JMLR: Workshop and Conference Proceedings vol 30:1–16, 2013

Learning a set of directions*

Wouter M. Koolen WMKOOLEN@CWI.NL
Centrum Wiskunde & Informatica, P.O. Box 94079, NL-1090 GB Amsterdam, The Netherlands

Jiazhong Nie NIEJIAZHONG@SOE.UCSC.EDU

Manfred K. Warmuth MANFRED@SOE.UCSC.EDU
Department of Computer Science, University of California, Santa Cruz, CA 95064

Abstract

Assume our data consists of unit vectors (directions) and we are to find a small orthogonal set of the "the most important directions" summarizing the data. We develop online algorithms for this type of problem. The techniques used are similar to Principal Component Analysis which finds the most important small rank subspace of the data. The new problem is significantly more complex since the online algorithm maintains uncertainty over the most relevant subspace as well as directional information.

1. Introduction

In this paper we consider learning directions. Let us fix the dimensionality n throughout. Then a direction is simply a vector $\boldsymbol{u} \in \mathbb{R}^n$ of unit length. We model the learning problem as a sequential game where each round the learner predicts by playing a direction $\boldsymbol{u}$ and nature responds with an instance direction $\boldsymbol{x}$. We define the resulting *directional gain* as

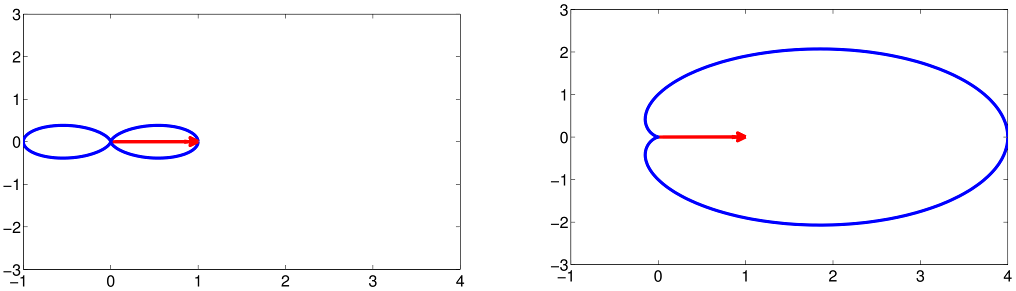
$$\left(\boldsymbol{u}^\intercal \boldsymbol{x} + c\right)^2 \tag{1.1}$$

where the constant c is a fixed design parameter known to the learner. We choose to study this gain because it is a simple and smooth trade-off (governed by c) between two intuitively reasonable criteria of closeness: the angle cosine and the subspace similarity. To see this, we expand our gain as:

$$\left(\boldsymbol{u}^\intercal \boldsymbol{x} + c\right)^2 \;=\; (\boldsymbol{u}^\intercal \boldsymbol{x})^2 + 2c\,\boldsymbol{u}^\intercal \boldsymbol{x} + c^2. \tag{1.2}$$

- As $c \to \infty$, then our gain becomes the angle cosine $\boldsymbol{u}^\intercal \boldsymbol{x} = \cos(\boldsymbol{u}, \boldsymbol{x})$. There is a simple minimax algorithm for this angle gain by Kotłowski and Warmuth (2011).
- When $c = 0$, then our gain becomes the subspace similarity $(\boldsymbol{u}^\intercal \boldsymbol{x})^2$. This gain is optimized in rank one (un-centered) PCA. (Warmuth and Kuzmin, 2008). The main disadvantage of the PCA gain $(\boldsymbol{u}^\intercal \boldsymbol{x})^2$ is that it is fundamentally bidirectional, i.e. reversing either $\boldsymbol{u}$ or $\boldsymbol{x}$ does not affect this gain.

[*] The last two authors were supported by NSF grant IIS-0917397, and the first author was supported by the same grant while visiting Santa Cruz.



(a) PCA gain $(\boldsymbol{u}^\intercal \boldsymbol{x})^2$ (b) Directional gain $\big(\boldsymbol{u}^\intercal \boldsymbol{x} + 1\big)^2$

Figure 1.1: Comparison of PCA gain and directional gain (for $c = 1$): The target direction $\boldsymbol{x}$ is depicted by a red arrow. In each case the blue curve is $\boldsymbol{u}$ scaled by the directional gain of $\boldsymbol{u}$, as the prediction $\boldsymbol{u}$ goes around the unit circle.

- For general c, the directional gain (1.2) is a trade-off between the above two gains. Unfortunately the algorithms for the linear and quadratic gains cannot just be merged somehow. As we shall see the tools needed for the trade-off gain are much more involved. The new directional gain (for $c = 1$) as well as the original PCA gain are plotted in Figure 1. Note that the directional gain is highly sensitive[1] to the direction of the prediction $\boldsymbol{u}$ as well as the target instance $\boldsymbol{x}$: it attains maximum value 4 when $\boldsymbol{x}$ is the same direction as $\boldsymbol{u}$ (i.e. $\boldsymbol{x} = \boldsymbol{u}$) and minimum value 0 at the opposite $\boldsymbol{x} = -\boldsymbol{u}$. For other values of c the range of the gain is discussed in Appendix A.

- Note that the quadratic Taylor approximation of any gain $g(\boldsymbol{u}^\intercal \boldsymbol{x})$ at $\boldsymbol{u} = \boldsymbol{0}$ has the form $g(0) + g'(0)\, \boldsymbol{u}^\intercal \boldsymbol{x} + \frac{1}{2} g''(0)\, (\boldsymbol{u}^\intercal \boldsymbol{x})^2$. Dividing by $\frac{1}{2} g''(0)$ results in our gain (1.2) except for an immaterial constant shift.

So how can we get away with maximizing a quadratic gain? Note that the gain is linear in $\boldsymbol{u}$ and $\boldsymbol{u}\boldsymbol{u}^\intercal$, and therefore the underlying optimization problems become linear semi-definite.

We think of a sequence of instances $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ as "easy" if there is a single direction $\boldsymbol{u}$ with high cumulative gain. The goal of the learner is to predict well if the data are easy. To this end, we evaluate the performance of the learner after T rounds by measuring its *regret*:

$$\underbrace{\max_{\text{unit } \boldsymbol{u}} \sum_{t=1}^{T} \big(\boldsymbol{u}^\intercal \boldsymbol{x}_t + c\big)^2}_{\text{offline gain}} - \underbrace{\sum_{t=1}^{T} \big(\boldsymbol{u}_t^\intercal \boldsymbol{x}_t + c\big)^2}_{\text{online gain}}.$$

Here $\boldsymbol{u}_t$ denotes the direction of the online algorithm chosen at trial t. To be able to guarantee low regret in an adversarial environment, it is sometimes advantageous to choose

[1] The bidirectional PCA gain is essentially the average of the directional gain for $\boldsymbol{x}$ and $-\boldsymbol{x}$:

$$\underbrace{(\boldsymbol{u}^\intercal \boldsymbol{x})^2}_{\text{PCA gain}} = \frac{1}{2}\Big(\underbrace{(\boldsymbol{u}^\intercal \boldsymbol{x} + c)^2}_{\text{directional gain of } \boldsymbol{x}} + \underbrace{(\boldsymbol{u}^\intercal (-\boldsymbol{x}) + c)^2}_{\text{directional gain of } -\boldsymbol{x}} \Big) - c^2.$$

Thus the algorithms of this paper retain PCA as a special case when the instance directions are doubled.

the direction $\boldsymbol{u}_t$ probabilistically and define the regret as the offline gain minus the *expected* online gain. A probability distribution $\mathbb{P}$ on predictions $\boldsymbol{u}$ has expected gain given by

$$\mathbb{E}\left[\left(\boldsymbol{x}^\intercal\boldsymbol{u}+c\right)^2\right] \;=\; \mathbb{E}\left[\boldsymbol{x}^\intercal\boldsymbol{u}\boldsymbol{u}^\intercal\boldsymbol{x}+2c\boldsymbol{x}^\intercal\boldsymbol{u}+c^2\right] \;=\; \boldsymbol{x}^\intercal\,\mathbb{E}\big[\boldsymbol{u}\boldsymbol{u}^\intercal\big]\boldsymbol{x}+2c\boldsymbol{x}^\intercal\,\mathbb{E}\left[\boldsymbol{u}\right]+c^2.$$

This shows that most of $\mathbb{P}$ is irrelevant. The expected gain is determined by just the first moment (mean) $\mathbb{E}[\boldsymbol{u}]$ and second moment $\mathbb{E}[\boldsymbol{u}\boldsymbol{u}^\intercal]$. In this paper we never work with full distributions, but always with these simple two statistics. That is, the parameter of the algorithm has the form $(\boldsymbol{\mu},\boldsymbol{D})$, s.t. $(\boldsymbol{\mu},\boldsymbol{D})=\mathbb{E}\left[(\boldsymbol{u},\boldsymbol{u}\boldsymbol{u}^\intercal)\right]$ for some $\mathbb{P}$. It is hence important to characterize which pairs of first and second moments can arise from distributions: We will show that a vector $\boldsymbol{\mu}$ and symmetric matrix $\boldsymbol{D}$ are the first and second moment of some distribution on directions iff $\operatorname{tr}(\boldsymbol{D})=1$ and $\boldsymbol{D}\succeq\boldsymbol{\mu}\boldsymbol{\mu}^\intercal$. Note that these conditions imply that $\boldsymbol{D}$ is a density matrix, i.e. a positive semi-definite matrix of unit trace.

Our algorithm has the following outline. At the beginning of each trial we decompose the current parameter $(\boldsymbol{\mu}_t,\boldsymbol{D}_t)$ into a sparse mixture of pure events $(\boldsymbol{u},\boldsymbol{u}\boldsymbol{u}^\intercal)$ and choose a direction $\boldsymbol{u}_t$ at random from this mixture. We then update the parameter based on the observed instance $\boldsymbol{x}_t$ and project the updated parameter back into the parameter space.

We also consider the direction learning problem where each round the learner plays a *set* of k orthogonal directions $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k$. The set size k is a fixed design parameter known to the learner. After nature reveals its instance $\boldsymbol{x}$, the algorithm now achieves the total gain over the set:

$$\sum_{i=1}^{k}\left(\boldsymbol{u}_i^\intercal\boldsymbol{x}+c\right)^2. \tag{1.3}$$

The online algorithm chooses such a set probabilistically in each trial. If $\mathbb{P}$ is a probability distribution on such sets $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k$, then the expectation of the gain (1.3) expands to

$$\mathbb{E}\left[\sum_{i=1}^{k}\left(\boldsymbol{x}^\intercal\boldsymbol{u}_i+c\right)^2\right] \;=\; \boldsymbol{x}^\intercal\;\mathbb{E}\left[\sum_{i=1}^{k}\boldsymbol{u}_i\boldsymbol{u}_i^\intercal\right]\boldsymbol{x}+2c\boldsymbol{x}^\intercal\;\mathbb{E}\left[\sum_{i=1}^{k}\boldsymbol{u}_i\right]+kc^2.$$

We see that the expected gain is again determined by the first moment $\mathbb{E}\left[\sum_{i=1}^{k}\boldsymbol{u}_i\right]$ and second moment $\mathbb{E}\left[\sum_{i=1}^{k}\boldsymbol{u}_i\boldsymbol{u}_i^\intercal\right]$. We will show that a vector $\boldsymbol{\mu}$ and matrix $\boldsymbol{D}$ are the first two moments of a distribution on sets of k orthogonal directions iff $\operatorname{tr}(\boldsymbol{D})=k$ and $\boldsymbol{\mu}\boldsymbol{\mu}^\intercal/k\preceq\boldsymbol{D}\preceq\boldsymbol{I}$. The parameter space of our algorithm hence consists of all $(\boldsymbol{\mu},\boldsymbol{D})$ with these properties. Again we present an algorithm for decomposing an arbitrary parameter $(\boldsymbol{\mu},\boldsymbol{D})$ into a sparse mixture of pure events $(\sum_{i=1}^{k}\boldsymbol{u}_i,\sum_{i=1}^{k}\boldsymbol{u}_i\boldsymbol{u}_i^\intercal)$ with orthonormal $\boldsymbol{u}_i$ and sample from this mixture at the beginning of each trial. We also generalize our projection algorithm to the $k>1$ case.

The gain (1.1) (and set generalization (1.3)) are quadratic in their natural parametrization by the direction $\boldsymbol{u}$. However by expanding the square, we find that they are *linear* in the two parts $\boldsymbol{u}$ and $\boldsymbol{u}\boldsymbol{u}^\intercal$. Our setup exploits this linear reformulation of the gain.

We still need to discuss which type of algorithms should be used for updating the parameter matrix after processing the current direction. There are two families of algorithms to consider: the Matrix Exponentiated Gradient family that is based on regularizing with the Quantum Relative Entropy (Tsuda et al., 2005) and the Gradient Descent family which uses the squared Frobenius norm as a regularizer. For our problem the representatives from both families have the same regret bound (not shown for MEG) as a function of the number of examples. However, MEG has a budget bound as well (not shown), i.e. the time horizon in the regret can be replaced by an upper bound on the total gain of the comparator. We only discuss the simpler GD algorithm in this paper even though we don't have a budget bound for this algorithm.[2]

Related work

The outline of our algorithm is similar to Component Hedge (Koolen et al., 2010) which deals with distributions on exponentially many combinatorial concepts by maintaining the expectation of their constituent components. The key two pieces are the convex decomposition and the projection algorithm. This method was lifted to the matrix domain in the work on online PCA (Warmuth and Kuzmin, 2008). However each piece is significantly more complicated in our setting because our gain trades off first and second order parts.

Our work is related to centered PCA (Warmuth and Kuzmin, 2008) which also uses a mean and a density matrix as the parameter. However in that case the mean is unconstrained and can be optimized independently from the density, leading to a much simpler problem.

Our gain $(\boldsymbol{u}^\intercal \boldsymbol{x} + c)^2$ is a simple polynomial kernel with the feature map $\phi(\boldsymbol{u})$ being comprised of the n components of $\boldsymbol{u}$, the n^2 components of $\boldsymbol{u}\boldsymbol{u}^\intercal$ and a constant feature. However our methods are decidedly different from kernel methods (including Kernel PCA (Kuzmin and Warmuth, 2007)). Our algorithms don't just rely on dot products $\phi(\boldsymbol{x}_t)^\intercal \phi(\boldsymbol{x}_q)$ in feature space (the kernel paradigm). Instead, our parameter is always a convex combination of $\phi(\boldsymbol{u})$ and we project back into this parameter space. This projection step clearly violates the kernel paradigm.

Outline

We warm up by optimizing the gain offline in Section 2. We then present the online algorithm in Section 3 and analyze its regret. The essential building block in both these sections is the characterization of the parameter space. We prove the difficult direction of the characterization theorem in Section 4 by presenting our new decomposition algorithm. We conclude by discussing the big picture in Section 5.

[2] The issue of how to prove budget bounds for GD is an independent problem.

2. The offline problem

Given a sequence of directions $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$, the offline problem is to optimize the total gain:

$$\begin{aligned}
&\max_{\text{orthonormal } \boldsymbol{u}_1 \ldots \boldsymbol{u}_k} \sum_{t=1}^{T} \sum_{i=1}^{k} (\boldsymbol{u}_i^\intercal \boldsymbol{x}_t + c)^2 \\
= &\max_{\text{orthonormal } \boldsymbol{u}_1 \ldots \boldsymbol{u}_k} \operatorname{tr}\Big(\sum_{i=1}^{k} \boldsymbol{u}_i \boldsymbol{u}_i^\intercal \underbrace{\sum_{t=1}^{T} \boldsymbol{x}_t \boldsymbol{x}_t^\intercal}_{=:\boldsymbol{R}}\Big) \;+\; 2c\Big(\sum_{i=1}^{k} \boldsymbol{u}_i\Big)^\intercal \underbrace{\sum_{t=1}^{T} \boldsymbol{x}_t}_{=:\boldsymbol{r}} + Tc^2.
\end{aligned}$$

We will reformulate the above as a semi-definite optimization problem. Instead of maximizing over a single orthonormal set, we maximize the expected value of the objective over distributions on such sets. This does not change the value of the optimization problem. For any probability distribution on sets of k orthogonal directions, we can characterize the first moment $\mathbb{E}\left[\sum_{i=1}^{k} \boldsymbol{u}_i\right]$ and second moment $\mathbb{E}\left[\sum_{i=1}^{k} \boldsymbol{u}_i \boldsymbol{u}_i^\intercal\right]$ as follows:

Theorem 2.1 *A vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and symmetric matrix $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ are the first and second moment of a probability distribution on sets of k orthogonal directions if and only if*

$$\operatorname{tr}(\boldsymbol{D}) \;=\; k \qquad \textit{and} \qquad \boldsymbol{\mu}\boldsymbol{\mu}^\intercal/k \preceq \boldsymbol{D} \preceq \boldsymbol{I}. \tag{2.1}$$

Proof For the $\Longrightarrow$ direction, it suffices to show that (2.1) is satisfied for "pure" distributions, i.e. when $\boldsymbol{D} = \sum_{i=1}^{k} \boldsymbol{u}_i \boldsymbol{u}_i^\intercal$ and $\boldsymbol{\mu} = \sum_{i=1}^{k} \boldsymbol{u}_i$, for some set of orthogonal directions. The result then extends to all distributions by convexity. Since the condition is invariant under basis transformations, we may as well verify it for the set of standard basis vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_k$. Its first and second moment are

$$\boldsymbol{\mu} \;=\; \begin{bmatrix} \boldsymbol{1}_k \\ \boldsymbol{0}_{n-k} \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{D} \;=\; \begin{bmatrix} \boldsymbol{I}_k & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}.$$

Clearly, $\operatorname{tr}(\boldsymbol{D}) = k$ and $\boldsymbol{D} \preceq \boldsymbol{I}$. To show that $\boldsymbol{\mu}\boldsymbol{\mu}^\intercal/k \preceq \boldsymbol{D}$, note that $\boldsymbol{\mu}$ is the only eigenvector of $\boldsymbol{\mu}\boldsymbol{\mu}^\intercal/k$, and its associated eigenvalue is 1. However, $\boldsymbol{\mu}$ is also an eigenvector of $\boldsymbol{D}$, again with eigenvalue 1. The $\Longleftarrow$ direction is much harder. It follows from the decomposition procedure presented in Section 4. ■

This means that our offline problem becomes the following semi-definite program:

$$\max_{(\boldsymbol{\mu},\boldsymbol{D}) \text{ s.t. } \operatorname{tr}(\boldsymbol{D})=k \text{ and } \boldsymbol{\mu}\boldsymbol{\mu}^\intercal/k \preceq \boldsymbol{D} \preceq \boldsymbol{I}} \quad \operatorname{tr}(\boldsymbol{D}\boldsymbol{R}) + 2c\,\boldsymbol{\mu}^\intercal \boldsymbol{r} + Tc^2.$$

In Appendix B we discuss a special condition on $(\boldsymbol{r}, \boldsymbol{R})$ when the solution of the k directions problem can be constructed from the solution to the k-PCA problem.

Note that the solution $(\boldsymbol{\mu}^*, \boldsymbol{D}^*)$ returned for the above optimization problem might not be a pure set of k directions but the first and second moment of a distribution on sets of k orthogonal directions, all of which have the same gain. In that case we can employ the decomposition algorithm of Section 4 which decomposes the moments $(\boldsymbol{\mu}^*, \boldsymbol{D}^*)$ into a mixture of pure solutions. To obtain one set, simply run this greedy algorithm for one step.

3. Online algorithm

The algorithm maintains the two moments $(\boldsymbol{\mu}_t, \boldsymbol{D}_t)$ as its parameter. It follows the protocol:

```
At trial $t = 1 \dots T$,
1.  Learner *decomposes* parameter $(\boldsymbol{\mu}_t, \boldsymbol{D}_t)$ into a mixture of $2(n+1)$ sets
    of $k$ orthonormal directions and chooses a set $\boldsymbol{u}_1, \dots, \boldsymbol{u}_k$ at random from from it
2.  Nature reveals direction $\boldsymbol{x}_t \in \mathbb{R}^n$
3.  Learner receives expected gain $\mathbb{E}\left[\sum_{i=1}^k (\boldsymbol{u}_i^\intercal \boldsymbol{x}_t + c)^2\right]$
4.  Learner *updates* $(\boldsymbol{\mu}_t, \boldsymbol{D}_t)$ to $(\widehat{\boldsymbol{\mu}}_{t+1}, \widehat{\boldsymbol{D}}_{t+1})$ based on the gradient of the gain on $\boldsymbol{x}_t$
5.  Learner produces new parameter $(\boldsymbol{\mu}_{t+1}, \boldsymbol{D}_{t+1})$ by
    *projecting* $(\widehat{\boldsymbol{\mu}}_{t+1}, \widehat{\boldsymbol{D}}_{t+1})$ back into the parameter space.
```

The goal of the learner is to minimize the regret which is the gain of the offline algorithm minus the expected gain of the online algorithm. We first show how to update and project (steps 4 and 5) and defer the decomposition step 1 to the end, since it is the hardest.

3.1. The update and projection

We update using the Gradient Descent algorithm (see e.g. Kivinen and Warmuth (1997); Zinkevich (2003))

$$\widehat{\boldsymbol{\mu}}_{t+1} \;\coloneqq\; \boldsymbol{\mu}_t + 2\eta c\, \boldsymbol{x}_t \qquad \text{and} \qquad \widehat{\boldsymbol{D}}_{t+1} \;\coloneqq\; \boldsymbol{D}_t + \eta\, \boldsymbol{x}_t \boldsymbol{x}_t^\intercal,$$

and project back into the parameter space as follows:

$$(\boldsymbol{\mu}_{t+1}, \boldsymbol{D}_{t+1}) \;\coloneqq\; \operatorname*{argmin}_{(\boldsymbol{\mu}, \boldsymbol{D}) \text{ s.t. } \operatorname{tr}(\boldsymbol{D}) = k \text{ and } \boldsymbol{\mu}\boldsymbol{\mu}^\intercal / k \preceq \boldsymbol{D} \preceq \boldsymbol{I}} \quad \|\boldsymbol{D} - \widehat{\boldsymbol{D}}_{t+1}\|_F^2 + \|\boldsymbol{\mu} - \widehat{\boldsymbol{\mu}}_{t+1}\|^2.$$

Since both the objective and the constraint set are convex, this projection can be efficiently computed using a convex optimization package.[3]

The above GD update and the projection are based on regularizing with the square Frobenius norm. An alternate would be the Matrix Exponentiated Gradient update which uses the Quantum Relative Entropy as a regularizer. Since the MEG update has the same regret bound (not shown) for our specific problem based on unit instance vectors, we chose to only present the simpler GD update.

The following theorem develops a regret bound for the GD algorithm. Note that the squared Frobenius norm is used as a measure of progress. We don't need to be concerned with the projection step since the Pythagorean Theorem implies that the projection step does not hurt (Herbster and Warmuth, 2001).

Theorem 3.1 *Fix dimension n, set size k and gain constant c. The regret after T trials of the GD algorithm with learning rate $\eta = \sqrt{\frac{k + \frac{k(n-k)}{n}}{(4c^2+1)T}}$ and initial parameters $\boldsymbol{\mu}_1 = \boldsymbol{0}$ and $\boldsymbol{D}_1 = \frac{k}{n}\boldsymbol{I}$ is upper bounded by $\sqrt{2(4c^2+1)\left(\frac{n-k}{n} + 1\right) kT}$.*

[3] There are several SDP packages (such as CVX) that are guaranteed to output the value of the SDP up to an additive error of ϵ in time polynomial in the size of the program description and $\log \frac{1}{\epsilon}$.

Proof Let $\boldsymbol{W} = \begin{bmatrix}\boldsymbol{\mu} & \boldsymbol{D}\end{bmatrix}$ denote the matrix formed by concatenating column vector $\boldsymbol{\mu}$ and matrix $\boldsymbol{D}$. Similarly, let $\boldsymbol{X} = \begin{bmatrix}2c\boldsymbol{x} & \boldsymbol{x}\boldsymbol{x}^\intercal\end{bmatrix}$. With this notation, the expected gain $\operatorname{tr}(\boldsymbol{D}\boldsymbol{x}\boldsymbol{x}^\intercal) + 2c\boldsymbol{\mu}^\intercal\boldsymbol{x} + kc^2$ of parameter $(\boldsymbol{\mu}, \boldsymbol{D})$ on instance $\boldsymbol{x}$ becomes $\operatorname{tr}(\boldsymbol{W}\boldsymbol{X}^\intercal) + kc^2$.

For any offline comparator $\boldsymbol{W}^* = \begin{bmatrix}\sum_{i=1}^k \boldsymbol{u}_i & \sum_{i=1}^k \boldsymbol{u}_i\boldsymbol{u}_i^\intercal\end{bmatrix}$, we have

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

where the inequality follows from the Pythagorean Theorem (Herbster and Warmuth, 2001). Since $\boldsymbol{x}_t$ has unit length, $\|\boldsymbol{X}_t\|_F^2 = \left\|\begin{bmatrix}2c\boldsymbol{x}_t & \boldsymbol{x}_t\boldsymbol{x}_t^\intercal\end{bmatrix}\right\|_F^2 = 4c^2\|\boldsymbol{x}_t\|^2 + \|\boldsymbol{x}_t\boldsymbol{x}_t^\intercal\|_F^2 = 4c^2 + 1$. By rearranging terms, we have

$$\operatorname{tr}(\boldsymbol{W}^*\boldsymbol{X}_t^\intercal) - \operatorname{tr}(\boldsymbol{W}_t\boldsymbol{X}_t^\intercal) \;\le\; \frac{\|\boldsymbol{W}_t - \boldsymbol{W}^*\|_F^2 - \|\boldsymbol{W}_{t+1} - \boldsymbol{W}^*\|_F^2}{2\eta} + \frac{(4c^2+1)\eta}{2}. \tag{3.1}$$

Note that the LHS of (3.1) is the regret in trial t. Summing the inequality over all T trials, we have that the total regret is upper bounded by

$$\frac{\|\boldsymbol{W}_1 - \boldsymbol{W}^*\|_F^2 - \cancel{\|\boldsymbol{W}_{T+1} - \boldsymbol{W}^*\|_F^2}}{2\eta} + \frac{(4c^2+1)\eta T}{2} \le \frac{k + \frac{k(n-k)}{n}}{2\eta} + \frac{(4c^2+1)\eta T}{2},$$

since $\|\boldsymbol{W}_1 - \boldsymbol{W}^*\|_F^2 = \left\|\begin{bmatrix}\mathbf{0} \; \frac{k}{n}\boldsymbol{I}\end{bmatrix} - \begin{bmatrix}\sum_i \boldsymbol{u}_i \; \sum_i \boldsymbol{u}_i\boldsymbol{u}_i^\intercal\end{bmatrix}\right\|_F^2$ is by the rotation invariance of $\|.\|_F^2$ equal to $\left\|\begin{bmatrix}\mathbf{0} \; \frac{k}{n}\boldsymbol{I}\end{bmatrix} - \begin{bmatrix}\mathbf{1}_k \; \boldsymbol{I}_k\end{bmatrix}\right\|_F^2 = \frac{k(n-k)}{n} + k$. Choosing $\eta = \sqrt{\frac{k + \frac{k(n-k)}{n}}{(4c^2+1)T}}$ proves the theorem. ■

We now reason that the above regret bound for GD (expressed as a function of the time horizon) cannot be improved by more than a constant factor. We first consider the original online PCA problem, where $c = 0$. In this case our regret bound for GD becomes $\sqrt{2\left(\frac{n-k}{n} + 1\right)kT}$ and a matching lower bound (up to a constant factor) was shown in Nie et al. (2013).[4] For the directional case $c \neq 0$, we prove the following matching lower bound in Appendix C.

Theorem 3.2 *The minimax regret of the T-round directional gain game with constant $c \neq 0$ and orthonormal sets of size k is $\Omega(\sqrt{c^2kT})$.*

For linear losses, all known regret bound for the GD update grow with the time horizon (including the bound we just proved). Slightly better bounds can be proven for the MEG update (not shown), where the time horizon is replaced by an upper bound on the offline comparator's gain.

4. The decomposition

In this section we decompose any parameter $(\boldsymbol{\mu}, \boldsymbol{D})$ satisfying (2.1), that is, we write it as a convex combination of (first and second moments of) sets of k orthogonal directions.

[4] One can show that the original MEG algorithm for online PCA given in Warmuth and Kuzmin (2008) also achieves the optimal regret bound as a function of the time horizon (up to a constant factor) (See discussion in Nie et al. (2013)).

Our algorithm is a greedy iterative removal scheme, like the decomposition algorithms for sets and subspaces (Warmuth and Kuzmin, 2008), permutations (Helmbold and Warmuth, 2009), paths and trees (Koolen et al., 2010).

Note that the condition $\boldsymbol{\mu}\boldsymbol{\mu}^\intercal/k \preceq \boldsymbol{D}$ of Theorem 2.1 is equivalent to the following, where $\boldsymbol{D}^\dagger$ denotes the pseudo-inverse: $\boldsymbol{D} \succeq \boldsymbol{0}$, $\boldsymbol{\mu}\boldsymbol{D}^\dagger\boldsymbol{\mu} \leq k$ and $\boldsymbol{\mu} \in \operatorname{range}(\boldsymbol{D})$(see e.g. Bernstein (2011, Proposition 8.2.4)). It will be convenient to assume that the mean is extreme, i.e. $\boldsymbol{\mu}^\intercal\boldsymbol{D}^\dagger\boldsymbol{\mu} = k$. If instead $\boldsymbol{\mu}^\intercal\boldsymbol{D}^\dagger\boldsymbol{\mu} < k$ we may decompose by mixing the two decompositions[5] of the extreme opposites $\left(\pm\boldsymbol{\mu}\sqrt{\frac{k}{\boldsymbol{\mu}^\intercal\boldsymbol{D}^\dagger\boldsymbol{\mu}}}, \boldsymbol{D}\right)$ with probabilities $\frac{k\pm\boldsymbol{\mu}^\intercal\boldsymbol{D}^\dagger\boldsymbol{\mu}}{2k}$. (If the mean $\boldsymbol{\mu}$ is zero we may choose any pair of opposites in the range of $\boldsymbol{D}$.) So we henceforth assume that

$$\operatorname{tr}(\boldsymbol{D}) \;=\; k, \qquad \boldsymbol{0} \;\preceq\; \boldsymbol{D} \;\preceq\; \boldsymbol{I}, \qquad \boldsymbol{\mu} \in \operatorname{range}(\boldsymbol{D}) \qquad \text{and} \qquad \boldsymbol{\mu}^\intercal\boldsymbol{D}^\dagger\boldsymbol{\mu} \;=\; k. \tag{4.1}$$

This equation implies that the eigenvalues of $\boldsymbol{D}$ lie in $[0,1]$. We proceed by recursion on

$$\chi(\boldsymbol{D}) \;\coloneqq\; \text{the number of eigenvalues of } \boldsymbol{D} \text{ in } (0,1).$$

In the base case $\chi(\boldsymbol{D}) = 0$ all eigenvalues of $\boldsymbol{D}$ are either 0 or 1, and since $\operatorname{tr}(\boldsymbol{D}) = k$ there must be k ones and $n-k$ zeroes. In particular this means that $\boldsymbol{\mu}^\intercal\boldsymbol{\mu} = k$. To obtain an orthonormal set with mean $\boldsymbol{\mu}$ and second moment $\boldsymbol{D}$, we may choose $\boldsymbol{U}$ to be any orthonormal basis spanning the range of $\boldsymbol{D}$ with sum equal to $\boldsymbol{\mu}$.

If $\chi(\boldsymbol{D}) > 0$ we find an orthonormal set $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ (with moments $(\sum_{i=1}^k \boldsymbol{u}_i, \sum_{i=1}^k \boldsymbol{u}_i\boldsymbol{u}_i^\intercal)$ that are abbreviated as $(\boldsymbol{s}, \boldsymbol{S})$ throughout), a probability $\rho \in (0,1)$, and decompose

$$(\boldsymbol{\mu}, \boldsymbol{D}) \;=\; \rho\,(\boldsymbol{s}, \boldsymbol{S}) + (1-\rho)\,(\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{D}}),$$

where the normalized remainder $(\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{D}}) \coloneqq \left(\frac{\boldsymbol{\mu}-\rho\boldsymbol{s}}{1-\rho}, \frac{\boldsymbol{D}-\rho\boldsymbol{S}}{1-\rho}\right)$ again satisfies (4.1) so that it can be decomposed recursively and moreover $\chi(\widetilde{\boldsymbol{D}}) < \chi(\boldsymbol{D})$. This recursive process must therefore terminate in at most $n+1$ steps.

A similar but simpler recursive process is used in the original online PCA problem (where $c = 0$) (Warmuth and Kuzmin, 2008). In this case, the learner only needs to decompose the parameter matrix $\boldsymbol{D}$ into a small mixture of orthonormal sets of size k. These orthonormal sets can always be chosen as subsets of the eigenvectors of $\boldsymbol{D}$. In the general case (when $c \neq 0$), the sets need to simultaneously decompose the mean parameter $\boldsymbol{\mu}$, and the additional constraints this imposes are not generally satisfied by the eigenvectors of $\boldsymbol{D}$.

The rest of this section will be concerned with finding the set $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]$ and the probability ρ and proving that $\chi(\widetilde{\boldsymbol{D}}) < \chi(\boldsymbol{D})$. First in Theorem 4.2 we prove that Algorithm 1 will find an orthonormal set of k so-called *tangent* directions. We call a direction $\boldsymbol{u}$ *tangent* to $(\boldsymbol{\mu}, \boldsymbol{D})$ if $\boldsymbol{u}^\intercal\boldsymbol{D}^\dagger\boldsymbol{\mu} = 1$. Then in Lemma 4.3 we show that splitting off a tangent set $\boldsymbol{U}$ preserves (4.1). Finally in Theorem 4.6 we show that the probability $\rho \in (0,1)$ can be found, and that $\chi(\widetilde{\boldsymbol{D}}) < \chi(\boldsymbol{D})$.

4.1. Finding a tangent set

In this section we present Algorithm 1 for finding a tangent set. The algorithm will make use of the following simple lemma.

[5] Each decomposition will be of size $n+1$, for a total of $2(n+1)$.

Lemma 4.1 *A linear equation $\boldsymbol{v}^\intercal \boldsymbol{x} = a$ of dimension at least 2 has a solution for $\boldsymbol{x}$ of unit length if $\|\boldsymbol{v}\| \geq |a|$.*

Proof Let $\boldsymbol{v}^\perp$ be a unit vector perpendicular to $\boldsymbol{v}$. If $\|\boldsymbol{v}\| = a = 0$, return $\boldsymbol{v}^\perp$. Otherwise $\frac{a}{\|\boldsymbol{v}\|^2}\boldsymbol{v} + \sqrt{1 - \frac{a^2}{\|\boldsymbol{v}\|^2}}\boldsymbol{v}^\perp$ is a unit length solution. ■

We are now ready to show that the algorithm indeed produces a tangent set.

Theorem 4.2 *Let $\boldsymbol{\mu}$ and $\boldsymbol{D}$ satisfy (4.1). Let $[\boldsymbol{A}\ \boldsymbol{B}\ \boldsymbol{C}]$ be an orthonormal eigenbasis for $\boldsymbol{D}$, with $\boldsymbol{A}$ associated to the eigenvalue 1, $\boldsymbol{C}$ to eigenvalue 0 and $\boldsymbol{B}$ to the remaining intermediate eigenvalues. (Any of them can be empty). Then Algorithm 1 applied to $(\boldsymbol{\mu}, \boldsymbol{D})$ produces a set $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k]$ of k orthonormal vectors with moments $(\boldsymbol{s}, \boldsymbol{S})$ such that*

$$\boldsymbol{U}^\intercal \boldsymbol{D}^\dagger \boldsymbol{\mu} = \boldsymbol{1}_k \qquad \boldsymbol{U} \text{ is a tangent set} \tag{4.2a}$$

$$\boldsymbol{S} = \boldsymbol{D}\boldsymbol{D}^\dagger \boldsymbol{S} \qquad \boldsymbol{U} \text{ avoids the } 0 \text{ eigenspace of } \boldsymbol{D} \tag{4.2b}$$

$$\boldsymbol{I} - \boldsymbol{S} = (\boldsymbol{I} - \boldsymbol{D})(\boldsymbol{I} - \boldsymbol{D})^\dagger(\boldsymbol{I} - \boldsymbol{S}) \qquad \boldsymbol{U} \text{ contains the } 1 \text{ eigenspace of } \boldsymbol{D} \tag{4.2c}$$

The algorithm can be implemented in time $O(kn^2)$ when $\boldsymbol{C}\boldsymbol{C}^\intercal$ is precomputed.

Proof We first show that $\widehat{\boldsymbol{A}}$ consists of k orthonormal vectors and that $\|\widehat{\boldsymbol{A}}^\intercal \boldsymbol{D}^\dagger \boldsymbol{\mu}\|^2 = k$. When $\operatorname{rank}(\boldsymbol{A}) = k$, since $\boldsymbol{I} \succeq \boldsymbol{D}$ and $\operatorname{tr}(\boldsymbol{D}) = k$, $\boldsymbol{B}$ is empty and $\boldsymbol{D} = \boldsymbol{D}^\dagger = \boldsymbol{A}\boldsymbol{A}^\intercal$.

$$\|\boldsymbol{A}^\intercal \boldsymbol{D}^\dagger \boldsymbol{\mu}\|^2 = \boldsymbol{\mu}^\intercal \boldsymbol{D}^\dagger \boldsymbol{A}\boldsymbol{A}^\intercal \boldsymbol{D}^\dagger \boldsymbol{\mu} = \boldsymbol{\mu}^\intercal \boldsymbol{D}^\dagger \boldsymbol{\mu} = k.$$

When $\operatorname{rank}(\boldsymbol{A}) < k$, $\boldsymbol{D}$ can be eigendecomposed as $\boldsymbol{D} = \boldsymbol{A}\boldsymbol{A}^\intercal + \boldsymbol{B}\widehat{\boldsymbol{D}}\boldsymbol{B}^\intercal$ where $\widehat{\boldsymbol{D}}$ is a diagonal matrix and $\boldsymbol{0} \prec \widehat{\boldsymbol{D}} \prec \boldsymbol{I}$. We rewrite $\boldsymbol{D}^\dagger$, $\boldsymbol{v}_{\boldsymbol{A}}$ and $\boldsymbol{v}_{\boldsymbol{B}}$ with the decomposition as:

$$\boldsymbol{D}^\dagger = \boldsymbol{A}\boldsymbol{A}^\intercal + \boldsymbol{B}\widehat{\boldsymbol{D}}^\dagger \boldsymbol{B}^\intercal, \qquad \boldsymbol{v}_A = \boldsymbol{A}\boldsymbol{A}^\intercal \boldsymbol{D}^\dagger \boldsymbol{\mu} = \boldsymbol{A}\boldsymbol{A}^\intercal \boldsymbol{\mu}, \qquad \boldsymbol{v}_B = \boldsymbol{B}\boldsymbol{B}^\intercal \boldsymbol{D}^\dagger \boldsymbol{\mu} = \boldsymbol{B}\widehat{\boldsymbol{D}}^\dagger \boldsymbol{B}^\intercal \boldsymbol{\mu}.$$

Now we show that the conditions for using Lemma 4.1 to compute $\hat{\boldsymbol{v}}$ are met.

- $\operatorname{rank}(\boldsymbol{B}) = \underbrace{\operatorname{rank}(\boldsymbol{A}) + \operatorname{rank}(\boldsymbol{B})}_{>k} - \underbrace{\operatorname{rank}(\boldsymbol{A})}_{<k} \geq 2$. The lower bound on $\operatorname{rank}(\boldsymbol{A}) + \operatorname{rank}(\boldsymbol{B})$ follows from

$$\underbrace{\operatorname{rank}(\boldsymbol{A}) + \operatorname{rank}(\boldsymbol{B}) = \overbrace{\operatorname{tr}(\boldsymbol{A}\boldsymbol{A}^\intercal + \boldsymbol{B}\boldsymbol{B}^\intercal)}_{}}_{\boldsymbol{A} \text{ and } \boldsymbol{B} \text{ consist of orthonormal vectors}} \overbrace{\vphantom{)} > \operatorname{tr}(A\boldsymbol{A}^\intercal + \boldsymbol{B}\widehat{\boldsymbol{D}}\boldsymbol{B}^\intercal)}^{\boldsymbol{I} \succ \widehat{\boldsymbol{D}}} = k.$$

- To show $k \geq \|\boldsymbol{v}_A\|^2$, notice that

$$k = \boldsymbol{\mu}^\intercal \boldsymbol{D}^\dagger \boldsymbol{\mu} = \boldsymbol{\mu}^\intercal(\boldsymbol{A}\boldsymbol{A}^\intercal + \boldsymbol{B}\widehat{\boldsymbol{D}}^\dagger \boldsymbol{B}^\intercal)\boldsymbol{\mu}^\intercal = \underbrace{\boldsymbol{\mu}^\intercal \boldsymbol{A}\boldsymbol{A}^\intercal \boldsymbol{A}\boldsymbol{A}^\intercal \boldsymbol{\mu}}_{\|\boldsymbol{v}_A\|^2} + \underbrace{\boldsymbol{\mu}^\intercal \boldsymbol{B}\widehat{\boldsymbol{D}}^\dagger \boldsymbol{B}^\intercal \boldsymbol{\mu}}_{\geq 0}$$

- Finally $\|\boldsymbol{v}_B\| \geq \sqrt{k - \|\boldsymbol{v}_A\|^2}$ follows from $(\widehat{\boldsymbol{D}}^\dagger)^2 \succeq \widehat{\boldsymbol{D}}^\dagger$ and

$$\|\boldsymbol{v}_B\|^2 = \boldsymbol{\mu}^\intercal \boldsymbol{B}(\widehat{\boldsymbol{D}}^\dagger)^2 \boldsymbol{B}^\intercal \boldsymbol{\mu} \geq \boldsymbol{\mu}^\intercal \boldsymbol{B}\widehat{\boldsymbol{D}}^\dagger \boldsymbol{B}^\intercal \boldsymbol{\mu} = k - \|v_A\|^2.$$

The next step is to show that finding $\widehat{\boldsymbol{B}}$ is always possible. This follows simply from $k - \operatorname{rank}(\boldsymbol{A}) - 1 \leq (\operatorname{rank}(\boldsymbol{A}) + \operatorname{rank}(\boldsymbol{B}) - 1) - \operatorname{rank}(\boldsymbol{A}) - 1 = \operatorname{rank}(\boldsymbol{B}) - 2$. Since $\boldsymbol{A}$, $\hat{\boldsymbol{v}}$ and $\widehat{\boldsymbol{B}}$ are orthogonal to each other and $\widehat{\boldsymbol{B}}$ is also orthogonal to $\boldsymbol{D}^\dagger\boldsymbol{\mu}$, $\|\widehat{\boldsymbol{A}}^\top\boldsymbol{D}^\dagger\boldsymbol{\mu}\|^2 = \|\boldsymbol{v}_A\|^2 + (\hat{\boldsymbol{v}}^\top\boldsymbol{D}^\dagger\boldsymbol{\mu})^2 = k$.

So in both cases, $\widehat{\boldsymbol{A}}^\top\boldsymbol{D}^\dagger\boldsymbol{\mu}$ is a vector in $\mathbb{R}^k$ with length $\sqrt{k}$. By a rotation matrix $\widehat{\boldsymbol{U}}$ in $\mathbb{R}^{k\times k}$, we can rotate $\widehat{\boldsymbol{A}}^\top\boldsymbol{D}^\dagger\boldsymbol{\mu}$ to a vector of the same length, $\mathbf{1}_k$(see e.g. Hazan et al. (2011)). As a result, $\boldsymbol{U}^\top\boldsymbol{D}^\dagger\boldsymbol{\mu} = \widehat{\boldsymbol{U}}\widehat{\boldsymbol{A}}^\top\boldsymbol{D}\boldsymbol{\mu} = \mathbf{1}_k$ and $\boldsymbol{U}^\top\boldsymbol{U} = \widehat{\boldsymbol{U}}\widehat{\boldsymbol{A}}^\top\widehat{\boldsymbol{A}}\widehat{\boldsymbol{U}} = \boldsymbol{I}_k$.

Finally, noticing that by construction of $\widehat{\boldsymbol{A}}$, $\operatorname{range}(\boldsymbol{U}) \in \operatorname{range}(\boldsymbol{D})$, $\boldsymbol{U} = \boldsymbol{D}\boldsymbol{D}^\dagger\boldsymbol{U}$. Also,

$$\boldsymbol{I} - \boldsymbol{S} = \boldsymbol{I} - \boldsymbol{A}\boldsymbol{A}^\top - \hat{\boldsymbol{v}}\hat{\boldsymbol{v}}^\top - \widehat{\boldsymbol{B}}\widehat{\boldsymbol{B}}^\top = \boldsymbol{B}\boldsymbol{B} + \boldsymbol{C}\boldsymbol{C} - \hat{\boldsymbol{v}}\hat{\boldsymbol{v}}^\top - \widehat{\boldsymbol{B}}\widehat{\boldsymbol{B}}^\top$$

means $\operatorname{range}(\boldsymbol{I} - \boldsymbol{S}) \in \operatorname{range}(\boldsymbol{B}\boldsymbol{B}^\top + \boldsymbol{C}\boldsymbol{C}^\top) = \operatorname{range}((\boldsymbol{I} - \boldsymbol{D})(\boldsymbol{I} - \boldsymbol{D})^\dagger)$ as required.

Now we show how to implement the algorithm in $O(kn^2)$ with precomputed $\boldsymbol{C}\boldsymbol{C}^\top$. First computing $\boldsymbol{A}$ can be done in $O(kn^2)$ time. Noticing that $\boldsymbol{A}\boldsymbol{A}^\top + \boldsymbol{B}\boldsymbol{B}^\top + \boldsymbol{C}\boldsymbol{C}^\top = \boldsymbol{I}$, we obtain $\boldsymbol{B}\boldsymbol{B}^\top$ in $O(n^2)$. Using columns of $\boldsymbol{B}\boldsymbol{B}^\top$ as a basis of $\boldsymbol{B}$, $\widehat{\boldsymbol{B}}$ can be computed in (k^2n) with a Gram-Schmidt process. Finally, computing a rotation matrix in $\mathbb{SO}(k)$ needs time $O(k^2)$ and computing $\widehat{\boldsymbol{A}}\widehat{\boldsymbol{U}}$ needs time $O(kn^2)$. ■

```
input  : parameter $(\boldsymbol{\mu}, \boldsymbol{D})$ satisfying (4.1)
output: orthonormal $k$-set $\boldsymbol{U}$ satisfying (4.2)

Compute orthonormal eigenbasis $\boldsymbol{A}$ and $\boldsymbol{B}$ of $\boldsymbol{D}$ as described in Theorem 4.2
if $\operatorname{rank}(\boldsymbol{A}) = k$ then
    $\widehat{\boldsymbol{A}} = \boldsymbol{A}$
else
    $\boldsymbol{v}_A = \boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{D}^\dagger\boldsymbol{\mu}$                                   // Project $\boldsymbol{D}^\dagger\boldsymbol{\mu}$ on $\boldsymbol{A}$
    $\boldsymbol{v}_B = \boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{D}^\dagger\boldsymbol{\mu}$                                   // Project $\boldsymbol{D}^\dagger\boldsymbol{\mu}$ on $\boldsymbol{B}$
    Compute a unit vector $\hat{\boldsymbol{v}}$ in $\boldsymbol{B}$ satisfying $\hat{\boldsymbol{v}}^\top\boldsymbol{v}_B = \sqrt{k - \|\boldsymbol{v}_A\|^2}$ via Lemma 4.1
    Pick $k - \operatorname{rank}(\boldsymbol{A}) - 1$ orthonormal basis $\widehat{\boldsymbol{B}}$ from the complementary of $\boldsymbol{v}_B$ and $\hat{\boldsymbol{v}}$ in $\boldsymbol{B}$
    $\widehat{\boldsymbol{A}} = \begin{bmatrix}\boldsymbol{A} & \hat{\boldsymbol{v}} & \widehat{\boldsymbol{B}}\end{bmatrix}$
end
Compute a rotation matrix $\widehat{\boldsymbol{U}} \in \mathbb{SO}(k)$ which rotates $\widehat{\boldsymbol{A}}^\top\boldsymbol{D}^\dagger\boldsymbol{\mu}$ to $\mathbf{1}_k$
return $\boldsymbol{U} = \widehat{\boldsymbol{A}}\widehat{\boldsymbol{U}}^\top$
```

Algorithm 1: Find a removable set $\boldsymbol{U}$

4.2. Removing a tangent set preserves the mean constraints

At this point we have a tangent set $\boldsymbol{U}$ to split off. We now show that the remainder $(\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{D}})$ satisfies (4.1). We start with the rightmost two conditions, which will be satisfied for any weight $\rho > 0$. Lemma 4.3 covers a single tangent vector, whereas Lemma 4.4 covers sets.

Lemma 4.3 *Fix a matrix $\boldsymbol{D} \in \mathbb{R}^{n\times n}$, vectors $\boldsymbol{\mu}, \boldsymbol{u} \in \operatorname{range}(\boldsymbol{D})$ with $\boldsymbol{u}\boldsymbol{D}^\dagger\boldsymbol{\mu} = 1$ and a weight $\rho \in \mathbb{R}$. Define $\widetilde{\boldsymbol{D}} \coloneqq \boldsymbol{D} - \rho\boldsymbol{u}\boldsymbol{u}^\top$ and $\widetilde{\boldsymbol{\mu}} \coloneqq \boldsymbol{\mu} - \rho\boldsymbol{u}$. Then*

$$\widetilde{\boldsymbol{\mu}}^\top\widetilde{\boldsymbol{D}}^\dagger\widetilde{\boldsymbol{\mu}} = \boldsymbol{\mu}^\top\boldsymbol{D}^\dagger\boldsymbol{\mu} - \rho \quad \text{and} \quad \widetilde{\boldsymbol{\mu}} \in \operatorname{range}(\widetilde{\boldsymbol{D}}).$$

If $\operatorname{rank}(\widetilde{\boldsymbol{D}}) = \operatorname{rank}(\boldsymbol{D})$ *then* $\widetilde{\boldsymbol{D}}^\dagger\widetilde{\boldsymbol{\mu}} = \boldsymbol{D}^\dagger\boldsymbol{\mu}$*. Otherwise there is a real number* α *such that*

$$\widetilde{\boldsymbol{D}}^\dagger\widetilde{\boldsymbol{\mu}} = \boldsymbol{D}^\dagger\boldsymbol{\mu} + \alpha\boldsymbol{D}^\dagger\boldsymbol{u} \quad \textit{and} \quad \widetilde{\boldsymbol{D}}\boldsymbol{D}^\dagger\boldsymbol{u} = 0.$$

Proof First notice that $\operatorname{rank}(\boldsymbol{D}) - 1 \le \operatorname{rank}(\widetilde{\boldsymbol{D}}) \le \operatorname{rank}(\boldsymbol{D})$, since $\boldsymbol{u} \in \mathbb{R}^n$ and $\rho\boldsymbol{u}^\intercal\boldsymbol{u}$ is a rank one modification. So $\operatorname{rank}(\widetilde{\boldsymbol{D}})$ equals either $\operatorname{rank}(\boldsymbol{D})$ or $\operatorname{rank}(\boldsymbol{D}) - 1$. We cover these two cases separately. In the first case when $\operatorname{rank}(\widetilde{\boldsymbol{D}}) = \operatorname{rank}(\boldsymbol{D})$ we have

$$\widetilde{\boldsymbol{D}}^\dagger\widetilde{\boldsymbol{\mu}} \;=\; (\boldsymbol{D} - \rho\boldsymbol{u}\boldsymbol{u}^\intercal)^\dagger(\boldsymbol{\mu} - \rho\boldsymbol{u}) \;=\; \left(\boldsymbol{D}^\dagger + \rho\frac{\boldsymbol{D}^\dagger\boldsymbol{u}\boldsymbol{u}^\intercal\boldsymbol{D}^\dagger}{1 - \rho\boldsymbol{u}^\intercal\boldsymbol{D}^\dagger\boldsymbol{u}}\right)(\boldsymbol{\mu} - \rho\boldsymbol{u}) \;=\; \boldsymbol{D}^\dagger\boldsymbol{\mu}$$

by Bernstein (2011, Fact 6.4.2). And so $\widetilde{\boldsymbol{\mu}}\widetilde{\boldsymbol{D}}^\dagger\widetilde{\boldsymbol{\mu}} = (\boldsymbol{\mu} - \rho\boldsymbol{u})^\intercal\boldsymbol{D}^\dagger\boldsymbol{\mu} = \boldsymbol{\mu}^\intercal\boldsymbol{D}^\dagger\boldsymbol{u} - \rho$. Also in this case $\widetilde{\boldsymbol{\mu}} \in \operatorname{range}(\boldsymbol{D}) = \operatorname{range}(\widetilde{\boldsymbol{D}})$.

In the second case $\operatorname{rank}(\widetilde{\boldsymbol{D}}) = \operatorname{rank}(\boldsymbol{D}) - 1$ or equivalently $\rho\boldsymbol{u}^\intercal\boldsymbol{D}^\dagger\boldsymbol{u} = 1$. We first show $\boldsymbol{D}^\dagger\boldsymbol{u}$ is a null vector of $\widetilde{\boldsymbol{D}}$.

$$\widetilde{\boldsymbol{D}}\boldsymbol{D}^\dagger\boldsymbol{u} \;=\; (\boldsymbol{D} - \rho\boldsymbol{u}\boldsymbol{u}^\intercal)\boldsymbol{D}^\dagger\boldsymbol{u} \;=\; \boldsymbol{D}\boldsymbol{D}^\dagger\boldsymbol{u} - \boldsymbol{u}\rho\boldsymbol{u}^\intercal\boldsymbol{D}^\dagger\boldsymbol{u} \;=\; \boldsymbol{u} - \boldsymbol{u} = 0$$

Notice that $\boldsymbol{D}\boldsymbol{D}^\dagger\boldsymbol{u} = \boldsymbol{u} \neq 0$, so $\operatorname{range}(\widetilde{\boldsymbol{D}})$ is exactly the complementary space of $\boldsymbol{D}^\dagger\boldsymbol{u}$ in $\operatorname{range}(\boldsymbol{D})$. This implies $\widetilde{\boldsymbol{\mu}} \in \operatorname{range}(\widetilde{\boldsymbol{D}})$ since $\boldsymbol{D}^\dagger\boldsymbol{u}$ is also null to $\widetilde{\boldsymbol{\mu}}$:

$$\widetilde{\boldsymbol{\mu}}^\intercal\boldsymbol{D}^\dagger\boldsymbol{u} = (\boldsymbol{\mu} - \rho\boldsymbol{u})^\intercal\boldsymbol{D}^\dagger\boldsymbol{u} = \boldsymbol{\mu}^\intercal\boldsymbol{D}^\dagger\boldsymbol{u} - \rho\boldsymbol{u}^\intercal\boldsymbol{D}^\dagger\boldsymbol{u} = 1 - 1 = 0$$

We now use Bernstein (2011, Fact 6.4.2) to rewrite $\widetilde{\boldsymbol{D}}^\dagger$ (α_i are unimportant scalars)

$$\begin{aligned}
\widetilde{\boldsymbol{D}}^\dagger\widetilde{\boldsymbol{\mu}} =& [\boldsymbol{D}^\dagger + \alpha_1\boldsymbol{D}^\dagger\boldsymbol{u}\boldsymbol{u}^\intercal(\boldsymbol{D}^\dagger)^2 + \underbrace{\alpha_2(\boldsymbol{D}^\dagger)^2\boldsymbol{u}\boldsymbol{u}^\intercal\boldsymbol{D}^\dagger + \alpha_3\boldsymbol{D}^\dagger\boldsymbol{u}\boldsymbol{u}^\intercal\boldsymbol{D}^\dagger}_{\text{(become 0 after distribution)}}]\widetilde{\boldsymbol{\mu}} \\
=& [\boldsymbol{D}^\dagger + \alpha_1\boldsymbol{D}^\dagger\boldsymbol{u}\boldsymbol{u}^\intercal(\boldsymbol{D}^\dagger)^2](\boldsymbol{\mu} - \rho\boldsymbol{u}) \\
=& \boldsymbol{D}^\dagger\boldsymbol{\mu} - \rho\boldsymbol{D}^\dagger\boldsymbol{u} + \alpha_1\boldsymbol{D}^\dagger\boldsymbol{u}\underbrace{\boldsymbol{u}^\intercal(\boldsymbol{D}^\dagger)^2(\boldsymbol{\mu} - \rho\boldsymbol{u})}_{\text{a number}} = \boldsymbol{D}^\dagger\boldsymbol{\mu} + \alpha\boldsymbol{D}^\dagger\boldsymbol{u}.
\end{aligned}$$

The last thing to show is $\widetilde{\boldsymbol{\mu}}\widetilde{\boldsymbol{D}}^\dagger\widetilde{\boldsymbol{\mu}} = \boldsymbol{\mu}\boldsymbol{D}^\dagger\boldsymbol{\mu} - \rho$ which follows by

$$\widetilde{\boldsymbol{\mu}}\widetilde{\boldsymbol{D}}^\dagger\widetilde{\boldsymbol{\mu}} = \widetilde{\boldsymbol{\mu}}^\intercal(\boldsymbol{D}^\dagger\boldsymbol{\mu} + \alpha\boldsymbol{D}^\dagger\boldsymbol{u}) = (\boldsymbol{\mu} - \rho\boldsymbol{u})^\intercal\boldsymbol{D}^\dagger\boldsymbol{\mu} = \boldsymbol{\mu}^\intercal\boldsymbol{D}^\dagger\boldsymbol{\mu} - \rho.$$ ■

The previous lemma covered single tangent vectors. Next we take out a full tangent set.

Lemma 4.4 *Let* $\boldsymbol{\mu}, \boldsymbol{D}$ *satisfy* (4.1)*, and let the orthonormal set* $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ *(with moments* $(\boldsymbol{s}, \boldsymbol{S})$*) be tangent. Then for any* $\rho > 0$ *if* $\boldsymbol{D} \succeq \rho\boldsymbol{S}$*, we have*

$$(\boldsymbol{\mu} - \rho\boldsymbol{s})^\intercal(\boldsymbol{D} - \rho\boldsymbol{S})(\boldsymbol{\mu} - \rho\boldsymbol{s}) \;=\; \boldsymbol{\mu}^\intercal\boldsymbol{D}\boldsymbol{\mu} - k\rho \qquad \textit{and} \qquad \boldsymbol{\mu} - \rho\boldsymbol{s} \in \operatorname{range}(\boldsymbol{D} - \rho\boldsymbol{S}).$$

Proof For $1 \le d \le k$, define the intermediate remainder as $\widetilde{\boldsymbol{\mu}}_d \coloneqq \boldsymbol{\mu} - \rho\sum_{i=1}^d \boldsymbol{u}_i$ and $\widetilde{\boldsymbol{D}}_d \coloneqq \boldsymbol{D} - \rho\sum_{i=1}^d \boldsymbol{u}_i\boldsymbol{u}_i^\intercal$. Also $\widetilde{\boldsymbol{D}}_0 = \boldsymbol{D}$ and $\widetilde{\boldsymbol{\mu}}_0 = \boldsymbol{\mu}$. We show by induction that $\boldsymbol{u}_i$ remains tangent to $(\widetilde{\boldsymbol{\mu}}_d, \widetilde{\boldsymbol{D}}_d)$ for $d < i \le k$ and

$$\widetilde{\boldsymbol{\mu}}_d^\intercal\widetilde{\boldsymbol{D}}_d\widetilde{\boldsymbol{\mu}}_d = \boldsymbol{\mu}^\intercal\boldsymbol{D}\boldsymbol{\mu} - d\rho \qquad \text{and} \qquad \widetilde{\boldsymbol{\mu}}_d \in \operatorname{range}(\widetilde{\boldsymbol{D}}_d).$$

The base case $d = 0$ is trivial. Let us, to simplify notation, show the induction step for $d = 1$. The last two claims follow directly from Lemma 4.3. We now show that for $2 \leq i \leq k$, $\boldsymbol{u}_i$ is also tangent to $\widetilde{\boldsymbol{\mu}}_1$ and $\widetilde{\boldsymbol{D}}_1$. When $\operatorname{rank}(\widetilde{\boldsymbol{D}}_1) = \operatorname{rank}(\boldsymbol{D})$ we have $\boldsymbol{u}_i^\intercal \widetilde{\boldsymbol{D}}_1^\dagger \widetilde{\boldsymbol{\mu}}_1 = \boldsymbol{u}_i^\intercal \boldsymbol{D} \boldsymbol{\mu} = 1$ as required. When $\operatorname{rank}(\widetilde{\boldsymbol{D}}_1) = \operatorname{rank}(\boldsymbol{D}) - 1$,

$$\boldsymbol{u}_i^\intercal \widetilde{\boldsymbol{D}}_1^\dagger \widetilde{\boldsymbol{\mu}}_1 = \boldsymbol{u}_i^\intercal \boldsymbol{D}^\dagger \boldsymbol{\mu} - \alpha \boldsymbol{u}_i^\intercal \boldsymbol{D}^\dagger \boldsymbol{u}_1 = 1 - \alpha \boldsymbol{u}_i^\intercal \boldsymbol{D}^\dagger \boldsymbol{u}_1.$$

Note that $\boldsymbol{u}_i^\intercal \boldsymbol{D}^\dagger \boldsymbol{u}_1 = 0$, for otherwise, $(\boldsymbol{D}^\dagger \boldsymbol{u}_1)^\intercal (\widetilde{\boldsymbol{D}}_1 - \rho \boldsymbol{u}_i \boldsymbol{u}_i^\intercal)(\boldsymbol{D}^\dagger \boldsymbol{u}_1) = -\rho(\boldsymbol{u}_i^\intercal \boldsymbol{D}^\dagger \boldsymbol{u}_1)^2 < 0$, which contradicts $\widetilde{\boldsymbol{D}}_1 - \rho \boldsymbol{u}_i \boldsymbol{u}_i^\intercal \succeq \widetilde{\boldsymbol{D}}_k \succeq \boldsymbol{0}$. This also implies $\boldsymbol{u}_i \in \operatorname{range}(\widetilde{\boldsymbol{D}}_1)$ which means $\boldsymbol{u}_i$ is tangent to $\widetilde{\boldsymbol{\mu}}_1$ and $\widetilde{\boldsymbol{D}}_1$. ■

4.3. Choosing the weight ρ

We know that taking out a tangent set $\boldsymbol{U}$ preserves the rightmost two constraints of (4.1) on the remainder for any weight ρ. To satisfy the leftmost two, we investigate how semi-definiteness and rank of $\widetilde{\boldsymbol{D}}$ are related to ρ.

Lemma 4.5 *Let $\boldsymbol{D}, \boldsymbol{S} \in \mathbb{R}^{n \times n}$ be non-zero positive semi-definite matrices with $\boldsymbol{S} = \boldsymbol{D}\boldsymbol{D}^\dagger \boldsymbol{S}$. Define $\rho_s := \frac{1}{\lambda_{\max}(\boldsymbol{D}^\dagger \boldsymbol{S})}$ where $\lambda_{\max}(\boldsymbol{M})$ is the largest eigenvalue of $\boldsymbol{M}$. Then the following hold for $\widetilde{\boldsymbol{D}} = \boldsymbol{D} - \rho \boldsymbol{S}$:*

- $0 < \rho_s < \infty$
- $\widetilde{\boldsymbol{D}} \succeq \boldsymbol{0}$ *for any* $\rho \leq \rho_s$
- $\operatorname{rank}(\widetilde{\boldsymbol{D}}) \leq \operatorname{rank}(\boldsymbol{D})$, *and* $\operatorname{rank}(\widetilde{\boldsymbol{D}}) < \operatorname{rank}(\boldsymbol{D})$ *when* $\rho = \rho_s$.

Proof First notice that $\boldsymbol{S} = \boldsymbol{D}\boldsymbol{D}^\dagger \boldsymbol{S}$ implies both $\boldsymbol{S} \in \operatorname{range}(\boldsymbol{D})$ and $\widetilde{\boldsymbol{D}} = \boldsymbol{D} - \rho \boldsymbol{S} \in \operatorname{range}(\boldsymbol{D})$. So $\operatorname{rank}(\widetilde{\boldsymbol{D}}) \leq \operatorname{rank}(\boldsymbol{D})$. Next, $0 < \rho_s < \infty$ follows from that $\boldsymbol{D}^\dagger \boldsymbol{S}$ is non-zero and positive semi-definite. To show $\widetilde{\boldsymbol{D}} \succeq \boldsymbol{0}$, consider an eigenpair $(\boldsymbol{v}, p)$ of $\widetilde{\boldsymbol{D}}$ where $\boldsymbol{v}$ is a unit vector.

$$\boldsymbol{v}^\intercal \boldsymbol{D}^\dagger \boldsymbol{S} \boldsymbol{v} = \frac{\boldsymbol{v}^\intercal \boldsymbol{D}^\dagger}{\rho}(\boldsymbol{D}\boldsymbol{v} - (\boldsymbol{D} - \rho \boldsymbol{S})\boldsymbol{v}) = \frac{\boldsymbol{v}^\intercal \boldsymbol{D}^\dagger}{\rho}(\boldsymbol{D}\boldsymbol{v} - p\boldsymbol{v}) = \frac{1}{\rho} - \frac{p}{\rho}\boldsymbol{v}^\intercal \boldsymbol{D}^\dagger \boldsymbol{v}.$$

When $\rho \leq \rho_s$, $\frac{1}{\rho} \geq \lambda_{\max}(\boldsymbol{D}^\dagger \boldsymbol{S}) \geq \boldsymbol{v}^\intercal \boldsymbol{D}^\dagger \boldsymbol{S} \boldsymbol{v}$ which implies $p \geq 0$. So $\widetilde{\boldsymbol{D}} \succeq \boldsymbol{0}$.

When $\rho = \rho_s$, let $\boldsymbol{x}$ be a eigenvector of eigenvalue $\frac{1}{\rho_s}$: $\boldsymbol{D}^\dagger \boldsymbol{S} \boldsymbol{x} = \frac{1}{\rho_s}\boldsymbol{x} \neq 0$ and notice that

$$\boldsymbol{D}(\boldsymbol{D}^\dagger \boldsymbol{S} \boldsymbol{x}) = \boldsymbol{S}\boldsymbol{x} \neq 0 \qquad \widetilde{\boldsymbol{D}}(\boldsymbol{D}^\dagger \boldsymbol{S} \boldsymbol{x}) = (\boldsymbol{D} - \rho \boldsymbol{S})\boldsymbol{D}^\dagger \boldsymbol{S} \boldsymbol{x} = \boldsymbol{D}\boldsymbol{D}^\dagger \boldsymbol{S} \boldsymbol{x} - \rho \boldsymbol{S} \frac{1}{\rho} \boldsymbol{x} = 0.$$

So $\widetilde{\boldsymbol{D}}$ has at least one more null dimension than $\boldsymbol{D}$. Together with $\widetilde{\boldsymbol{D}} \in \operatorname{range}(\boldsymbol{D})$, this implies $\operatorname{rank}(\widetilde{\boldsymbol{D}}) < \operatorname{rank}(\boldsymbol{D})$. ■

Finally, we are able to choose ρ to in addition satisfy the leftmost two conditions of (4.1), and reduce the complexity $\chi(\widetilde{\boldsymbol{D}})$.

Theorem 4.6 *Let* $(\boldsymbol{\mu}, \boldsymbol{D})$ *satisfy* (4.1)*. Let* $\boldsymbol{U}$ *be the output of Algorithm* 1*, and let*

$$\rho \;=\; \min\left\{\frac{1}{\lambda_{\max}(\boldsymbol{D}^\dagger \boldsymbol{S})}, \frac{1}{\lambda_{\max}\big((\boldsymbol{I}-\boldsymbol{D})^\dagger(\boldsymbol{I}-\boldsymbol{S})\big)}\right\}.$$

Then the normalized remainder $(\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{D}}) = \big(\frac{\boldsymbol{\mu}-\rho \boldsymbol{s}}{1-\rho}, \frac{\boldsymbol{D}-\rho\boldsymbol{S}}{1-\rho}\big)$ *satisfies* (4.1) *and* $\chi(\widetilde{\boldsymbol{D}}) < \chi(\boldsymbol{D})$*.*

Proof If $\boldsymbol{I} \succeq \boldsymbol{D} \succeq \boldsymbol{0}$, since $\rho \le \frac{1}{\lambda_{\max}(\boldsymbol{D}^\dagger\boldsymbol{S})}$, by Lemma 4.5 $\boldsymbol{D} - \rho\boldsymbol{S} \succeq \boldsymbol{0}$ and so $\widetilde{\boldsymbol{D}} \succeq \boldsymbol{0}$. Also since $(\boldsymbol{I}-\boldsymbol{D}) \succeq \boldsymbol{0}$ and $\rho \le \frac{1}{\lambda_{\max}((\boldsymbol{I}-\boldsymbol{D})^\dagger(\boldsymbol{I}-\boldsymbol{S}))}$, $(\boldsymbol{I}-\boldsymbol{D}) - \rho(\boldsymbol{I}-\boldsymbol{S}) \succeq \boldsymbol{0}$ which is equivalent to $\boldsymbol{I} \succeq \widetilde{\boldsymbol{D}}$. Also

$$\operatorname{tr}(\widetilde{\boldsymbol{D}}) = \frac{\operatorname{tr}(\boldsymbol{D}) - \rho\operatorname{tr}(\boldsymbol{S})}{1-\rho} = \frac{k-\rho k}{1-\rho} = k.$$

Since all $\boldsymbol{u}_1, \dots, \boldsymbol{u}_k$ are tangent, we may apply Lemma 4.4 to show that $\widetilde{\boldsymbol{\mu}}\widetilde{\boldsymbol{D}}^\dagger\widetilde{\boldsymbol{\mu}} = k$ and $\widetilde{\boldsymbol{\mu}} \in \operatorname{range}(\widetilde{\boldsymbol{D}})$. By Lemma 4.5, $\operatorname{rank}(\widetilde{\boldsymbol{D}}) \le \operatorname{rank}(\boldsymbol{D})$ and $\operatorname{rank}(\boldsymbol{I}-\widetilde{\boldsymbol{D}}) \le \operatorname{rank}(\boldsymbol{I}-\boldsymbol{D})$, where at least one inequality is strict since ρ equals the minimum of $\frac{1}{\lambda_{\max}(\boldsymbol{D}^\dagger\boldsymbol{S})}$ and $\frac{1}{\lambda_{\max}((\boldsymbol{I}-\boldsymbol{D})^\dagger(\boldsymbol{I}-\boldsymbol{S}))}$. Finally observe that $\chi(\boldsymbol{D}) = \operatorname{rank}(\boldsymbol{D}) + \operatorname{rank}(\boldsymbol{I}-\boldsymbol{D}) - n$ so that $\chi(\widetilde{\boldsymbol{D}}) < \chi(\boldsymbol{D})$. $\blacksquare$

To implement the decomposition efficiently, one may want to compute $\boldsymbol{D}^\dagger$ incrementally by doing k rank one pseudo-inverse updates for each set peeled off. Since each of these updates needs $O(n^2)$, peeling one set off can be completed in $O(kn^2)$. Notice that Algorithm 1 can also be implemented in $O(kn^2)$ (see Theorem 4.2) with a projector of the null space of $\boldsymbol{D}$ incrementally maintained using Lemma 4.3. Combining the two parts gives a $O(kn^3)$ implementation for the entire decomposition process.

5. Conclusion

A new use of kernels is emerging from this line of research: The gain/loss is a kernel $k(\boldsymbol{u}, \boldsymbol{x}) = \phi(\boldsymbol{u})^\intercal\phi(\boldsymbol{x})$, the parameter space consists of all possible expectations $\mathbb{E}\left[\phi(\boldsymbol{u})\right]$, and after the update, the algorithm projects back into this parameter space. Finally any parameter is decomposed into a small mixture of $\phi(\boldsymbol{u})$, and thus the parameter is expressed in terms of the original domain of the feature map ϕ. We showed here how to do this for a simple quadratic kernel, and the work on Component Hedge can be reinterpreted as following this outline. However, what are the ingredients needed for the method to succeed in general? For example can this be done for higher order polynomial kernels?

In our treatment all instances $\boldsymbol{x}$ were assumed to be unit length. Ideally we want to learn vectors of varying length. To do this, more work first needs to be done on developing expert updates that can handle unbounded losses (see e.g. McMahon (2013) for a start). This work should be transferable to the matrix domain.

We believe that the richer modeling capability developed in this paper will make the use of matrix parameters imperative. However, one of the main criticism of this line of research is that it relies on eigendecompositions that require $O(n^3)$ time. The key open problem is to develop $O(n^2)$ algorithms without degrading the regret bounds too much (See e.g. discussions in (Hazan et al., 2010)).

References

Dennis S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas (Second Edition)*. Princeton reference. Princeton University Press, 2011. ISBN 9780691140391.

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

Elad Hazan, Satyen Kale, and Manfred K. Warmuth. Learning rotations with little regret. Unpublished journal submission, February 2011.

David P. Helmbold and Manfred K. Warmuth. Learning permutations with exponential weights. *Journal of Machine Learning Research*, 10:1705–1736, July 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, January 2011.

Wouter M. Koolen, Manfred K. Warmuth, and Jyrki Kivinen. Hedging structured concepts. In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT)*, pages 93–105, June 2010.

Wojciech Kotłowski and Manfred K. Warmuth. Minimax algorithm for learning rotations. *Journal of Machine Learning Research - Proceedings Track*, 19:821–824, 2011.

Dima Kuzmin and Manfred K. Warmuth. Online Kernel PCA with entropic matrix updates. In *Proceedings of the 24rd international conference on Machine learning (ICML '07)*, pages 465–471. ACM International Conference Proceedings Series, June 2007.

Brendan McMahon. Minimax optimal algorithms for unconstrained linear optimization. Unpublished manuscript arXiv:1302.2176v1, February 2013.

Jiazhong Nie, Wojciech Kotłowski, and Manfred K. Warmuth. On-line PCA with optimal regrets. Submitted, 2013.

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

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, October 2008.

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

Appendix A. Range of the gain

We first determine the range of the gain during a single trial. Fix any set of k orthogonal directions $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ and let $\boldsymbol{\mu} = \sum_{i=1}^k \boldsymbol{u}_i$ so that $\|\boldsymbol{\mu}\| = \sqrt{k}$. Let $\boldsymbol{x}$ be a direction, and let $\hat{\boldsymbol{x}} = \sum_{i=1}^k (\boldsymbol{u}_i^\intercal \boldsymbol{x})\boldsymbol{u}_i$ be the projection of $\boldsymbol{x}$ on the set. This notation allows us to write

$$\sum_{i=1}^k (\boldsymbol{x}^\intercal \boldsymbol{u}_i + c)^2 \;=\; \|\hat{\boldsymbol{x}}\|^2 + 2c\,\hat{\boldsymbol{x}}^\intercal \boldsymbol{\mu} + kc^2.$$

Using Cauchy-Schwartz, i.e. $(\hat{\boldsymbol{x}}^\intercal \boldsymbol{\mu})^2 \leq \|\hat{\boldsymbol{x}}\|\|\boldsymbol{\mu}\|$, the gain can be sandwiched as follows:

$$(\|\hat{\boldsymbol{x}}\| - \sqrt{k}c)^2 \;=\; \|\hat{\boldsymbol{x}}\|^2 - 2\sqrt{k}c\|\hat{\boldsymbol{x}}\| + kc^2 \;\leq\; \sum_{i=1}^k (\boldsymbol{x}^\intercal \boldsymbol{u}_i + c)^2 \;\leq\; \|\hat{\boldsymbol{x}}\|^2 + 2\sqrt{k}c\|\hat{\boldsymbol{x}}\| + kc^2 \;=\; (\|\hat{\boldsymbol{x}}\| + \sqrt{k}c)^2.$$

Recall that Cauchy-Schwartz holds with equality when $\hat{\boldsymbol{x}}$ and $\boldsymbol{\mu}$ are parallel. For $c \geq 0$, the gain is hence maximized at $\boldsymbol{x} = \hat{\boldsymbol{x}} = \boldsymbol{\mu}/\sqrt{k}$, where it takes value $(1 + \sqrt{k}c)^2$. Minimization is slightly more complicated. If $\sqrt{k}c \geq 1$, the gain is minimized at $\boldsymbol{x} = \hat{\boldsymbol{x}} = -\boldsymbol{\mu}/\sqrt{k}$, i.e. the reverse of the maximizer, where it takes value $(1 - \sqrt{k}c)^2$. If on the other hand $\sqrt{k}c \leq 1$, the gain is minimized when $\hat{\boldsymbol{x}} = -c\boldsymbol{\mu}$. This means that we can choose any $\boldsymbol{x} = \hat{\boldsymbol{x}} + \boldsymbol{x}_\perp$, where $\boldsymbol{x}_\perp$ is any vector of length $\sqrt{1 - kc^2}$ that is perpendicular to all $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$. Now the gain takes value 0.

Appendix B. When do solutions to the problems of learning k directions and k-PCA coincide?

Let $\boldsymbol{R}$ and $\boldsymbol{r}$ denote $\sum_t \boldsymbol{x}_t \boldsymbol{x}_t^\intercal$ and $\sum_t \boldsymbol{x}_t$, respectively. Let $\underbrace{\boldsymbol{U}\boldsymbol{U}^\intercal}_{(n,k)\times(k,n)}$ be the rank k projection matrix for the solution subspace of the PCA problem.

Lemma B.1 *If $\boldsymbol{r}$ lies in the subspace of the k-PCA solution, i.e. $\boldsymbol{U}\boldsymbol{U}^\intercal \boldsymbol{r} = \boldsymbol{r}$, then there is an orthonormal basis $\widehat{\boldsymbol{U}}$ s.t. $\widehat{\boldsymbol{U}}\widehat{\boldsymbol{U}}^\intercal = \boldsymbol{U}\boldsymbol{U}^\intercal$ which is also the solution of learning of k directions problem.*

Proof The gains relate as follows:

$$\overbrace{\underbrace{\mathrm{tr}(\boldsymbol{U}\boldsymbol{U}^\intercal \boldsymbol{R})}_{\text{PCA gain}} + 2c\,\boldsymbol{1}^\intercal \boldsymbol{U}^\intercal \boldsymbol{r}}^{\text{directional gain}}.$$

Since $\boldsymbol{U}\boldsymbol{U}^\intercal \boldsymbol{r} = \boldsymbol{r}$, there is an orthonormal basis $\widehat{\boldsymbol{U}}$ s.t. $\widehat{\boldsymbol{U}}\widehat{\boldsymbol{U}}^\intercal = \boldsymbol{U}\boldsymbol{U}^\intercal$, and $\widehat{\boldsymbol{U}}\boldsymbol{1}$ and $\boldsymbol{r}$ point in the same direction. So $\boldsymbol{U} = \widehat{\boldsymbol{U}}$ maximizes $\boldsymbol{1}^\intercal \boldsymbol{U}^\intercal \boldsymbol{r}$. On the other hand, since $\boldsymbol{U}\boldsymbol{U}^\intercal$ is the solution subspace of PCA, and $\boldsymbol{U}\boldsymbol{U}^\intercal = \widehat{\boldsymbol{U}}\widehat{\boldsymbol{U}}^\intercal$, $\boldsymbol{U} = \widehat{\boldsymbol{U}}$ also maximizes the PCA gain $\mathrm{tr}(\boldsymbol{U}\boldsymbol{U}^\intercal \boldsymbol{R})$. This means that $\widehat{\boldsymbol{U}}$ maximizes both terms of the directional PCA gain and is a solution to both problems. ■

Appendix C. Proof of the lower bound (Theorem 3.2)

First notice that for any distribution $\mathcal{P}$ on instance sequences $\boldsymbol{x}_{1\ldots T}$, the minimax regret of the game is lower bounded by the difference

$$\mathbb{E}_{\boldsymbol{x}_{1\ldots T}\sim\mathcal{P}}[G_C] - \max_{\text{alg. A}} \mathbb{E}_{\boldsymbol{x}_{1\ldots T}\sim\mathcal{P}}[G_A], \tag{C.1}$$

where G_C is the gain of the comparator (i.e. the best set of k orthogonal directions) chosen in hindsight and G_A is the gain of algorithm A.

In our lower bound, we use a $\mathcal{P}$ that is i.i.d. between trails and at each trial gives probability $\frac{1}{2}$ to each of the following two opposite instances,

$$\boldsymbol{x}_+ \;\coloneqq\; (\,\underbrace{1/\sqrt{k},\ldots,1/\sqrt{k}}_{k},\;\underbrace{0,\ldots,0}_{n-k}\,) \quad\text{and}\quad \boldsymbol{x}_- \;\coloneqq\; -(\,\underbrace{1/\sqrt{k},\ldots,1/\sqrt{k}}_{k},\;\underbrace{0,\ldots,0}_{n-k}\,).$$

Now we lower bound the difference in (C.1) for this particular choice of $\mathcal{P}$. We first lower bound the gain of the comparator by the gain of the best of two orthonormal sets, either $\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_k\}$ or $\{-\boldsymbol{e}_1,\ldots,-\boldsymbol{e}_k\}$ (these sets maximize the gain on $\boldsymbol{x}_+$ and $\boldsymbol{x}_-$ respectively).

$$\mathbb{E}_{\boldsymbol{x}_{1\ldots T}\sim\mathcal{P}}[G_C] \;\geq\; \mathbb{E}_{\boldsymbol{x}_{1\ldots T}\sim\mathcal{P}}\left[\sum_{t=1}^{T}\boldsymbol{x}_t^\intercal\boldsymbol{D}\boldsymbol{x}_t + 2c\max\left\{\boldsymbol{\mu}_+^\intercal\sum_{t=1}^{T}\boldsymbol{x}_t, \boldsymbol{\mu}_-^\intercal\sum_{t=1}^{T}\boldsymbol{x}_t\right\}\right] + Tc^2,$$

where $\boldsymbol{\mu}_+$ and $\boldsymbol{\mu}_-$ are the first moments of the two sets, that is

$$\boldsymbol{\mu}_+ \;=\; \{\underbrace{1,\ldots,1}_{k},0,\ldots,0\} \qquad\text{and}\qquad \boldsymbol{\mu}_- \;=\; \{\underbrace{-1,\ldots,-1}_{k},0,\ldots 0\},$$

and $\boldsymbol{D} \;=\; \begin{bmatrix}\boldsymbol{I}_k & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}$ is the common second moment of both sets. Since we only compare to two sets, the first moment part of the gain is essentially the two experts setting with loss per round equal to $\pm 2c\sqrt{k}$. With analysis in Koolen (2011), one can show that the first moment part is hence lower bounded by $\Omega(\sqrt{c^2kT})$. The second moment part, noticing that both instances $\boldsymbol{x}_+$ and $\boldsymbol{x}_-$ lie in the span of $\boldsymbol{D}$, always attains its maximum T. Finally, since instances are generated independently between trials with expectation zero ($\mathbb{E}[\boldsymbol{x}_t] = \boldsymbol{0}$), any algorithm has expected gain 0 in the first moment part, and so

$$\max_{\text{alg. A}} \mathbb{E}_{\boldsymbol{x}_{1\ldots T}\sim\mathcal{P}}[G_A] \;\leq\; T(1+c^2).$$

By combining the bounds on comparator and algorithms, we show a $\Omega(\sqrt{c^2kT})$ lower bound of the difference in (C.1) which concludes our proof.