Learning rotations with little regret

Elad Hazan
IBM Almaden
650 Harry Road
San Jose, CA 95120
ehazan@cs.princeton.edu

Satyen Kale
Yahoo! Research
4301 Great America Parkway
Santa Clara, CA 95054
skale@yahoo-inc.com

Manfred K. Warmuth*
Department of Computer Science
UC Santa Cruz
manfred@cse.ucsc.edu

Abstract

We describe online algorithms for learning a rotation from pairs of unit vectors in $\mathbb{R}^n$. We show that the expected regret of our online algorithm compared to the best fixed rotation chosen offline is $O(\sqrt{nL})$, where L is the loss of the best rotation. We also give a lower bound that proves that this expected regret bound is optimal within a constant factor. This resolves an open problem posed in COLT 2008. Our online algorithm for choosing a rotation matrix in each trial is based on the Follow-The-Perturbed-Leader paradigm. It adds a random spectral perturbation to the matrix characterizing the loss incurred so far and then chooses the best rotation matrix for that loss. We also show that any deterministic algorithm for learning rotations has $\Omega(T)$ regret in the worst case.

1 Introduction

Rotations are a fundamental object in robotics and vision. The problem of learning rotations, or finding the underlying rotation from a given set of examples, has numerous applications in these areas (see [Aro09] for a summary of application areas). As a motivating example, in optical character recognition, rotational (or skew) correction is an important and challenging problem. Optically read characters need to be aligned before they can be recognized. A fast, low-regret online learning algorithm for rotations can be used for detecting skew using a small number of examples.

Besides their practical importance, rotations have been shown to be powerful enough to capture seemingly more general mappings. Rotations can represent arbitrary Euclidean transformations via a conformal embedding by adding two special dimensions [WCL05]. Also [DHSA93] showed the rotation group provides a universal representation for all Lie groups.

The batch learning problem has a simple and well known solution [Wah65, Sch66], but the question of whether there are online algorithms for this problem was posed in [SW08] as an open problem. Recently, an algorithm was given in [Aro09] based on the Matrix Exponentiated Gradient update [TRW05]. This algorithm elegantly exploits the Lie group/Lie algebra relationship between rotation matrices and skew symmetric matrices, respectively, and the matrix exponential and matrix logarithm that maps between these domains. However, this algorithm deterministically predicts with a single rotation matrix in each trial. In this paper, we prove that any such deterministic algorithm can be forced to have regret at least $\Omega(T)$, where T is the number of trials.

To achieve regret bounds that are sublinear in T, it is necessary to "hedge our bets" and predict randomly from a suitable distribution over rotation matrices. There are two ways in which this can be done: either the algorithm predicts deterministically with a parameter that represents a convex combination of rotation matrices or it explicitly produces a rotation matrix based on its internal randomization. Our algorithm is of the latter type and is in the Follow-The-Perturbed-Leader (FPL) [KV05] family of algorithms. At this point we do not know how to design algorithms of the former type. One promising approach makes use of the von Mises-Fisher distribution, which is an exponential family distribution over rotations (See discussion in [SW10]).

In this paper we bypass the differential geometry of rotations altogether and hedge by predicting with a random rotation matrix. The key insight is that the loss for our rotation learning problem is linear in the chosen rotation matrix. We add a suitably chosen random perturbation matrix to the matrix characterizing the loss incurred so far and then simply choose the best rotation matrix for the perturbed matrix. A good choice

*Supported by NSF grant IIS-0917397

of the perturbation matrix turns out to be a one that has exponentially distributed singular values and random orthogonal left and right singular vectors. Surprisingly, for this choice, we can show that the regret bound for the resulting algorithm is optimal within a constant factor.

Outline of paper: We begin with some preliminaries in the next section, the precise problem statement, and basic properties of rotations. In this section we also prove a lower bound for any deterministic algorithm. In section 3 we describe our randomized algorithm and give a bound on its expected regret. Following that, in section 4 we give the lower bound which applies to any algorithm, randomized or otherwise, and which matches the regret bound of our algorithm up to a constant factor.

2 Preliminaries and Problem Statement

2.1 Notation.

In this paper, all vectors lie in $\mathbb{R}^n$ and all matrices in $\mathbb{R}^{n\times n}$. We use $\mathcal{SO}(n)$ to denote the special orthogonal group, i.e. the set of all rotation matrices $\mathbf{R}$. These are all orthogonal matrices of determinant one. For any vector $\mathbf{x}$, $\|\mathbf{x}\|$ denotes its ℓ_2 norm. For any matrix $\mathbf{A}$, $\|\mathbf{A}\|$ denotes its spectral norm (or Schatten-∞ norm) which is the maximum singular value of $\mathbf{A}$. Furthermore, $\|\mathbf{A}\|_\star$ denotes the trace norm (also known as the nuclear norm or Schatten-1 norm) which is the sum of all singular values. For two matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A}\bullet\mathbf{B}$ denotes the trace product $\mathbf{Tr}(\mathbf{A}^\top\mathbf{B}) = \sum_{ij} A_{ij}B_{ij}$.

2.2 Online Learning of Rotations problem.

Learning proceeds in a series of trials. In every iteration for $t = 1, 2, \ldots, T$:

1. The online learner is given a unit[1] vector $\mathbf{x}_t$ (i.e. $\|\mathbf{x}_t\| = 1$).
2. The learner is then required to commit (either deterministically or probabilistically), to a rotation matrix $\mathbf{R}_t \in \mathcal{SO}(n)$. The choice of $\mathbf{R}_t$ gives the predicted vector $\hat{\mathbf{y}}_t = \mathbf{R}_t\mathbf{x}_t$.
3. Finally the algorithm obtains true result, a unit vector $\mathbf{y}_t$ (which is presumably the result of some unknown rotation applied to $\mathbf{x}_t$).
4. The loss to the learner then is half the squared norm of the difference between her predicted vector and the "true" rotated vector:

$$L_t(\mathbf{R}_t) \;=\; \tfrac{1}{2}\|\mathbf{R}_t\mathbf{x}_t - \mathbf{y}_t\|^2 \;=\; \tfrac{1}{2}[\|\mathbf{R}_t\mathbf{x}_t\|^2 + \|\mathbf{y}_t\|^2 - 2\mathbf{y}_t^\top\mathbf{R}_t\mathbf{x}_t] \;=\; 1 - (\mathbf{y}_t\mathbf{x}_t^\top)\bullet\mathbf{R}_t. \qquad (1)$$

The last equality uses the fact that rotation preserves the ℓ_2 norm. Note that the loss is always in the range $[0, 2]$. The goal of the learner is to choose rotations $\mathbf{R}_t$ in such a way as to minimize the regret on all T examples given by

$$\text{Regret}_T \;=\; \sum_{t=1}^{T} L_t(\mathbf{R}_t) - \min_{\mathbf{R}\in\mathcal{SO}(n)} \sum_{t=1}^{T} L_t(\mathbf{R}).$$

We aim to find online algorithms with regret sublinear in T (Such algorithms are called Hannan consistent [CBL06]). Henceforth we give an algorithm whose regret is optimal within a constant factor.

We can also consider a setting of the problem that is more difficult for the learner in which $\mathbf{x}_t$ is not known at the point when the rotation matrix must be chosen. In this setting the learner first commits to a distribution over rotations and only then sees a pair $\mathbf{x}_t, \mathbf{y}_t$ and incurs the associated expected loss. Our algorithm works even in this setting and the bound on its expected regret remains valid. However our lower bounds apply to the above definition that is easier for the learner.

2.3 Main Result.

Our main result is a Follow-The-Perturbed-Leader type algorithm (see Section 3) with the following guarantee on its expected regret:

Theorem 1 *There is a randomized online algorithm which for any sequence of T examples for which the loss L of the best fixed rotation in hindsight, i.e. $L = \min_{\mathbf{R}\in\mathcal{SO}(n)} \sum_{t=1}^{T} L_t(\mathbf{R})$, is at least $16n$, achieves regret*

$$\mathbf{E}[\textit{Regret}_T] \;\le O(\sqrt{nL}).$$

This algorithm can be implemented to run in $O(n^3)$ time per trial.

[1] Note that there is no loss of generality in restricting $\mathbf{x}_t$ and $\mathbf{y}_t$ to be unit vectors. Our algorithm and its analysis, work unchanged assuming only $\|\mathbf{y}_t\mathbf{x}_t^\top\|_\star = \|\mathbf{x}_t\|\|\mathbf{y}_t\| \le 1$.

We also can prove a matching lower bound.

Theorem 2 *For any integer $T > n$, and for any online algorithm for learning rotations, there is a sequence of T examples, such that the algorithm incurs regret at least $\Omega(\sqrt{nT})$.*

Since the per trial loss of any rotation is at most 2, the loss L of the best rotation chosen in hindsight is at most $2T$, and therefore any lower bound of $\Omega(\sqrt{nT})$ on the regret implies a lower bound of $\Omega(\sqrt{nL})$. This shows that the regret of our algorithm (given in Theorem 1 above) is tight up to constant factors.

2.4 Solving the Offline Problem.

Before describing our online algorithm, we need to understand how to solve the optimization problem of offline (batch) algorithm:

$$\operatorname*{argmin}_{\mathbf{R}\in\mathcal{SO}(n)} \sum_{t=1}^{T} L_t(\mathbf{R}) \;=\; \operatorname*{argmax}_{\mathbf{R}\in\mathcal{SO}(n)} \sum_{t=1}^{T} (\mathbf{y}_t \mathbf{x}_t^\top) \bullet \mathbf{R}.$$

The equality follows from rewriting the loss function L_t as in (1). In general, an optimization problem of the form

$$\operatorname*{argmax}_{\mathbf{R}\in\mathcal{SO}(n)} \mathbf{M} \bullet \mathbf{R}$$

for some matrix $\mathbf{M}$, is a classical problem known as Wahba's problem [Wah65]. Figure 1 gives a simple example of the challenges in solving Wahba's problem: degeneracies can arise unexpectedly.

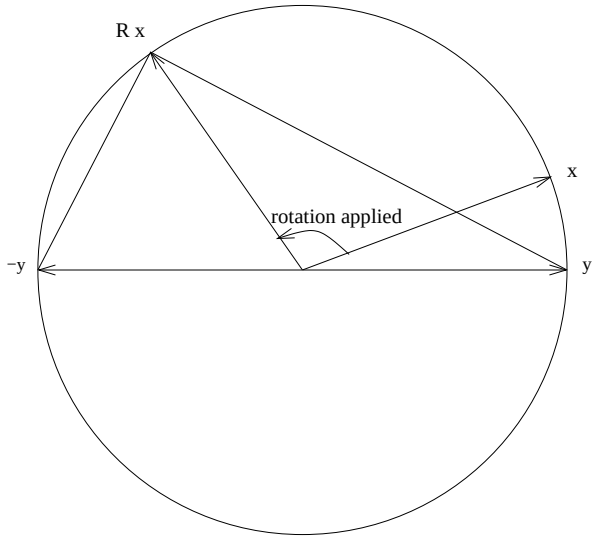

Figure 1: If we have two examples $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}, -\mathbf{y})$, then regardless of which rotation $\mathbf{R}$ we choose, $\mathbf{Rx}$ has loss exactly 2. This a consequence of the geometric fact that the diameter connecting $\mathbf{y}$ to $-\mathbf{y}$ subtends a right angle at $\mathbf{Rx}$, and therefore Pythagoras' theorem applies. Algebraically, for any $\mathbf{R}$, we have $L_1(\mathbf{R}) + L_2(\mathbf{R}) = 2 - (\mathbf{y}\mathbf{x}^\top - \mathbf{y}\mathbf{x}^\top) \bullet \mathbf{R} = 2 - 0 = 2$.

Nevertheless, Wahba's problem has a very elegant solution[2] via any singular value decomposition (SVD) of $\mathbf{M}$.

Lemma 3 *Let $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be any SVD of $\mathbf{M}$, i.e. $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, and $\mathbf{\Sigma} = diag(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is the diagonal matrix of non-negative singular values. Assume that σ_n is the smallest singular value. Let $s := \det(\mathbf{U})\det(\mathbf{V})$, where $\det$ denotes determinant. Since $\mathbf{U}$ and $\mathbf{V}$ are orthogonal, $s \in \{+1, -1\}$. Now if $\mathbf{W} := diag(1, 1, \ldots, 1, s)$, then $\mathbf{U}\mathbf{W}\mathbf{V}^\top$ is a solution to Wahba's problem, i.e.*

$$\mathbf{U}\mathbf{W}\mathbf{V}^\top \in \operatorname*{argmax}_{\mathbf{R}\in\mathcal{SO}(n)} \mathbf{M} \bullet \mathbf{R},$$

and the value of the optimal solutions is $\sum_{i=1}^{n-1} \sigma_i + s\sigma_n$, which is always non-negative.

[2]Note that the solution to Wahba's problem may not be unique, in which case we are satisfied with any solution amongst the optimal set of solutions.

This solution is a rotation matrix since it is a product of three orthogonal matrices, and its determinant equals $\det(\mathbf{U})\det(\mathbf{V})\det(\mathbf{W}) = 1$. The solution can be found in $O(n^3)$ time by constructing a SVD of $\mathbf{M}$.

We have been unable to find a complete proof of this lemma for dimensions more than 3 in the literature and for the sake of completeness we give a self-contained proof in Appendix A.

Note that if we are simply optimizing over all orthogonal matrices $\mathbf{R} \in \mathcal{O}(n)$, with no condition on $\det(\mathbf{R})$, then the construction becomes simpler (also proven in Appendix A):

Lemma 4 *Let $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be a SVD of $\mathbf{M}$ as in Lemma 3. Then*

$$\mathbf{U}\mathbf{V}^\top \in \operatorname*{argmax}_{\mathbf{R}\in\mathcal{O}(n)} \mathbf{M} \bullet \mathbf{R}$$

and the value of the optimum solutions is $\sum_{i=1}^n \sigma_i$.

This is also a classical problem known as the "Orthogonal Procrustes Problem", first solved by Schönemann [Sch66].

2.5 The necessity of randomization in learning rotations.

We give an adversary strategy that forces any deterministic algorithm such as the simple Follow-The-Leader algorithm (which predicts with the rotation matrix minimizing the loss on the past examples) or Arora's deterministic algorithm [Aro09] to have linear regret. This shows that randomization is essential for obtaining good regret bounds for learning rotations.

Theorem 5 *Any algorithm which deterministically predicts with a single rotation matrix at each trial has worst case regret at least T on example sequences of length T.*

Proof: Consider the following problem instance. In each iteration, the adversary always sets $\mathbf{x}_t = \mathbf{e}_1$. Since the algorithm is deterministic, the adversary can then compute the matrix $\mathbf{R}_t$ in every iteration. The algorithm predicts with $\hat{\mathbf{y}}_t = \mathbf{R}_t\mathbf{x}_t$ and the adversary chooses $\mathbf{y}_t$ as $-\hat{\mathbf{y}}_t$. Therefore the algorithm's per trial loss is

$$\tfrac{1}{2}\|\hat{\mathbf{y}}_t - \mathbf{y}_t\|^2 \;=\; \tfrac{1}{2}\|2\hat{\mathbf{y}}_t\|^2 \;=\; 2,$$

amounting to a loss $2T$ in all trials.

On the other hand the loss of the optimum rotation is

$$\min_{\mathbf{R}\in\mathcal{SO}(n)} \tfrac{1}{2}\|\mathbf{R}\mathbf{x}_t - \mathbf{y}_t\|^2 \;=\; T - \max_{\mathbf{R}\in\mathcal{SO}(n)} \mathbf{W}_T \bullet \mathbf{R},$$

where $\mathbf{W}_T = \sum_{t=1}^T \mathbf{y}_t\mathbf{x}_t^\top$. By Lemma 3, $\max_{\mathbf{R}\in\mathcal{SO}(n)} \mathbf{W}_T \bullet \mathbf{R} \geq 0$. Thus, the loss of the optimum rotation is at most T. Hence, the algorithm has regret at least $2T - T = T$. ■

In view of this lower bound, only randomized algorithms can hope to achieve sublinear regret. As we observed in equation (1), the square loss over the (non-convex) set $\mathcal{SO}(n)$ is linear. Therefore it is natural to apply the Follow-The-Perturbed-Leader (FPL) type algorithm [KV05], as this generic template is capable of handling non-convex decision sets. A direct application of the Kalai-Vempala result gives the following algorithm:

$$\mathbf{R}_t \;=\; \operatorname*{argmin}_{\mathbf{R}\in\mathcal{SO}(n)} \sum_{i=1}^{t-1} L_i(\mathbf{R}) - \mathbf{N} \bullet \mathbf{R},$$

where $\mathbf{N}$ is a random matrix whose entries are i.i.d. uniform random numbers in the range $[0, \frac{1}{\varepsilon}]$ for some parameter ε. However, tuning ε to its optimal value gives a suboptimal regret bound of $O(n^{5/4}\sqrt{T})$. The reason for this suboptimality is that uniform sampling of the components of $\mathbf{N}$ does not match the characteristics of our problem.

2.6 Sampling Random Orthogonal Matrices.

A fundamental task in our better implementation of FPL is sampling of random orthogonal matrices "uniformly", in the sense that the density of the distribution at any two matrices $\mathbf{U}$ and $\mathbf{V}$ in $\mathcal{O}(n)$ is the same. Technically, this distribution is given by the Haar measure ν on $\mathcal{O}(n)$ scaled so that $\nu(\mathcal{O}(n)) = 1$. The distribution ν has the property that for any fixed orthogonal matrix $\mathbf{U} \in \mathcal{O}(n)$, if $\mathbf{V}$ is sampled from ν, then the distribution on $\mathcal{O}(n)$ induced by $\mathbf{U}\mathbf{V}$ is also ν. This implies the desired uniformity property.

To sample a random orthogonal matrix from such a uniform distribution is essentially to choose a random orthogonal basis. The following process generates such basis incrementally: start with a random unit vector, and then repeatedly pick a random unit vector orthogonal to all vectors seen so far. This process can be implemented by running the Gram-Schmidt process on a set of n random vectors. Indeed, the efficient way of doing essentially this is by using the QR-decomposition of a random matrix with independent standard Gaussian entries (see [Ste80] for more details).

3 Algorithm and Analysis

Instead of using uniform randomness, the correct perturbation matrix turns out to be one with singular values chosen from an exponential distribution with randomly chosen left and right singular vectors, as given in the algorithm below. The online algorithm can be implemented in $O(n^3)$ time per trial by uniformly sampling orthogonal matrices as outlined in Section 2.6 and by optimizing a linear function over rotations as done in the solution of the off-line problem (see Sections 2.4).

Algorithm 1 FSPL: Follow-The-Spectrally-Perturbed-Leader

1: Select n non-negative real numbers $\sigma_1, \sigma_2, \ldots, \sigma_n$ independently from the exponential distribution with rate parameter ε, i.e. with density $\varepsilon \exp(-\varepsilon\sigma)d\sigma$. Let $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$.
2: Select two random orthogonal matrices $\mathbf{U}, \mathbf{V} \in \mathcal{O}(n)$ uniformly from the Haar measure.
3: Define $\mathbf{N} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$.
4: **for** $t = 1$ to T **do**
5: Let $\mathbf{R}_t$ be the result of the following optimization problem:

$$\mathbf{R}_t = \operatorname*{argmin}_{\mathbf{R}\in\mathcal{SO}(n)} \sum_{i=1}^{t-1} L_i(\mathbf{R}) - \mathbf{N} \bullet \mathbf{R} = \operatorname*{argmax}_{\mathbf{R}\in\mathcal{SO}(n)} \left[\sum_{i=1}^{t-1} (\mathbf{y}_i \mathbf{x}_i^\top) + \mathbf{N}\right] \bullet \mathbf{R}.$$

6: Obtain vector $\mathbf{x}_t$. Predict $\hat{\mathbf{y}}_t = \mathbf{R}_t\mathbf{x}_t$ and observe the result vector $\mathbf{y}_t$. Suffer loss $L_t(\mathbf{R}_t)$.
7: **end for**

We point out that our algorithm is *basis-invariant* in the following sense: fix an orthogonal matrix (i.e. an orthonormal basis) $\mathbf{B} \in \mathcal{O}(n)$. Consider two problem instances, one with examples $(\mathbf{x}_t, \mathbf{y}_t)$ for $t = 1, 2, \ldots, T$, and another with the same examples expressed in the alternate basis $\mathbf{B}$, viz $(\mathbf{B}\mathbf{x}_t, \mathbf{B}\mathbf{y}_t)$, and consider running the algorithm on these two sequences of examples. Geometrically, the instances are the same, so it is desirable for the algorithm to behave the same way, modulo the change of basis.

In our algorithm, the orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$ are drawn from the uniform Haar measure over orthogonal matrices, the distribution of $\mathbf{N}$ and that of $\mathbf{B}\mathbf{N}\mathbf{B}^\top$ is the same. This fact allows us to correlate the choice of the perturbations in the algorithms running over the two instances by choosing the perturbation $\mathbf{N}$ for the first instance, and the perturbation $\mathbf{B}\mathbf{N}\mathbf{B}^\top$ for the second instance.

Thus, if

$$\mathbf{R}_t = \operatorname*{argmax}_{\mathbf{R}\in\mathcal{SO}(n)} \left[\sum_{i=1}^{t-1} (\mathbf{y}_i \mathbf{x}_i^\top) + \mathbf{N}\right] \bullet \mathbf{R},$$

then

$$\mathbf{B}\mathbf{R}_t\mathbf{B}^\top = \operatorname*{argmax}_{\mathbf{R}\in\mathcal{SO}(n)} \left[\sum_{i=1}^{t-1} (\mathbf{B}\mathbf{y}_i(\mathbf{B}\mathbf{x}_i)^\top) + \mathbf{B}\mathbf{N}\mathbf{B}^\top\right] \bullet \mathbf{R}.$$

Note that $\mathbf{B}\mathbf{R}_t\mathbf{B}^\top$ is simply the rotation matrix $\mathbf{R}_t$ expressed in basis $\mathbf{B}$. The prediction of $\mathbf{B}\mathbf{R}_t\mathbf{B}^\top$ on the transformed example $(\mathbf{B}\mathbf{x}_t, \mathbf{B}\mathbf{y}_t)$ is $\mathbf{B}\mathbf{R}_t\mathbf{B}^\top\mathbf{B}\mathbf{x}_t$ which simplifies to $\mathbf{B}\mathbf{R}_t\mathbf{x}_t$. This prediction is the same as the prediction of $\mathbf{R}_t$ on the original example $(\mathbf{x}_t, \mathbf{y}_t)$, i.e. $\mathbf{R}_t\mathbf{x}_t$, and while premultiplying with the transformation $\mathbf{B}$. Geometrically, our algorithms is doing the same thing on the original and the transformed sequence of examples, and the losses are the same well.

Before launching into the analysis, we make a few observations regarding the noise matrix $\mathbf{N}$. Essentially, the noise matrix is sampled by constructing each component of its SVD randomly. With probability 1, the singular values of $\mathbf{N}$, viz. $\sigma_1, \sigma_2, \ldots, \sigma_n$ are all distinct, so in the sequel we assume this is the case whenever we talk about the noise matrix $\mathbf{N}$.

In the induced probability distribution over matrices $\mathbf{N}$, if the SVD of $\mathbf{N}$ is $\mathbf{N} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, the density of the distribution at $\mathbf{N}$ is

$$d\mu(\mathbf{N}) = 2^n n! \varepsilon^n \exp\left[-\varepsilon \sum_{i=1}^{n} \sigma_i\right] d\sigma_1 d\sigma_2 \ldots d\sigma_n d\nu(\mathbf{U}) d\nu(\mathbf{V}),$$

where $d\nu(\mathbf{U})$ is the density at $\mathbf{U}$ in the Haar measure over $\mathcal{O}(n)$. The leading factor of $2^n n!$ is because the SVD is uniquely determined by the ordering of the singular values and the sign multiplying the n pairs of right and left singular vectors. Note that $\sum_{i=1}^{n} \sigma_i = \|\mathbf{N}\|_\star$, the trace norm of N. Thus, since ν is the uniform Haar measure on $\mathcal{O}(n)$, we may (informally) say that

$$d\mu(\mathbf{N}) \propto \exp(-\varepsilon\|\mathbf{N}\|_\star).$$

Now, we can analyze FSPL. The following Theorem implies Theorem 1.

Theorem 6 *For any target rotation matrix $\mathbf{R}^\star$, the algorithm FSPL attains the following regret bound:*

$$\mathbf{E}\left[\sum_{t=1}^{T} L_t(\mathbf{R}_t)\right] - \sum_{t=1}^{T} L_t(\mathbf{R}^\star) \;\le\; \frac{\varepsilon}{1-\varepsilon}\sum_{t=1}^{T} L_t(\mathbf{R}^\star) + \frac{2n}{(1-\varepsilon)\varepsilon}.$$

If $L \ge 16n$ and ε is set to $\sqrt{\frac{n}{L}}$, where $L = \sum_{t=1}^{T} L_t(\mathbf{R}^\star)$, then the expected regret is bounded[3] by $4\sqrt{nL}$.

Proof: The analysis of FSPL is based on the technique of Kalai and Vempala [KV05] for analyzing the Follow-The-Perturbed-Leader style algorithms.

For convenience of notation, define $\mathbf{W}_t = \sum_{i=1}^{t-1} \mathbf{y}_i \mathbf{x}_i^\top$, and the function $\mathcal{R} : \mathbb{R}^{n\times n} \to \mathcal{SO}(n)$ to be

$$\mathcal{R}(\mathbf{M}) \;:=\; \operatorname*{argmax}_{\mathbf{R}\in\mathcal{SO}(n)} \mathbf{M} \bullet \mathbf{R}.$$

Now the rotation matrix $\mathbf{R}_t$ chosen by the algorithm is denoted as $\mathcal{R}(\mathbf{W}_t + \mathbf{N})$. The first step of analyzing any Follow-The-Perturbed-Leader algorithm, is the following inequality which is essentially proved in [KV05], but we give the proof in Appendix B for the sake of completeness.

Claim 7 *For any target rotation matrix $\mathbf{R}^\star$, we have*

$$\sum_{t=1}^{T} L_t(\mathbf{R}_t) - \sum_{t=1}^{T} L_t(\mathbf{R}^\star) \;\le\; \sum_{t=1}^{T} L_t(\mathbf{R}_t) - L_t(\mathbf{R}_{t+1}) + \mathbf{N}\bullet\mathbf{R}_1 - \mathbf{N}\bullet\mathbf{R}^\star.$$

Now, to bound the *expected* regret, we prove the following two bounds:

$$\mathbf{E}[L_t(\mathbf{R}_t) - L_t(\mathbf{R}_{t+1})] \;\le\; \varepsilon\mathbf{E}[L_t(\mathbf{R}_t)], \tag{2}$$

and

$$\mathbf{E}[\mathbf{N}\bullet(\mathbf{R}_1 - \mathbf{R}^\star)] \;\le\; \frac{2n}{\varepsilon}. \tag{3}$$

Combining these two bounds, and some algebra, we get the stated bound on the regret:

$$\mathbf{E}\left[\sum_{t=1}^{T} L_t(\mathbf{R}_t)\right] - \sum_{t=1}^{T} L_t(\mathbf{R}^\star) \;\le\; \frac{\varepsilon}{1-\varepsilon}\sum_{t=1}^{T} L_t(\mathbf{R}^\star) + \frac{2n}{(1-\varepsilon)\varepsilon}.$$

If we set $\varepsilon = \sqrt{\frac{n}{L}}$, where $L = \sum_{t=1}^{T} L_t(\mathbf{R}^\star)$, then the expected regret is bounded by $\mathbf{E}[\text{Regret}] \;\le\; 4\sqrt{nL}$, assuming[4] $L \ge 16n$.

We prove the inequality (2) now. Since we are only interested in expected values, it doesn't matter whether we choose the noise at the beginning, or if we re-randomize every trial. Thus, denoting by $\mathbf{N}$ and $\mathbf{N}'$ the noise chosen in the t^{th} and $(t+1)^{\text{st}}$ trials respectively, we have

$$\begin{aligned}
&\mathbf{E}[L_t(\mathbf{R}_t)] - \mathbf{E}[L_t(\mathbf{R}_{t+1})] \\
&= \int_{\mathbf{N}} L_t(\mathcal{R}(\mathbf{W}_t + \mathbf{N}))d\mu(\mathbf{N}) - \int_{\mathbf{N}'} L_t(\mathcal{R}(\mathbf{W}_{t+1} + \mathbf{N}'))d\mu(\mathbf{N}') \\
&= \int_{\mathbf{N}} L_t(\mathcal{R}(\mathbf{W}_t + \mathbf{N}))d\mu(\mathbf{N}) - L_t(\mathcal{R}(\mathbf{W}_t + \mathbf{N}))d\mu(\mathbf{N} - \mathbf{y}_t\mathbf{x}_t^\top) \\
&\quad (\text{doing a change of variables in the integration: } \mathbf{N}' = \mathbf{N} - \mathbf{y}_t\mathbf{x}_t^\top) \\
&\le \int_{\mathbf{N}} \varepsilon L_t(\mathcal{R}(\mathbf{W}_t + \mathbf{N}))d\mu(\mathbf{N}) \\
&= \varepsilon\mathbf{E}[L_t(\mathbf{R}_t)].
\end{aligned}$$

The last inequality follows from the fact that for any $\mathbf{N}$, we have

$$d\mu(\mathbf{N}) - d\mu(\mathbf{N} - \mathbf{y}_t\mathbf{x}_t^\top) \;\le\; \varepsilon d\mu(\mathbf{N}),$$

which we prove now:

$$\frac{d\mu(\mathbf{N} - \mathbf{y}_t\mathbf{x}_t^\top)}{d\mu(\mathbf{N})} \;=\; \frac{\exp(-\varepsilon\|\mathbf{N} - \mathbf{y}_t\mathbf{x}_t^\top\|_\star)}{\exp(-\varepsilon\|\mathbf{N}\|_\star)} \;\ge\; \exp(-\varepsilon\|\mathbf{y}_t\mathbf{x}_t^\top\|_\star) \;\ge\; \exp(-\varepsilon) \;\ge\; 1-\varepsilon.$$

[3] Standard techniques such as the "doubling trick" (see e.g. [CBFH$^+$97]), can be used to obtain the same type of regret bound even if L is not known in advance while only slightly increasing the constant in front of the square root.

[4] A slightly better constant is obtainable via some further optimization over ε.

We use the triangle inequality for the $\|\cdot\|_\star$ norm in the first inequality, and $\|\mathbf{y}_t\mathbf{x}_t^\top\|_\star \leq 1$ in the second inequality.

Now, we prove inequality (3). To this end, we show that for any rotation matrix $\mathbf{R}$, we have $\mathbf{E}[|\mathbf{N} \bullet \mathbf{R}|] \leq \frac{n}{\varepsilon}$. To prove this, note that

$$|\mathbf{N} \bullet \mathbf{R}| \;\leq\; \|\mathbf{N}\|_\star \|\mathbf{R}\|,$$

by the Hölder inequality for trace norm (Schatten 1-norm) and its dual matrix norm (Schatten ∞-norm). Clearly, $\|\mathbf{R}\| = 1$ since all singular values of an orthogonal matrix are one. For the noise matrix $\mathbf{N}$, if $\sigma_1, \sigma_2, \ldots, \sigma_n$ are its singular values, then

$$\mathbf{E}[\|\mathbf{N}\|_\star] \;=\; \mathbf{E}\left[\sum_{i=1}^{n} \sigma_i\right] \;=\; \sum_{i=1}^{n} \mathbf{E}[\sigma_i] \;=\; \frac{n}{\varepsilon}$$

since the σ_i's are independently distributed exponential random variables with mean $\frac{1}{\varepsilon}$. Putting the bounds together, we get that

$$\mathbf{E}[\mathbf{N} \bullet (\mathbf{R}_1 - \mathbf{R}^\star)] \;\leq\; \frac{2n}{\varepsilon}.$$

■

4 Lower Bound

We now show a lower bound against any algorithm (including probabilistic ones). This matches the upper bound for the FSPL algorithm up to constant factors.

Proof of Theorem 2. We assume for convenience that the dimension is $n+1$, rather than n. For any online algorithm for the rotations problem we construct an example sequence of length T for which this algorithm has regret $\Omega(\sqrt{nT})$.

Let $\mathbf{e}_i$ denote the i-th standard basis vector, i.e. the vector with 1 in its i-th coordinate and 0 everywhere else. In trial $t < T$, set $\mathbf{x}_t = \mathbf{e}_{f(t)}$, where $f(t) = (t \bmod n) + 1$ (i.e., cycle through the coordinates $1, 2, \ldots, n$), and $\mathbf{y}_t = \sigma_t \mathbf{e}_{f(t)}$, where $\sigma_t \in \{-1, 1\}$ uniformly at random. For any coordinate $i \in 1, 2, \ldots, n$, let $X_i = \sum_{t:\ f(t)=i} \sigma_t$. For the final trial T, set $\mathbf{x}_T = \mathbf{e}_{n+1}$, and $\mathbf{y}_T = \sigma_T \mathbf{e}_{n+1}$, where $\sigma_T \in \{-1, 1\}$ is chosen in a certain way specified momentarily. First, note that

$$\mathbf{W}_{T+1} \;=\; \sum_{t=1}^{T} \mathbf{y}_t \mathbf{x}_t^\top \;=\; \mathrm{diag}(X_1, X_2, \ldots, X_n, \sigma_T).$$

We choose σ_T so that

$$\det(\mathbf{W}_{T+1}) \;=\; \sigma_T \prod_{i=1}^{n} X_i \;>\; 0.$$

In other words, $\sigma_T = \mathrm{sgn}(\prod_{i=1}^{n} X_i)$.

By Lemma 3, the solution to the offline problem is the rotation matrix $\mathbf{R}^\star = \mathrm{argmax}_{\mathbf{R} \in \mathcal{SO}(n)} \mathbf{W}_{T+1} \bullet \mathbf{R}$, where

$$\mathbf{R}^\star = \mathrm{diag}(\mathrm{sgn}(X_1), \mathrm{sgn}(X_2), \ldots, \mathrm{sgn}(X_n), \sigma_T),$$

and the loss of this matrix is

$$\sum_{t=1}^{T} L_t(\mathbf{R}^\star) \;=\; T - \mathbf{W}_{T+1} \bullet R^\star \;=\; T - \sum_{i=1}^{n} |X_i| - 1.$$

Since each X_i is a sum of $\lfloor \frac{T-1}{n} \rfloor$ Rademacher variables (give or take 1), standard probabilistic bounds (such as Khintchine's inequality [Haa82]) imply that $\mathbf{E}[|X_i|] = \Omega(\sqrt{T/n})$, where the expectation is taken over the choice of the σ_t's. Thus, the expected loss of the optimal rotation is bounded as follows:

$$\mathbf{E}\left[\sum_{t=1}^{T} L_t(\mathbf{R}^\star)\right] \;=\; T - \mathbf{E}[\mathbf{W}_{T+1} \bullet \mathbf{R}^\star] \;=\; T - \Omega(\sqrt{T/n} \cdot n) \;=\; T - \Omega(\sqrt{nT}).$$

Finally, note that for $t < T$, regardless of which specific rotation matrix $\mathbf{R}_t$ is selected by the algorithm,

$$\mathbf{E}_{\sigma_t}[L_t(\mathbf{R}_t)] \;=\; 1 - \mathbf{E}_{\sigma_t}[\mathbf{y}_t \mathbf{x}_t^\top \bullet \mathbf{R}_t] \;=\; 1 - \mathbf{E}_{\sigma_t}[\sigma_t] \cdot \mathbf{e}_{f(t)}^\top \mathbf{R}_t \mathbf{e}_{f(t)} \;=\; 1.$$

In trial T, the algorithm might at best have a loss of 0. Thus, the expected loss of the algorithm is at least $T - 1$, and hence its expected regret is $\Omega(\sqrt{nT})$. This implies that there is a choice of the σ_t's so that the actual regret of the algorithm is $\Omega(\sqrt{nT})$, as required. ■

5 Conclusions

We have presented tight bounds on the regret for online learning of rotation matrices. Our main technique is a Follow-The-Perturbed-Leader type algorithm with spectral perturbations. Essentially the same algorithm works in a different setting as well: online learning of a basis (or more colorfully, the Online Orthogonal Procrustes problem). Here, the learner is presented the example $\mathbf{x}_t$, and now the learner chooses an orthogonal matrix $\mathbf{U}_t \in \mathcal{O}(n)$ instead of a rotation matrix and predicts $\hat{\mathbf{y}}_t = \mathbf{U}_t\mathbf{x}_t$. This corresponds to a change of basis. Then the actual vector $\mathbf{y}_t$ is revealed, and the loss is defined the same way: $\frac{1}{2}\|\hat{\mathbf{y}}_t - \mathbf{y}_t\|^2$. The goal is to minimize regret with respect to the best change of basis in hindsight. It is apparent from the algorithm's analysis that we can use the same algorithm with the same perturbations, except that we optimize over $\mathcal{O}(n)$ rather than $\mathcal{SO}(n)$, an even simpler task (c.f. Lemma 4). Our regret bounds carry over to this setting.

It would be very interesting to find other applications of the spectral perturbations idea. In particular, the matrix version of the FPL algorithm might lead to speedups of the Matrix Hedge [WK06] and more generally the Matrix Exponentiated Gradient algorithm which both optimize over density matrices. To realize these speedups one would either have to get away with a single perturbation matrix or perturbation matrices that can be sampled in $O(n^2)$ time per trial. We expand on these ideas in an open problem posed in this conference [HKW10].

Acknowledgement

Our work was motivated by preliminary research done with Adam Smith [SW10, SW08] and we greatly benefited from discussions with him. We are also thankful to Abhishek Kumar for allowing us to include a simplified proof of Wahba's problem which was worked out in collaboration with him.

References

[Aro09] R. Arora. On learning rotations. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 55–63. MIT Press, 2009.

[CBFH+97] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, May 1997.

[CBL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

[DHSA93] C. Doran, D. Hestenes, F. Sommen, and N. Van Acker. Lie groups as spin groups. *J. Math. Phys.*, 34(8):3642–3669, August 1993.

[Haa82] U. Haagerup. The best constants in the Khintchine inequality. *Studia Math.*, 70(3):427–485, 1982.

[HKW10] E. Hazan, S. Kale, and M. K. Warmuth. On-line variance minimization in $O(n^2)$ per trial? In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT '10)*, 2010.

[KV05] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, 2005.

[Sch66] P. Schonemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1), March 1966.

[Ste80] G. W. Stewart. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM J. Numer. Anal.*, 17(3):403–409, 1980.

[SW08] A. Smith and M. K. Warmuth. Learning rotations. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT '08)*, page 517, July 2008.

[SW10] A. M. Smith and M. K. Warmuth. Learning rotations online. Technical Report UCSC-SOE-10-08, Department of Computer Science, University of California, Santa Cruz, February 2010.

[TRW05] K. Tsuda, G. Rätsch, and M. K. Warmuth. Matrix exponentiated gradient updates for on-line learning and Bregman projections. *Journal of Machine Learning Research*, 6:995–1018, June 2005.

[Wah65] G. Wahba. Problem 65-1, a least squares estimate of satellite attitude. *SIAM Review*, 7(3), July 1965.

[WCL05] R. Wareham, J. Cameron, and J. Lasenby. Applications of conformal geometric algebra in computer vision and graphics. *6th International Workshop IWMM 2004*, pages 329–349, 2005.

[WK06] M. K. Warmuth and D. Kuzmin. Online variance minimization. In *Proceedings of the 19th Annual Conference on Learning Theory (COLT '06)*, 2006.

A Solutions of Wahba's problem and the Orthogonal Procrustes Problem

We first prove Lemma 4, since it is simpler and it gives a reduction which will be useful for the proof of Lemma 3.

Proof of Lemma 4: Recall that we want to compute

$$\max_{\mathbf{R}\in\mathcal{O}(n)} \mathbf{M}\bullet\mathbf{R}.$$

Let $\mathbf{M}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be an SVD of $\mathbf{M}$, so that $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, and $\mathbf{\Sigma}=\mathrm{diag}(\sigma_1,\sigma_2,\ldots,\sigma_n)$ is a diagonal matrix of the non-negative singular values σ_i.

Now, we do a change of variables. Instead of maximizing over an orthogonal matrix $\mathbf{R}$, we maximize over orthogonal matrix $\mathbf{W}=\mathbf{U}^\top\mathbf{R}\mathbf{V}$. This lets us rewrite the dot product we are minimizing over

$$\mathbf{M}\bullet\mathbf{R} \;=\; \mathbf{Tr}(\mathbf{M}^\top\mathbf{U}\mathbf{W}\mathbf{V}^\top) \;=\; \mathbf{Tr}(\mathbf{V}\mathbf{\Sigma}^\top\mathbf{U}^\top\mathbf{U}\mathbf{W}\mathbf{V}^\top) \;=\; \mathbf{Tr}(\mathbf{\Sigma}^\top\mathbf{W}) \;=\; \mathbf{\Sigma}\bullet\mathbf{W}. \tag{4}$$

Since $\mathbf{W}$ is an orthogonal matrix, we have $|W_{ii}|\le 1$ for all i. Hence, the linear expression $\mathbf{\Sigma}\bullet\mathbf{W}=\sum_i\sigma_i W_{ii}$ is maximized when $\mathbf{W}=\mathbf{I}$, the identity matrix. We conclude that $\mathbf{R}=\mathbf{U}\mathbf{V}^\top$ is an optimal solution to $\max_{\mathbf{R}\in\mathcal{O}(n)}\mathbf{M}\bullet\mathbf{R}$ and all maxima has the value $\sum_{i=1}^n\sigma_i$. ■

We have been unable to find a complete, rigorous solution of Wahba's problem in the literature for dimensions more than 3. For the sake of completeness, we give a complete proof. This proof was obtained in collaboration with Abhishek Kumar, simplifying a previous version given in the submitted version of this paper.

Proof of Lemma 3: Recall that we want to compute

$$\max_{\mathbf{R}\in\mathcal{SO}(n)} \mathbf{M}\bullet\mathbf{R}.$$

As in the proof of Lemma 4, let $\mathbf{M}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be an SVD of $\mathbf{M}$. As before, we do a change of variables from a rotation matrix $\mathbf{R}$ to an orthogonal matrix $\mathbf{W}=\mathbf{U}^\top\mathbf{R}\mathbf{V}$, with the following condition on the determinant of $\mathbf{W}$:

$$\det(\mathbf{W}) \;=\; \det(\mathbf{U})\det(\mathbf{R})\det(\mathbf{V}) \;=\; \det(\mathbf{U})\det(\mathbf{V}) \;=:\; s\in\{+1,-1\}.$$

Using equation (4), the problem now reduces to:

$$\max_{\mathbf{W}\in\mathcal{O}(n),\det(\mathbf{W})=s} \mathbf{\Sigma}\bullet\mathbf{W}. \tag{5}$$

The case $\det(\mathbf{W})=1$ is easy. We already showed in the previous lemma that

$$\mathbf{I}\in\operatorname*{argmax}_{\mathbf{W}\in\mathcal{O}(n)} \mathbf{\Sigma}\bullet\mathbf{W}.$$

Thus in this case the constraint on the determinant of $\mathbf{W}$ is immaterial and the value of the maxima is $\sum_{i=1}^n\sigma_i=\sum_{i=1}^{n-1}\sigma_i+s\sigma_n$, where σ_n is the smallest singular value.

The case $\det(\mathbf{W})=-1$ is considerably harder. We need to show $\mathbf{W}=\mathrm{diag}(1,1,\ldots,1,-1)$ is an optimal solution which has value $\sum_{i=1}^{n-1}\sigma_i-\sigma_n=\sum_{i=1}^{n-1}\sigma_i+s\sigma_n$, where σ_n is the smallest singular value.

Let $\mathbf{W}$ be an arbitrary orthogonal matrix of determinant -1. We make the following observations regarding $\mathbf{W}$. First, if $\lambda_1,\lambda_2,\ldots,\lambda_n$ are the n (real or complex) eigenvalues of $\mathbf{W}$, then we have

$$\prod_{i=1}^n\lambda_i=\det(\mathbf{W})=-1.$$

Since $\mathbf{W}\in\mathcal{O}(n)$, all eigenvalues λ_i have magnitude $|\lambda_i|=1$: this is because if λ is an eigenvalue of $\mathbf{W}$ with eigenvector $\mathbf{v}$, i.e. $\mathbf{W}\mathbf{v}=\lambda\mathbf{v}$, then

$$1=\|\mathbf{v}\| \;=\; \|\mathbf{W}\mathbf{v}\| \;=\; \|\lambda\mathbf{v}\| \;=\; |\lambda|\|\mathbf{v}\|=|\lambda|,$$

where the second equality uses $\mathbf{W}\in\mathcal{O}(n)$.

We claim that at least one eigenvalue of the matrix $\mathbf{W}$ is -1. This is so because all the complex eigenvalues of the *real* matrix $\mathbf{M}$ must occur in complex conjugate pairs, $a + ib$ and $a - ib$, for some $b \neq 0$ and $a^2 + b^2 = 1$. Now, the product of any such complex conjugate pair of eigenvalues is $(a + ib)(a - ib) = a^2 + b^2 = 1$. Hence, the product of all complex eigenvalues is 1. Since the product of *all* eigenvalues (real or complex) is -1, and all real eigenvalues are either $+1$ or -1, we must have at least one eigenvalue being -1.

For convenience of notation, let $\lambda_n = -1$. Now, we have

$$\mathrm{tr}(\mathbf{W}) \;=\; \sum_{i=1}^{n-1} \lambda_i + \lambda_n \;=\; \sum_{i=1}^{n-1} \mathrm{real}(\lambda_i) - 1 \;\leq\; n - 1 - 1 \;=\; n - 2.$$

Here, $\mathrm{real}(z)$ is the real part of a complex number z, and we use the fact that the sum of two complex conjugate eigenvalues $a + ib$ and $a - ib$ is $2a$, which is the sum of their real parts. We also used the fact that for any eigenvalue λ_i, $\mathrm{real}(\lambda_i) \leq 1$ since $|\lambda_i| = 1$.

Finally, note that $|W_{ii}| \leq 1$ since $\mathbf{W}$ is an orthogonal matrix. Now, consider the following linear program which is a relaxation of the optimization problem (5) (This is a relaxation since the last inequality holds for all solutions of (5) and we drop the constraint that $\mathbf{W}$ is an orthogonal matrix of determinant -1):

$$\begin{aligned} & \max \sum_{i=1}^{n} \sigma_i W_{ii} \\ \forall i: \quad & -1 \;\leq\; W_{ii} \;\leq 1 \\ & \sum_{i=1}^{n} W_{ii} \;\leq\; n - 2. \end{aligned}$$

The optimal solution to this linear program is obtained at a vertex of the polytope defined by the constraints. We now characterize the vertices of the polytope as follows:

Claim 8 *Any vertex of the polytope defined by the constraints of the above linear program satisfies $W_{ii} \in \{+1, -1\}$ for all i, with at least one W_{ii} set to -1.*

Proof: Any vertex is obtained by setting n of the inequalities to equalities.

Case 1: n of the $-1 \leq W_{ii} \leq 1$ inequalities are tight. Then all $W_{ii} \in \{-1, +1\}$, and to satisfy $\sum_{i=1}^{n} W_{ii} \leq n - 2$, we must have at least one -1.

Case 2: $\sum_{i=1}^{n} W_{ii}$ equals the integer $n - 2$, exactly $n - 1$ of the inequalities $-1 \leq W_{ii} \leq 1$ are tight for say $1 \leq i \leq n - 1$, and the last one is not tight, i.e. $-1 < W_{nn} < 1$. Then for all $1 \leq i \leq n - 1$, we have $W_{ii} \in \{+1, -1\}$, since $\sum_{i=1}^{n} W_{ii} = n - 2$, an integer, W_{nn} is also an integer, and hence must be zero. But then with $W_{ii} \in \{+1, -1\}$ for $1 \leq i \leq n - 1$ the sum $\sum_{i=1}^{n-1} W_{ii}$ is either $n - 1$ or at most $n - 3$. Thus $\sum_{i=1}^{n} W_{ii} = n - 2$ can't be satisfied and case 2 does not give any more vertices. ■

With this characterization of the vertices, since the σ_n is the smallest singular value, the optimal vertex for the linear program is the one where $W_{ii} = 1$ for $1 \leq i \leq n-1$, and $W_{nn} = -1$. Thus the optimum value of the linear program is $\sum_{i=1}^{n-1} \sigma_i - \sigma_n$. Since this is a relaxation to the original problem, this optimum value is only larger than the optimum of the original problem. However, by setting $\mathbf{W} = \mathrm{diag}(1, 1, \ldots, 1, -1)$, which is an orthogonal matrix of determinant -1, we achieve the same value in the original problem as in the relaxed LP, and hence the optimal solution to the original problem is given by this $\mathbf{W}$. ■

B Proof of Claim 7

For notational convenience, define a "hallucinated" 0-th trial with loss function $L_0(\mathbf{R}) := -\mathbf{N} \bullet \mathbf{R}$ over rotation matrices. With this notation, $\mathbf{R}_t = \arg\min_{\mathbf{R} \in \mathcal{SO}(n)} \sum_{\tau=0}^{t-1} L_t(\mathbf{R})$, for any $t \geq 1$. We prove by induction that for any $T \geq 0$, we have

$$\sum_{t=0}^{T} L_t(\mathbf{R}_{t+1}) \;\leq\; \sum_{t=0}^{T} L_t(\mathbf{R}_{T+1}).$$

The statement holds trivially for $T = 0$. So assume that it holds for some $T \geq 0$, and now we prove it for $T + 1$. For this, we have

$$\sum_{t=0}^{T+1} L_t(\mathbf{R}_{t+1}) \;\leq\; \sum_{t=0}^{T} L_t(\mathbf{R}_{T+1}) + L_{T+1}(\mathbf{R}_{T+2}) \;\leq\; \sum_{t=0}^{T} L_t(\mathbf{R}_{T+2}) + L_{T+1}(\mathbf{R}_{T+2}) \;=\; \sum_{t=0}^{T+1} L_t(\mathbf{R}_{T+2}).$$

Here, the first inequality follows from the induction hypothesis, and the second from the fact that $\mathbf{R}_{T+1} = \arg\min_{\mathbf{R}\in\mathcal{SO}(n)} \sum_{t=0}^{T} L_t(\mathbf{R})$. Thus, the induction is complete.

We now continue by using $\mathbf{R}_{T+1} = \arg\min_{\mathbf{R}\in\mathcal{SO}(n)} \sum_{t=0}^{T} L_t(\mathbf{R})$ a second time:

$$\sum_{t=0}^{T} L_t(\mathbf{R}_{t+1}) \;\leq\; \sum_{t=0}^{T} L_t(\mathbf{R}_{T+1}) \;\leq\; \sum_{t=0}^{T} L_t(\mathbf{R}^\star).$$

This implies that

$$\sum_{t=1}^{T} L_t(\mathbf{R}_t) - \sum_{t=1}^{T} L_t(\mathbf{R}^\star) \;\leq\; \sum_{t=1}^{T} (L_t(\mathbf{R}_t) - L_t(\mathbf{R}_{t+1})) - L_0(\mathbf{R}_1) + L_0(\mathbf{R}^\star),$$

as required.