warmuth (1997); Warmuth and Vishwanathan (2005); Sridharan and Tewari (2010); Srebro et al. (2011)). It is simplest to compare algorithms in the case when the loss is

1. Note that for $\boldsymbol{x} \in \boldsymbol{R}^n$, $\|x\|_\infty \leq \|x\|_2 \leq \|x\|_1$.

linear. Linear losses are the least convex losses and in the regret bounds, convex losses are often approximated by first-order Taylor approximations which are linear, and the gradient of the loss function serves as the linear "loss/gain vector" (Kivinen and Warmuth, 1997; Zinkevich, 2003). In this case it is often assumed that the gradient of the loss lies in an L_p ball (which is a symmetric constraint) and the results are as expected: EG is optimal when the parameter space is L_1-bounded and the gradient vectors are L_∞-bounded, and GD is optimal when the both spaces are L_2-bounded (Sridharan and Tewari, 2010; Srebro et al., 2011).

In contrast for PCA, the gradient of the loss $\mathrm{tr}(\boldsymbol{W}_t\boldsymbol{X}_t)$ is the instance matrix $\boldsymbol{X}_t$ which is assumed to be positive semi-definite. None of the previous work exploits this special property of the PCA setup, where the gradient of the loss satisfies some non-negativity property. In this paper we carefully study this case and show that MEG is optimal.

We also made significant technical progress on the lower bounds for online PCA. The previous lower bounds (Warmuth and Kuzmin (2008) and Koolen et al. (2010)) were incomplete in the following three ways: First, the lower bounds only apply to the case of L_∞-bounded instances and not to the more restricted case of L_1-bounded instances. Second, the previous lower bounds assume that the dimension k of target subspace is at least $\frac{n}{2}$ and in common PCA problems, k is much smaller than $\frac{n}{2}$. Third, the proofs rely on the Central Limit Theorem and therefore the resulting lower bounds only hold in the limit as T and n go to infinity (See Cesa-Bianchi et al. (1997); Cesa-Bianchi and Lugosi (2006); Abernethy et al. (2009) for details). In this paper, we circumvent all three weak aspects of the previous proofs: We give lower bounds for all four combinations of L_1 or L_∞-bounded instance matrices versus $k \leq \frac{n}{2}$ or $k \geq \frac{n}{2}$, respectively. All our lower bounds are non-asymptotic, i.e. they hold for all values of the variables T and n. The new lower bounds use a novel probabilistic bounding argument for the minimum of n random variables. Alternate methods for obtaining non-asymptotic lower bound for label efficient learning problems in the expert setting were given in (Audibert and Bubeck, 2010). However those techniques are more complicated and it is not clear how to adapt them to the online PCA problem.

In summary, our contribution consists of proving tight upper bounds on the regret of the two main online PCA algorithms, as well as proving lower bounds on the regret of any algorithm for solving online PCA. For the case when the regret is expressed as a function of the number of trials T, we show that MEG's and GD's regret bounds are independent of the dimension n of the problem and are within a constant factor of the lower bound on the regret of any online PCA algorithm. This means the both algorithms are optimal in this case. For the case when the regret is a function of the loss budget, we prove that MEG remains optimal, while we show that the regret of GD is suboptimal by a $\sqrt{k}$ factor.

Furthermore, for the generalization of the PCA with L_∞-bounded instance matrices, we improve the known regret bound significantly by switching from a loss version to a gain version of MEG depending on the dimension k of the subspace. If $k \geq \frac{n}{2}$, then the gain version of MEG is optimal for L_∞-bounded instances, and when $k \leq \frac{n}{2}$, then the loss version is optimal. On the other hand, GD is non-optimal for both ranges of k.

A much shorter preliminary version of this manuscript appeared in the 24th International Conference on Algorithmic Learning Theory (2013) (Nie et al., 2013). In this more detailed journal version we give more background and complete proofs of all of our results (mostly omitted or only sketched in the conference version). This paper also has the following

additional material: A proof of the budget bound (3.5) for the gain version of MEG; an extension of the lower bound on the regret of GD (Theorem 4.1) to the case of small budgets; the analysis of the Follow the Regularized Leader variant of GD (Section 4.2) and a discussion of its final parameter matrix (Appendix E); lower bounds on the regret when the number of trials is small (Appendix G).

1.2 Outline of the Paper:

In Section 2, we start with describing the MEG and GD algorithms for online PCA. In particular, we present two versions of the MEG algorithm: the Loss MEG algorithm introduced in (Warmuth and Kuzmin, 2008), and the Gain MEG algorithm, which is the same as Loss MEG except for a sign change in the exponential. Following the description of each algorithm, we then derive in Section 3 their regret bounds expressed as functions of the number of trials T. These bounds are compared in Section 3.2 for all four combinations of L_1 or L_∞-bounded instance matrices versus $k \leq \frac{n}{2}$ or $k \geq \frac{n}{2}$, respectively (see Table 3.2). Next we consider regret bounds expressed as functions of the loss budget. In Section 4, we prove a lower bound on GD's regret which shows that the regret of GD is at least $\sqrt{k}$ times larger than the regret of Loss EG. A similar lower bound is proved for the Follow the Regularized Leader variant of GD in Section 4.2. In Section 5 we prove lower bounds for online PCA with L_1 and L_∞-bounded instances that hold for any online algorithm, and in Section 6 we conclude with a summary of which algorithms are optimal.

2. The Online Algorithms

Online uncentered PCA uses the following protocol in each trial $t = 1, \ldots, T$: the algorithm probabilistically chooses a projection matrix $\boldsymbol{P}_t \in \mathbb{R}^{n \times n}$ of rank k. Then a point $\boldsymbol{x}_t \in \mathbb{R}^n$ is received and the algorithm suffers the *loss* $\mathrm{tr}((\boldsymbol{I} - \boldsymbol{P}_t)\boldsymbol{x}_t\boldsymbol{x}_t^\top)$.

We also consider the generalization where the instance matrix is any positive definite matrix $\boldsymbol{X}_t$ instead of an outer product $\boldsymbol{x}_t\boldsymbol{x}_t^\top$. In that case the loss of the algorithm is $\mathrm{tr}((\boldsymbol{I} - \boldsymbol{P}_t)\boldsymbol{X}_t)$. As discussed in the introduction (e.g. Equation (1.2)), this linear loss has a compression loss interpretation. It is "complementary" to the *gain* $\mathrm{tr}(\boldsymbol{P}_t\boldsymbol{X}_t)$, i.e.

$$\underbrace{\mathrm{tr}((\boldsymbol{I} - \boldsymbol{P}_t)\boldsymbol{X}_t)}_{\text{loss}} \;=\; \underbrace{\mathrm{tr}(\boldsymbol{X}_t)}_{\text{constant}} \;-\; \underbrace{\mathrm{tr}(\boldsymbol{P}_t\boldsymbol{X}_t)}_{\text{gain}},$$

and the $n-k$ dimensional projection matrix $\boldsymbol{I}-\boldsymbol{P}_t$ is "complementary" to the k dimensional projection matrix $\boldsymbol{P}_t$. These two complementations are inherent to our problem and will be present throughout the paper.

In the above protocol, the algorithm is allowed to choose its k dimensional subspace $\boldsymbol{P}_t$ probabilistically. Therefore we use the expected compression loss $\mathbb{E}[\mathrm{tr}((\boldsymbol{I} - \boldsymbol{P}_t)\boldsymbol{X}_t)]$ as the loss of the algorithm. The regret of the algorithm is then the difference between its cumulative loss and the loss of the best k subspace:

$$\mathcal{R} = \sum_{t=1}^{T} \mathbb{E}[\mathrm{tr}((\boldsymbol{I} - \boldsymbol{P}_t)\boldsymbol{X}_t)] \;-\; \min_{\substack{\boldsymbol{P} \text{ projection} \\ \text{matrix of rank } k}} \sum_{t=1}^{T} \mathrm{tr}((\boldsymbol{I} - \boldsymbol{P})\boldsymbol{X}_t).$$

The regret can also be rewritten in terms of gain, but this gives the same value of the regret. Therefore, throughout the paper we use (expected) losses and "loss" regrets (as defined above) to evaluate the algorithms.

Now we rewrite the loss of the algorithm as $\mathrm{tr}(\mathbb{E}[\boldsymbol{I} - \boldsymbol{P}_t]\boldsymbol{X}_t)$ which shows that for any random prediction $\boldsymbol{P}_t$ of rank k, this loss is fully determined by $\mathbb{E}[\boldsymbol{I} - \boldsymbol{P}_t]$, a convex combination of rank $m = n - k$ projection matrices. Hence it is natural to choose the set $\boldsymbol{\mathcal{W}}_m$ of convex combinations of rank m projection matrices as the parameter set of the algorithm. By the definition of projection matrices, $\boldsymbol{\mathcal{W}}_m$ is the set of positive semi-definite matrices of trace m and eigenvalues not larger than 1. The current parameter $\boldsymbol{W}_t \in \boldsymbol{\mathcal{W}}_m$ of the online algorithm expresses its "uncertainty" about which subspace of rank m is best for the online data stream seen so far and the (expected) loss in trial t becomes $\mathrm{tr}(\boldsymbol{W}_t\boldsymbol{X}_t)$. Alternatively, the complementary set $\boldsymbol{\mathcal{W}}_k$ of rank k projection matrices can be used as the parameter set (In that case the loss is $\mathrm{tr}((\boldsymbol{I} - \boldsymbol{W}_t)\boldsymbol{X}_t)$). As discussed, there is a one-to-one correspondence between the two parameter sets: Given $\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_k$, then $\boldsymbol{I} - \boldsymbol{W}$ is the corresponding convex combination in $\boldsymbol{\mathcal{W}}_m$.

The second reason why convex combinations are natural parameter spaces is that since the loss is linear, the convex combination with the minimum loss occurs at a "pure" projection matrix, i.e.

$$\begin{aligned}
\min_{\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_m} \sum_{t=1}^{T} \mathrm{tr}(\boldsymbol{W}\boldsymbol{X}_t) &= \min_{\substack{\boldsymbol{P} \text{ projection} \\ \text{matrix of rank } k}} \sum_{t=1}^{T} \mathrm{tr}((\boldsymbol{I} - \boldsymbol{P})\boldsymbol{X}_t) \quad \text{and} \\
\min_{\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_k} \sum_{t=1}^{T} \mathrm{tr}((\boldsymbol{I} - \boldsymbol{W})\boldsymbol{X}_t) &= \min_{\substack{\boldsymbol{P} \text{ projection} \\ \text{matrix of rank } k}} \sum_{t=1}^{T} \mathrm{tr}((\boldsymbol{I} - \boldsymbol{P})\boldsymbol{X}_t). \qquad (2.1)
\end{aligned}$$

Our protocol requires the algorithm to predict with a rank k projection matrix. Therefore, given a parameter matrix $\boldsymbol{W}_t$ in say $\boldsymbol{\mathcal{W}}_m$, the online algorithm still needs to produce a random projection matrix $\boldsymbol{P}_t$ of rank k at the beginning of trial t such that $\mathbb{E}[\boldsymbol{I} - \boldsymbol{P}_t] = \boldsymbol{W}_t$. A simple greedy algorithm for achieving this is given in (Warmuth and Kuzmin, 2008) (Algorithm 2) which efficiently decomposes $\boldsymbol{W}_t$ into a convex combination of up to n projection matrices of rank m (the algorithm requires the eigenvalue decomposition of $\boldsymbol{W}_t$, which has $O(n^3)$ time complexity in general, followed by a *mixture decomposition* of the eigenvalues which runs in $O(n^2)$ time). Using the mixture coefficients it is now easy to sample a projection matrix $\boldsymbol{I} - \boldsymbol{P}_t$ from parameter matrix $\boldsymbol{W}_t$.

We now motivate the two main online algorithms used in this paper: the GD and MEG algorithms. The GD algorithm is straightforward and the MEG algorithm was introduced in Tsuda et al. (2005). Both are examples of the *Mirror Descent* family of algorithms developed much earlier in the area of convex optimization (Nemirovski and Yudin, 1978). The Mirror Descent algorithms update their parameter by minimizing a trade-off function of a divergence between the new and old parameter and the loss of the new parameter on the current instance, while constraining the new parameter to lie in the parameter set.

For the problem of online PCA, the update specializes into the following two versions depending on the choice of the parameter set:

Loss update on parameter set $\boldsymbol{\mathcal{W}}_m$ (i.e., $\boldsymbol{W}_{t+1}, \boldsymbol{W}, \boldsymbol{W}_t \in \boldsymbol{\mathcal{W}}_m$):

$$\boldsymbol{W}_{t+1} = \operatorname*{argmin}_{\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_m} \left(\Delta(\boldsymbol{W}, \boldsymbol{W}_t) + \eta \operatorname{tr}(\boldsymbol{W}\boldsymbol{X}_t)\right). \tag{2.2}$$

Gain update on parameter set $\boldsymbol{\mathcal{W}}_k$ (i.e., $\boldsymbol{W}_{t+1}, \boldsymbol{W}, \boldsymbol{W}_t \in \boldsymbol{\mathcal{W}}_k$):

$$\begin{aligned}\boldsymbol{W}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_k} \left(\Delta(\boldsymbol{W}, \boldsymbol{W}_t) + \eta \operatorname{tr}((\boldsymbol{I} - \boldsymbol{W})\boldsymbol{X}_t)\right) \\ &= \operatorname*{argmin}_{\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_k} \left(\Delta(\boldsymbol{W}, \boldsymbol{W}_t) - \eta \operatorname{tr}(\boldsymbol{W}\boldsymbol{X}_t)\right). \end{aligned} \tag{2.3}$$

Here $\Delta(\boldsymbol{W}, \boldsymbol{W}_t)$ is the motivating Bregman divergence that will be different for the MEG and GD algorithms. The *Loss update* minimizes a trade-off with the expected loss $\operatorname{tr}(\boldsymbol{W}\boldsymbol{X}_t)$ which is a matrix version of the dot loss used for motivating the Hedge algorithm (Freund and Schapire, 1995). Note that in the *gain* version, minimizing the loss $-\operatorname{tr}(\boldsymbol{W}\boldsymbol{X}_t)$ is the same as maximizing the gain $\operatorname{tr}(\boldsymbol{W}\boldsymbol{X}_t)$. Recall that there is a one-to-one correspondence between $\boldsymbol{\mathcal{W}}_m$ and $\boldsymbol{\mathcal{W}}_k$, i.e. $\boldsymbol{I}$ minus a parameter in $\boldsymbol{\mathcal{W}}_m$ gives the corresponding parameter in $\boldsymbol{\mathcal{W}}_k$ and vice versa. Therefore, one can for example rewrite the Gain update (2.3) with the parameter set $\boldsymbol{\mathcal{W}}_m$ as well:

$$\widetilde{\boldsymbol{W}}_{t+1} = \operatorname*{argmin}_{\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_m} \left(\Delta(\boldsymbol{I} - \boldsymbol{W}, \boldsymbol{I} - \widetilde{\boldsymbol{W}}_t) + \eta \operatorname{tr}(\boldsymbol{W}\boldsymbol{X}_t)\right), \tag{2.4}$$

where the above solution $\widetilde{\boldsymbol{W}}_{t+1} \in \boldsymbol{\mathcal{W}}_m$ of the Gain update is related to the solution $\boldsymbol{W}_{t+1} \in \boldsymbol{\mathcal{W}}_k$ of (2.3) by the same complimentary relationship, i.e. $\widetilde{\boldsymbol{W}}_{t+1} = \boldsymbol{I} - \boldsymbol{W}_{t+1}$, for $t = 1, \ldots, T$. Notice that the Loss update is motivated by the divergence $\Delta(\boldsymbol{W}, \boldsymbol{W}_t)$ on parameter space $\boldsymbol{\mathcal{W}}_m$ (2.2). On the other hand, when the Gain update is formulated with parameter $\boldsymbol{\mathcal{W}}_m$, then it is motivated by the divergence $\Delta(\boldsymbol{I} - \boldsymbol{W}, \boldsymbol{I} - \widetilde{\boldsymbol{W}}_t)$ (2.4).

Now we define the GD and MEG algorithms for online PCA. For the GD algorithm, the motivating Bregman divergence is the squared Frobenius norm between the old and new parameters: $\Delta(\boldsymbol{W}, \boldsymbol{W}_t) = \frac{1}{2}\|\boldsymbol{W} - \boldsymbol{W}_t\|_F^2$ (Kivinen and Warmuth, 1997; Zinkevich, 2003). With this divergence, the Loss update is solved in the following two steps:

$$\text{GD update:} \qquad \begin{aligned} &\text{Descent step:} && \widehat{\boldsymbol{W}}_{t+1} = \boldsymbol{W}_t - \eta \boldsymbol{X}_t, \\ &\text{Projection step:} && \boldsymbol{W}_{t+1} = \operatorname*{argmin}_{\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_m} \|\boldsymbol{W} - \widehat{\boldsymbol{W}}_{t+1}\|_F^2. \end{aligned} \tag{2.5}$$

Note, that the split into two steps happens whenever a Bregman divergence is traded off with a linear loss and domain is convex (See Helmbold and Warmuth (2009), Section 5.2, for a discussion). For the squared Frobenius norm, the Gain update is equivalent to the Loss update, since when formulating both updates on parameter set $\boldsymbol{\mathcal{W}}_m$, then the divergence $\|\boldsymbol{W} - \boldsymbol{W}_t\|_F^2$ of the Loss update (2.2) and the divergence $\|(\boldsymbol{I} - \boldsymbol{W}) - (\boldsymbol{I} - \boldsymbol{W}_t)\|_F^2$ of the Gain update (2.4) are the same. A procedure for projecting $\widehat{\boldsymbol{W}}_{t+1}$ into $\boldsymbol{\mathcal{W}}_m$ with respect to the squared Frobenius norm is given in Algorithm 2 of Arora et al. (2013). The expensive part of this procedure is obtaining the eigendecomposition of $\widehat{\boldsymbol{W}}_{t+1}$.

The MEG algorithm uses the (un-normalized) quantum relative entropy $\Delta(\boldsymbol{W}, \boldsymbol{W}_t) = \operatorname{tr}(\boldsymbol{W}(\log \boldsymbol{W} - \log \boldsymbol{W}_t) + \boldsymbol{W}_t - \boldsymbol{W})$ as its motivating Bregman divergence (Tsuda et al.,

T	Number of trials
n	Dimension of data points $\boldsymbol{x}_t \in \mathbb{R}^n$ and instance matrices $\boldsymbol{X}_t \in \mathbb{R}^{n \times n}$
k	Rank of the subspace of PCA into which the data is projected
m	Complement of k, $m = n - k$ (used for the rank of subspace of Loss MEG)
L_1-bounded instances	positive semi-definite matrices $\boldsymbol{X}_t$ s.t. $\mathrm{tr}(\boldsymbol{X}_t) \leq 1$ (subsumes the special case when the $\boldsymbol{X}_t$ are of the form $\boldsymbol{x}_t \boldsymbol{x}_t^\top$, w. $\|\boldsymbol{x}_t\| \leq 1$)
L_∞-bounded instances	positive semi-definite matrices $\boldsymbol{X}_t$ with spectral norm at most one, that is $\lambda_{\max}(\boldsymbol{X}_t) \leq 1$
B_L	Upper bound on loss of best subspace of rank $n - k$, c.f. (3.1)
B_G	Upper bound on gain of best subspace of rank k, c.f. (3.2).

Table 3.1: Summary of various symbols and terms used in Section 3.

2005) which is based on the matrix logarithm **log**. With this divergence the solutions to the Loss update (2.2) and Gain update (2.3) are the following expressions which make use of the matrix exponential **exp** (the inverse of **log**):

$$\text{Loss MEG update:} \quad \begin{aligned} &\text{Descent step:} && \widehat{\boldsymbol{W}}_{t+1} = \mathbf{exp}(\mathbf{log}\, \boldsymbol{W}_t - \eta \boldsymbol{X}_t), \\ &\text{Projection step:} && \boldsymbol{W}_{t+1} = \operatorname*{argmin}_{\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_m} \Delta(\boldsymbol{W}, \widehat{\boldsymbol{W}}_{t+1}). \end{aligned} \tag{2.6}$$

$$\text{Gain MEG update:} \quad \begin{aligned} &\text{Descent step:} && \widehat{\boldsymbol{W}}_{t+1} = \mathbf{exp}(\mathbf{log}\, \boldsymbol{W}_t + \eta \boldsymbol{X}_t), \\ &\text{Projection step:} && \boldsymbol{W}_{t+1} = \operatorname*{argmin}_{\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_k} \Delta(\boldsymbol{W}, \widehat{\boldsymbol{W}}_{t+1}). \end{aligned} \tag{2.7}$$

Note that the only difference between the gain and loss versions of MEG is a sign flip in the exponential. The projection steps in the algorithms are with respect to the quantum relative entropy. An efficient procedure for solving such projections is given in Algorithm 4 of Warmuth and Kuzmin (2008): it does a projection with respect to the standard relative entropy on the vector of eigenvalues of the parameter matrix. Finally note that the computational complexity of all described updates (GD, Loss MEG, Gain MEG) is dominated by the time required for obtaining the eigendecomposition of the parameter matrix $\boldsymbol{W}_{t+1}$ (or $\widehat{\boldsymbol{W}}_{t+1}$), which is $O(n^3)$ in general.

3. Upper Bounds on the Regret

Recall that the instance matrices $\boldsymbol{X}_t$ are always assumed to be positive semi-definite matrices. We call such instance matrices L_1-*bounded*, if the trace norm of the instance matrices is at most one, i.e. $\mathrm{tr}(\boldsymbol{X}_t) \leq 1$ always holds. In particular, this happens for the vanilla PCA setting where the data received at trial is a point $\boldsymbol{x}_t \in \boldsymbol{R}^n$ s.t. $\|\boldsymbol{x}_t\|^2 \leq 1$. In this case the instance matrices have the form $\boldsymbol{X}_t = \boldsymbol{x}_t \boldsymbol{x}_t^\top$ and $\mathrm{tr}(\boldsymbol{x}_t \boldsymbol{x}_t^\top) = \boldsymbol{x}_t^\top \boldsymbol{x}_t = \|\boldsymbol{x}_t\|^2 \leq 1$. Note that in the L_1-bounded case, the sums of the eigenvalues of the $\boldsymbol{X}_t$ are at most one. We also study the case when the maximum eigenvalue of the instance matrices $\boldsymbol{X}_t$ is at most one and call the latter the L_∞-*bounded* case.

In this section, we present regret upper bounds for the three online algorithms introduced in the previous section, which are Loss MEG, Gain MEG and GD. All three algorithms are examples from the Mirror Descent family of algorithms. Our proof techniques require us to use different restrictions on the worst-case sequences that the adversary can produce. For the Loss MEG algorithm, we give the adversary a *loss budget*, i.e. the adversary must produce a sequence of instances $\boldsymbol{X}_1 \dots \boldsymbol{X}_T$ for which the loss of the best subspace is upper bounded by the loss budget B_L:

$$\min_{\substack{\boldsymbol{P} \text{ projection} \\ \text{matrix of rank } k}} \sum_{t=1}^{T} \operatorname{tr}((\boldsymbol{I} - \boldsymbol{P})\boldsymbol{X}_t) \leq B_L. \tag{3.1}$$

We call a regret bound that depends on this parameter a *loss budget dependent* bound. A bound of this type was first proved for Loss MEG in Warmuth and Kuzmin (2008). The latter paper is the precursor of this paper in which the analysis of online algorithms for PCA was started.

For the algorithm of Gain MEG, we give the adversary a *gain budget* B_G, i.e. an upper bound on the gain of the best subspace:

$$\max_{\substack{\boldsymbol{P} \text{ projection} \\ \text{matrix of rank } k}} \sum_{t=1}^{T} \operatorname{tr}(\boldsymbol{P}\boldsymbol{X}_t) \leq B_G. \tag{3.2}$$

Now the adversary can only produce sequences for which all subspaces have gain at most B_G. We call this type of bound a *gain budget dependent* bound.

Finally we prove regret bounds of a third type for the GD algorithm. For this type the regret is a function of the number of trials T, and we call such a regret bound a *time dependent* regret bound.

We present the three regret bounds in the next subsection and compare them in the following subsection. As we shall see, upper bounds of the regret in terms of a budget imply time dependent bounds, and for lower bounds the implication is reversed. The main symbols and terms used throughout this section are summarized in Table 3.1.

3.1 Upper Bounds on the Regret of Loss MEG, Gain MEG, and GD

The Loss MEG algorithm (2.6) is the original MEG algorithm developed in the precursor paper of Warmuth and Kuzmin (2008) for online PCA. This paper proves a loss budget dependent upper bound on the regret of Loss MEG. The is done by exploiting the fact that PCA learning has the so called expert setting as a special case (See extensive discussion at the beginning of Section 4). More precisely the following bound is proven by lifting a regret bound developed for learning well compared to the best subset of $m = n - k$ experts to the matrix case, where subsets of size m generalize to projection matrices of rank m.

Loss budget dependent bound of Loss MEG:

$$\mathcal{R}_{\text{Loss MEG}} \;\leq\; \sqrt{2B_L\, m \log \frac{n}{m}} + m \log \frac{n}{m}. \tag{3.3}$$

This bound follows from Theorem 6 of Warmuth and Kuzmin (2008), and holds for any sequence of instance matrices (L_∞ as well as L_1-bounded) for which the total compression loss of the best rank m subspace does not exceed the loss budget B_L (Condition (3.1)).

We begin by showing that the right-hand side of (3.3) is bounded above by an expression that does not depend on the dimension n of the data points:

$$\mathcal{R}_{\text{Loss MEG}} \;\leq\; \sqrt{2B_L\, k} + k. \tag{3.4}$$

This follows immediately from the following inequality and the relationship $m = n - k$ ($n = m + k$):

$$m \log \frac{n}{m} = m \log \left(\frac{k+m}{m} \right) = m \log \left(1 + \frac{k}{m} \right) \leq m \frac{k}{m} = k.$$

As mentioned at the beginning of this subsection (and discussed in more detail later in Section 4), online PCA specializes to the problem of learning well compared to the best set of $m = n-k$ experts. Regret bounds for the expert setting typically depend logarithmically on the number of experts n. Therefore the above dimension free regret bound might seem puzzling at first. However there is no contradiction. In the current setup we have $m = 1$ and $k = n - m = n - 1$ for the vanilla single expert case, and the above dimension free bound (3.4) becomes $\sqrt{2B_L(n-1)}$. This bound is not close to the optimum loss budget dependent regret bound for the single expert case which is $O(\sqrt{B_L \log n} + \log n)$. This latter bound is obtained by plugging $m = 1$ into *the original* regret bound (3.3). Thus for $m = 1$, the above dimension free approximation (3.4) of the original bound is loose. However, when $k \leq \frac{n}{2}$, then as we shall see in Section 5, the dimension free approximation actually is tight. In the precursor paper (Warmuth and Kuzmin, 2008), a different but weaker approximation of the original bound was proved that still has an additional logarithmic dependence when $k \leq \frac{n}{2}$: $O(\sqrt{B_L k \log \frac{n}{k}} + k \log \frac{n}{k})$.

We next develop a regret bound for Gain MEG (2.7). The proof technique is a variation of the original regret bound for Loss MEG (and is given for the sake of completeness in Appendix A).

Gain budget dependent bound of Gain MEG:

$$\mathcal{R}_{\text{Gain MEG}} \;\leq\; \sqrt{2B_G\, k \log \frac{n}{k}}. \tag{3.5}$$

This bound holds for any sequence of instance matrices (L_1 as well as L_∞-bounded) for which the total gain of the best rank k subspace does not exceed the gain budget B_G (Condition (3.2)).

Finally, we give a simple regret bound for the GD algorithm. This bound (also observed in Arora et al. (2013) and proved for the sake of completeness in Appendix B) is based on two standard techniques: the use of the squared Frobenius norm (Kivinen and Warmuth, 1997) as a measure of progress and the use of the Pythagorean Theorem for handling the projection step (Herbster and Warmuth, 2001).

Time dependent regret bound of GD:

$$\mathcal{R}_{\text{GD}} \;\leq\; \begin{cases} \sqrt{T\,\frac{km}{n}} & \text{for } L_1\text{-bounded instances} \\ \sqrt{T\,km} & \text{for } L_\infty\text{-bounded instances} \end{cases} . \tag{3.6}$$

Note that each regret bound is expressed as a function of a loss budget, a gain budget or a time bound. They are obtained by setting the fixed learning rate of the algorithm as a function of one of these three parameters. The resulting basic algorithms can be used as sub-modules: For example the algorithm can be stopped as soon as the loss budget is reached and restarted with twice the budget and the corresponding re-tuned learning rate. This heuristic is known as the "doubling trick" (Cesa-Bianchi et al., 1997). Much fancier tuning schemes are explored in (van Erven et al., 2011; de Rooij et al., 2014) and are not the focus of this paper.

3.2 Comparison of the Regret Upper Bounds

Our goal is to find algorithms that achieve the optimal loss budget dependent and time dependent regret bounds where optimal means that the bound is within a constant factor of optimum. We are not interested in *gain dependent* regret bounds per se, i.e. bounds in terms of a gain budget B_G, because the maximal gain is typically much larger than the minimal loss. However when the gain budget restricted regret bounds are converted to time bounds, then for some setting (discussed below) the resulting algorithm becomes the only optimal algorithm we are aware of.

The only known *loss budget dependent* regret bound is bound (3.3) for Loss MEG obtained in the original paper for online learning of PCA (Warmuth and Kuzmin, 2008). We will show later in Section 5 that this upper bound on the regret is optimal. There are no known loss budget dependent upper bounds on the regret of GD. However in Section 4, we prove a lower bound on GD's regret in terms of the loss budget which shows that GD's regret is suboptimal by at least a factor of $\sqrt{k}$ when the regret is expressed as a function of the loss budget. The discussion of the *time dependent* regret upper bounds is more involved. We first convert the budget dependent regret bounds of the MEG algorithms into time dependent bounds. We shall see later, for lower bounds on the regret, time dependent bounds lead to budget dependent bounds (see Corollary 5.7). Before we do this, recall that the instance matrices $\boldsymbol{X}_t$ are L_1-bounded if their trace is at most one, and for L_∞-bounded instance matrices, their maximum eigenvalue is at most one. Note that for any vector $\boldsymbol{x}_t$ of length at most one, $\text{tr}(\boldsymbol{x}_t\boldsymbol{x}_t^\top) \leq 1$, and therefore vanilla PCA belongs to the case of L_1-bounded instance matrices.

Theorem 3.1 *When the instances are L_1-bounded, then for the online PCA with T trials, the following regret bounds hold for the Loss MEG and Gain MEG algorithms, respectively:*

$$\mathcal{R}_{\text{Loss MEG}} \;\leq\; m\sqrt{\frac{2T}{n}\log\frac{n}{m}} + m\log\frac{n}{m}, \qquad\qquad \mathcal{R}_{\text{Gain MEG}} \leq \sqrt{2T\;k\log\frac{n}{k}}. \tag{3.7}$$

Similarly, when the instances are L_∞-bounded, then the following regret bounds hold:

$$\mathcal{R}_{\text{Loss MEG}} \leq m\sqrt{2T\log\frac{n}{m}} + m\log\frac{n}{m}, \qquad \mathcal{R}_{\text{Gain MEG}} \leq k\sqrt{2T\log\frac{n}{k}}. \tag{3.8}$$

Proof The theorem will be proved by developing simple upper bounds on the loss/gain of the best rank k subspace that depend on the sequence length T. These upper bounds are then used as budgets in the previously obtained budget dependent bounds.

The best rank k subspace picks k eigenvectors of the covariance matrix $\boldsymbol{C} = \sum_{t=1}^{T} \boldsymbol{X}_t$ with the largest eigenvalues. Hence the total compression loss equals the sum of the smallest m eigenvalues of $\boldsymbol{C}$. If $\omega_1, \ldots, \omega_n$ denote all the eigenvalues of $\boldsymbol{C}$, then:

$$\sum_{i=1}^{n} \omega_i = \text{tr}(\boldsymbol{C}) = \sum_{t=1}^{T} \text{tr}\left(\boldsymbol{X}_t\right) \leq \begin{cases} T & \text{for } L_1\text{-bounded instances} \\ Tn & \text{for } L_\infty\text{-bounded instances} \end{cases}.$$

where the inequality follows from our definition of L_1-bounded and L_∞-bounded instance matrices. This implies that the sum of the m smallest eigenvalues is upper bounded by $\frac{Tm}{n}$ and Tm, respectively. By using these two bounds as the loss budget B_L in (3.3), we get the time dependent bound for Loss MEG for L_1-bounded and L_∞-bounded instances, respectively.

For the regret bounds of Gain MEG, we use the fact that B_G is upper bounded by T when instances are L_1-bounded and upper bounded by kT when the instances are L_∞-bounded, and plug these values for B_G into (3.5). ■

Table 3.2 compares time dependent upper bounds for each of the three algorithms (Loss MEG, Gain MEG, GD) where we consider each of the 4 variants of the problem: L_1-bounded or L_∞-bounded instance matrices versus $k \leq \frac{n}{2}$ or $k \geq \frac{n}{2}$.

As far as time dependent bounds are concerned, no single algorithm is optimal in all cases. In Table 3.2, the optimum bounds are shown in bold. The lower bounds matching these bold bounds within a constant factor will be proved in Section 5. Note that one version of MEG (either the loss or gain version) is optimal in each case, while GD is optimal only in first case (This is the most important case in practice: vanilla online PCA with $k \ll n$). For the remaining three cases, consider the ratio between the GD's bound and the better of the two MEG bounds, which is

- $\sqrt{\frac{n}{m}/\left(\log\frac{n}{m}\right)}$, when the instances are L_1-bounded and $k \geq \frac{n}{2}$,
- $\sqrt{\frac{n}{k}/\left(\log\frac{n}{k}\right)}$, when the instances are L_∞-bounded and $k \leq \frac{n}{2}$ and
- $\sqrt{\frac{n}{m}/\left(\log\frac{n}{m}\right)}$, when the instances are L_∞-bounded and $k \geq \frac{n}{2}$.

Since none of these three ratios can be upper bounded by a constant, GD is clearly suboptimal in each of the remaining three cases.

	L_1-bounded instances		L_∞-bounded instances	
	$k \le \frac{n}{2}$	$k \ge \frac{n}{2}$	$k \le \frac{n}{2}$	$k \ge \frac{n}{2}$
Loss MEG	$\boldsymbol{\sqrt{Tk}}$	$\boldsymbol{\sqrt{Tm\left(\log \frac{n}{m}\right) / \frac{n}{m}}}$	$\sqrt{Tkm}$	$\boldsymbol{\sqrt{Tm^2 \log \frac{n}{m}}}$
Gain MEG	$\sqrt{Tk \log \frac{n}{k}}$	$\sqrt{Tm}$	$\boldsymbol{\sqrt{Tk^2 \ln \frac{n}{k}}}$	$\sqrt{Tkm}$
GD	$\boldsymbol{\sqrt{Tk}}$	$\sqrt{Tm}$	$\sqrt{Tkm}$	$\sqrt{Tkm}$

Table 3.2: Comparison of the time dependent upper bounds on the regret of the Loss MEG, Gain MEG, and GD algorithms. Each column corresponds to one of the four combinations of L_1-bounded or L_∞-bounded instance matrices versus $k \le \frac{n}{2}$ or $k \ge \frac{n}{2}$, respectively. All bounds were given in Section 3.1 and Section 3.2: constants are omitted, we only show the leading term of each bound, and when we compare Loss and Gain MEG bounds, we use $m \ln \frac{n}{m} = \Theta(k)$ when $k \le \frac{n}{2}$ and $k \ln \frac{n}{k} = \Theta(m)$ when $k \ge \frac{n}{2}$. Recall that m is shorthand for $n - k$. The best (smallest) bound for each case (column) is shown in bold. In Section 5, all bold bounds will be shown to be optimal (within constant factors).

4. Lower Bounds on the Regret of GD

Recall that vanilla online PCA uses L_1-bounded instance matrices and the subspace dimension k is typically at most $\frac{n}{2}$. In this case Loss MEG has regret $O(\sqrt{Tk})$ and the regret of GD is $O(\sqrt{Tk})$ as well. As for loss budget dependent regret bounds, Loss MEG has regret $O(\sqrt{B_L k} + k)$ and we initially conjectured that GD has the same bound. However, this is not true: we will now show in this section an $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\}\})$ *lower bound* on the regret of GD for L_1-bounded instance sequences when $k \le \frac{n}{2}$. In contrast, Loss MEG's regret bound of $O(\sqrt{B_L k} + k)$ will be shown to be optimal in Section 5 for this case. It follows that GD is suboptimal by at least a factor of $\sqrt{k}$ when $B_L = \Omega(k^2)$. A detailed comparison of the lower bound for GD and the optimum upper bound is given in Table 4.1.

It suffices to prove lower bounds on GD's regret on a restricted class of instance matrices: We assume that all instance matrices are in the same eigensystem, i.e. they are diagonal matrices $\boldsymbol{X} = \mathrm{diag}(\boldsymbol{\ell})$ with $\boldsymbol{\ell} \in \mathbb{R}^n_{\ge 0}$. We call the diagonals $\boldsymbol{\ell}$ the *loss vectors*. All loss vectors in our lower bounds are restricted to be bit vectors in $\{0,1\}^n$. In the L_1-bounded instance case, the loss vectors are further restricted to be one of the n unit bit vectors $\boldsymbol{e}_i$, i.e. $\boldsymbol{X} = \mathrm{diag}(\boldsymbol{e}_i) = \boldsymbol{e}_i \boldsymbol{e}_i^\top$. In the L_∞-bounded instance case, the loss vectors $\boldsymbol{\ell}$ are arbitrary n-dimensional bit vectors.

When all instance matrices are diagonal then the off-diagonal elements in a parameter matrix $\boldsymbol{W}$ are irrelevant and therefore the algorithm's loss and regret is determined by the diagonals of the parameter matrices $\boldsymbol{W}$ which is of trace m. Therefore without loss of generality we can assume that the parameter matrices are diagonal as well, i.e. $\boldsymbol{W} = \mathrm{diag}(\boldsymbol{w})$ where $\boldsymbol{w}$ is a *weight vector* in $[0,1]^n$ with total weight m. Note that the loss becomes

Regret bounds for L_1-bounded instances, $k \leq \frac{n}{2}$	$B_L \leq k$	$k \leq B_L \leq k^2$	$k^2 \leq B_L$
Upper bound on regret of Loss MEG (see (3.3))	$\boldsymbol{O(k)}$	$\boldsymbol{O(\sqrt{B_L k})}$	$\boldsymbol{O(\sqrt{B_L k})}$
Lower bound on regret of GD (see Theorem 4.1)	$\Omega(k)$	$\Omega(B_L)$	$\Omega(k\sqrt{B_L})$

Table 4.1: Comparison of the loss budget dependent regret bounds for online PCA with $k \leq \frac{n}{2}$. Given dimension k of the subspace, each column shows the values of the two bounds for a specific range of the loss budget B_L. The first row gives the upper bound on the regret of Loss MEG in bold, which will be shown to be optimal in Section 5. The second row gives the lower bound on the regret of GD, which is suboptimal whenever $B_L \geq k$.

a dot product between the weight vector and the loss vector:

$$\mathrm{tr}(\boldsymbol{W}\boldsymbol{X}) = \mathrm{tr}(\mathrm{diag}(\boldsymbol{w})\,\mathrm{diag}(\boldsymbol{\ell})) = \boldsymbol{w} \cdot \boldsymbol{\ell}.$$

What is the prediction of the algorithm with a diagonal parameter matrix $\boldsymbol{W} = \mathrm{diag}(\boldsymbol{w})$? It probabilistically predicts with an m dimensional projection matrix $\boldsymbol{P}$ s.t. $\mathbb{E}[\boldsymbol{P}] = \mathrm{diag}(\boldsymbol{w})$. This means $\boldsymbol{P}$ is a subset of size m from $\{\boldsymbol{e}_1\boldsymbol{e}_1^\top, \boldsymbol{e}_2\boldsymbol{e}_2^\top, \ldots, \boldsymbol{e}_n\boldsymbol{e}_n^\top\}$. The diagonals of such projection matrices consists of exactly m ones and $n - m = k$ zeros. In other words the diagonals are indicator vectors of the chosen *subsets of size* m and the expected indicator vector equals the weight vector $\boldsymbol{w}$.

We just outlined one of the main insights of (Warmuth and Kuzmin, 2008): The restriction of the PCA problem to diagonal matrices corresponds to learning a subset of size m. The n components of the vectors are usually called *experts*. At trial t the algorithm chooses a subset of m experts. It then receives a loss vector $\boldsymbol{\ell} \in \mathbb{R}^n_{\geq 0}$ for the experts and incurs the total loss of the chosen m experts. The algorithm maintains its uncertainty over the m-sets by means of a parameter vector $\boldsymbol{w} \in [0,1]^n$ with total weight m, and it chooses the subset of size m probabilistically so that the expected indicator vector equals $\boldsymbol{w}$. We denote the set of such parameter vectors as $\boldsymbol{\mathcal{S}}_m$. In the L_1-bounded instance case, the loss vector is a unit bit vector (only one expert incurs a unit of loss). In the L_∞-bounded instance case, the loss vectors are restricted to be n-dimensional bit vectors.

4.1 Lower Bound on the Regret of the GD Algorithm

The GD algorithm for online PCA (2.5) specializes to the following update of the parameter vector for learning sets:

$$\begin{array}{ll} \text{Descent step:} & \hat{\boldsymbol{w}}_{t+1} = \boldsymbol{w}_t - \eta\boldsymbol{\ell}_t, \\ \text{Projection step:} & \boldsymbol{w}_{t+1} = \mathrm{argmin}_{\boldsymbol{w} \in \boldsymbol{\mathcal{S}}_m} \|\boldsymbol{w} - \hat{\boldsymbol{w}}_{t+1}\|^2. \end{array} \tag{4.1}$$

We now give a lower bound on the regret of the GD algorithm for the m-set problem. This lower bound is expressed as a function of the loss budget.

Theorem 4.1 *Consider the $m = n - k$ set problem with $k \leq n/2$ and unit bit vectors as loss vectors. Then for any fixed learning rate $\eta \geq 0$, the GD algorithm* (4.1) *can be forced to have regret* $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$.

We prove this theorem in Appendix C. From the fact that m-set problem is a special case of PCA problem, we get the following corollary, which shows that the GD algorithm is suboptimal (see Table 4.1 for an overview):

Corollary 4.2 *Consider the PCA problem with $k \leq n/2$ and L_1-bounded instance matrices. Then for any fixed learning rate $\eta \geq 0$, the GD algorithm (2.5) can be forced to have regret $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$.*

4.2 Lower Bound on the Regret of the Follow the Regularized Leader GD Algorithm (FRL-GD)

In the previous section, we showed that for online PCA with L_1-bounded instance matrices and $k \leq \frac{n}{2}$, the GD algorithm is suboptimal for loss budget dependent regret bounds. However, our lower bounds are only for the Mirror Descent version of GD given in (2.5). This algorithm is prone to "forgetting" lots of information about the past losses when projections with respect to inequality constraints are involved. Recall that at the end of each trial t, the mirror descent algorithm uses the last parameter $\boldsymbol{W}_t$ as a summary of the knowledge attained so far, and minimizes a trade-off between a divergence to the $\boldsymbol{W}_t$ and the loss on the last data point $\boldsymbol{x}_t$ to determine the next parameter $\boldsymbol{W}_{t+1}$. When the parameter resulting from the trade-off lies outside the parameter set, then it is projected back into the parameter set (see update (2.5)). In the case when the projection enforces inequality constraints on the parameters, information about the past losses may be lost. This issue was first discussed in Section 5.5 of Helmbold and Warmuth (2009). Curiously enough, Bregman projections with respect to only equality constraints do not lose information.

We now demonstrate in more detail the "forgetting" issue for the Mirror Descent GD algorithm when applied to online PCA. First recall that the batch PCA solution consists of the subspace spanned by the k eigenvectors belonging to the k largest values of the covariance matrix $\boldsymbol{C} = \sum_{t=1}^{T} \boldsymbol{x}_t \boldsymbol{x}_t^\top$. The complementary space is the $m = n-k$ dimensional subspace formed by the m eigenvectors of m largest eigenvalues of $-\boldsymbol{C}$. Hence, the final parameter $\boldsymbol{W}_{T+1}$ of the on-line algorithm should have the same eigenvectors as $-\boldsymbol{C}$, as well as the order of their corresponding eigenvalues. The descent step of (2.5) accumulates the scaled negated instance matrices $\boldsymbol{X}_t = \boldsymbol{x}_t \boldsymbol{x}_t^\top$, i.e. $\widehat{\boldsymbol{W}}_{t+1} = \boldsymbol{W}_t - \eta \boldsymbol{X}_t$. In the projection step of (2.5), the parameter matrix $\widehat{\boldsymbol{W}}_{t+1}$ is projected back to the parameter set $\boldsymbol{\mathcal{W}}_m$ by enforcing an equality constraint $\mathrm{tr}(\boldsymbol{W}_{t+1}) = m$ and inequality constraints that keep all the eigenvalues of $\boldsymbol{W}_{t+1}$ are in the range $[0, 1]$. The equality constraint on $\widehat{\boldsymbol{W}}_{t+1}$ results in adding to $\widehat{\boldsymbol{W}}_{t+1}$ a scaled version of the identity matrix $\boldsymbol{I}$ (See Appendix C). These iterated shifts do not affect either the eigenvectors or the order of their corresponding eigenvalues. However, when the inequality constraints are enforced, then at trial t the eigenvalues of $\widehat{\boldsymbol{W}}_{t+1}$ that are larger than 1 or less than 0 are capped at 1 and 0, respectively. Performing such a non-uniform capping of $\widehat{\boldsymbol{W}}_{t+1}$'s eigenvalues in each trial will result in a final parameter $\boldsymbol{W}_{T+1}$ with an eigensystem that is typically different from $-\boldsymbol{C}$. Therefore the PCA solution extracted from $\boldsymbol{W}_{T+1}$ and the covariance matrix $\boldsymbol{C}$ will not be the same.

There is another version of the GD algorithm that does not "forget": The Follow the Regularized Leader GD (FRL-GD) algorithm (see, e.g., Shalev-Shwartz and Singer (2007)[2])

2. This algorithm is also called as the Incremental Off-line Algorithm in (Azoury and Warmuth, 2001).

trades off the total loss on all data points against the Frobenius norm of the parameter matrix:

$$\begin{aligned}
&\text{Follow the regularized leader:} \\
&\quad \widehat{\boldsymbol{W}}_{t+1} = \operatorname{argmin} \left(\|\boldsymbol{W}\|_F^2 + \eta \sum_{q=1}^{t} \operatorname{tr}(\boldsymbol{W}\boldsymbol{X}_q) \right) = -\eta \sum_{q=1}^{t} \boldsymbol{X}_q, \\
&\text{Projection step:} \\
&\quad \boldsymbol{W}_{t+1} = \operatorname*{argmin}_{\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_m} \|\boldsymbol{W} - \widehat{\boldsymbol{W}}_{t+1}\|_F^2 = \operatorname*{argmin}_{\substack{\text{Eigenvalues of } \boldsymbol{W} \text{ in} \\ [0,1] \text{ and } \operatorname{tr}(\boldsymbol{W}) = m}} \|\boldsymbol{W} - \widehat{\boldsymbol{W}}_{t+1}\|_F^2.
\end{aligned} \tag{4.2}$$

Note that in each trial, the update (4.2) projects a parameter $\widehat{\boldsymbol{W}}_{t+1}$ that accumulates all the past scaled negated instance matrices $(-\eta \boldsymbol{X}_t)$ back to trial one. In contrast, the Mirror Descent update in (2.5) performs projection iteratively, i.e. it projects parameter matrices of previous trials that are projections themselves. Therefore, the FRL-GD algorithm circumvents the forgetting issue introduced by iterative projections with respect to inequality constraints. In fact the final parameter $\boldsymbol{W}_{T+1}$ of the FRL-GD is the projection of the scaled negated covariance matrix $\widehat{\boldsymbol{W}}_{T+1} = -\eta \sum_{t=1}^{T} \boldsymbol{x}_t \boldsymbol{x}_t^\top = -\eta \boldsymbol{C}$. We will show essentially in Appendix E that a single projection operation does not change the set of eigenvectors belonging to the m largest eigenvalues. This means that the eigenvectors belonging to the k smallest eigenvalues of $\boldsymbol{W}_{T+1}$ agree with the eigenvectors of $\boldsymbol{C}$ belonging to the k largest eigenvalues of $\boldsymbol{C}$.

Encouraged by this observation, we initially conjectured that the FRL-GD is strictly better than the commonly studied Mirror Descent version. More concretely, we conjectured that the FRL-GD has the optimal loss budget dependent regret bound for vanilla online PCA (as Mirror Descent MEG does which enforces the non-negativity constraints with its divergence). Unfortunately, we are able to show the opposite: The $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$ lower bound we showed for (Mirror Descent) GD in Theorem 4.1 also holds for FRL-GD. To be precise, we have the following theorem and corollary:

Theorem 4.3 *Consider the $m = n - k$ set problem with $k \leq n/2$ and unit bit vectors as loss vectors. Then for any fixed learning rate $\eta \geq 0$, the vector version of the FRL-GD algorithm* (4.2) *can be forced to have regret $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$.*

The proof is given in Appendix D. Theorem 4.3 immediately gives the lower bound on the regret of FRL-GD algorithm for the online PCA:

Corollary 4.4 *Consider the PCA problem with $k \leq n/2$ and L_1-bounded instance matrices. Then for any fixed learning rate $\eta \geq 0$, the FRL-GD algorithm* (4.2) *can be forced to have regret $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$.*

This shows that the worst case regret of the FRL-GD algorithm is the same as that of (Mirror Descent) GD, and hence suboptimal.

5. General Lower Bounds and Optimal Algorithms

In the previous section, we presented lower bounds on the regret of the GD algorithms. In this section we present lower bounds on the regret of *any* algorithm that solves the online PCA problem with L_1 and L_∞-bounded instance matrices. More importantly, these lower bounds match all our upper bounds on the regret of the MEG algorithms within a constant factor (See bold entries in Table 3.2 and Table 4.1). To be precise, we will prove in this section a series of regret lower bounds that match our loss budget dependent upper bound (3.3) on the regret of Loss MEG, and our time dependent upper bounds (Theorem 3.1) on the regret of Loss MEG and Gain MEG, respectively. For the time dependent bounds, our lower bounds will match the lower of the two MEG bounds in each of the four sub-cases of the problem, i.e. L_1 and L_∞-bounded instance matrices versus $k \leq \frac{n}{2}$ or $k \geq \frac{n}{2}$ (See Table 3.2 for a summary). Note that in one case the GD algorithm is also optimal: time dependent regret bounds for PCA with L_1-bound instances when $k \leq \frac{n}{2}$.

We begin with an overview of our proof techniques for proving lower bounds that hold for any algorithm. When proving *upper bounds* on the regret (in Section 3), we first proved upper bounds as a function of the loss budget B_L and then converted them into time dependent upper bounds. For *lower* bounds on the regret, the order is reversed: we first will show time dependent lower bounds and then convert them into loss budget dependent lower bounds. As discussed in Section 4, it suffices to prove lower bounds for the m-set problem, which is the hard special case when all instances are diagonal.

Let $\boldsymbol{\mathcal{A}}$ be the set of all online algorithms for the m-set problem. Such algorithms maintain a weight vector in $\boldsymbol{\mathcal{S}}_m$ (consisting of all vectors in $[0,1]^n$ of total weight m). For an algorithm $A \in \boldsymbol{\mathcal{A}}$, we denote its regret by $\mathcal{R}(A, \boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T)$ where $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T$ is a sequence of T loss vectors. The loss vectors $\boldsymbol{\ell}_t$ lie in a constraint set $\boldsymbol{\mathcal{L}}$. The constraint set $\boldsymbol{\mathcal{L}}$ either consists of all n dimensional unit bit vectors (the restriction of the L_1-bounded case we use in the lower bounds), or $\boldsymbol{\mathcal{L}} = \{0,1\}^n$ (the restriction used for the L_∞-bounded case). We use the standard method of lower bounding the regret for worst case loss sequences from $\boldsymbol{\mathcal{L}}$ by the expected regret when the loss vectors are generated i.i.d. with respect to a distribution $\mathcal{P}$ on $\boldsymbol{\mathcal{L}}$:

$$\min_{\substack{\text{over any} \\ \text{alg. } A \in \boldsymbol{\mathcal{A}}}} \left\{ \max_{\substack{\text{over loss vectors} \\ \boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T \in \boldsymbol{\mathcal{L}}}} \mathcal{R}(A, \boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T) \right\} \geq \min_{\substack{\text{over any} \\ \text{alg. } A \in \boldsymbol{\mathcal{A}}}} \left\{ \mathbb{E}_{\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T \sim \mathcal{P}^T} \left[\, \mathcal{R}(A, \boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T) \, \right] \right\}.$$

Each lower bound is proved as follows: Choose a distribution $\mathcal{P}$ on $\boldsymbol{\mathcal{L}}$, and then show a lower bound on the expected regret of any algorithm $A \in \boldsymbol{\mathcal{A}}$. Note that this expectation becomes the expected loss of A minus the expected loss of the best comparator (i.e. the best m-set). We first prove time dependent regret lower bounds with L_1 and L_∞-bounded instance vectors in sections 5.1 and 5.2, respectively. Finally we convert these lower bounds into loss budget dependent lower bounds (in Section 5.3).

5.1 Time Dependent Lower Bounds for Online PCA

Recall that $m = n - k$. First, we give a lower bound on the regret of any algorithm for the m-set problem, when $k \leq \frac{n}{2}$:

Theorem 5.1 *Consider the m-set problem with unit bit vectors as loss vectors. Then for $k \leq \frac{n}{2}$ and $T \geq k$, any online algorithm suffers worst case regret at least $\Omega(\sqrt{Tk})$.*

The proof is given in Appendix F. We lower bound the expected loss w.r.t. the distribution $\mathcal{P}$ which is uniform on the first $2k$ unit bit vectors. Note that Theorem 5.1 requires the condition $T \geq k$. For the case $T < k$, there is a lower bounds of $\Omega(T)$ (See Theorem G.1 in Appendix G). When the loss vectors are bit vectors, then any algorithm has loss (and regret) $O(T)$. Therefore when $T < k$, any algorithm achieves the minimax regret up to a constant factor.

We now consider the uncommon case when $k \geq \frac{n}{2}$:

Theorem 5.2 *Consider the m-set problem with unit bit vectors as loss vectors. Then for $k \geq \frac{n}{2}$ and $T \geq n \log_2(n/m)$, any online algorithm suffers worst case regret of at least $\Omega(m\sqrt{\frac{T}{n} \ln \frac{n}{m}})$.*

We now set $\mathcal{P}$ to the uniform distribution on all n unit bit vectors (See Appendix F). The small T case (here $T < n \log_2(n/m)$) is slightly more involved. There is a lower bound of $\Omega(\frac{m}{n}T)$ regret for any algorithm (see Theorem G.3 in Appendix G). Also the algorithm which predicts with the uniform weight $\frac{m}{n}$ on all experts achieves the matching regret of $O(\frac{m}{n}T)$.

Recall that the m-set problem with unit bit vectors as loss vectors is a special case of the online PCA problem with L_1-bounded instance matrices. Combining the above two lower bounds for different ranges of k with our upper bound (Theorem 3.1) on the regret of Loss MEG for online PCA with L_1-bounded instances gives the following corollary:

Corollary 5.3 *Consider the problem of online PCA with L_1-bounded instance matrices. Then for $T \geq n \log_2(n/m)$, the $\Theta(m\sqrt{\frac{T}{n} \ln \frac{n}{m}})$ regret of Loss MEG is within a constant factor of the minimax regret.*

Note that we do not use the condition $T \geq k$ of Theorem 5.1, since when $k \leq \frac{n}{2}$, $k = \Theta(n \log_2(n/m))$.

5.2 Time Dependent Lower Bound for the Generalization with L_∞-Bounded Instance Matrices

We first give the time dependent lower bound for the m-set problem with bit vectors.

Theorem 5.4 *Consider the m-set problem with bit vectors as loss vectors. Then for $T \geq \log_2 \frac{n}{\min\{k,m\}}$, any online algorithm suffers worst case regret of at least*

$$\Omega(k\sqrt{T \ln \frac{n}{k}}) \text{ when } k \leq \frac{n}{2} \quad \text{or} \quad \Omega(m\sqrt{T \ln \frac{n}{m}}) \text{ when } k \geq \frac{n}{2}.$$

The proof is given in Appendix F. The distribution $\mathcal{P}$ is such that each expert incurs a unit of loss with probability $1/2$ independently from the other experts. For the small T case ($T < \log_2 \frac{n}{\min\{k,m\}}$), there is a lower bound of $\Omega(\min\{Tm, Tk\})$ (See Theorem G.4 and Theorem G.5 in Appendix G). A matching upper bound of $O(\min\{Tm, Tk\})$ on the

regret of any algorithm can be reasoned as follows: At each trial, the algorithm plays with $\boldsymbol{W}_t \in \boldsymbol{\mathcal{W}}_m$ and suffers loss $\mathrm{tr}(\boldsymbol{W}_t \boldsymbol{X}_t)$. Since $\mathrm{tr}(\boldsymbol{W}_t) = m$, the algorithm suffers loss at most m per trial and for T trials, and the cumulative loss (and thus regret) is at most Tm. The Tk upper bound can be showed similarly by considering the "gain" of the best rank k projector $\boldsymbol{P}^*$, which is $\sum_{t=1}^{T} \mathrm{tr}(\boldsymbol{P}^* \boldsymbol{X}_t) \leq Tk$. Combining the lower bounds of Theorem 5.4 with the upper bounds on the regret of Loss MEG and Gain MEG when the instance matrices are L_∞-bounded (Inequality (3.8)), results in the following corollary, which states that the Gain MEG is optimal for $k \leq \frac{n}{2}$ while the Loss MEG is optimal for $k \geq \frac{n}{2}$.

Corollary 5.5 *Consider the generalization of online PCA where the instance matrices are L_∞-bounded.*

- *When $k \leq \frac{n}{2}$ and $T \geq \log_2 \frac{n}{k}$, then the regret $\Theta(k\sqrt{T \log \frac{n}{k}})$ of Gain MEG is within a constant factor of the minimax regret.*
- *When $k \geq \frac{n}{2}$ and $T \geq \log_2 \frac{n}{m}$, then the regret $\Theta(m\sqrt{T \log \frac{n}{m}})$ of Loss MEG is within a constant factor of the minimax regret.*

5.3 Loss Budget Dependent Lower Bounds

In this subsection, we give regret lower bounds that are functions of the loss budget B_L (defined in (3.1)). Similar to our loss budget dependent upper bound (3.3) on the regret of Loss MEG, the loss dependent lower bounds are the same for both unit and arbitrary bit vectors:

Theorem 5.6 *For the m-set problem with either unit or arbitrary bit vectors as loss vectors, any online algorithm suffers worst case regret at least $\Omega(\sqrt{B_L m \ln \frac{n}{m}}) + m \ln \frac{n}{m})$.*

The proof of the theorem is given in Appendix H. We convert the time dependent lower bounds given in Theorem 5.1 and Theorem 5.2 into loss budget dependent ones. Note that unlike our time dependent lower bounds, Theorem 5.6 is stated for the full range of the loss budget parameter B_L. The proof also distinguishes between a small and a large budget case depending on whether $B_L \leq m \ln \frac{n}{m}$. The lower bound of $\Theta(m \ln \frac{n}{m})$ follows from a conversion. However the upper bound of $O(m \ln \frac{n}{m})$ for the small budget case is non-trivial. Incidentally, this upper bound is achieved by Loss MEG.

Finally, combining this lower bound with the upper bounds (3.3) on the regret of Loss MEG, gives the following corollary, which establishes the optimality of Loss MEG no matter if the instance matrices are L_1 or L_∞-bounded.

Corollary 5.7 *Consider the problem of online PCA with L_1 or L_∞-bounded instance matrices. Then the regret $\Theta(\sqrt{B_L m \ln \frac{n}{m}} + m \ln \frac{n}{m})$ of Loss MEG is within a constant factor of the minimax regret.*

6. Conclusion

In this paper, we carefully studied two popular online algorithms for PCA: the Gradient Descent (GD) and Matrix Exponentiated Gradient (MEG) algorithms. For the case when the instance matrices are L_1-bounded, we showed that both algorithms are optimal to within

a constant factor when the worst-case regret is expressed as a function of the number of trials. Furthermore, when considering regret bounds as a function of a loss budget, then MEG remains optimal and strictly outperforms GD for L_1-bounded instances. We also studied the case when the instance matrices are L_∞-bounded. Again we show MEG to be optimal and strictly better than GD in this case. It follows that MEG is the algorithm of choice for both cases. Note that that vanilla PCA (where the instances are outer products of vectors of length at most one) is subsumed by the case of L_1-bounded instance matrices.

In this paper we focused on obtaining online algorithms with optimal regret and we ignored efficiency concerns. Straightforward implementations of both the GD and MEG online PCA updates required $O(n^3)$ computation per trial (because they require an eigen-decomposition of the parameter matrices). This leads to a major open problem for online PCA (Hazan et al., 2010): Is there any algorithm that can achieve optimal regret with $O(n^2)$ computation per trial. To this end, Arora et al. (2013) considers the Gain version of GD (Equation (2.3), with the squared Euclidean distance as the divergence) where the projection enforces the additional constraint that the parameter matrix $\boldsymbol{W}_t$ has rank $\widehat{k}$. Encouraging experimental results are provided for the choice $\widehat{k} = k + 1$. However, as we shall see immediately, in the most basic case when the instance matrices are outer products of unit length vectors $\boldsymbol{x}_t$ that are chosen by an adversary, then any algorithm that uses parameter matrices of rank $\widehat{k}$ less than n can be forced to suffer worst case regret linear in T. Recall that the parameter matrix $\boldsymbol{W}_t$ at trial t is simply the expected projection matrix of rank k chosen by the algorithm and this matrix is defined for any (deterministic or randomized) algorithm. We give an adversary argument for any algorithm for which the rank of the parameter matrix $\boldsymbol{W}_t$ at any trial t is at most $\widehat{k}$. The parameter matrices are known to the adversary. Also the initial parameter matrix $\boldsymbol{W}_1$ must have rank $\widehat{k}$ and be known to the adversary. For any algorithm following this setup the adversary argument proceeds as follows: At the beginning of the game the adversary fixes any subspace $\mathcal{Q}$ of dimension $\widehat{k} + 1$. In each trial, the adversary picks a unit length vector $\boldsymbol{x}_t \in \mathcal{Q}$, which is in the null space of the parameter matrix $\boldsymbol{W}_t$ of the algorithm (This is always possible, because the dimension of $\mathcal{Q}$ is larger than the rank of $\boldsymbol{W}_t$). After T trials, the algorithm has zero gain, while the total gain T is accumulated within subspace $\mathcal{Q}$. This means that there are k orthogonal directions within $\mathcal{Q}$ with the total gain at least $\frac{k}{\widehat{k}+1}T$ and therefore, the algorithm suffers regret at least $\frac{k}{\widehat{k}+1}T$.

Besides restricting the rank of the parameter matrix, a second approach is to add perturbations to the current covariance matrix and then find the eigenvectors of the k-largest eigenvalues (Hazan et al., 2010). So far this approach has not led to algorithms with optimal regret bounds and $O(n^2)$ update time. Some partial results recently appeared in Garber et al. (2015) and Kotlowski and Warmuth (2015).

Acknowledgments. Jiazhong Nie and Manfred K. Warmuth were support by NSF grant IIS-1118028. Wojciech Kotłowski was supported by the Polish National Science Centre grant 2013/11/D/ST6/03050 and by the Foundation for Polish Science grant Homing Plus 2012-5/5.

Appendix A. Proof of Upper Bound (3.5) on the Regret of Gain MEG

Proof The proof is based on the by now standard proof techniques of Tsuda et al. (2005). Let $\boldsymbol{W}_t \in \boldsymbol{\mathcal{W}}_k$ be the parameter of the Gain MEG algorithm at trial t and $\boldsymbol{X}_t$ be the instance matrix at this trial. Now plugging the (un-normalized) relative entropy $\Delta(\boldsymbol{W}, \boldsymbol{W}_t) = \mathrm{tr}(\boldsymbol{W}(\log \boldsymbol{W} - \log \boldsymbol{W}_t) + \boldsymbol{W}_t - \boldsymbol{W})$ into the descent step of the Gain MEG algorithm (2.7) gives:

$$\widehat{\boldsymbol{W}}_{t+1} = \exp(\log \boldsymbol{W}_t + \eta \boldsymbol{X}_t) \quad \text{where } \eta \geq 0 \text{ is the learning rate.}$$

Take any projection matrix $\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_k$ as a comparator and use $\Delta(\boldsymbol{W}, \boldsymbol{W}_t) - \Delta(\boldsymbol{W}, \boldsymbol{W}_{t+1})$ as a measure of progress towards $\boldsymbol{W}$:

$$\begin{aligned}
\Delta(\boldsymbol{W}, \boldsymbol{W}_t) - \Delta(\boldsymbol{W}, \boldsymbol{W}_{t+1}) &\geq \Delta(\boldsymbol{W}, \boldsymbol{W}_t) - \Delta(\boldsymbol{W}, \widehat{\boldsymbol{W}}_{t+1}) \\
&= \mathrm{tr}(\boldsymbol{W}(\log \widehat{\boldsymbol{W}}_{t+1} - \log \boldsymbol{W}_t) + \boldsymbol{W}_t - \widehat{\boldsymbol{W}}_{t+1}) \\
&= \mathrm{tr}(\eta \boldsymbol{W} \boldsymbol{X}_t) + \mathrm{tr}(\boldsymbol{W}_t - \exp(\log \boldsymbol{W}_t + \eta \boldsymbol{X}_t)) \\
&\geq \mathrm{tr}(\eta \boldsymbol{W} \boldsymbol{X}_t) + \mathrm{tr}(\boldsymbol{W}_t - \boldsymbol{W}_t \exp(\eta \boldsymbol{X}_t)) \\
&= \mathrm{tr}(\eta \boldsymbol{W} \boldsymbol{X}_t) + \mathrm{tr}(\boldsymbol{W}_t(\boldsymbol{I} - \exp(\eta \boldsymbol{X}_t)),
\end{aligned} \tag{A.1}$$

where the first inequality follows from the Pythagorean Theorem and the second from the Golden-Thompson inequality: $\mathrm{tr}(\exp(\log \boldsymbol{W}_t + \eta \boldsymbol{X}_t) \leq \mathrm{tr}(\boldsymbol{W}_t \exp(\eta \boldsymbol{X}_t))$. By Lemma 2.1 of Tsuda et al. (2005),

$$\mathrm{tr}(\boldsymbol{W}_t(\boldsymbol{I} - \exp(\eta \boldsymbol{X}_t))) \geq (1 - e^{\eta})\, \mathrm{tr}(\boldsymbol{W}_t \boldsymbol{X}_t),$$

and therefore

$$\Delta(\boldsymbol{W}, \boldsymbol{W}_t) - \Delta(\boldsymbol{W}, \boldsymbol{W}_{t+1}) \geq \eta \underbrace{\mathrm{tr}(\boldsymbol{W} \boldsymbol{X}_t)}_{\substack{\text{gain of the} \\ \text{comparator}}} + (1 - e^{\eta}) \underbrace{\mathrm{tr}(\boldsymbol{W}_t \boldsymbol{X}_t)}_{\substack{\text{gain of the} \\ \text{algorithm}}}.$$

Summing over trials gives:

$$\eta \underbrace{\sum_{t=1}^{T} \mathrm{tr}(\boldsymbol{W} \boldsymbol{X}_t)}_{\substack{\text{total gain } G_{\boldsymbol{W}} \text{ of} \\ \text{the comparator } \boldsymbol{W}}} + (1 - e^{\eta}) \underbrace{\sum_{t=1}^{T} \mathrm{tr}(\boldsymbol{W}_t \boldsymbol{X}_t)}_{\substack{\text{total gain } G_A \\ \text{of Gain MEG}}} \leq \underbrace{\Delta(\boldsymbol{W}, \boldsymbol{W}_1)}_{\substack{\leq k \log \frac{n}{k} \\ \text{with initialization} \\ \boldsymbol{W}_1 = \frac{k}{n} \boldsymbol{I}}} - \underbrace{\Delta(\boldsymbol{W}, \boldsymbol{W}_{T+1})}_{\geq 0}.$$

We now rearrange the terms to bound the regret of Gain MEG:

$$G_{\boldsymbol{W}} - G_A \leq \frac{1}{e^{\eta} - 1}\, k \log \frac{n}{k} + \left(1 - \frac{\eta}{e^{\eta} - 1}\right) G_{\boldsymbol{W}}. \tag{A.2}$$

Since $e^{\eta} \geq 1 + \eta$, the coefficient $\frac{1}{e^{\eta} - 1}$ of the first term on the RHS is upper bounded by $\frac{1}{\eta}$. Next we upper bound the coefficient of the second term by η:

$$1 - \frac{\eta}{e^{\eta} - 1} = 1 - \frac{\eta e^{-\eta}}{1 - e^{-\eta}} \leq 1 - \frac{\eta e^{-\eta}}{\eta} = 1 - e^{-\eta} \leq \eta.$$

The inequality (3.5) on the regret of Gain MEG now follows from these two upper bounds, the budget inequality $G_{\boldsymbol{W}} \leq B_G$ and from tuning the learning rate as a function of B_G:

$$\mathcal{R}_{\text{Gain EG}} \quad \leq \quad \frac{k \log \frac{n}{k}}{\eta} + \eta B_G \quad \overset{\eta=\sqrt{\frac{\log \frac{n}{k}}{B_G}}}{=} \quad \sqrt{2 B_G \, k \log \frac{n}{k}}.$$

■

Appendix B. Proof of Upper Bound (3.6) on the Regret of GD

Proof This proof is also standard (Herbster and Warmuth, 2001). Minor alterations are needed because we have matrix parameters. Let $\boldsymbol{W}_t \in \boldsymbol{\mathcal{W}}_m$ be the parameter of the GD algorithm at trial t and $\boldsymbol{X}_t$ be the instance matrix at this trial. Then for the best comparator $\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_m$ and any learning rate $\eta \geq 0$, the following holds

$$\|\boldsymbol{W}_{t+1} - \boldsymbol{W}\|_F^2 \leq \|\widehat{\boldsymbol{W}}_{t+1} - \boldsymbol{W}\|_F^2 = \|\boldsymbol{W}_t - \boldsymbol{W}\|_F^2 - 2\eta \operatorname{tr}((\boldsymbol{W}_t - \boldsymbol{W})\boldsymbol{X}_t^\top) + \eta^2 \|\boldsymbol{X}_t\|_F^2,$$

where the inequality follows from the Pythagorean Theorem (Herbster and Warmuth, 2001) and the equality follows from the descent step of the GD algorithm (see (2.5)). By rearranging terms, we have

$$\operatorname{tr}(\boldsymbol{W}_t \boldsymbol{X}_t^\top) - \operatorname{tr}(\boldsymbol{W} \boldsymbol{X}_t^\top) \;\leq\; \frac{\|\boldsymbol{W}_t - \boldsymbol{W}\|_F^2 - \|\boldsymbol{W}_{t+1} - \boldsymbol{W}\|_F^2}{2\eta} + \frac{\eta \|\boldsymbol{X}_t\|_F^2}{2}.$$

Note that the LHS is the regret in trial t w.r.t. $\boldsymbol{W}$. By summing all trials, we have that the (total) regret $\mathcal{R}_{GD} = \sum_{t=1}^T \operatorname{tr}(\boldsymbol{W}_t \boldsymbol{X}_t^\top)$ is upper bounded by

$$\frac{\|\boldsymbol{W}_1 - \boldsymbol{W}\|_F^2 - \cancel{\|\boldsymbol{W}_{T+1} - \boldsymbol{W}\|_F^2}}{2\eta} + \frac{\eta \sum_{t=1}^T \|\boldsymbol{X}_t\|_F^2}{2} \;\leq\; \frac{k(n-k)}{2n\eta} + \frac{\eta \sum_{t=1}^T \|\boldsymbol{X}_t\|_F^2}{2}, \quad \text{(B.1)}$$

where we used $\|\boldsymbol{W}_1 - \boldsymbol{W}\|_F^2 \leq \frac{k(n-k)}{n}$ since $\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_m$ and $\boldsymbol{W}_1 = \frac{n-k}{n}\boldsymbol{I}$. In the L_1-bounded instance matrix case (when $\|\boldsymbol{X}\|_F^2 \leq 1$), (B.1) can be further simplified as

$$\mathcal{R}_{GD} \;\leq\; \frac{k(n-k)}{2n\eta} + \frac{\eta T}{2}.$$

By setting $\eta = \frac{k(n-k)}{nT}$, we obtain the $\sqrt{\frac{k(n-k)}{n}T}$ regret bound for the L_1-bounded instance case. When the instance matrices are L_∞-bounded, then $\|\boldsymbol{X}_t\|_F^2 \leq n$ and hence, $\mathcal{R}_{GD} \leq \sqrt{k(n-k)T}$ with $\eta = \frac{k(n-k)}{T}$. ■

Appendix C. Proof of Theorem 4.1

Theorem 4.1 gives a lower bound on the regret of the GD algorithm for the m-set problem with unit bit vectors as loss vectors. At each trial of the m-set problem, the online algorithm

first predicts with a weight vector $\boldsymbol{w}_t \in [0,1]^n$, the coordinates of which sum to m. Then the algorithm receives a unit bit vector $\boldsymbol{\ell}_t$ and suffers loss $\boldsymbol{w}_t \cdot \boldsymbol{\ell}_t$. The GD algorithm for online PCA (2.5) specializes to the following updates of the parameter vector for learning m-sets:

$$\begin{array}{ll} \text{Descent step:} & \hat{\boldsymbol{w}}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{\ell}_t, \\ \text{Projection step:} & \boldsymbol{w}_{t+1} = \operatorname{argmin}_{\boldsymbol{w} \in \boldsymbol{\mathcal{S}}_m} \|\boldsymbol{w} - \hat{\boldsymbol{w}}_{t+1}\|^2, \end{array} \tag{C.1}$$

where $\eta > 0$ is the learning rate and $\boldsymbol{\mathcal{S}}_m = \{\boldsymbol{w} \in [0,1]^n : \sum_{i=1}^n w_i = m\}$.

Since our lower bound for GD must hold no matter what the *fixed* learning rate η is, we construct two adversarial loss sequences: The first causes the GD algorithm to suffer large regret when η is small and the second causes large regret when η is large. Specifically, we will show that the GD algorithm suffers regret at least $\Omega(k/\eta)$ on the first sequence, and at least $\Omega(\min\{B_L, kB_L\eta\})$ on the second sequence. We will then show that the lower bound of the theorem follows by taking the maximum of these two bounds and by solving for the learning rate that minimizes this maximum. The first sequence consists of unit losses assigned to the first k experts. At each trial, the adversary gives a unit of loss to the expert (out of the first k) with the largest current weight. If the learning rate η is small, then the weights assigned to the first k experts decrease too slowly (Lemma C.2). This causes the algorithm to suffer a substantial amount of loss on the first sequence, while the loss of the remaining m experts remains zero. The second sequence consists of unit losses assigned to the first $k+1$ experts. As before, the adversary always gives the expert with the largest weight (now out of the first $k+1$) a unit of loss. Intuitively, the GD algorithm will give high weight to the $m-1 = n-(k+1)$ loss free experts and the best out of the first $k+1$ experts. As the η gets larger, the algorithm puts more and more weight on the *current* best out of the $k+1$ experts instead of hedging its bets over all $k+1$ experts. So the algorithm becomes more and more deterministic and the adversary strategy of hitting the expert with the largest weight (out of the first $k+1$) causes the algorithm to suffer a substantial loss (Lemma C.3). Formalizing these findings is not simple as the projection step of the GD algorithm does not have a closed form. Hence, we need to resort to the Karush-Kuhn-Tucker optimality conditions and prove a sequence of lemmas before assembling all the pieces for proving Theorem 4.1.

Let α_i be a dual variable for the constraint $w_{t+1,i} \geq 0$ $(i = 1, \ldots, n)$, β_i be a dual variable for the constraint $w_{t+1,i} \leq 1$ $(i = 1, \ldots, n)$, and γ be a dual variable for the constraint $\sum_{i=1}^n w_{t+1,i} = m$. Then the KKT conditions on the projection step of (C.1) have the following form: For $i = 1, \ldots, n$,

$$\begin{array}{lll} \text{Stationarity:} & w_{t+1,i} = -w_{t,i} - \eta \ell_{t,i} + \gamma + \alpha_i - \beta_i, & \\ \text{Complementary slackness:} & w_{t+1,i}\,\alpha_i = 0, \qquad (w_{t+1,i} - 1)\beta_i = 0, & \\ \text{Primal feasibility:} & \sum_{i=1}^n w_{t+1,i} = m, \qquad 0 \leq w_{t+1,i} \leq 1, & \\ \text{Dual feasibility:} & \alpha_i \geq 0, \qquad \beta_i \geq 0. & \end{array} \tag{C.2}$$

Note that since the projection step of (C.1) is a convex optimization problem, these conditions are necessary and sufficient for the optimality of a solution. Hence, for any intermediate weight vector $\hat{\boldsymbol{w}}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{\ell}_t$, if a set of primal and dual variables $\boldsymbol{w}_{t+1}, \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n), \boldsymbol{\beta} = (\beta_1, \ldots, \beta_n), \gamma$ satisfy all the conditions (C.2), then they are the unique primal and dual solutions of the projection step.

We start with a special case where the GD update (C.1) actually has a closed form solution:

Lemma C.1 *Consider a trial of the m-set problem with n experts, when only one expert incurs a unit of loss. If this expert has weight w and all remaining experts have weight at most $1 - \min\{\frac{\eta}{n}, \frac{w}{n-1}\}$, then the GD algorithm with learning rate $\eta > 0$ will decrease w by $\min\{\frac{(n-1)\eta}{n}, w\}$ and increase all the other weights by $\min\{\frac{\eta}{n}, \frac{w}{n-1}\}$.*

Proof W.l.o.g., the first expert incurs a unit of loss in trial t, i.e. $w_{t,1} = w$ and $\boldsymbol{\ell}_t = \boldsymbol{e}_1$, where $\boldsymbol{e}_1$ is the unit bit vector with first coordinate equal to 1 (and all other coordinates equal to 0). To solve the projection step of the GD update (C.1), we distinguish two cases based on the value of $w_{t,1}$. In each case we propose a solution to the projection step and show that it is a valid solution by verifying the KKT conditions (C.2).

Case $w_{t,1} = w \geq \frac{n-1}{n}\eta$: The proposed solution is $\gamma = \frac{\eta}{n}$ and for $1 \leq i \leq n$, $\alpha_i = \beta_i = 0$,

$$w_{t+1,i} = \begin{cases} w_{t,1} - \frac{n-1}{n}\eta & \text{for } i = 1 \\ w_{t,i} + \frac{\eta}{n} & \text{for } i \geq 2 \end{cases}.$$

All KKT conditions are easy to check, except for the primal feasibility condition: $w_{t+1,i} \leq 1$, for $i \geq 2$. By the assumption of the lemma, $w_{t,i} \leq 1 - \min\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\}$. Since we are in the case $w_{t,1} \geq \frac{n-1}{n}\eta$, we have $w_{t,i} \leq 1 - \frac{\eta}{n}$ and therefore

$$w_{t+1,i} = w_{t,i} + \frac{\eta}{n} \leq 1 - \frac{\eta}{n} + \frac{\eta}{n} = 1.$$

We conclude that in this case, the first weight decreases by $\frac{n-1}{n}\eta$ and all the other weights increase by $\frac{\eta}{n}$.

Case $w_{t,1} = w < \frac{n-1}{n}\eta$: The proposed solution is $\gamma = \frac{w_{t,1}}{n-1}$ and for $1 \leq i \leq n$, $\beta_i = 0$,

$$\alpha_i = \begin{cases} \eta - \frac{n}{n-1}w_{t,1} & \text{for } i = 1 \\ 0 & \text{for } i \geq 2 \end{cases}, \qquad w_{t+1,i} = \begin{cases} 0 & \text{for } i = 1 \\ w_{t,i} + \frac{w_{t,1}}{n-1} & \text{for } i \geq 2 \end{cases}.$$

Again, all KKT conditions are easy to check, except for the primal feasibility condition $w_{t+1,i} \leq 1$ for $i \geq 2$. By the assumption of the lemma $w_{t,i} \leq 1 - \min\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\}$. Since we are in the case $w_{t,1} < \frac{n-1}{n}\eta$, we have $w_{t,i} \leq 1 - \frac{w_{t,1}}{n-1}$ and therefore

$$w_{t+1,i} = w_{t,i} + \frac{w_{t,1}}{n-1} \leq 1 - \frac{w_{t,1}}{n-1} + \frac{w_{t,1}}{n-1} = 1.$$

We conclude that in this case, the first weight decreases by $w_{t,1}$ and all the other weights increase by $\frac{w_{t,1}}{n-1}$. Combining the above two cases proves the lemma. ■

Our next lemma considers the general case when the weight vector before the update does not necessarily satisfy the assumption in Lemma C.1, i.e. the weights of the experts not incurring loss may be larger than $1 - \min\{\frac{\eta}{n}, \frac{w}{n-1}\}$ (where w is the weight of the only expert incurring loss).

Lemma C.2 *Consider a trial of the m-set problem with n experts, when only one expert incurs a unit of loss. If this expert has weight w, then the GD algorithm with learning rate $\eta > 0$ will decrease w by at most η and will not decrease the weights of any other experts. Furthermore, if any expert not incurring loss has weight at least $1 - \min\{\frac{\eta}{n}, \frac{w}{n-1}\}$, then its weight will be set to 1 by the capping constraint.*

Proof Let $\boldsymbol{w}_t$ be the weight vector at the beginning of the trial and assume w.l.o.g. that the first expert incurs one unit of loss, i.e. $\boldsymbol{\ell}_t = \boldsymbol{e}_1$. Let $\boldsymbol{w}_{t+1}, \boldsymbol{\alpha}, \boldsymbol{\beta}$ and γ denote the variables satisfying the KKT conditions (C.2). The lemma now states that:

$$w_{t+1,1} \geq w_{t,1} - \eta \qquad \text{and} \qquad w_{t+1,i} \geq w_{t,i}, \quad \text{for } 2 \leq i \leq n, \tag{C.3}$$

and furthermore

$$w_{t+1,i} = 1, \quad \text{for any } 2 \leq i \leq n \text{ such that } w_{t,i} \geq 1 - \min\left\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\right\}. \tag{C.4}$$

We first prove (C.3). By the stationarity condition of (C.2) and the assumption $\boldsymbol{\ell}_t = \boldsymbol{e}_1$, we have that

$$w_{t+1,1} - w_{t,1} \;=\; \cancel{w_{t,1}} - \eta + \alpha_1 - \beta_1 + \gamma - \cancel{w_{t,1}} \;=\; -\eta + \alpha_1 - \beta_1 + \gamma,$$

$$\text{and for } 2 \leq i \leq n\text{:} \quad w_{t+1,i} - w_{t,i} \;=\; \cancel{w_{t,1}} + \alpha_i - \beta_i + \gamma - \cancel{w_{t,1}} \;=\; \alpha_i - \beta_i + \gamma.$$

Therefore, to prove (C.3), it suffices to show $\alpha_i - \beta_i + \gamma \geq 0$ for $1 \leq i \leq n$. By the dual feasibility condition of (C.2), $\alpha_i \geq 0$ but $-\beta_i \leq 0$. However, when $-\beta_i < 0$, we have $w_{t+1,i} = 1$ by the complementary slackness condition, and therefore (C.3) holds trivially in this case (noting that $w_{t,i} \leq 1$). Now we only need to show $\gamma \geq 0$. We do this by summing $w_{t,i} - \eta\ell_{t,i} + \gamma$ over indices i such that $w_{t+1,i} > 0$:

$$\begin{aligned}
\sum_{i:1\leq i\leq n, w_{t+1,i}>0} (w_{t,i} - \eta\ell_{t,i} + \gamma) &\overset{\substack{\alpha_i = 0 \text{ since} \\ w_{t+1,i} > 0}}{=} \sum_{i:1\leq i\leq n, w_{t+1,i}>0} (w_{t,i} - \eta\ell_{t,i} + \gamma + \alpha_i) \\
&\geq \sum_{i:1\leq i\leq n, w_{t+1,i}>0} (w_{t,i} - \eta\ell_{t,i} + \gamma + \alpha_i - \beta_i) \\
&\geq \sum_{i:1\leq i\leq n, w_{t+1,i}>0} (w_{t,i}) \quad = \quad m.
\end{aligned} \tag{C.5}$$

Furthermore, since both the learning rate η and the loss vector $\boldsymbol{\ell}_t$ are non-negative, we have that for all $1 \leq i \leq n$,

$$\sum_{i:1\leq i\leq n, w_{t+1,i}>0} (w_{t,i} - \eta\ell_{t,i}) \quad \leq \quad \sum_{i:1\leq i\leq n, w_{t+1,i}>0} (w_{t,i}) \quad \leq \quad m.$$

Combining the above inequality with (C.5) implies that $\gamma \geq 0$, which completes our proof of (C.3).

Next we prove (C.4). By the stationarity condition of (C.2) and the assumption $\boldsymbol{\ell}_t = \boldsymbol{e}_1$, we have that for $2 \leq i \leq n$,

$$w_{t+1,i} \;=\; w_{t,i} - \eta\ell_{t,i} + \alpha_i - \beta_i + \gamma \;=\; w_{t,i} + \alpha_i - \beta_i + \gamma. \tag{C.6}$$

Now if we further assume that $w_{t,i} \geq 1 - \min\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\}$, then (C.6) is lower bounded by

$$w_{t+1,i} \;=\; w_{t,i} + \alpha_i - \beta_i + \gamma \;\geq\; 1 - \min\left\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\right\} + \alpha_i - \beta_i + \gamma.$$

Thus to prove (C.4), it suffices to show that $-\min\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\} + \alpha_i - \beta_i + \gamma \geq 0$. By the dual feasibility condition of (C.2), $\alpha_i \geq 0$ but $-\beta_i \leq 0$. However, when $-\beta_i < 0$, then $w_{t+1,i} = 1$ follows directly from the complementary slackness condition. Therefore w.l.o.g., we assume $\beta_i = 0$. Now all that remains is to show $\gamma \geq \min\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\}$, for which we distinguish the following 2 cases.

Case $w_{t+1,1} > 0$: We will show $\gamma \geq \frac{\eta}{n}$ for this case. First note that

$$m \overset{\text{(C.5)}}{\leq} \sum_{i:1\leq i\leq n, w_{t+1,i}>0} (w_{t,i} - \eta\ell_{t,i} + \gamma) \overset{\gamma\geq 0}{\leq} \sum_{i:1\leq i\leq n, w_{t+1,i}>0} (w_{t,i} - \eta\ell_{t,i}) + n\gamma. \tag{C.7}$$

Now since we assume $w_{t+1,1} > 0$ and $\boldsymbol{\ell}_t = \boldsymbol{e}_1$, the first term on RHS of (C.7) is upper bounded by:

$$\sum_{i:1\leq i\leq n, w_{t+1,i}>0} (w_{t,i} - \eta\ell_{t,i}) \quad = \sum_{i:1\leq i\leq n, w_{t+1,i}>0} (w_{t,i}) - \eta \quad \leq \quad m - \eta.$$

Together, we get $m \leq n\gamma - \eta + m$, and this gives $\gamma \geq \frac{\eta}{n}$.

Case $w_{t+1,1} = 0$: We will show $\gamma \geq \frac{w_{t,1}}{n-1}$ for this case. Since $w_{t+1,1} = 0$, the summation $\sum_{i:1\leq i\leq n, w_{t+1,i}>0} (w_{t,i} - \eta\ell_{t,i} + \gamma)$ does not include the case $i = 1$, i.e.

$$\sum_{i:1\leq i\leq n, w_{t+1,i}>0} (\hat{w}_{t,i} - \eta\ell_{t,i} + \gamma) \quad = \sum_{i:2\leq i\leq n, w_{t+1,i}>0} (\hat{w}_{t,i} - \eta\ell_{t,i} + \gamma)\,.$$

Therefore, (C.7) can be tightened as follows:

$$m \overset{\text{(C.5)}}{\leq} \sum_{i:2\leq i\leq n, w_{t+1,i}>0} (w_{t,i} - \eta\ell_{t,i} + \gamma) \overset{\gamma\geq 0}{\leq} \sum_{i:1\leq i\leq n, w_{t+1,i}>0} (w_{t,i} - \eta\ell_{t,i}) + (n-1)\gamma.$$

Again, by the assumption $\boldsymbol{\ell}_t = \boldsymbol{e}_1$, we have

$$\sum_{i:2\leq i\leq n, w_{t+1,i}>0} (w_{t,i} - \eta\ell_{t,i}) \quad = \sum_{i:2\leq i\leq n, w_{t+1,i}>0} (w_{t,i}) \quad \leq \quad m - w_{t,1}.$$

Together, we get $m \leq (n-1)\gamma + m - w_{t,1}$, which gives $\gamma \geq \frac{w_{t,1}}{n-1}$ and completes the proof. ■

Our third lemma lower bounds the loss of the GD algorithm with respect to a particular adversarial loss sequence of n trials (instead of the above lower bounds for single trials). We argue this lower bound for the special case of the m-set problem when $m = 1$, i.e. the vanilla expert setting. As we shall see shortly in the main proof of Theorem 4.1, the lower bound of the general m-set problem degenerates into this special case for a certain loss sequence. Note that the assumptions of Lemma C.1 are always met when $m = 1$, because in this case any expert not incurring loss has weight at most $1 - w$, where w is the weight of the expert incurring loss.

Lemma C.3 *Consider the m-set problem with n experts, and $m = 1$. If at each trial, only the expert with the largest weight incurs a unit of loss, then after n consecutive such trials, the GD algorithm with learning rate $\eta > 0$ suffers loss at least $1 + \frac{1}{32}\min\{n\eta, 1\}$.*

Proof First notice that when $m = 1$, the largest of the n expert weights at each trial is at least $\frac{1}{n}$. Therefore, any algorithm suffers total loss at least 1 in n trials. To show the extra loss of $\frac{1}{32}\min\{n\eta, 1\}$, we claim that in at least $\frac{n}{4}$ of these n trials, the largest expert weight assigned by the GD algorithm is at least $\frac{1}{n} + \frac{1}{8}\min\{\eta, \frac{1}{n}\}$. This claim is proved as follows.

Let $\eta' = \min\{\eta, \frac{1}{n}\}$ and t_0 be the first trial that the largest expert weight of the trial is less than $\frac{1}{n} + \frac{1}{8}\eta'$. If $t_0 > \frac{n}{4}$, the claim holds trivially. Hence, we assume $t_0 \le \frac{n}{4}$. Now call any expert with weight at least $\frac{1}{n} - \frac{1}{8}\eta'$ at trial t_0 a *candidate*. We will show that the number of candidates s is at least $\frac{n}{2}$. To show this we first upper bound the expert weights at trial t_0 as follows:

$$\begin{aligned} \text{sum of non-candidates' weights} &\le (n-s)\left(\frac{1}{n} - \frac{1}{8}\eta'\right), \\ \text{sum of candidates' weights} &\le s\left(\frac{1}{n} + \frac{1}{8}\eta'\right). \end{aligned}$$

The first inequality follows from the fact that non-candidates have weight at most $\frac{1}{n} - \frac{1}{8}\eta'$ and the second inequality follows from the definition of t_0, i.e. the maximum weight at that trial is less than $\frac{1}{n} + \frac{1}{8}\eta'$. Now, since all the expert weights at a trial sum to 1, we have

$$1 \le s\left(\frac{1}{n} + \frac{1}{8}\eta'\right) + (n-s)\left(\frac{1}{n} - \frac{1}{8}\eta'\right) = 1 + \frac{s}{4}\eta' - \frac{n}{8}\eta',$$

which gives $s \ge \frac{n}{2}$ since $\eta' \ge \eta > 0$.

Next, we show that at trial $t_0 + \frac{n}{4}$, there will be a subset of at least $\frac{n}{4}$ candidates whose weight will be at least $\frac{1}{n} + \frac{1}{8}\eta'$. First recall that at each trial, only one expert incurs a unit of loss. Therefore, in the $\frac{n}{4}$ trials from t_0 to $t_0 + \frac{n}{4} - 1$, there will be at least $\frac{n}{2} - \frac{n}{4} = \frac{n}{4}$ candidates that do not incur any loss. By Lemma C.1, the weight of an expert not incurring loss is increased at each trial by $\min\{\frac{\eta}{n}, \frac{w}{n-1}\}$, where w is the weight of the expert incurring loss at that trial. Note that $w \ge \frac{1}{n}$ always hold since the expert incurring loss has the largest weight among the n experts. Therefore, at trial $t_0 + \frac{n}{4}$, each of the $\frac{n}{4}$ candidates that do not incur any loss from trial t_0 to trial $t_0 + \frac{n}{4} - 1$ has weight at least:

$$\underbrace{\frac{1}{n} - \frac{1}{8}\eta'}_{\substack{\text{lower bound on the} \\ \text{weight at trial } t_0}} + \underbrace{\frac{n}{4}\min\{\frac{\eta}{n}, \frac{w_t}{n-1}\}}_{\substack{\text{lower bound on the increase} \\ \text{from trial } t_0 \text{ to trial } t_0 + \frac{n}{4} - 1}} \overset{\frac{w_t}{n-1} \ge \frac{1}{n^2}}{\ge} \frac{1}{n} - \frac{1}{8}\eta' + \frac{n}{4}\min\{\frac{\eta}{n}, \frac{1}{n^2}\} = \frac{1}{n} + \frac{\eta'}{8}.$$

Finally, consider the next $\frac{n}{4}$ trials from $t_0 + \frac{n}{4}$ to $t_0 + \frac{n}{2} - 1$. (The game must have more than $t_0 + \frac{n}{2}$ trials, since we assume $t_0 \le \frac{n}{4}$.) The maximum weights at these trials are always at least $\frac{1}{n} + \frac{1}{8}\eta'$, because only one expert incurs loss at a time, and the weights of the remaining experts are never decreased. This completes the proof of the claim and the lemma. ■

Now we are ready to give the lower bound on the regret of the GD algorithm for the m-set problem. For the sake of readability, we repeat the statement of Theorem 4.1 below:

Theorem 4.1 *Consider the m-set problem with $k \le n/2$ and unit bit vectors as loss vectors. Then for any fixed learning rate η, the GD algorithm* (C.1) *can be forced to have regret at least* $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$.

Proof The lower bound $\Omega(k)$ directly follows from Lemma G.2 proven later: If we set the variable i in the statement of the lemma to k, then this results in a lower bound of $\Omega(m \log \frac{n}{m})$ for any algorithm. Now, $m \log \frac{n}{m} = m \log(\frac{k}{m} + 1) \ge k$. Hence to prove this theorem, we only need to show a lower bound of $\Omega(\{\min\{B_L, k\sqrt{B_L}\})$, where B_L is the loss budget (defined in (3.1)). Also, w.l.o.g., assume $B_L \ge 4k$ since when $B_L \le 4k$, the claimed bound is in fact $\Omega(k)$, which always holds as we just argued.

The hard part (deferred to later) in proving the $\Omega(\{\min\{B_L, k\sqrt{B_L}\})$ lower bound for GD is to show that the algorithm suffers regret at least $\Omega(k/\eta)$ and $\Omega(\min\{B_L, kB_L\eta\})$ on two different loss sequences, respectively. Clearly, it follows that the regret of GD is then at least the maximum of these two bounds. By a case analysis, one can show that $\max\{a, \min\{b, c\}\} \ge \min\{b, \max\{a, c\}\}$ for any $a, b, c \in \mathbb{R}$. (We prove this as Lemma I.1 in Appendix I.) Therefore we get the lower bound of $\Omega(\min\{B_L, \max\{k/\eta, kB_L\eta\}\})$. The lower bound for GD with any fixed learning rate now follows from fact that $\max\{k/\eta, kB_L\eta\}$ is minimized at $\eta = \Theta(1/\sqrt{B_L})$. The value of the lower bound with this choice of η is the target lower bound of $\Omega(k\sqrt{B_L})$.

We still need to describe the two loss sequences and prove the claimed lower bounds on the regret. The first loss sequence forces GD to suffer regret $\Omega(k/\eta)$. It consists of $\left\lfloor \frac{km}{n\eta} \right\rfloor + 1$ trials in which only the first k experts incur losses. More precisely, at each trial, the expert with the largest weight (within the first k experts) incurs one unit of loss (In the case of tied weights, only the expert with the smallest index incurs loss). The last m experts have loss 0. Therefore the regret is simply the total loss of the GD algorithm. The loss of the algorithm at each trial is equal to the largest weight of the first k experts. Therefore the loss is lower bounded by the average of the first k weights. With a uniform initial weight vector, this average is $\frac{m}{n}$ at the beginning of the first trial, and by Lemma C.2, it is decreased by at most $\frac{\eta}{k}$ after each of the following $\left\lfloor \frac{km}{n\eta} \right\rfloor + 1$ trials. Therefore, at the beginning of trial t, the average is at least $\frac{m}{n} - (t-1)\frac{\eta}{k}$. Summing up the arithmetic series from trial 1 to trial $\left\lfloor \frac{km}{n\eta} \right\rfloor + 1$ gives the following lower bound on the total loss of GD:

$$\frac{1}{2}\left(\left\lfloor \frac{km}{n\eta} \right\rfloor + 1\right)\left(\frac{m}{n} + \frac{m}{n} - \left(\left\lfloor \frac{km}{n\eta} \right\rfloor + 1 - 1\right)\frac{\eta}{k}\right) \overset{\frac{m}{n} \ge \frac{1}{2}}{\ge} \frac{1}{4}\left(\left\lfloor \frac{k}{2\eta} \right\rfloor + 1\right) = \Omega\left(\frac{k}{\eta}\right).$$

Now we describe the second loss sequence which forces the GD algorithm to suffer regret $\Omega(\min\{B_L, kB_L\eta\})$. For the sake of clarity, we assume that B_L is integer (otherwise replace B_L by $\lfloor B_L \rfloor$ in the proof). The sequence consists of $(k+1)B_L$ trials, where the expert with the largest weight among first $k+1$ experts incurs a unit of loss. The best comparator of this sequence consists of the last $m-1$ experts that have 0 total loss and the best of the first $k+1$ experts which has total loss at most B_L.

Next we lower bound the loss of GD with respect to this loss sequence. First observe, that the last $m-1$ experts do not incur any loss in the $(k+1)B_L$ trials. Therefore their weight may increase (from their initial value of $\frac{m}{n}$), but at any trial the weights of these

experts always have the same value. The value of this block of equal weights is always the maximum weight of any expert, since the weight value of the block is never decreased by the algorithm. More precisely, at each trial the block's value is increased as given in Lemma C.1, until it becomes 1 at trial t_{cap} and stay at 1 till the end of the game. If no such trial t_{cap} exists (i.e. the value of the block remains less than 1 at the end of the game), then let $t_{cap} = \infty$. In the degenerate case when $m = 1$ (i.e. the block has size $m - 1 = 0$), we simply set $t_{cap} = 1$ from the beginning.

Depending on the value of t_{cap}, we distinguish two cases in which GD suffers loss at least $B_L + \Omega(B_L)$ and $B_L + \Omega(\min\{B_L, kB_L\eta\})$, respectively.

Case $t_{cap} > (k+1)B_L/2$: We will show that GD suffers loss at least $B_L + \Omega(B_L)$ in this case. First recall that at the beginning of the proof we assumed $B_L \geq 4k$. Therefore in the case $t_{cap} > (k+1)B_L/2$ we have $t_{cap} > 4$. From our definition of t_{cap} this means that $m \geq 2$. Next we argue that since $t_{cap} > (k+1)B_L/2$, we have $\eta \leq \frac{1}{k+1}$. Let W_t denote the sum of the first $k+1$ weights at trial t and let w_t be their maximum. By Lemma C.1, we know that in each trial prior to t_{cap}, the weight w_t of the expert incurring loss is decreased by $\min\{\frac{(n-1)\eta}{n}, w_t\}$ and all other weights are increased by $\min\{\frac{\eta}{n}, \frac{w_t}{n-1}\}$. Since the expert incurring loss is always one of the first $k+1$ experts, we have that in each trial prior to t_{cap}, the total weight W_t is decreased by at least

$$\min\left\{\frac{(n-1)\eta}{n}, w_t\right\} - k\min\left\{\frac{\eta}{n}, \frac{w_t}{n-1}\right\} \;\geq\; \frac{m-1}{n}\min\{\eta, w_t\} \;\geq\; \frac{m-1}{n}\min\left\{\eta, \frac{1}{k+1}\right\}.$$

The second inequality follows from the fact that since w_t is the largest of the first $k+1$ expert weights, it must be at least $\frac{1}{k+1}$. Together with the fact that $W_1 = \frac{(k+1)m}{n}$, we have

$$W_{(k+1)B_L/2} \leq \frac{(k+1)m}{n} - \frac{(k+1)B_L}{2}\frac{m-1}{n}\min\left\{\eta, \frac{1}{k+1}\right\}. \tag{C.8}$$

Now if $\eta \geq \frac{1}{k+1}$, the upper bound (C.8) becomes $\frac{(k+1)m}{n} - \frac{(m-1)B_L}{2n}$, which can be further upper bounded by $\frac{m}{n}$ using the fact $m \geq 2$ and the assumption $B_L \geq 4k$. However, the upper bound of $W_{(k+1)B_L/2} \leq \frac{m}{n}$ is less than 1 and all W_t are at least 1 since $m - W_t$ is the total weight of the last $m-1$ experts which is at most $m-1$. Therefore we have $\eta < \frac{1}{k+1}$ in this case.

Now we lower bound the loss of GD by lower bounding the average weight $W_t/(k+1)$. We have $\eta < \frac{1}{k+1}$ and $t_{cap} > (k+1)B_L/2$. Also by Lemma C.1, W_t decreases by exactly $\frac{(m-1)\eta}{n}$ at each trial for $1 \leq t \leq (k+1)B_L/2$. Therefore the total average weight in trials 1 through $(k+1)B_L/2$ is at least

$$\frac{1}{2}\frac{1}{k+1}\left(\frac{(k+1)m}{n} + 1\right)\frac{(k+1)B_L}{2} = \left(\frac{(k+1)m}{n} + 1\right)\frac{B_L}{4}. \tag{C.9}$$

Now with $m \geq 2$, $k \geq 1$ and $n = m + k$, it is easy to verify that $\frac{(k+1)m}{n}$ is at least $1 + \Omega(1)$, which along with (C.9) results in a $\frac{B_L}{2} + \Omega(B_L)$ lower bound on the loss of GD for $1 \leq t \leq (k+1)B_L/2$. In trials $(k+1)B_L/2 < t \leq (k+1)B_L$, GD suffers loss at least $\frac{(k+1)B_L}{2}\frac{1}{k+1} = \frac{B_L}{2}$ since the weight of the expert incurring loss is at least $\frac{1}{k+1}$. Thus in trial

1 through $(k+1)B_L$ the loss of GD is at least $B_L + \Omega(B_L)$ which concludes the proof of the case $t_{cap} \geq (k+1)B_L/2$.

Case $t_{cap} \leq (k+1)B_L/2$: We will show that GD suffers total loss at least $B_L + \Omega(\min\{B_L, kB_L\eta\})$ in this case. First note that GD suffers loss at least $B_L/2$ in the first $(k+1)B_L/2$ trials. This follows from the fact that in each trial, the expert with the largest weight among first $k+1$ experts incurs a unit of loss. Since the sum of all weights is equal to m, and none of the remaining $m-1$ weights can exceed 1, the sum of weights of the first $k+1$ experts must be at least 1, and hence the largest weight among the first $k+1$ experts is at least $\frac{1}{k+1}$. This means that in a sequence of $(k+1)B_L/2$ trials, the loss of the GD algorithm is at least $B_L/2$.

Thus, it suffices to show that GD suffers loss at least $B_L/2 + \Omega(\min\{B_L, kB_L\eta\})$ in trials $(k+1)B_L/2+1$ through $(k+1)B_L$. First note that since $t_{cap} \leq (k+1)B_L/2$, in each of these trials the weights of the $m-1$ loss free experts have reached the cap 1. This means that GD updates the weights of the first $k+1$ experts as in the vanilla expert setting (i.e. $m=1$). Therefore by Lemma C.3, the loss of GD in the second $(k+1)B_L/2$ trials is at least $\frac{B_L}{2}(1 + \frac{1}{32}\min\{(k+1)\eta, 1\}) = \frac{B_L}{2} + \Omega(\min\{B_L, kB_L\eta\})$.

We conclude that for the second loss sequence, the loss of the best comparator is at most B_L and the loss of GD is at least $B_L + \Omega(\min\{B_L, kB_L\eta\})$. Therefore, the regret of GD is at least $\Omega(\min\{B_L, kB_L\eta\})$ for the second loss sequence and this completes our proof of the theorem. ■

Appendix D. Proof of Theorem 4.3

Theorem 4.3 gives a lower bound on the regret of the FRL-GD algorithm for the m-set problem with unit bit vectors as loss vectors. In this case, the FRL-GD algorithm (4.2) specializes to the following:

$$\begin{aligned} \text{Follow the regularized leader:} \quad & \hat{\boldsymbol{w}}_{t+1} = -\eta \sum_{q=1}^{t} \boldsymbol{\ell}_q, \\ \text{Projection step:} \quad & \boldsymbol{w}_{t+1} = \operatorname*{argmin}_{\boldsymbol{w} \in \boldsymbol{\mathcal{S}}_m} \|\boldsymbol{w} - \hat{\boldsymbol{w}}_{t+1}\|^2. \end{aligned} \tag{D.1}$$

The proof has the same structure as the lower bound for the GD algorithm (Appendix C). Again we use two adversarial loss sequences (one for low and high learning rates) and give three technical lemmas that reason with the KKT conditions. The details are different because the intermediate weight vector $\hat{\boldsymbol{w}}_{t+1}$ has a different form than for vanilla GD. The KKT conditions are the same as the KKT condition for GD (C.1) except for a slight change in the stationarity condition. For $i = 1, \ldots, n$,

$$\begin{aligned} &\text{Stationarity:} && w_{t+1,i} = -\eta \ell_{\leq t,i} + \gamma + \alpha_i - \beta_i, \\ &\text{Complementary slackness:} && w_{t+1,i}\alpha_i = 0, \quad (w_{t+1,i} - 1)\beta_i = 0, \\ &\text{Primal feasibility:} && \textstyle\sum_{i=1}^{n} w_{t+1,i} = m, \qquad 0 \leq w_{t+1,i} \leq 1, \\ &\text{Dual feasibility:} && \alpha_i \geq 0, \qquad \beta_i \geq 0, \end{aligned} \tag{D.2}$$

where $\ell_{\leq t,i} = \sum_{q=1}^{t} \ell_{q,i}$ is the cumulative loss of expert i up to trial t. Again we prove three technical lemmas before assembling them into the main proof.

Lemma D.1 *Consider the m-set problem with n experts, where at the beginning of trial $t+1$, each of the first $k+1$ experts (where $k = n-m$) have incurred the same cumulative loss ℓ, and all the remaining experts are loss free, i.e.*

$$\ell_{\leq t,i} = \begin{cases} \ell & \text{for } i \leq k+1 \\ 0 & \text{for } i > k+1 \end{cases}.$$

Now the FRL-GD algorithm predicts at trial $t+1$ with the weight vector $\boldsymbol{w}_{t+1}$ given by:

$$w_{t+1,i} = \begin{cases} \text{if } \eta\ell < \frac{k}{k+1} \text{ then } \begin{cases} \frac{m-\eta\ell(m-1)}{n} & \text{for } i \leq k+1 \\ \frac{m+\eta\ell(k+1)}{n} & \text{for } i > k+1 \end{cases} \\ \text{if } \eta\ell \geq \frac{k}{k+1} \text{ then } \begin{cases} \frac{1}{k+1} & \text{for } i \leq k+1 \\ 1 & \text{for } i > k+1 \end{cases} \end{cases}.$$

Proof We prove this lemma by verifying the KKT conditions (D.2). If $\eta\ell < \frac{k}{k+1}$, we have:

$$1 > \frac{m-\eta\ell(m-1)}{n} > 0, \quad \text{and} \quad 0 < \frac{m+\eta\ell(k+1)}{n} < 1.$$

Therefore $0 < w_{t+1,i} < 1$, for all i. By taking $\boldsymbol{\alpha} = \boldsymbol{\beta} = \mathbf{0}$, and $\gamma = \frac{m+\eta\ell(k+1)}{n}$, the KKT conditions can easily be verified to hold. If $\eta\ell \geq \frac{k}{k+1}$, the KKT conditions are satisfied by taking $\alpha_i = 0$ for $i \leq k+1$ and $\alpha_i = \frac{k}{k+1} - \eta\ell$ for $i > k+1$, $\boldsymbol{\beta} = \mathbf{0}$ and $\gamma = \frac{1}{k+1} + \eta\ell$. ■

Lemma D.2 *Consider a trial of the m-set problem with n experts, when only one expert incurs a unit of loss. Then the FRL-GD algorithm with learning rate $\eta > 0$ decreases the weight of this expert by at most η and none of the other weights are decreased in this trial.*

Proof Let $\boldsymbol{\ell}_{\leq t-1}$ be the cumulative loss vector at the beginning of the trial, and let $\boldsymbol{w}_t, \boldsymbol{\alpha}_t, \boldsymbol{\beta}_t$ and γ_t be the corresponding primal and dual variables satisfying KKT conditions (D.2) with respect to $\boldsymbol{\ell}_{\leq t-1}$. W.l.o.g., we assume the first expert incurs a unit of loss, i.e. $\boldsymbol{\ell}_{\leq t} = \boldsymbol{\ell}_{\leq t-1} + \boldsymbol{e}_1$. Let $\boldsymbol{w}_{t+1}, \boldsymbol{\alpha}_{t+1}, \boldsymbol{\beta}_{t+1}$ and γ_{t+1} denote the variables satisfying the KKT conditions with respect to the updated loss vector $\boldsymbol{\ell}_{\leq t}$. The lemma now states that $w_{t+1,1} - w_{t,1} \geq -\eta$.

The lemma holds trivially when $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t$. When $\boldsymbol{w}_{t+1} \neq \boldsymbol{w}_t$, we first show that $\gamma_{t+1} \geq \gamma_t$. Since both $\boldsymbol{w}_t$ and $\boldsymbol{w}_{t+1}$ sum to m, there must be an expert j, such that $w_{t,j} < w_{t+1,j}$. By the stationarity condition of (D.2), we have:

$$0 < w_{t+1,j} - w_{t,j} = (-\eta\ell_{\leq t,j} + \alpha_{t+1,j} - \beta_{t+1,j} + \gamma_{t+1}) - (-\eta\ell_{\leq t-1,j} + \alpha_{t,j} - \beta_{t,j} + \gamma_t),$$

or, equivalently,

$$\gamma_{t+1} - \gamma_t > \eta(\ell_{\leq t,j} - \ell_{\leq t-1,j}) + (\alpha_{t,j} - \alpha_{t+1,j}) + (\beta_{t+1,j} - \beta_{t,j}). \tag{D.3}$$

Since $w_{t+1,j} > w_{t,j}$, and the weights must be non-negative, we have $w_{t+1,j} > 0$, and thus $\alpha_{t+1,j} = 0$ due to the complementary slackness condition of (D.2). Since $\alpha_{t,j}$ must be

non-negative due to the dual feasibility condition of (D.2), we have $\alpha_{t,j} \geq \alpha_{t+1,j}$. A similar argument gives $\beta_{t+1,j} \geq \beta_{t,j}$. Moreover, since $\ell_{\leq t,j} - \ell_{\leq t-1,j} \geq 0$ (due to $\boldsymbol{\ell}_{\leq t} = \boldsymbol{\ell}_{\leq t-1} + \boldsymbol{e}_1$), the RHS of (D.3) is non-negative, and thus $\gamma_{t+1} \geq \gamma_t$.

By the stationary condition of (D.2), we have:

$$\begin{aligned} w_{t+1,1} - w_{t,1} &= (-\eta\ell_{\leq t,1} + \alpha_{t+1,1} - \beta_{t+1,1} + \gamma_{t+1}) \ - \ (-\eta\ell_{\leq t-1,1} + \alpha_{t,1} - \beta_{t,1} + \gamma_t) \\ &= -\eta \ + \ (\gamma_{t+1} - \gamma_t) \ + \ (\alpha_{t+1,1} - \alpha_{t,1}) \ + \ (\beta_{t,1} - \beta_{t+1,1}), \end{aligned} \tag{D.4}$$

where we used $\ell_{\leq t,1} = \ell_{\leq t-1,1} + 1$. If $\alpha_{t,1} \neq 0$, then $w_{t,1} = 0$ due to complementary slackness, and the lemma trivially holds. Similarly if $\beta_{t+1,1} \neq 0$, then $w_{t+1,1} = 1$, and again the lemma holds trivially. Thus, we may assume that $\alpha_{t,1} = \beta_{t+1,1} = 0$. However then (D.4) becomes

$$w_{t+1,1} - w_{t,1} \quad = \quad -\eta \ + \ (\gamma_{t+1} - \gamma_t) \ + \ \alpha_{t+1,1} \ + \ \beta_{t,1} \quad \geq \quad -\eta.$$

We now show the second statement of the lemma, that $w_{t+1,i} \geq w_{t,i}$ for all $i > 1$. First note that if $\alpha_{t,i} > 0$, then by the complementary slackness condition of (D.2), $w_{t,i} = 0$, and the statement trivially holds. Similarly, if $\beta_{t+1,i} > 0$, then by the complementary slackness condition, $w_{t+1,i} = 1$, and, again the statement trivially holds. Therefore we prove the statement assuming that $\alpha_{t,i} = 0$ and $\beta_{t+1,i} = 0$. Since $\ell_{\leq t,i} = \ell_{\leq t-1,i}$, the complementary slackness condition of (D.2) implies:

$$\begin{aligned} w_{t+1,i} - w_{t,i} &= (\cancel{-\eta\ell_{\leq t,i}} + \alpha_{t+1,i} - \beta_{t+1,i} + \gamma_{t+1}) \ - \ (\cancel{-\eta\ell_{\leq t-1,i}} + \alpha_{t,i} - \beta_{t,i} + \gamma_t) \\ &= (\alpha_{t+1,i} - \underbrace{\alpha_{t,i}}_{=0}) \ + \ (\beta_{t,i} - \underbrace{\beta_{t+1,i}}_{=0}) \ + \ (\underbrace{\gamma_{t+1} - \gamma_t}_{\geq 0}) \\ &\geq \ \alpha_{t+1,i} + \beta_{t,i} \ \geq \ 0, \end{aligned}$$

where the last inequality is by the dual feasibility condition of (D.2). This finishes the proof. ■

Lemma D.3 *Consider the m-set problem with n experts, and $m = 1$. Assume at the end of trial t, the cumulative losses of all experts are the same. Assume further that the loss sequence in trials $t+1, \ldots, n$ is $\boldsymbol{\ell}_{t+1} = \boldsymbol{e}_1, \boldsymbol{\ell}_{t+2} = \boldsymbol{e}_2, \ldots, \boldsymbol{\ell}_{t+n} = \boldsymbol{e}_n$, i.e. each expert subsequently incurs a unit of loss. Then the cumulative loss incurred by the FRL-GD algorithm in iterations $t+1, \ldots, n$ is at least $1 + \frac{1}{4}\min\{n\eta, 1\}$.*

Proof The proof goes by providing primal and dual variables satisfying the KKT conditions (D.2). Since the solution $\boldsymbol{w}_{t+1}$ to (D.2) does not change if we shift all cumulative losses $\ell_{\leq t,i}$ by a constant we can assume w.l.o.g. that the cumulative loss of all experts at the end of trial t is 0.

Take trial $t + j + 1$ ($j \geq 0$), at the beginning of which each of the first j experts have already incurred a unit of loss and the remaining $n - j$ experts are loss free. If $\eta \leq \frac{1}{n-j}$, then the KKT conditions (D.2) are satisfied by taking $\alpha_i = \beta_i = 0$ for all $i = 1, \ldots, n$, $\gamma = \frac{j}{n}\eta + \frac{1}{n}$, and

$$w_{t+j+1,i} = \begin{cases} \frac{1}{n} - \frac{n-j}{n}\eta & \text{for } i \leq j \\ \frac{1}{n} + \frac{j}{n}\eta & \text{for } i > j \end{cases}.$$

In this trial, expert $j+1$ incurs a unit of loss, and hence the algorithm's loss is $\frac{1}{n}+\frac{j}{n}\eta$.

If $\eta > \frac{1}{n-j}$, then the KKT conditions (D.2) are satisfied by taking $\gamma = \frac{1}{n-j}$ and for $1 \le i \le n$, $\beta_i = 0$,

$$w_{t+j+1,i} = \begin{cases} 0 & \text{for } i \le j \\ \frac{1}{n-j} & \text{for } i > j \end{cases}, \qquad \alpha_i = \begin{cases} \eta - \frac{1}{n-j} & \text{for } i \le j \\ 0 & \text{for } i > j \end{cases}.$$

The loss of the algorithm in such a case is $\frac{1}{n-j}$.

Thus depending on η, the algorithm's loss at trial $t+j+1$ is equal to

$$\begin{cases} \frac{1}{n}+\frac{j}{n}\eta & \text{if } \eta \le \frac{1}{n-j} \\ \frac{1}{n-j} = \frac{1}{n}+\frac{j}{n}\frac{1}{n-k} & \text{if } \eta > \frac{1}{n-j} \end{cases},$$

which can be concisely written as: $\frac{1}{n}+\frac{j}{n}\min\left\{\eta, \frac{1}{n-j}\right\}$. Summing the above over $j = 0, \ldots, n$ gives the cumulative loss of the algorithm incurred at trials $t+1, \ldots, t+n$:

$$\begin{aligned} \sum_{j=0}^{n-1} \frac{1}{n} + \frac{j}{n}\min\left\{\eta, \frac{1}{n-j}\right\} &\ge \sum_{j=0}^{n-1} \frac{1}{n} + \frac{j}{n}\min\left\{\eta, \frac{1}{n}\right\} \\ &= 1 + \frac{n-1}{2}\min\left\{\eta, \frac{1}{n}\right\} \\ &\ge 1 + \frac{1}{4}\min\{\eta n, 1\}, \end{aligned}$$

where the last inequality is due to $n-1 > \frac{n}{2}$ for $n \ge 2$. ■

We are now ready to give the proof of Theorem 4.3:

Theorem 4.3 *Consider the m-set problem with $k \le n/2$ and unit bit vectors as loss vectors. Then for any fixed learning rate η, the FRL-GD algorithm* (D.1) *can be forced to have regret at least $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$.*

Proof Theorem 5.6 gives a lower bound of $\Omega(\sqrt{B_L m \log\frac{n}{m}} + m\log\frac{n}{m})$ that holds for any algorithm. This lower bound is at least $\Omega(k)$ since $m\log\frac{n}{m} = m\log(\frac{k}{m}+1) \ge k$. Hence to prove this theorem, we only need to show a lower bound of $\Omega(\{\min\{B_L, k\sqrt{B_L}\})$. Similarly as in the proof of Theorem 4.1, we show this in two steps: First, we give two loss sequences that force FRL-GD to have regret at least $\Omega(k/\eta)$ and $\Omega(\min\{B_L, kB_L\eta\})$, respectively. Then, the lower bound follows by taking the maximum between the two lower bounds.

The first loss sequence is exactly the same as in the proof of Theorem 4.1, i.e. the sequence consists of $\left\lfloor \frac{km}{n\eta} \right\rfloor + 1$ trials and in each trial, the expert with the largest weight (within the first k experts) incurs one unit of loss. With Lemma D.2 in place of Lemma C.2, one can easily show an $\Omega(k/\eta)$ regret lower bound for FRL-GD by repeating the argument from the proof of Theorem 4.1.

Now we describe the second loss sequence which forces the FRL-GD algorithm to suffer regret $\Omega(\min\{(B_L), kB_L\eta\})$. For the sake of clarity, we assume that B_L is integer (otherwise

replace B_L by $\lfloor B_L \rfloor$ in the proof). The sequence consists of B_L "rounds", and each round consist of $k+1$ trials (so that there are $(k+1)B_L$ trials in total). In each round, one unit of loss is given alternately to each of the first $k+1$ experts, one at a time. In other words, in trial t, the loss vector $\boldsymbol{\ell}_t$ equals to $\boldsymbol{e}_r$ where $r = t \mod (k+1)$. The best comparator of this sequence consists of the last $m-1$ loss free experts and any of the first $k+1$ experts, which incurs cumulative loss B_L.

To lower bound the loss of the algorithm, first notice that in each round, each of the first $k+1$ experts incurs exactly one unit of loss. The sum of weights of these experts at the beginning of a round lower bounds the algorithm's loss in this round. This is because the weight of a given expert cannot decrease if the expert does not incur any loss (Lemma D.2); hence, the weight of a given expert at a trial, in which that expert receives a unit of loss, will be at least as large as the weight of that expert at the beginning of a round. Since the weights are initialized uniformly, this sum is $m(k+1)/n$ before round 1, and by Lemma D.1, each of the following rounds decreases it by $(m-1)(k+1)\eta/n$ until it is lower capped at 1 (Since the total sum of the weights is m, and none of the remaining $m-1$ weights can exceed 1, the sum of weights of the first $k+1$ experts must be at least 1).

We first assume that after $B_L/2$ rounds, this sum is strictly larger than 1 which means the sum decreases as an arithmetic series for all the first $B_L/2$ rounds and the algorithm's loss can be lower bounded by

$$\frac{1}{2}(m(k+1)/n+1)\frac{B_L}{2} \overset{\text{Use the same argument as in (C.9)}}{=} B_L/2 + \Omega(B_L).$$

Since the sum of the first $k+1$ weights at the beginning of any trial is at least 1, the algorithm incurs loss at least $B_L/2$ in the remaining $B_L/2$ rounds. Summing up the algorithm's loss on both halves of the sequence, we get a regret lower bound of $\Omega(B_L)$.

Now consider the case, when after the first $B_L/2$ rounds, the sum of the first $k+1$ weights is 1. This implies that the weights of $m-1$ remaining experts are all equal to 1, and will stay at this value, since only the first $k+1$ experts incur any loss (and, by Lemma D.2, the weight of an expert cannot decrease if that expert does not incur any loss). Thus, we can disregard the loss free $m-1$ experts, and in the remaining $B_L/2$ rounds, the first $k+1$ expert weights are updated as in the m-set problem with $m=1$. Notice that the algorithm suffers loss at least $B_L/2$ in the first $B_L/2$ rounds and by Lemma D.3, suffers loss at least $B_L/2 + B_L \min\{(k+1)\eta, 1\}/8$ in the second $B_L/2$ rounds. Summing up the algorithm's loss on both halves of the sequence, we get a regret lower bound of $\Omega(\min\{B_L, kB_l\eta\})$. ■

Appendix E. A Discussion on the Final Parameter of FRL-GD

In this appendix, we show that the final parameter matrix of the FRL-GD algorithm essentially contains the solution to the batch PCA problem. First recall that given n dimensional data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$, the batch version of the k-PCA problem is solved by the eigenvectors of the k largest eigenvalues of the covariance matrix $\boldsymbol{C} = \sum_{t=1}^{T} \boldsymbol{x}_t \boldsymbol{x}_t^\top$. Let $\boldsymbol{W}_{T+1}$ be the final parameter matrix of the FRL-GD algorithm when the instance matrices are $\boldsymbol{X}_1 = \boldsymbol{x}_1\boldsymbol{x}_1^\top, \ldots, \boldsymbol{X}_T = \boldsymbol{x}_T\boldsymbol{x}_T^\top$. We will show that the eigenvectors of the $m = n-k$ largest eigenvalues of $\boldsymbol{W}_{T+1}$ are the same as the eigenvectors of the m largest eigenvalues of the

negated covariance matrix $-\boldsymbol{C}$. Thus, by computing the complementary subspace of rank k, one finds the solution of the batch PCA problem with respect to data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$.

Recall that the final parameter $\boldsymbol{W}_{T+1}$ of FRL-GD is the projection of the $-\boldsymbol{C}$ into the parameter set $\boldsymbol{\mathcal{W}}_m$:

$$\boldsymbol{W}_{T+1} = \operatorname*{argmin}_{\boldsymbol{W} \in \boldsymbol{\mathcal{W}}_m} \| - \boldsymbol{C} - \boldsymbol{W} \|_F^2.$$

Let $-\boldsymbol{C}$ have eigendecomposition $-\boldsymbol{C} = \boldsymbol{U} diag(\boldsymbol{\lambda}) \boldsymbol{U}^\top$, where $\boldsymbol{\lambda}$ is the vector of the eigenvalues of $-\boldsymbol{C}$. Arora et al. (2013, Lemma 3.2) shows that the projection of $-\boldsymbol{C}$ is solved by projecting the eigenvalues $\boldsymbol{\lambda}$ into $\boldsymbol{\mathcal{S}}_m$ while keeping its eigensystem unchanged:

$$\boldsymbol{W}_{T+1} = \boldsymbol{U} diag(\boldsymbol{\lambda}') \boldsymbol{U}^\top \qquad \text{and} \qquad \boldsymbol{\lambda}' = \operatorname*{argmin}_{\boldsymbol{v} \in \boldsymbol{\mathcal{S}}_m} \|\boldsymbol{\lambda} - \boldsymbol{v}\|_2^2.$$

W.l.o.g., assume the elements of $\boldsymbol{\lambda}$ are in descending order, i.e. $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n$. To prove that the eigenvectors of the m largest eigenvalues are the same in $\boldsymbol{W}_{T+1}$ and $-\boldsymbol{C}$, we only need to show the following: for any integers pair i and j such that $1 \leq i \leq m < j \leq n$, if $\lambda_i > \lambda_j$, then $\lambda'_i > \lambda'_j$. First note that by the KKT analysis of the problem of projecting into $\boldsymbol{\mathcal{S}}_m$ (see (D.2)), it is easy to see that if $\lambda_i > \lambda_j$, then exactly one of the following three cases holds.

$$\lambda'_i > \lambda'_j \qquad \text{or} \qquad \lambda'_i = \lambda'_j = 0 \qquad \text{or} \qquad \lambda'_i = \lambda'_j = 1.$$

Now we show that when i and j further satisfy $i \leq m < j$, the latter two cases can never happen. Suppose $\lambda'_i = \lambda'_j = 1$ for some $i \leq m < j$. In this case for any $i' \leq m$, $\lambda'_{i'} = \lambda'_j = 1$ also holds. Therefore, the sum of all the coordinates of $\boldsymbol{\lambda}'$ will be at least $m+1$ which contradicts $\boldsymbol{\lambda}' \in \boldsymbol{\mathcal{S}}_m$. Now assume $\lambda'_i = \lambda'_j = 0$ for some $i \leq m < j$. In this case for any $m < j'$, $\lambda'_i = \lambda'_{j'} = 0$ also holds. This implies that the sum of all the coordinates of $\boldsymbol{\lambda}'$s will be at most $m-1$ which again contradicts $\boldsymbol{\lambda}' \in \boldsymbol{\mathcal{S}}_m$.

Appendix F. Regret Lower Bounds When the Number of Trials Is Large

This appendix proves lower bounds on the regret of any online algorithm for the m-set problem: Theorem 5.1 and Theorem 5.2 prove lower bounds for unit bit vectors as loss vectors and Theorem 5.4 proves lower bounds for arbitrary bit vectors as loss vectors. In all of these lower bounds, we assume that the number of the trials T is larger than either the number of experts n or some function of n, m and k (see details of the assumptions in individual theorems). The regret lower bounds for small number of trials are given in the next Appendix G.

All the lower bounds given in this appendix are proved with the probabilistic bounding technique described in Section 5, i.e. in each case, we first choose a probability distribution $\mathcal{P}$ and then show a lower bound on the expected regret of any algorithm when the loss vectors are generated i.i.d. from $\mathcal{P}$. Our lower bounds on the expected regret make use of the following lemma which gives an upper bound on the expected loss of the best comparator in a two expert game.

Lemma F.1 *Consider a two expert game in which the random loss pairs of both experts are i.i.d. between trials, and at each trial the random pair follows the distribution:*

value of the loss pair	$(0,1)$	$(1,0)$	$(1,1)$	$(0,0)$
probability	p	p	q	$1-2p-q$

(F.1)

where non-negative parameters p and q satisfy $2p+q \le 1$. Let M be the minimum total loss of the two experts in T such trials. If T and p satisfy $Tp \ge 1/2$, then

$$\mathbb{E}[\,M\,] \le\ T(p+q) - c\sqrt{Tp}$$

for some constant $c > 0$ independent of T, p and q.

Later we will use the case $q = 0$ of the two expert distribution (F.1) as a submodule for building distributions over the n unit bit vectors and $p = q = 1/4$ for building distributions over $\{0,1\}^n$. To prove Lemma F.1, we need the following lemma from (Koolen, 2011, Theorem 2.5.3):

Lemma F.2 *Let a_t and b_t be two binary random variables following the distribution*

value of (a_t, b_t)	$(0,1)$	$(1,0)$
probability	0.5	0.5

For T independent such pairs, we have

$$\frac{T}{2} - \sqrt{\frac{T-1}{2\pi}} \quad\le\quad \mathbb{E}\left[\min\left\{\sum_{t=1}^{T} a_t, \sum_{t=1}^{T} b_t\right\}\right] \quad\le\quad \frac{T}{2} - \sqrt{\frac{T+1}{2\pi}}.$$

Proof of Lemma F.1 Denote the experts' losses at trials $1 \le t \le T$ by $\tilde{a}_t$ and $\tilde{b}_t$. In this notation, the statement of Lemma F.1 is equivalent to:

$$\mathbb{E}\left[\min\left\{\sum_t \tilde{a}_t, \sum_t \tilde{b}_t\right\}\right] \le T(p+q) - c\sqrt{Tp}.$$

At each trial, the random variable pair $(\tilde{a}_t, \tilde{b}_t)$ has four possible values: $(1,0)$, $(0,1)$, $(1,1)$ or $(0,0)$. If $\tilde{a}_t \ne \tilde{b}_t$, then this trial is "covered by" Lemma F.2. If $\tilde{a}_t = \tilde{b}_t$, then this trial affects $\sum_t \tilde{a}_t$ and $\sum_t \tilde{b}_t$ the same way and therefore can be excluded from the minimization. We formalize this observation as follows:

$$\begin{aligned}\mathbb{E}\left[\min\left\{\sum_t \tilde{a}_t, \sum_t \tilde{b}_t\right\}\right] \quad &= \quad \mathbb{E}\left[\min\left\{\sum_{t:\tilde{a}_t\ne\tilde{b}_t} \tilde{a}_t, \sum_{t:\tilde{a}_t\ne\tilde{b}_t} \tilde{b}_t\right\}\right] + \mathbb{E}\left[\sum_{t:\tilde{a}=\tilde{b}} \tilde{a}_t\right] \\ &\overset{\text{Lemma F.2}}{\le} \quad \mathbb{E}\left[\frac{R}{2} - \sqrt{\frac{R-1}{2\pi}}\right] + Tq,\end{aligned}$$

where R is a binomial random variable with T draws and success probability $2p$. Clearly $\mathbb{E}[R] = 2Tp$ and therefore $\mathbb{E}[\frac{R}{2}] = Tp$. Moreover under the assumption that $Tp \ge 1/2$, we will show in Lemma I.2 of Appendix I (using an application of the Chernoff bound) that $\mathbb{E}\left[\ \sqrt{\frac{R-1}{2\pi}}\ \right] \ge c\sqrt{Tp}$ for some constant c that does not depend on T, p and q. ■

We now use Lemma F.1 to prove the following theorem which addresses the m-set problem with unit bit vectors for the case $k \le \frac{n}{2}$.

Theorem 5.1 *Consider the m-set problem with unit bit vectors as loss vectors, where $m = n - k$. Then for $k \leq \frac{n}{2}$ and $T \geq k$, any online algorithm suffers worst case regret at least $\Omega(\sqrt{Tk})$.*

Proof In this proof, each loss vector is uniformly sampled from the first $2k$ unit bit vectors, i.e. at each trial, one of the first $2k$ experts is uniformly chosen to incur a unit of loss. To show an upper bound on the loss of the comparator, we group these $2k$ experts into k pairs and note that the loss of each expert pair follows the joint distribution described Lemma F.1 with $p = \frac{1}{2k}$ and $q = 0$. Furthermore, the condition $Tp \geq 1/2$ of Lemma F.1 is also satisfied because of the assumption $T \geq k$. Hence, by applying Lemma F.1 we know that the expected loss of the winner in each pair is at most $T/2k - c\sqrt{T/2k}$, and the total expected loss for k winners from all k pairs is upper bounded by $T/2 - c\sqrt{kT/2}$. Now recalling that the comparator consists of the $m = n - k$ best experts, its total expected loss is upper bounded by the expected loss of the k winners, which is at most $T/2 - c\sqrt{kT/2}$, plus the expected loss of the remaining $n - 2k$ experts, which is zero. Therefore, we have an upper bound of $T/2 - c\sqrt{kT/2}$ on the expected loss of the comparator. On the other hand, since losses are generated independently between trials, any online algorithm suffers loss at least $T/2$. The difference between the lower bound on the expected loss of the algorithm and the upper bound on the expected loss of the best m-set gives the regret lower bound of the theorem. ■

The case $k \geq \frac{n}{2}$ is more complicated. Recall that $k = n - 1$ reproduces the vanilla single expert case. Therefore additional $\log n$ factor must appear in the square root of the lower bound. We need the following lemma, which is a generalization of Lemma F.1 to n experts. In the proof, we upper bound the minimum loss of the experts by the loss of the winner of a tournament among the n experts. The tournament winner does not necessarily have the lowest loss. However as we shall see, its expected loss is close enough to the expected loss of the best expert so that this bounding technique is still useful for obtaining lower bounds on the regret.

Lemma F.3 *Choose any n, S and T, such that $n = 2^S$ and S divides T. If the loss sequence of length T is generated from a distribution $\mathcal{P}$, such that:*

- *at each trial t, the distribution of losses on n experts is exchangeable, i.e. for any permutation π on a set $\{1, \ldots, n\}$, and for any t, $\boldsymbol{\ell}_t = (\ell_{t,1}, \ell_{t,2}, \ldots, \ell_{t,n})$ and $\boldsymbol{\ell}_t^\pi = (\ell_{t,\pi(1)}, \ell_{t,\pi(2)}, \ldots, \ell_{t,\pi(n)})$ have the same distribution,*
- *and the distribution of losses is i.i.d. between trials,*

then,

$$\mathbb{E}\left[\text{ minimum loss among } n \text{ experts in } T \text{ trials }\right] \leq S\,\mathbb{E}\left[\text{ minimum loss among experts 1 and 2 in } T/S \text{ trials }\right].$$

Proof The key idea is to upper bound the minimum loss of any expert by the loss of the expert that wins an S round tournament. In the first round, we start with n experts and pair each expert with a random partner. The round lasts for T/S trials. For each pair, the expert with the smaller loss wins in this round (tie always broken randomly). The

	first round			second round		
	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$
[expert 1	1	0	1	0	0	0
[expert 4	**0**	**1**	**0**	1	1	1]
[expert 2	**0**	**0**	**1**	**1**	**1**	**0**]
[expert 3	1	1	0	1	0	1

Table F.1: A tournament with $T = 6$ trials, $S = 2$ rounds, and $n = 4$ experts. The bits in the table are the binary losses of the experts in each trial. The brackets show the pairing in each round. The losses of the winners are in bold

$n/2$ winners continue to the second round. At round s, the remaining $n/2^{s-1}$ experts are randomly paired and the winners are determined based on the losses in another set of T/S trials. After S rounds and $T = ST/S$ trials we are left with 1 overall winner.

For example for $S = 2$ rounds, $n = 2^2 = 4$ experts and $T = 6$ trials, consider the sequence of losses shown in Table F. Each of the two tournament consists of $6/2 = 3$ trials. In the first round, expert 1 is paired with expert 4 and expert 2 with expert 3. The cumulative losses of experts $1, 2, 3, 4$ in this round are $2, 1, 2, 1$, respective. So expert 4 is the winner of the first pair and expert 2 is the winner of the second pair. In the second round, the two winners (experts 2 and 4) are paired, and they incur cumulative loss 2 and 3, respectively. Hence, expert 2 wins the tournament. The total loss of the tournament winner in all 6 trials is 3. Note that this is larger than the minimum total loss of the 4 experts since expert 1 incurred total loss 2. Nevertheless we shall see that for our probabilistic lower bound proof, the total loss of the tournament winner is close enough to the total loss of the best expert.

To complete the proof it suffices to show that

$$\begin{aligned}&\mathbb{E}\left[\text{ total loss of tournament winner in } T \text{ trials }\right]\\ &\qquad = S\,\mathbb{E}\left[\text{ minimum loss among experts 1 and 2 in } T/S \text{ trials }\right].\end{aligned}$$

Due to linearity of expectation:

$$\begin{aligned}&\mathbb{E}\left[\text{ total loss of tournament winner in } T \text{ trials }\right]\\ &\qquad = \sum_{i=1}^{S} \mathbb{E}\left[\text{ total loss of tournament winner in } i\text{-th round }\right].\end{aligned}$$

The exchangeability of the losses and the symmetry of the tournament guarantees that each expert is equally likely to be the overall winner. Therefore w.l.o.g., expert 1 is the overall winner. Consider i-th round of the tournament ($1 \le i \le S$), and let (w.l.o.g.) expert 2 be

the partner of expert 1 in this round. We have:

$$\begin{aligned}
&\mathbb{E}\left[\text{ total loss of tournament winner in } i\text{-th round }\right] \\
&\quad= \mathbb{E}\left[\text{ total loss of exp. 1 in } i\text{-th round} \;\middle|\; \begin{array}{l}\text{exp. 1 is the tournament winner,}\\ \text{exp. 2 won all past competitions}\\ \text{at rounds } 1,\ldots,i-1.\end{array}\right] \\
&\quad= \mathbb{E}\left[\text{ total loss of exp. 1 in } i\text{-th round} \mid \text{exp. 1 wins over exp. 2 in } i\text{-th round }\right].
\end{aligned}$$

The second equality is due to the fact that the distribution of losses is i.i.d. between trials, and therefore the future and past rounds are independent of the current round. Since the last expression is the same for each of the S rounds we have:

$$\begin{aligned}
&\mathbb{E}\left[\text{ total loss of tournament winner in } T \text{ trials }\right] \\
&\quad= S\,\mathbb{E}\left[\text{ expected loss of expert 1 in } T/S \text{ trials} \mid \text{expert 1 wins over expert 2 }\right].
\end{aligned}$$

Remains to be shown that the latter expectation is simple the expectation of the minimum of the two experts losses in a single round. Let L_1 and L_2 be the total losses of both experts in the T/S trials and let "$L_1 > L_2$" denote the event that 1 wins over 2 (ties broken uniformly, so that, e.g., $\Pr(L_1 > L_2) + \Pr(L_2 > L_1) = 1$). Then

$$\begin{aligned}
\mathbb{E}\left[\, L_1 | L_2 > L_1 \,\right] &= \Big(\Pr(L_2 > L_1) + \Pr(L_1 > L_2)\Big)\,\mathbb{E}\left[\, L_1 | L_2 > L_1 \,\right], \\
&= \Pr(L_2 > L_1)\,\mathbb{E}\left[\, L_1 | L_2 > L_1 \,\right] \;+\; \Pr(L_1 > L_2)\,\mathbb{E}\left[\, \mathbf{L_1} | \mathbf{L_2} > \mathbf{L_1} \,\right] \\
(\textbf{exchangeability}) \quad &= \Pr(L_2 > L_1)\,\mathbb{E}\left[\, L_1 | L_2 > L_1 \,\right] \;+\; \Pr(L_1 > L_2)\,\mathbb{E}\left[\, \mathbf{L_2} | \mathbf{L_1} > \mathbf{L_2} \,\right] \\
&= \Pr(L_2 > L_1)\,\mathbb{E}\left[\, \min\{L_1, L_2\} | L_2 > L_1 \,\right] \\
&\quad \;+\; \Pr(L_1 > L_2)\,\mathbb{E}\left[\, \min\{L_1, L_2\} | L_1 > L_2 \,\right] \\
&= \mathbb{E}\left[\, \min\{L_1, L_2\} \,\right].
\end{aligned}$$

■

Now, we use this lemma to prove a lower bound for the m-set problem with $k \geq \frac{n}{2}$:

Theorem 5.2 *Consider the m-set problem with unit bit vectors as loss vectors, where $m = n - k$. Then for $k \geq \frac{n}{2}$ and $T \geq n \log_2(n/m)$, any online algorithm suffers worst case regret at least $\Omega(m\sqrt{T \ln(n/m)/n})$.*

Proof Let us first assume that $n = 2^j m$ for some integer $j > 0$, i.e. $\log_2(n/m)$ is a positive integer, and that $\frac{T}{\log_2(n/m)}$ is an integer value as well.

At each trial, a randomly chosen expert out of n experts incurs a unit of loss. To show an upper bound on the loss of the comparator, we partition the n experts into m groups (n divides m from the assumption), and notice that the losses of the n/m experts in each group are exchangeable. Applying Lemma F.3 to each group of n/m experts with $S = \log_2(n/m)$ rounds and T/S trials per round, we obtain:

$$\begin{aligned}
&\mathbb{E}\left[\text{ Loss of the winner in a given group in } T \text{ trials }\right] \\
&\quad\leq \log_2\left(\frac{n}{m}\right)\;\; \mathbb{E}\left[\text{ Loss of the winner of two experts in } \frac{T}{\log_2(n/m)} \text{ trials }\right]. \qquad \text{(F.2)}
\end{aligned}$$

The expectation on the RHS is the two expert game considered in Lemma F.1 with parameters $p = 1/n$ and $q = 0$. Note that $q = 0$ because only one expert suffers loss in each trial. Applying this lemma bounds the expectation on the RHS as

$$\frac{T}{\log_2(n/m)n} - c\sqrt{\frac{T}{\log_2(n/m)n}}.$$

Plugging this into (F.2) gives $T/n - c\sqrt{T\log_2(n/m)/n}$ upper bound on the expected loss of the winner in a given group. We upper bound the expected loss of the comparator by the total loss of m winners from the m groups, which in expectation is at most $Tm/n - cm\sqrt{T\log_2(n/m)/n}$.

Finally the loss of the algorithm is lower bounded as follows: Every expert incurs loss $1/n$ in expectation at each trial and losses are i.i.d. between trials. Therefore any online algorithm suffers loss at least mT/n. and the expected regret is lower bounded by $cm\sqrt{T\log_2(n/m)/n}$. This concludes the proof when $n = 2^j m$ and $\log_2(n/m)$ divides T.

If n is not of this form, we take the largest $n_0 < n$, such that $n_0 = 2^j m$ for some integer j, i.e. $n_0 = \max_{j\in\mathbb{N}}\{2^j m\colon 2^j m \le n\}$. We then apply the reasoning above to n_0 experts, while the remaining $n - n_0$ will incurs loss 1 all the time, which can only increase the loss of the algorithm, but this will not affect the loss of the comparator (comparator will never pick these experts). Since $n_0 \ge n/2$ (otherwise n_0 would not be the largest integer of the form $2^j n$, smaller than n), this does not change the rate under $\Omega(\cdot)$ for the lower bound in the statement of the theorem. Finally, if $\frac{T}{\log_2(n/m)}$ is not an integer value, we can choose the largest $T_0 < T$, such $\frac{T_0}{\log_2(n/m)}$ is integer, and use the proof with T_0 rounds, while in the remaining $T - T_0$ rounds all losses are zero. Since $T_0 \ge T/2$, this, again, does not change the rate under $\Omega(\cdot)$. ■

Finally, we consider the m-set problems with L_∞-bounded loss vectors. The following theorem proves lower bounds for such problems when $k \le \frac{n}{2}$ and when $k \ge \frac{n}{2}$.

Theorem 5.4 *Consider the m-set problem with loss vectors in $\{0,1\}^n$, where $m = n - k$. Then for $T \ge \log_2 \frac{n}{\min\{k,m\}}$, any online algorithm suffers worst case regret of at least*

$$\Omega(k\sqrt{T\ln\frac{n}{k}}) \text{ when } k \le \frac{n}{2} \quad \text{or} \quad \Omega(m\sqrt{T\ln\frac{n}{m}}) \text{ when } k \ge \frac{n}{2}.$$

Proof The proof is similar to the proof of Theorem 5.2, except that at each trial, the losses of all n experts are i.i.d. Bernoulli random variable with probability $p = 1/2$. For such a distribution over losses, any algorithm suffers expected cumulative loss at least $m\,T/2$ for the m-set problem.

For the sake of simplicity, we make some assumptions about n, k and T that avoid rounding issues. When $k \le n/2$, we assume that $n = 2^j k$ for some integer $j \ge 1$ and that $\frac{T}{\log_2(n/k)}$ is an integer. When $k \ge n/2$, i.e. $m = n - k \le n/2$, we assume that $n = 2^j m$ for some integer $j \ge 1$ and that $\frac{T}{\log_2(n/m)}$ is an integer. As in the proof of Theorem 5.2, it is easy to generalize the theorem to arbitrary n, k and T satisfying $T \ge \log_2 \frac{n}{\min\{k,m\}}$.

Now, we prove our regret lower bounds for each of the two cases: When $m \leq n/2$, we group the experts into m groups of size n/m and upper bound the loss of the comparator using the m group winners. As before, the loss of each winner can be upper bounded by the lemmas F.1 (with $p = q = 1/4$) and F.3:

$$\begin{aligned}
&\mathbb{E}\left[\text{ Loss of the winner in a given group in } T \text{ trials }\right] \\
&\overset{\text{Lemma F.3}}{\leq} \log_2 \frac{n}{m}\ \mathbb{E}\left[\text{ Loss of the winner of two experts in } \frac{T}{\log_2(n/m)} \text{ trials }\right] \\
&\overset{\text{Lemma F.1}}{\leq} \frac{T}{2} - c\sqrt{\frac{T}{4}\log_2 \frac{n}{m}}.
\end{aligned}$$

Note that since the experts here incur i.i.d. $Bernoulli(\frac{1}{2})$ losses, the above application of Lemma F.1 requires $p = q = 1/4$. Next, summing up m winners, we have the expected loss of the comparator upper bounded by $Tm/2 - cm\sqrt{T\log_2(n/m)/4}$. Taking the difference between this upper bound and the $Tm/2$ lower bound on loss of any algorithm results in the claimed $\Omega(m\sqrt{T\ln(n/m)})$ lower bound on the regret.

When $k \leq n/2$, we group the experts into k groups and consider a *loser* out of each group which is the expert which incurs the *largest* loss in each group. One can flip around the content of Lemma F.1 and F.3 to show that the loser in a group of n/k experts incurs loss in expectation at least $T/2 + c\sqrt{T\log_2(n/k)/4}$. Therefore, the expected loss of all k losers is lower bounded by $Tk/2 + ck\sqrt{T\log_2(n/k)/4}$. Now note that the expected loss of the comparator is upper bounded by the expected total loss of all the experts, which is $Tn/2$, minus the expected loss of the k losers, and hence upper bounded by

$$\frac{Tn}{2} - \left(\frac{Tk}{2} + ck\sqrt{\frac{T}{4}\log_2 \frac{n}{k}}\right) = \frac{Tm}{2} - ck\sqrt{\frac{T}{4}\log_2 \frac{n}{k}}.$$

Finally, the claimed regret bounds follows from taking the difference between this upper bound and the $Tm/2$ lower bound on the loss of any online algorithm. ■

Appendix G. Regret Lower Bounds When the Number of Trials Is Small

This appendix gives general regret lower bounds for the m-set problem when the number of trials T is small: Theorem G.1 and Theorem G.3 show lower bounds when the loss vectors are unit bit vectors; Theorem G.4 and Theorem G.5 show lower bounds when the loss vectors are bit vectors. Unlike the lower bounds for large T that are proved with probabilistic arguments (see previous Appendix F) all of the lower bounds in this appendix are proved by showing explicit adversary strategies that force large regret to any online algorithm. The matching upper bounds for small T are trivial and can be found in Section 5.

Theorem G.1 *Consider the m-set problem with unit bit vectors as loss vectors, where $m = n - k$. Then for $k \leq \frac{n}{2}$ and $T \leq k$, any online algorithm suffers worst case regret at least $\Omega(T)$.*

Proof Consider an adversary that at each trial gives a unit of loss to the expert with the largest weight assigned by the algorithm. Recall that $m = n - k$ and $k \leq \frac{n}{2}$. Therefore

all the weights assigned by the algorithm sum to $m \geq \frac{n}{2}$ and the largest weight out of n experts is at least $\frac{1}{2}$. Hence, after T trials, any algorithm suffers total loss at least $\frac{T}{2}$. On the other hand, since there are at least $n - T \geq m$ (because $T \leq k$) experts that are loss free, the loss of the best m-set of experts is zero. Therefore, the regret of any algorithm is at least $\frac{T}{2}$. ■

Now we consider the case when $k \geq \frac{n}{2}$. We start with a lemma which is parameterized by an integer $1 \leq i \leq k$ instead of the number of the trials T .

Lemma G.2 *Consider the m-set problem with unit bit vectors as loss vectors, where $m = n - k$. For any integer $1 \leq i \leq k$, an adversary can force any algorithm to suffer loss $\Omega(m \log_2 \frac{n}{n-i})$ in $O(n \log_2 \frac{n}{n-i})$ trials, and at the same time, keep a set of m experts with loss zero.*

Proof The adversary's strategy has i rounds, where the j-th round ($1 \leq j \leq i$) has at most $\left\lceil \frac{n}{n-j+1} \right\rceil$ trials and after it finishes, there will be at least $n - j$ experts that still have loss zero. The first round has only one trial, in which a unit of loss is given to the expert with the largest weight. Since all the weights assigned by the algorithm sum to m, the algorithm suffers loss at least $\frac{m}{n}$ in the first round.

Each of the following rounds may contain multiple trials and at the end of round $j-1$ ($2 \leq j \leq i$), there are still at least $n - j + 1$ loss free experts. In round j, the adversary uses a strategy with two subcases as follows: The adversary first considers the experts that are still loss free. If any of the first $n - j + 1$ of them has weight at least $\frac{m}{2(n-j+1)}$, then we are in case 1, where a unit of loss is given to this expert. Otherwise, we are in case 2, in which the adversary considers the remaining $j - 1$ experts (which may or may not be loss free) and gives a unit of loss to the one with the largest weight among them. The j-th round ends when the algorithm has suffered total loss at least $\frac{m}{2(n-j+1)}$ in that round. Note that whenever case 1 is reached, a round ends immediately. Our strategy guarantees that after round j, there are at least $n - j$ experts that are loss free. Next we upper bound the number of case 2 trials in a round by showing a lower bound on the loss of the algorithm in case 2 trials. Recall that in case 2, $n - j + 1$ experts have weight no more than $\frac{m}{2(n-j+1)}$ each, and the expert that has the largest weight in the remaining $j-1$ experts incurs a unit of loss. Using these facts as well as the fact that all the weights sum to m, we can lower bound the weight of the expert that incurs loss (which is also the loss of the algorithm) as follows:

$$\frac{\left(m - \frac{m}{2(n-j+1)}(n-j+1)\right)}{j-1} \;\geq\; \frac{m}{2(j-1)} \;\geq\; \frac{m}{2n}.$$

Recalling that the j-th round ends when the algorithm suffers total loss $\frac{m}{2(n-j+1)}$ in that round, we conclude that the j-th round can have at most $\left\lceil \frac{n}{n-j+1} \right\rceil$ trials.

Summing up over i rounds, the algorithm suffers total loss at least $\sum_{j=1}^{i} \frac{m}{2(n-j+1)} = \Omega(m \log \frac{n}{n+i})$ in at most $\sum_{j=1}^{i} \left\lceil \frac{n}{2(n-j-1)} \right\rceil = O(n \log \frac{n}{n-i})$ trials. On the other hand, the loss of the best m-set of experts is zero due to assumption $i \leq k$ and the fact that after $j = i$ rounds, there are at least $n - i$ loss free experts. Hence the lemma follows. ■

Theorem G.3 *Consider the m-set problem with unit bit vectors as loss vectors, where $m = n - k$. Then for $k \geq \frac{n}{2}$ and $T \leq n \log_2 \frac{n}{m}$, any algorithm suffers worst case regret at least $\Omega(\frac{m}{n}T)$.*

Proof Lemma G.2 states that there exist two positive constants c_1 and c_2, such that for any integer $1 \leq i \leq k$, the adversary can force any algorithm to suffer regret at least $c_1 m \log_2 \frac{n}{n-i}$ in at most $c_2 n \log_2 \frac{n}{n-i}$ trials. The proof splits into two cases, depending on the number of the trials T:

- When $T < c_2 n \log_2 \frac{n}{n-1}$, T is upper bounded by a constant as follows:

$$T < c_2 n \log_2 \frac{n}{n-1} = \frac{c_2 n}{\log 2} \log\left(1 + \frac{1}{n-1}\right) \leq \frac{c_2 n}{(n-1)\log 2} \overset{n \geq 2}{\leq} \frac{2c_2}{\log 2}.$$

  Since the adversary can always force any algorithm to suffer constant regret, the theorem holds trivially.

- When $T \geq c_2 n \log_2 \frac{n}{n-1}$, we set $i = \min\{\lfloor i' \rfloor, k\}$, where $i' = n(1 - 2^{-T/c_2 n})$ is the solution of $c_2 n \log_2 \frac{n}{n-i'} = T$. We note that the function $c_2 n \log_2 \frac{n}{n-i}$ is monotonically increasing in i, which results in two facts: first, $i \geq 1$, since we assumed $T \geq c_2 n \log_2 \frac{n}{n-1}$; second, $c_2 n \log_2 \frac{n}{n-i} \leq T$, since $i \leq \lfloor i' \rfloor$. We further show that $c_2 n \log_2 \frac{n}{n-i} \geq \min\{c_2, \frac{1}{3}\}T$ as follows:

  - When $i = \lfloor i' \rfloor$, first note that $\left(\frac{n}{n-i}\right)^3 \geq \frac{n}{n-i'}$, since:

$$(n-i')n^2 - (n-i)^3 \geq (n-i-1)n^2 - (n-i)^3 = 2n^2 i + 3ni^2 - i^3 - n^2 \overset{1 \leq i < n}{\geq} 0.$$

    Plugging $c_2 n \log_2 \frac{n}{n-i'} = T$, we have $c_2 n \log_2 \frac{n}{n-i} \geq \frac{1}{3}T$.
  - When $i = k$, $c_2 n \log_2 \frac{n}{n-i} = c_2 n \log_2 \frac{n}{m} \geq c_2 T$, since $T \leq n \log_2 \frac{n}{m}$ is assumed in the theorem.

  Now, using Lemma G.2 with i set as $i = \min\{\lfloor i' \rfloor, k\}$, results in an adversary that forces the algorithm to suffer regret at least $c_1 m \log \frac{n}{n-i} \geq \frac{m c_1}{n c_2} \min\{c_2, \frac{1}{3}\}T = \Omega\left(\frac{m}{n}T\right)$ in at most T trials. When the adversary uses less than T trials, then the game can be extended to last exactly T trials by playing zero loss vectors for the remaining trials.

■

Theorem G.4 *Consider the m-set problem with loss vectors in $\{0,1\}^n$, where $m = n - k$. Then for $k \geq \frac{n}{2}$ and $T \leq \log_2 \frac{n}{m}$, the worst case regret of any algorithm is at least $\Omega(Tm)$.*

Proof The proof uses an adversary which forces any algorithm to suffer loss $\Omega(Tm)$, and still keeps the best m-set of experts to be loss free. Note that at each trial, the adversary decides on the loss vector after the algorithm makes its prediction $\boldsymbol{w}_t$, where $\boldsymbol{w}_t \in [0,1]^n$ with $\sum_i w_{t,i} = m$.

At trial one, the adversary first sorts the n experts by their weights assigned by the algorithm, and then gives a unit of loss to each of the experts in the first half, i.e. the experts with larger weights. Since the weights sum to m, the total weight assigned to the experts in the first half is at least $\frac{m}{2}$. Hence in the first trial, the algorithm suffers loss at least $\frac{m}{2}$.

At each of the following trials, the adversary only sorts those experts that have not incur any loss so far and gives unit losses to the first half (the half with larger weights) of these experts, as well as all the experts that have already incurred losses before this trial. It is easy to see that in this way the algorithm suffers loss at least $\frac{m}{2}$ at each trial.

Since the number of the experts that are loss free halves at each trial, after $T \leq \log_2 \frac{n}{m}$ trials, there will still be at least m loss free experts. Now since the algorithm suffers loss at least $\frac{mT}{2}$ in T trials, the theorem follows. ■

Theorem G.5 *Consider the m-set problem with loss vectors in $\{0,1\}^n$, where $m = n - k$. Then for $k \leq \frac{n}{2}$ and $T \leq \log_2 \frac{n}{k}$, any algorithm suffers worst case regret at least $\Omega(Tk)$.*

Proof The proof becomes conceptually simpler if we use the notion of *gain* defined as the follows: if $\boldsymbol{w}_t$ is the parameter of the algorithm, we define its complement $\bar{\boldsymbol{w}}_t$ as $\bar{w}_{t,i} = 1 - w_{t,i}$. The gain of the algorithm at trial t is the inner product between the "gain" vector $\boldsymbol{\ell}_t$ and the complement $\bar{\boldsymbol{w}}_t$, i.e. $\bar{\boldsymbol{w}}_t \cdot \boldsymbol{\ell}_t$. Similarly, for any comparator $\boldsymbol{w} \in \boldsymbol{\mathcal{S}}_m$, we define its gain as $\bar{\boldsymbol{w}} \cdot \boldsymbol{\ell}_t = \sum_{i=1}^n (1 - w_i) l_{t,i}$. It is easy to verify that the regret of the algorithm can be written as the difference between the largest gain of any subset of k experts and the gain of the algorithm:

$$\mathcal{R} = \max_{\bar{\boldsymbol{w}} \in \boldsymbol{\mathcal{S}}_k} \sum_{t=1}^T \bar{\boldsymbol{w}} \cdot \boldsymbol{\ell}_t - \sum_{t=1}^T \bar{\boldsymbol{w}}_t \cdot \boldsymbol{\ell}_t,$$

where $\boldsymbol{\mathcal{S}}_k = \{\boldsymbol{w} \in [0,1]^n \colon \sum_i w_i = k\}$. At trial one, the adversary first sorts the n experts by their complementary weights and then gives a unit of gain to each of the experts in the second half, i.e. the experts with smaller complementary weights. Since the complementary weights sum to k, the gain of the algorithm is at most $\frac{k}{2}$ in the first trial.

At each of the following trials, the adversary only sorts the experts that received gains in all of the previous trials by their complementary weights. It then gives unit gains to the second half (the half with smaller complementary weights) of these experts. It is easy to see that in this way the gain of the algorithm is at most $\frac{k}{2}$ at each trial.

Note that half of the experts that always receive gain prior to a trial t will receive gain again in trial t. Hence, after $T \leq \log_2 \frac{n}{k}$ trials, there will be at least k experts that received gains in all of the T trials, which means that the total gain of the best k experts is Tk. Now, since the algorithm receives total gain at most $\frac{kT}{2}$ in T trials, the theorem follows. ■

Appendix H. Proof of Theorem 5.6

The following theorem gives a regret lower bound that is expressed as a function of the loss budget B_L. This lower bound holds for any online algorithm that solves the m-set problem

with either unit bit vectors or arbitrary bit vectors as loss vectors. The proof is based on the time dependent regret lower bounds proven in the previous appendices.

Theorem 5.6 *For the m set problem with either unit bit vectors or arbitrary bit vectors, any online algorithm suffers worst case regret of at least $\Omega(\max\{\sqrt{B_L m \ln(n/m)}, m\ln(n/m)\})$.*

Proof It suffices to prove the lemma for unit bit vectors. The lower bound $\Omega(m\ln(n/m))$ follows directly from Lemma G.2 by setting the variable i of the lemma to k.

What is left to show is the lower bound $\Omega(\sqrt{B_L m\ln(n/m)})$ when it dominates the bound $\Omega(m\ln(n/m))$, i.e. when $B_L = \Omega(m\ln\frac{n}{m})$. Thus, we assume $B_L \geq m\log_2\frac{n}{m}+1$ and we construct an instance sequence of loss budget B_L incurring regret at least $\Omega(\sqrt{B_L m\ln(n/m)})$ to any algorithm. This instance sequence is constructed via Theorem 5.1 and Theorem 5.2: For any algorithm, these theorems provide a sequence of T unit bit vectors that incurs regret at least $\Omega(m\sqrt{\frac{T\ln(n/m)}{n}})$. We apply these theorems with $T = \lfloor\frac{n}{m}B_L\rfloor \geq n\log_2\frac{n}{m}$. Since the produced sequence consists of unit bit vectors and has length $\lfloor\frac{n}{m}B_L\rfloor$, the total loss of the m best experts is at most B_L. Finally plugging $T = \lfloor\frac{n}{m}B_L\rfloor$ into the regret bounds guaranteed by the theorems results in the regret $\Omega(\sqrt{B_L m\ln(n/m)})$. ■

Appendix I. Auxiliary Lemmas

Lemma I.1 *Inequality $\max\{\min\{a,b\},c\} \geq \min\{\max\{a,c\},b\}$ holds for any real number a, b and c.*

Proof If $c \geq \max\{a,b\}$, LHS is c and RHS is b. Hence, the inequality holds. If $a \geq c \geq b$ or $b \geq c \geq a$, LHS is c while RHS is at most c. If $c \leq a$ and $c \leq b$, both sides are $\min\{a,b\}$. ■

Lemma I.2 *Let $X \sim$ Binomial(T,p). If $Tp \geq 8c$ for any positive constant c, then $\mathbb{E}[\sqrt{X}] \geq \frac{c}{\sqrt{2}(1+c)}\sqrt{Tp}$.*

Proof We use the following form of the Chernoff bound (DeGroot and Schervish, 2002):

$$\Pr(X \leq Tp - \delta) \leq e^{-\frac{\delta^2}{2Tp}}.$$

Setting $\delta = \frac{1}{2}Tp$, we have $\Pr(X \leq \frac{1}{2}Tp) \leq e^{-Tp/8} \leq e^{-c}$. Since for $c > 0$, $\log(c) \leq c-1$, this implies $e^{-c} \leq \frac{1}{1+c}$, so that we further have $\Pr(X \leq \frac{1}{2}Tp) \leq \frac{1}{1+c} = 1 - \frac{c}{1+c}$. Now we

calculate $\mathbb{E}[\sqrt{X}]$ from its definition,

$$\begin{aligned}
\mathbb{E}[\sqrt{X}] = \sum_{x=0}^{T} \Pr(X=x)\sqrt{x} &\geq \sum_{x=\left\lfloor \frac{Tp}{2} \right\rfloor + 1}^{T} \Pr(X=x)\sqrt{x} \\
&\geq \sum_{x=\left\lfloor \frac{Tp}{2} \right\rfloor + 1}^{T} \Pr(X=x)\sqrt{\left\lfloor \frac{Tp}{2} \right\rfloor + 1} \\
&= \Pr(X > \tfrac{1}{2}Tp)\sqrt{\left\lfloor \frac{Tp}{2} \right\rfloor + 1} \\
&\geq \frac{c}{\sqrt{2}(1+c)}\sqrt{TP}.
\end{aligned}$$

■

References

Jacob Abernethy, Manfred K. Warmuth, and Joel Yellin. When random play is optimal against an adversary. In *COLT*, pages 437–446, 2008.

Jacob Abernethy, Alekh Agarwal, Peter L. Bartlett, and Alexander Rakhlin. A stochastic view of optimal regret through minimax duality. In *COLT*, pages 56–64, 2009.

Raman Arora, Andrew Cotter, and Nati Srebro. Stochastic optimization of PCA with capped MSG. In *NIPS*, pages 1815–1823, 2013.

Jean-Yves Audibert and Sébastien Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11:2785–2836, 2010.

Katy S. Azoury and Manfred K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006. ISBN 978-0-521-84108-5.

Nicolò Cesa-Bianchi, Philip M. Long, and Manfred K. Warmuth. Worst-case quadratic loss bounds for prediction using linear functions and gradient descent. *IEEE Trans. Neural Netw. Learning Syst.*, 7(3):604–619, 1996.

Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

Steven de Rooij, Tim van Erven, Peter D. Grünwald, and Wouter M. Koolen. Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research*, 15: 1281–1316, 2014.

Morris H. DeGroot and Mark J. Schervish. *Probability and Statistics*. Addison-Wesley series in statistics. Addison-Wesley, 2002. ISBN 9780201524888.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT*, pages 23–37, 1995.

Dan Garber, Elad Hazan, and Tengyu Ma. Online learning of eigenvectors. In *ICML*, pages 560–568, 2015.

Elad Hazan, Satyen Kale, and Manfred K. Warmuth. On-line variance minimization in $o(n^2)$ per trial? In *COLT*, pages 314–315, 2010.

David P. Helmbold and Manfred K. Warmuth. Learning permutations with exponential weights. *Journal of Machine Learning Research*, 10:1705–1736, 2009.

Mark Herbster and Manfred K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.

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

Wouter M. Koolen. *Combining Strategies Efficiently: High-quality Decisions from Conflicting Advice*. PhD thesis, Institute of Logic, Language and Computation (ILLC), University of Amsterdam, 2011.

Wouter M. Koolen, Manfred K. Warmuth, and Jyrki Kivinen. Hedging structured concepts. In *COLT*, pages 93–105, 2010.

Wojciech Kotlowski and Manfred K. Warmuth. PCA with Guassian perturbation. Private communication, 2015.

Arkadi Nemirovski and D Yudin. On Cesaro's convergence of the gradient descent method for finding saddle points of convex-concave functions. *Doklady Akademii Nauk*, 4(249): 249, 1978.

Jiazhong Nie, Wojciech Kotlowski, and Manfred K. Warmuth. Online PCA with optimal regrets. In *ALT*, pages 98–112, 2013.

Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2-3):115–142, 2007.

Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. On the universality of online mirror descent. In *NIPS*, pages 2645–2653, 2011.

Karthik Sridharan and Ambuj Tewari. Convex games in Banach spaces. In *COLT*, pages 1–13, 2010.

Koji Tsuda, Gunnar Rätsch, and Manfred K. Warmuth. Matrix Exponential Gradient updates for on-line learning and Bregman projection. *Journal of Machine Learning Research*, 6:995–1018, 2005.

Tim van Erven, Peter Grünwald, Wouter M. Koolen, and Steven de Rooij. Adaptive Hedge. In *NIPS*, pages 1656–1664, 2011.

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

Manfred K. Warmuth and S. V. N. Vishwanathan. Leaving the span. In *COLT*, pages 366–381, 2005.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.