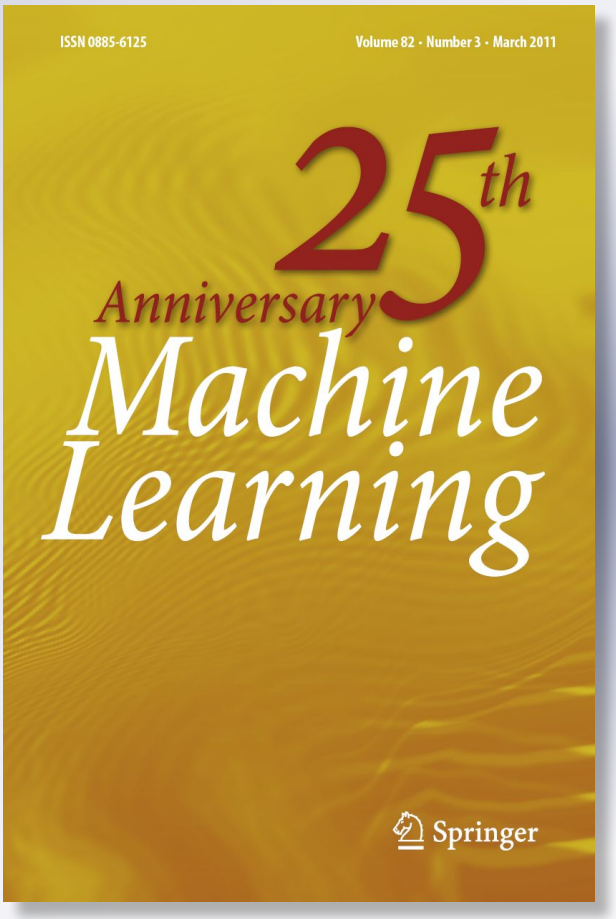
Online variance minimization

Manfred K. Warmuth & Dima Kuzmin

Machine Learning

ISSN 0885-6125
Volume 87
Number 1

Mach Learn (2012) 87:1-32
DOI 10.1007/s10994-011-5269-0



Springer

Mach Learn (2012) 87:1–32
DOI 10.1007/s10994-011-5269-0

Online variance minimization

Manfred K. Warmuth · Dima Kuzmin

Received: 28 September 2008 / Accepted: 25 October 2011 / Published online: 20 November 2011

Abstract We consider the following type of online variance minimization problem: In every trial t our algorithms get a covariance matrix $\boldsymbol{C}^t$ and try to select a parameter vector $\boldsymbol{w}^{t-1}$ such that the total variance over a sequence of trials $\sum_{t=1}^{T}(\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1}$ is not much larger than the total variance of the best parameter vector $\boldsymbol{u}$ chosen in hindsight. Two parameter spaces in $\mathbb{R}^n$ are considered—the probability simplex and the unit sphere. The first space is associated with the problem of minimizing risk in stock portfolios and the second space leads to an online calculation of the eigenvector with minimum eigenvalue of the total covariance matrix $\sum_{t=1}^{T}\boldsymbol{C}^t$. For the first parameter space we apply the Exponentiated Gradient algorithm which is motivated with a relative entropy regularization. In the second case, the algorithm has to maintain uncertainty information over all unit directions $\boldsymbol{u}$. For this purpose, directions are represented as dyads $\boldsymbol{u}\boldsymbol{u}^\top$ and the uncertainty over all directions as a mixture of dyads which is a density matrix. The motivating divergence for density matrices is the quantum version of the relative entropy and the resulting algorithm is a special case of the Matrix Exponentiated Gradient algorithm. In each of the two cases we prove bounds on the additional total variance incurred by the online algorithm over the best offline parameter.

Keywords Hedge algorithm · Weighted majority algorithm · Online learning · Expert setting · Density matrix · Matrix exponentiated gradient algorithm · Quantum relative entropy

Editor: Nicolo Cesa-Bianchi.

Supported by NSF grant IIS 0325363. Some of this work was done while visiting National ICT Australia in Canberra.

M.K. Warmuth (✉)
UC California, Santa Cruz, USA
e-mail: manfred@cse.ucsc.edu

D. Kuzmin
Google, Mountain View, USA
e-mail: dimakuzmin@google.com

1 Introduction

In one of the simplest models of online learning in the expert setting (Freund and Schapire 1997), the learner allocates a probability vector $\boldsymbol{w}$ over the experts at the beginning of each trial. It then receives a loss vector $\boldsymbol{l}$ and incurs loss $\boldsymbol{w} \cdot \boldsymbol{l} = \sum_i w_i l_i$. The goal is to design online algorithms with small *regret*, which is the total loss of the online algorithm minus the total loss of the best expert chosen in hindsight: i.e. $\sum_t \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t - \inf_i \sum_t l_i^t$, where t is the trial index.

In this paper we investigate online algorithms for minimizing the total variance over a sequence of trials. Instead of receiving a loss vector $\boldsymbol{l}$ in each trial, we now receive a covariance matrix $\boldsymbol{C}$ of a random loss vector $\boldsymbol{l}$, where $\boldsymbol{C}(i, j)$ is the covariance between l_i and l_j at the current trial. Intuitively the loss vector provides first-order information (means), whereas covariance matrices give second order information. The variance/risk of the loss for probability vector $\boldsymbol{w}$ when the covariance matrix is $\boldsymbol{C}$ can be expressed as $\boldsymbol{w}^\top \boldsymbol{C} \boldsymbol{w}$. If $\boldsymbol{C}(i, j)$ is the covariance between ℓ_i and ℓ_j, then this risk is also the variance of $\boldsymbol{w} \cdot \boldsymbol{l}$, i.e. $\boldsymbol{w}^\top \boldsymbol{C} \boldsymbol{w} = \mathrm{Var}(\boldsymbol{w} \cdot \boldsymbol{l})$. We want online algorithms with small regret in the variance, which is the total variance of the algorithm minus the total variance of the best probability vector $\boldsymbol{u}$ chosen in hindsight: $\sum_t (\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - \inf_{\boldsymbol{u}} \boldsymbol{u}^\top (\sum_t \boldsymbol{C}^t) \boldsymbol{u}$ (where the inf is over the n dimensional probability simplex).

In a more general setting one wants to optimize tradeoffs between first-order and second order terms, i.e. the loss in each trial has the form $\gamma\ \boldsymbol{w} \cdot \boldsymbol{l} + \boldsymbol{w}^\top \boldsymbol{C} \boldsymbol{w}$, where $\gamma \geq 0$ is a tradeoff parameter. Such problems arise in Markowitz portfolio optimization where a linear loss is traded off against a quadratic risk (see e.g. discussion in Boyd and Vandenberghe 2004, Sect. 4.4).

We develop an algorithm for this online variance minimization problem. The parameter space is the probability simplex in $\mathbb{R}^n$. We use the Exponentiated Gradient algorithm for solving this problem since it maintains a probability vector. The latter algorithm is motivated and analyzed using the relative entropy between probability vectors as a measure of progress (Kivinen and Warmuth 1997). The bounds we obtain are similar to the bounds of the Exponentiated Gradient algorithm when applied to linear regression with respect to the square loss.

In the second part of the paper we focus on a similar online variance minimization problem, but now the parameter space that we compare against is the unit sphere of direction vectors in $\mathbb{R}^n$ instead of the probability simplex and the total loss of the algorithm is to be close to $\inf_{\boldsymbol{u}} \boldsymbol{u}^\top (\sum_t \boldsymbol{C}^t) \boldsymbol{u}$, where the minimization is over unit vectors. The solution of the offline problem is an eigenvector that corresponds to a minimum eigenvalue of the total covariance $\sum_t \boldsymbol{C}^t$.

Note that the variance $\boldsymbol{u}^\top \boldsymbol{C} \boldsymbol{u}$ can be rewritten using the trace operator: $\boldsymbol{u}^\top \boldsymbol{C} \boldsymbol{u} = \mathrm{tr}(\boldsymbol{u}^\top \boldsymbol{C} \boldsymbol{u}) = \mathrm{tr}(\boldsymbol{u}\boldsymbol{u}^\top \boldsymbol{C})$. The outer product $\boldsymbol{u}\boldsymbol{u}^\top$ for unit $\boldsymbol{u}$ is called a *dyad* and the offline problem can be reformulated as minimizing trace of a product of a dyad with the total covariance matrix: $\inf_{\boldsymbol{u}} \mathrm{tr}(\boldsymbol{u}\boldsymbol{u}^\top (\sum_t \boldsymbol{C}^t))$ (where $\boldsymbol{u}$ is unit length).

In the original experts setting, the offline problem involved a minimum over n experts. Now this becomes a minimum over infinitely many dyads of dimension n and the best dyads correspond to eigenvectors with minimum eigenvalue. The algorithm for the original expert setting is the Hedge algorithm of Freund and Schapire (1997). It maintains its uncertainty over which expert is best as a probability vector $\boldsymbol{w}$, i.e. w_i is the current belief that expert i is best. Specifically, the probability vector of the Hedge algorithm has the following

exponential form:

$$w_i^{t-1} = \frac{e^{-\eta \sum_{q=1}^{t-1} l_i^q}}{Z^t},$$

$$\text{where } Z^{t-1} = \sum_j e^{-\eta \sum_{q=1}^{t-1} l_j^q} \text{ normalizes the total weight to one.} \tag{1.1}$$

Note that except for the normalization, the i-th weight w_i^{t-1} is exponentially decaying in the current total loss $\sum_{q=1}^{t-1} l_i^q$ of expert i.

In the generalized setting we need to maintain uncertainty over infinitely many dyads. The natural parameter space is therefore mixtures of dyads which are called density matrices in statistical physics (symmetric positive definite matrices of trace one). The vector of eigenvalues of such matrices is a probability vector. Using the methodology of Tsuda et al. (2005), Warmuth and Kuzmin (2006a) we develop a matrix version of the Hedge algorithm for solving our second online variance minimization problem. Now the density matrix parameter has the form

$$\boldsymbol{W}^{t-1} = \frac{\exp(-\eta \sum_{q=1}^{t-1} \boldsymbol{C}^q)}{Z^{t-1}},$$

$$\text{where } Z^{t-1} = \operatorname{tr}\left(\exp\left(-\eta \sum_{q=1}^{t-1} \boldsymbol{C}^q\right)\right) \text{ normalizes the trace to one.} \tag{1.2}$$

Here exp denotes the matrix exponential. This update was also independently developed in Arora and Kale (2007) in the context of convex optimization. When the covariance matrices $\boldsymbol{C}^q$ are the diagonal matrices $\operatorname{diag}(\boldsymbol{l}^q)$ then the density matrix parameter is a diagonalized probability vector and the matrix update (1.2) becomes the original expert update (1.1). In other words the original Hedge algorithm may be seen as a special case of the new matrix algorithm when the eigenvectors are fixed to the standard basis vectors and are not updated. We therefore call the new matrix version of the algorithm *Matrix Hedge*.

The parameter vector of the original Hedge algorithm (1.1) for a finite set of n experts may be seen as a softmin calculation of the total loss vector $\sum_{q=1}^{t-1} \boldsymbol{l}^q$, because as $\eta \to \infty$, the parameter vector $\boldsymbol{w}^{t-1}$ becomes the uniform distribution over the set $\arg\min_i \sum_{q=1}^{t-1} l_i^q$. Similarly, the matrix form (1.2) is a soft minimum eigenvector calculation of the total covariance matrix $\sum_{q=1}^{t-1} \boldsymbol{C}^q$, i.e. as $\eta \to \infty$, the density matrix $\boldsymbol{W}^{t-1}$ becomes the uniform density matrix over subspace spanned by eigenvectors of $\sum_{q=1}^{t-1} \boldsymbol{C}^q$ with minimum eigenvalue.

What replaces the loss $\boldsymbol{w} \cdot \boldsymbol{l}$ of the algorithm in the more general context? The dot product for matrices is a trace and we use the generalized loss $\operatorname{tr}(\boldsymbol{W}\boldsymbol{C})$. If the eigendecomposition of the parameter matrix $\boldsymbol{W}$ consists of the eigenvectors $\boldsymbol{w}_i$ and the associated eigenvalues ω_i, then this loss can be rewritten as

$$\operatorname{tr}(\boldsymbol{W}\boldsymbol{C}) = \operatorname{tr}\left(\left(\sum_i \omega_i \boldsymbol{w}_i \boldsymbol{w}_i^\top\right)\boldsymbol{C}\right) = \sum_i \omega_i \boldsymbol{w}_i^\top \boldsymbol{C} \boldsymbol{w}_i.$$

In other words, the trace may be seen as an expected variance along the eigenvectors $\boldsymbol{w}_i$ that is weighted by the eigenvalues ω_i. Curiously enough, this trace is also a quantum measurement, where $\boldsymbol{W}$ represents a mixture state of a particle and $\boldsymbol{C}$ the instrument (see Warmuth and Kuzmin 2010 for additional discussion). Again the dot product $\boldsymbol{w} \cdot \boldsymbol{l}$ is retained as special

case when the eigenvectors are the standard basis vectors, i.e.

$$\operatorname{tr}(\operatorname{diag}(\boldsymbol{w})\operatorname{diag}(\boldsymbol{l})) = \operatorname{tr}\left(\left(\sum_i w_i \boldsymbol{e}_i \boldsymbol{e}_i^\top\right)\operatorname{diag}(\boldsymbol{l})\right) = \sum_i w_i \boldsymbol{e}_i^\top \operatorname{diag}(\boldsymbol{l})\boldsymbol{e}_i = \sum_i w_i l_i.$$

In this paper we motivate and analyze the Matrix Hedge algorithm (1.2) using the quantum relative entropy (see e.g. Nielsen and Chuang 2000) as a measure of progress in place of the standard relative entropy that is used to analyze the original Hedge algorithm. We are able to generalize the original bounds to the density matrix case by employing the Golden-Thompson inequality and some lemmas developed in Tsuda et al. (2005).

It is important to note that we have no practical machine learning application for the second variance minimization problem over the unit sphere. However, the quantum relative entropy is a Bregman divergence and in a different paper (Warmuth and Kuzmin 2006c, 2008) we showed that by employing the Bregman projection methods developed in Herbster and Warmuth (2001), the Matrix Hedge algorithm leads to an on-line update for Principal Component Analysis. Also efficient semi-definite programming algorithms have been devised based on Matrix Hedge (Arora et al. 2005) and better approximation algorithms for NP-hard problems have been obtained with this algorithm (Arora and Kale 2007). Finally, in a recent paper Matrix Hedge was employed to collapse two important quantum complexity classes (QIP = PSPACE) (Jain et al. 2010).

1.1 Linear versus quadratic loss

In this section we discuss why the loss for our first variance minimization problem is quadratic, but for the second problem is linear. The two online learning problems in this paper have the same basic seemingly quadratic loss $\boldsymbol{u}^\top \boldsymbol{C}^t \boldsymbol{u}$ at trial t, but use a different *comparison class* from which the parameter $\boldsymbol{u}$ is chosen and against which we measure regret. The first problem uses the n dimensional probability simplex and the second problem the sphere of unit vectors $\{\boldsymbol{u} \in \mathbb{R}^n : \|\boldsymbol{u}\|_2 = 1\}$. In both cases we want to minimize the regret, which is the loss of the on-line algorithm on all covariance matrices $\boldsymbol{C}^1, \ldots, \boldsymbol{C}^T$ minus the loss of the best comparator, $\inf_{\boldsymbol{u}} \boldsymbol{u}^\top (\sum_{t=1}^T \boldsymbol{C}^t)\boldsymbol{u}$, chosen in hindsight.

Our algorithms are based on convex optimization and therefore our parameter class used by algorithm must be convex. The parameter class may be slightly larger than the comparison class, but it must be chosen so that we still obtain meaningful regret bounds.

For the first problem, the comparison class from which the off-line comparator $\boldsymbol{u}$ is chosen is the n-dimensional probability simplex S^n and the algorithm chooses its parameter $\boldsymbol{w}^{t-1}$ at the beginning of trial t from the same class. In that case, the regret becomes

$$\sum_{t=1}^T (\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - \inf_{\boldsymbol{u} \in S^n} \boldsymbol{u}^\top \left(\sum_{t=1}^T \boldsymbol{C}^t\right)\boldsymbol{u},$$

where the parameter vector $\boldsymbol{w}^{t-1}$ of the algorithm chosen at the beginning of trial t lies in S^n. Note that the simplex S^n consists of the convex hull of the n unit vectors in $\mathbb{R}^n$ and the losses are quadratic in the mixture coefficients. For the second problem we measure the regret against the set of unit vectors $\{\boldsymbol{u} : \|\boldsymbol{u}\|_2 = 1\}$. First note that this comparison class is not convex, because the average between any pair of units $\boldsymbol{u}$ and $-\boldsymbol{u}$ is $\boldsymbol{0}$, which is not unit. The key insight is to use the set of dyads $\{\boldsymbol{u}\boldsymbol{u}^\top : \|\boldsymbol{u}\|_2 = 1\}$ as our set of "elementary events" instead of the original set of units. The set of dyads is also not convex, but we use the convex hull of the set of dyads (i.e. density matrices) $\{\sum_i \omega_i^{t-1} \boldsymbol{w}_i^{t-1} (\boldsymbol{w}_i^{t-1})^\top : (\omega_1^{t-1}, \ldots,$

$\boldsymbol{\omega}_n^{t-1}) \in S^n$, and $\|\boldsymbol{w}_i^{t-1}\| = 1$, for $1 \leq i \leq n\}$ as the parameter class of our algorithms at the beginning of trial t. This convex hull (denoted as $\mathbb{S}^n$) is the cone of positive definite matrices intersected with the linear constraint that the trace of the matrices is one. We let the loss of a convex combination of dyads be the convex combination of their losses. This allows us to write the loss as the trace:

$$\sum_i \boldsymbol{\omega}_i^{t-1} (\boldsymbol{w}_i^{t-1})^T \boldsymbol{C} \boldsymbol{w}_i^{t-1} = \operatorname{tr}\Bigg(\underbrace{\sum_i \boldsymbol{\omega}_i^{t-1} \boldsymbol{w}_i^{t-1} (\boldsymbol{w}_i^{t-1})^\top}_{\boldsymbol{W}_{t-1}} \boldsymbol{C}\Bigg).$$

Note that for any covariance matrix, this linear loss over the set of positive definite matrices is minimized at a single dyad, and the dyads are the elementary events representing the unit vectors (the comparison vectors for our second problem). Thus we can readily extend the comparison class from the set of dyads to the set of density matrices because:

$$\inf_{\boldsymbol{u}:\|\boldsymbol{u}\|=1} \boldsymbol{u}^\top \left(\sum_{t=1}^T \boldsymbol{C}^t\right) \boldsymbol{u} = \inf_{\boldsymbol{u}:\|\boldsymbol{u}\|=1} \operatorname{tr}\left(\sum_{t=1}^T \boldsymbol{u}\boldsymbol{u}^\top \boldsymbol{C}^t\right) = \inf_{\boldsymbol{U}\in\mathbb{S}^n} \operatorname{tr}\left(\boldsymbol{U} \sum_{t=1}^T \boldsymbol{C}^t\right).$$

Now the regret for the second problem becomes

$$\sum_{t=1}^T \operatorname{tr}(\boldsymbol{W}^{t-1}\, \boldsymbol{C}^t) - \inf_{\boldsymbol{U}\in\mathbb{S}^n} \operatorname{tr}\left(\boldsymbol{U} \sum_{t=1}^T \boldsymbol{C}^t\right),$$

where the parameter matrix $\boldsymbol{W}^{t-1}$ of the algorithm chosen at the beginning of trial t lies in $\mathbb{S}^n$.

Note that predicting with the density matrix $\boldsymbol{W} = \sum_i \boldsymbol{\omega}_i \boldsymbol{w}_i (\boldsymbol{w}_i)^\top$ and incurring loss $\operatorname{tr}(\boldsymbol{W}\, \boldsymbol{C})$ is equivalent to predicting with dyad $\boldsymbol{w}_i (\boldsymbol{w}_i)^\top$ with probability $\boldsymbol{\omega}_i$ and incurring loss $(\boldsymbol{w}_i)^T \boldsymbol{C} \boldsymbol{w}_i$. This parallels what was done in the experts setting. In that case, the linear loss was extended to any convex combination of experts (unit vectors) but this linear loss over the probability simplex is still minimized at single experts. Also predicting with a probability vector $\boldsymbol{w}$ and incurring the linear loss $\boldsymbol{w} \cdot \boldsymbol{l}$ is the same as probabilistically choosing expert i with probability vector w_i and incurring the loss l_i of that expert. In contrast, the quadratic loss used in the first problem may be strictly smaller in the interior of the parameter class.

1.2 Relation to previous work

We now answer why the two problems we focus on in this paper are not covered by some of the recently developed general frameworks for proving regret bounds for on-line algorithms. On-line algorithms are typically motivated by minimization problems that trade off a regularization term against a loss on the last example (Kivinen and Warmuth 1997). There are two main families of on-line algorithms that are informally called the "additive" and "multiplicative" updates. The additive family results from regularizing with the squared Euclidean distance, whereas multiplicative family is based on regularizing with the relative entropy. Regret bounds for the additive family (which includes the perceptron and the Widrow-Hoff algorithms) are typically much easier to obtain. The algorithms analyzed in this paper belong to the multiplicative family (which includes the Winnow algorithm). In general, the two families of updates are incomparable. Depending on the loss function and the assumptions on the instances and the parameter class that we measure regret against, either of the

two families of updates has the advantage. This incomparability was discussed extensively in the case of linear regression in Kivinen and Warmuth (1997). Here we choose to focus on the multiplicative family because we wanted algorithms whose regret bounds have a logarithmic dependence in the matrix dimension n. Beginning with the work on Winnow (Littlestone 1988), regret bounds for multiplicative updates have been obtained that exhibit this logarithmic dependence. Here we generalize the multiplicative family to the matrix case while retaining the logarithm dependence. It has been shown recently that the Matrix Hedge algorithm that we developed for the second problem can be kernelized when the covariance matrices are outer products (Kuzmin and Warmuth 2007). Now the $\log n$ dependence of the algorithms lets us handle a large feature dimension n.

Clearly all the losses discussed in this paper are convex and therefore the general regret bounds of Zinkevich (2003) immediately apply since they only assume that the loss is convex. However, all bounds proven in that paper are for the gradient descent update (belonging to the additive family) and regret bound are linear in the dimension n of the problem and grow with square root of the number of trials T. All our regret bounds have the same dependence on T but grow logarithmically in n, the dimension of the covariance matrices.

More assumptions on the loss function in addition to convexity lead to stronger regret bounds. Let us recall the loss functions considered in this paper. The first problem uses the quadratic loss function $\boldsymbol{w}^\top \boldsymbol{C}\boldsymbol{w}$, where $\boldsymbol{w}$ is a parameter vector on the probability simplex and the instance matrix $\boldsymbol{C}$ is symmetric positive definite with the entries in a fixed range. The loss of our first problem is related to the loss used in linear regression. For an instance vector $\boldsymbol{x}$ and real label y, the linear regression loss is defined as $(\boldsymbol{w} \cdot \boldsymbol{x} - y)^2$. For $y = 0$ this loss becomes $\boldsymbol{w}^\top \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{w}$, which coincides with the loss of our first problem when $\boldsymbol{C}$ equals the dyad $\boldsymbol{x}\boldsymbol{x}'$. In this paper we prove relative loss bounds for arbitrary symmetric matrices $\boldsymbol{C}$ with bounded entries. The loss of the second problem is $\mathrm{tr}(\boldsymbol{W}\boldsymbol{C})$, where the parameter $\boldsymbol{W}$ is a density matrix and the instance matrix $\boldsymbol{C}$ is symmetric positive definite with a bounded range of eigenvalues. Clearly this loss is linear in the parameter $\boldsymbol{W}$.

Two assumption lead to loss bounds for the additive family that grow logarithmic with T and linearly in n: strong convexity and exp-concavity (Hazan et al. 2007). A loss function $\ell(\boldsymbol{w}, \boldsymbol{x}, y)$ is *strongly convex* if for any parameter vector $\boldsymbol{w}$, instance vector $\boldsymbol{x}$, and label y, the minimum eigenvalue of the Hessian of the loss with respect to the parameter $\boldsymbol{w}$ is at least α for some positive α. (Neither of our losses are strongly convex.) A function $f(z)$ is α-exp-concave if $\exp(-\alpha f(z))$ is concave for some $\alpha \geq 0$. The loss function of the first problem is exp-concave (along with the related square loss used for linear regression). Exp-concavity was first used for proving regret bounds for multiplicative updates against the best expert (Kivinen and Warmuth 1999) (see also Cesa-Bianchi and Lugosi 2006). More recently, regret bounds were proven for additive updates when the loss is exp-concave and regret is measured against the loss of the best parameter vector $\boldsymbol{w}$. However, these bounds are linear in n and logarithmic in T. In contrast, the regret bound proven here for the first problem are logarithmic in n and grow with square root of T.

For our second problem the loss is linear in the matrix parameter. Linear losses are neither strongly convex nor exp-concave. As discussed already, this problem has the Hedge setting for learning with experts as a special case. However linear losses are also special within the expert framework in that there is a square-root term in the regret bound. This term disappears when the loss is exp-concave (Kivinen and Warmuth 1999).

There are two common methods for proving relative loss bounds. The first method uses the motivating divergence as a measure of progress (Kivinen and Warmuth 1997). The regret bound for the first problem employs the standard relative entropy and is similar to the regret bound for the Exponentiated Gradient algorithm for the square loss. For the second problem

we use the quantum relative entropy between density matrices. Our initial proof is based on the first method and is similar to the regret bound proven in Tsuda et al. (2005) for a matrix generalization of Exponentiated Gradient update for the case when the loss function is a matrix generalization of the quadratic loss for linear regression. The second proof method makes use of a potential function (Kivinen and Warmuth 1999; Cesa-Bianchi and Lugosi 2006). For the sake of completeness we reprove our regret bounds for the more novel second problem based on this second method in Appendixes A, B and C.

In much earlier work, regret bounds for both the additive and the multiplicative family of algorithms have been proven for *matching loss functions* which are generalizations of the linear regression loss (Helmbold et al. 1999; Kivinen and Warmuth 2001). All these losses depend (non-linearly) on $\boldsymbol{w} \cdot \boldsymbol{x}$. The produced regret bound are logarithmic in the dimension of the vectors. However it is not clear how to generalize the matching loss setup to matrix instances.

Regret bounds based on rather general potential function for the linear loss in the standard expert setting were given in Gordon (2006). Changing the potential function amounts to changing the regularization. All bounds proven there are for the vector parameter case. These methods can probably be generalized to the matrix case discussed here, where the experts correspond to dyads. In doing so, the matrix inequalities used in this paper and in Tsuda et al. (2005) would invariably have to be used.

Rather general methodologies for proving regret bounds based on Fenchel duality and strong convexity have been launched in Shalev-Shwartz and Singer (2006), and recently been applied in the matrix domain (Kakade and Tewari 2010).

The regret bounds of this paper can most likely be proven using that methodology as well, but we already gave two alternated proof methods for the regret bounds of our second problem.

In this paper we show how different views of the same loss naturally lead to updates on the probability simplex verses the simplex of positive definite matrices.

1.3 Outline of the paper

The paper is organized as follows. The first half deals with variance minimization over the probability simplex. In this case the variance describes the risk of a stock portfolio and the goal is to minimize this risk. Section 2.1 gives the necessary definitions. Section 2.2 motivates the Exponentiated Gradient algorithm for this setting and describes the derivation of the algorithm. After this, in Sect. 2.3, we give a proof of relative loss bounds for this algorithm. Next in Sect. 2.4 we analyze a version of the algorithm that trades off the mean/return and the variance/risk. This generalization is again motivated by optimizing stock portfolios. The more involved proof of the relative loss bounds for the tradeoff case is relegated to Appendix A.

In the second half of the paper we deal with the problem of variance minimization over the unit sphere. Section 3.1 introduces the relevant matrix mathematics and other definitions. In Sect. 3.2 we introduce the online algorithm for this problem which maintains a density matrix as its parameter and is motivated by a quantum relative entropy as the parameter divergence. After this, in Sect. 3.3, we set the stage by re-deriving the relative loss bound for the Hedge algorithm using relative entropy as a measure of progress. Our new algorithm can be seen as a matrix generalization of the Hedge algorithm. An analogous bound for the matrix case is given in Sect. 3.4. For the sake of completeness we sketch in Appendix B a number of other alternate proofs of the same basic loss bound that are based on potentials and Bregman projections. The proof methods for the original Hedge algorithm go back to

the Continuous Weighted Majority algorithm and the original bounds were proven for a range of update factors. Analogously in Appendix C we show that the basic loss bound still holds if a range of matrix factors are used in the density matrix update. Concluding thoughts and observations are given in Sect. 4.

An earlier version of this paper appeared as a conference paper in Warmuth and Kuzmin (2006b). This journal version essentially has the following additional material: In the part that deals with portfolio optimization (Sect. 2), we added the analysis of an algorithm that trades off the mean/return and the variance/risk (Sect. 2.4). Also regarding the material about Matrix Hedge (Sect. 3), we added Appendices B and C. Appendix B contains alternate proof techniques for analyzing Matrix Hedge and Appendix C discusses alternate parameter updates that achieve the same loss bound.

2 Variance minimization over the probability simplex

The problem of minimizing the variance when the direction $\boldsymbol{w}$ lies in the probability simplex is connected to risk minimization in stock portfolios. In Markowitz portfolio theory, vector $\boldsymbol{p}$ denotes the relative price change of all assets in a given trading period. Let $\boldsymbol{w}$ be a probability vector that specifies the proportion of our capital invested into each asset (assuming short positions are not allowed). Then the relative capital change after a trading period is the dot product $\boldsymbol{w} \cdot \boldsymbol{p}$. If $\boldsymbol{p}$ is a random vector with known or estimated covariance matrix $\boldsymbol{C}$, then the variance of the relative capital change $\boldsymbol{w} \cdot \boldsymbol{p}$ for our portfolio $\boldsymbol{w}$ is $\boldsymbol{w}^\top \boldsymbol{C} \boldsymbol{w}$. This variance is clearly associated with the risk of our investment. We begin with some definitions, motivate an algorithm and then prove regret bound that quantify the additional total variance of the algorithm over the total variance of the best portfolio chosen in hindsight. We then the introduce a more general case where the variance is traded off against a linear loss. The bounds for the more general case are relegated to Appendixes A, B and C.

2.1 Definitions

In this paper we only consider symmetric matrices. Such matrices always have an eigendecomposition of the form $\boldsymbol{W} = \boldsymbol{\mathcal{W}} \boldsymbol{\omega} \boldsymbol{\mathcal{W}}^\top$, where $\boldsymbol{\mathcal{W}}$ is an orthogonal matrix of eigenvectors and $\boldsymbol{\omega}$ is a diagonal matrix of the corresponding eigenvalues. Alternatively, the decomposition can be written as $\boldsymbol{W} = \sum_i \omega_i \boldsymbol{w}_i \boldsymbol{w}_i^\top$, with the ω_i being the eigenvalues and the $\boldsymbol{w}_i$ the eigenvectors. Note that the dyads $\boldsymbol{w}_i \boldsymbol{w}_i^\top$ are square matrices of rank one.

Matrix $\boldsymbol{A}$ is called *positive definite* (denoted as $\boldsymbol{A} \succeq \boldsymbol{0}$) if for all vectors $\boldsymbol{w}$ we have $\boldsymbol{w}^\top \boldsymbol{A} \boldsymbol{w} \geq 0$. In terms of eigenvalues, positive definiteness means that all eigenvalues are non-negative. A matrix is *strictly positive definite* if all eigenvalues are positive.

Let $\boldsymbol{l}$ be a random vector, then $\boldsymbol{C} = \boldsymbol{E}((\boldsymbol{l} - \boldsymbol{E}(\boldsymbol{l}))(\boldsymbol{l} - \boldsymbol{E}(\boldsymbol{l}))^\top)$ is its *covariance matrix*. Such matrices are symmetric and positive definite. For any other vector $\boldsymbol{w}$ we can compute the variance of the dot product $\boldsymbol{w} \cdot \boldsymbol{l}$ as follows:

$$\begin{aligned}
\operatorname{Var}(\boldsymbol{w} \cdot \boldsymbol{l}) &= \boldsymbol{E}\big(\big(\boldsymbol{l}^\top \boldsymbol{w} - \boldsymbol{E}\big(\boldsymbol{l}^\top \boldsymbol{w}\big)\big)^2\big) \\
&= \boldsymbol{E}\big(\big(\big(\boldsymbol{l}^\top - \boldsymbol{E}\big(\boldsymbol{l}^\top\big)\big)\boldsymbol{w}\big)^\top \big(\big(\boldsymbol{l}^\top - \boldsymbol{E}\big(\boldsymbol{l}^\top\big)\big)\boldsymbol{w}\big)\big) \\
&= \boldsymbol{E}\big(\boldsymbol{w}^\top \big(\boldsymbol{l} - \boldsymbol{E}(\boldsymbol{l})\big)\big(\boldsymbol{l} - \boldsymbol{E}(\boldsymbol{l})\big)^\top\big)\boldsymbol{w}\big) \\
&= \boldsymbol{w}^\top \boldsymbol{C} \boldsymbol{w}.
\end{aligned}$$

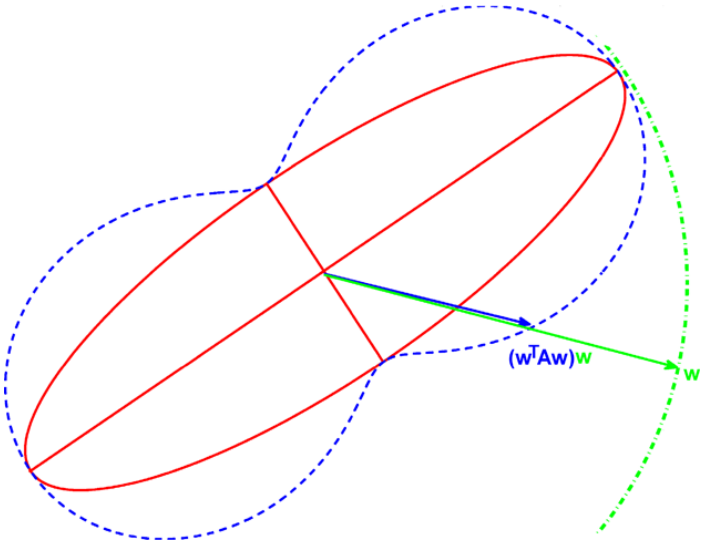


Fig. 1 An *ellipse* $\boldsymbol{A}$ in $\mathbb{R}^2$: The eigenvectors are the directions of the axes and the eigenvalues their lengths from the origin. *Ellipses* are weighted combinations of the one-dimensional degenerate ellipses (dyads) corresponding to the axes. (For unit direction $\boldsymbol{w}$, the dyad $\boldsymbol{w}\boldsymbol{w}^\top$ is a degenerate one-dimensional ellipse which is a line between $-\boldsymbol{w}$ and $\boldsymbol{w}$ on the *green dash-dotted unit circle*.) The *solid red curve* of the ellipse is a plot of direction vector $\boldsymbol{A}\boldsymbol{w}$ and the *blue dashed figure* eight is unit $\boldsymbol{w}$ on the *green dash-dotted unit circle* times the variance $\boldsymbol{w}^\top \boldsymbol{A}\boldsymbol{w}$. At the eigenvectors, this variance equals the eigenvalues and the *dashed curve* touches the ellipse

A covariance matrix can be depicted as an ellipse $\{\boldsymbol{C}\boldsymbol{w} \,:\, \|\boldsymbol{w}\|_2 = 1\}$ centered at the origin. The eigenvectors of $\boldsymbol{C}$ form the axes of the ellipse and eigenvalues are the lengths of the axes from the origin (see Fig. 1).

For two probability vectors $\boldsymbol{u}$ and $\boldsymbol{w}$ (e.g. vectors whose entries are nonnegative and sum to one) their relative entropy (or Kullback-Leibler divergence) is given by:

$$\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}) = \sum_{i=1}^{n} u_i \log \frac{u_i}{w_i}. \tag{2.1}$$

We call this a divergence (and not a distance) since its not symmetric and does not satisfy the triangle inequality. It is however nonnegative and convex in either argument. In the second part of the paper we deal with density matrix parameters and there we use the matrix generalization of the relative entropy (see Sect. 3.1).

2.2 Algorithm and motivation

Let us reiterate the setup and the goal for our algorithm. On every trial t it must produce a probability vector $\boldsymbol{w}^{t-1}$. It then gets a covariance matrix $\boldsymbol{C}^t$ and incurs a loss equal to the variance:

$$L^t(\boldsymbol{w}^{t-1}) := (\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1}.$$

Thus for a sequence of T trials, the total loss of the algorithm will be

$$L_{\mathrm{alg}} := \sum_{t=1}^{T} L^t(\boldsymbol{w}^{t-1}).$$

We want this loss to be comparable to the total loss of the best probability vector chosen in hindsight, where the loss of a probability vector $\boldsymbol{u}$ is

$$L_{\boldsymbol{u}} := \sum_{t=1}^{T} L^t(\boldsymbol{u}).$$

This offline problem is a quadratic optimization problem with non-negativity constraints which does not have a closed form solution. However we can still prove bounds for the online algorithm.

A natural choice for an online algorithm for this problem is the Exponentiated Gradient algorithm of Kivinen and Warmuth (1997) since it maintains a probability vector as its parameter. Recall that for a general loss function $L^t(\boldsymbol{w}^{t-1})$, the probability vector of Exponentiated Gradient algorithm is updated as

$$w_i^t = \frac{w_i^{t-1} e^{-\eta(\nabla L^t(\boldsymbol{w}^{t-1}))_i}}{\sum_i w_i^{t-1} e^{-\eta(\nabla L^t(\boldsymbol{w}^{t-1}))_i}}. \tag{2.2}$$

This update is motivated by considering the tradeoff between the relative entropy divergence to the old probability vector and the current loss, where $\eta > 0$ is the tradeoff parameter (which will become the learning rate of the algorithm):

$$\boldsymbol{w}^t = \operatorname*{arg\,min}_{\sum_{i=1}^n w_i = 1} \mathrm{d}\big(\boldsymbol{w}, \boldsymbol{w}^{t-1}\big) + \eta L^t(\boldsymbol{w}). \tag{2.3}$$

The exact solution to the above equation has the form (2.2) except that the gradient $\nabla L^t(\boldsymbol{w}^t))_i$ at the new parameter $\boldsymbol{w}^t$ appears in the exponents instead of the gradient $\nabla L^t(\boldsymbol{w}^{t-1}))_i$ at the old parameter $\boldsymbol{w}^{t-1}$. Since this exact update has $\boldsymbol{w}^t$ appearing on both sides of the update equation, it is called the *implicit* Exponentiated Gradient update (Kivinen et al. 2005). In practice, the approximation (2.2) is often used which is called the (*explicit*) Exponentiated Gradient update and this is the default here as well. Alternatively the explicit update results from approximating the loss $L^t(\boldsymbol{w})$ by its first-order linear expansion via the Taylor formula (see Kivinen and Warmuth 1997 for more discussion).

In our application, $L^t(\boldsymbol{w}^{t-1}) = \frac{1}{2}(\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1}$ and $\nabla L^t(\boldsymbol{w}^{t-1}) = \boldsymbol{C}^t \boldsymbol{w}^{t-1}$, leading to the following (explicit) Exponentiated Gradient update:

$$w_i^t = \frac{w_i^{t-1} e^{-\eta(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i}}{\sum_{i=1}^n w_i^{t-1} e^{-\eta(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i}}.$$

2.3 Proof of relative loss bounds

We now use the divergence $\mathrm{d}(\boldsymbol{u}, \boldsymbol{w})$ that motivated the update as a measure of progress in the analysis. The proof proceeds by bounding the progress to the comparison vector $\boldsymbol{u}$ on two successive trials, as measured by change of the divergence: $\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}^{t-1}) - \mathrm{d}(\boldsymbol{u}, \boldsymbol{w}^t)$. A somewhat different commonly used proof style uses a potential function, and bounds the increase of the potential on two successive trials. Many choices for the potential are possible, one common one is the value of optimization problem (2.3). We contrast the two proof styles in Appendix B when dealing with variance minimization over the sphere. Also, Helmbold and Warmuth (2007) compare the two proof styles for an online algorithm for learning permutations.

Lemma 2.1 *Let $\boldsymbol{w}^t$ be the weight vector of the Exponentiated Gradient algorithm computed at trial t and let $\boldsymbol{u}$ be an arbitrary probability vector that serves as a comparator. Also, let r be the bound on the range of elements in covariance matrix $\boldsymbol{C}^t$, specifically let $\max_{i,j}|\boldsymbol{C}^t(i,j)| \leq \frac{r}{2}$. For any constants a and b such that $0 < a \leq \frac{b}{1+rb}$ and a learning rate $\eta = \frac{2b}{1+rb}$ we have*:

$$a\big(\boldsymbol{w}^{t-1}\big)^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - b\boldsymbol{u}^\top \boldsymbol{C}^t \boldsymbol{u} \leq \mathrm{d}\big(\boldsymbol{u}, \boldsymbol{w}^{t-1}\big) - \mathrm{d}\big(\boldsymbol{u}, \boldsymbol{w}^t\big).$$

Proof The proof follows the same outline as Lemma 5.8 of Kivinen and Warmuth (1997) which gives an inequality for the Exponentiated Gradient algorithm when applied to linear regression with the square loss. We begin by analyzing the progress towards the comparison vector $\boldsymbol{u}$:

$$\begin{aligned}
\mathrm{d}\big(\boldsymbol{u}, \boldsymbol{w}^{t-1}\big) - \mathrm{d}\big(\boldsymbol{u}, \boldsymbol{w}^t\big) &= \sum_i u_i \log \frac{u_i}{w_i^{t-1}} - \sum_i u_i \log \frac{u_i}{w_i^t} \\
&= \sum_i u_i \log w_i^t - \sum_i u_i \log w_i^{t-1} \\
&= \sum_i u_i \log \frac{w_i^{t-1} e^{-\eta(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i}}{\sum_i w_i^{t-1} e^{-\eta(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i}} - \sum_i u_i \log w_i^{t-1} \\
&= \sum_i u_i \log w_i^{t-1} - \eta \sum_i u_i \big(\boldsymbol{C}^t \boldsymbol{w}^{t-1}\big)_i \\
&\quad - \log\Bigg(\sum_i w_i^{t-1} e^{-\eta(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i}\Bigg) - \sum_i u_i \log w_i^{t-1} \\
&= -\eta \sum_i u_i \big(\boldsymbol{C}^t \boldsymbol{w}^{t-1}\big)_i - \log\Bigg(\sum_i w_i^{t-1} e^{-\eta(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i}\Bigg).
\end{aligned}$$

Thus, our bound is equivalent to showing $F \leq 0$ with F given as:

$$F = a\big(\boldsymbol{w}^{t-1}\big)^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - b\boldsymbol{u}^\top \boldsymbol{C}\boldsymbol{u} + \eta \boldsymbol{u}^\top \boldsymbol{C}\boldsymbol{w}^{t-1} + \log\Bigg(\sum_i w_i^{t-1} e^{-\eta(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i}\Bigg).$$

We proceed by bounding the log term. The assumption on the range of elements of $\boldsymbol{C}^t$ and the fact that $\boldsymbol{w}^{t-1}$ is a probability vector allows us to conclude that $\max_i(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i - \min_i(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i \leq r$, since $(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i = \sum_j \boldsymbol{C}^t(i,j)\boldsymbol{w}_j^{t-1}$. Now, assume that l is a lower bound for $(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i$, then we have that $l \leq (\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i \leq l + r$, or $0 \leq \frac{(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i - l}{r} \leq 1$. This allows us to use the inequality $a^x \leq 1 - x(1-a)$ for $a \geq 0$ and $0 \leq x \leq 1$. Let $a = e^{-\eta r}$:

$$e^{-\eta(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i} = e^{-\eta l}\big(e^{-\eta r}\big)^{\frac{(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i - l}{r}} \leq e^{-\eta l}\Bigg(1 - \frac{(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i - l}{r}\big(1 - e^{-\eta r}\big)\Bigg).$$

Using this inequality we obtain:

$$\log\Bigg(\sum_i w_i^{t-1} e^{-\eta(\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i}\Bigg) \leq -\eta l + \log\Bigg(1 - \frac{(\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - l}{r}\big(1 - e^{-\eta r}\big)\Bigg).$$

This gives us $F \leq G$, with G given as:

$$G = a\left(\boldsymbol{w}^{t-1}\right)^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1} - b\boldsymbol{u}^{\top} \boldsymbol{C}^t \boldsymbol{u} + \eta \boldsymbol{u}^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1} - \eta l + \log\left(1 - \frac{(\boldsymbol{w}^{t-1})^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1} - l}{r}\left(1 - e^{-\eta r}\right)\right).$$

It is sufficient to show that $G \leq 0$. Let $\boldsymbol{z} = \sqrt{\boldsymbol{C}^t}\boldsymbol{u}$. Then $G(\boldsymbol{z})$ becomes:

$$G(\boldsymbol{z}) = -b\boldsymbol{z}^{\top}\boldsymbol{z} + \eta \boldsymbol{z}^{\top}\sqrt{\boldsymbol{C}^t}\boldsymbol{w}^{t-1} + \text{constant}.$$

The function $G(\boldsymbol{z})$ is concave quadratic and is maximized at:

$$\frac{\partial G}{\partial \boldsymbol{z}} = -2b\boldsymbol{z} + \eta\sqrt{\boldsymbol{C}^t}\boldsymbol{w}^{t-1} = 0, \quad \boldsymbol{z} = \frac{\eta}{2b}\sqrt{\boldsymbol{C}^t}\boldsymbol{w}^{t-1}.$$

We substitute this value of $\boldsymbol{z}$ into G and get $G \leq H$, where H is given by:

$$H = a\left(\boldsymbol{w}^{t-1}\right)^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1} + \frac{\eta^2}{4b}\left(\boldsymbol{w}^{t-1}\right)^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1} - \eta l + \log\left(1 - \frac{(\boldsymbol{w}^{t-1})^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1} - l}{r}\left(1 - e^{-\eta r}\right)\right).$$

Since $l \leq (\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i \leq l + r$, then obviously so is $(\boldsymbol{w}^{t-1})^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1}$, since the weighted average stays within the bounds. Now we can use the inequality $\log(1 - p(1 - e^q)) \leq pq + \frac{q^2}{8}$, for $0 \leq p \leq 1$ and $q \in \mathbb{R}$ (Helmbold et al. 1997)

$$\log\left(1 - \frac{(\boldsymbol{w}^{t-1})^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1} - l}{r}\left(1 - e^{-\eta r}\right)\right) \leq -\eta\left(\boldsymbol{w}^{t-1}\right)^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1} + \eta l + \frac{\eta^2 r^2}{8}.$$

We get $H \leq S$, where S is given as:

$$\begin{aligned} S &= a\left(\boldsymbol{w}^{t-1}\right)^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1} + \frac{\eta^2}{4b}\left(\boldsymbol{w}^{t-1}\right)^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1} - \eta\left(\boldsymbol{w}^{t-1}\right)^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1} + \frac{\eta^2 r^2}{8} \\ &= \frac{(\boldsymbol{w}^{t-1})^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1}}{4b}\left(4ab + \eta^2 - 4b\eta\right) + \frac{\eta^2 r^2}{8}. \end{aligned}$$

By our assumptions $(\boldsymbol{w}^{t-1})^{\top} \boldsymbol{C}^t \boldsymbol{w}^{t-1} \leq \frac{r}{2}$, and therefore:

$$S \leq Q = \eta^2\left(\frac{r^2}{8} + \frac{r}{8b}\right) - \frac{\eta r}{2} + \frac{ar}{2}.$$

We want to make this expression as small as possible, so that it stays below zero. To do so we minimize it over η:

$$2\eta\left(\frac{r^2}{8} + \frac{r}{8b}\right) - \frac{r}{2} = 0, \quad \eta = \frac{2b}{rb + 1}.$$

Finally we substitute this value of η into Q and obtain conditions on a that guarantees $Q \leq 0$:

$$a \leq \frac{b}{rb + 1}.$$

This concludes the proof. □

The following intermediate lemma gives a bound in terms of a tradeoff parameter c. Note that the learning rate η depends on this parameter.

Lemma 2.2 *Let* $\max_{i,j}|\boldsymbol{C}^t(i,j)| \leq \frac{r}{2}$ *as before. Then for arbitrary positive* c *and learning rate* $\eta = \frac{2c}{r(c+1)}$, *the following bound holds*:

$$L_{\text{alg}} \leq (1+c)L_{\boldsymbol{u}} + \left(1+\frac{1}{c}\right) r\,\mathrm{d}\big(\boldsymbol{u},\boldsymbol{w}^0\big).$$

Proof Let $b = \frac{c}{r}$, then for $a = \frac{b}{rb+1} = \frac{c}{r(c+1)}$ and $\eta = 2a = \frac{2c}{r(c+1)}$, we can use the inequality of Lemma 2.1 and obtain:

$$\frac{c}{c+1}\big(\boldsymbol{w}^{t-1}\big)^{\top}\boldsymbol{C}^t\boldsymbol{w}^{t-1} - c\boldsymbol{u}^{\top}\boldsymbol{C}^t\boldsymbol{u} \leq r\big(\mathrm{d}\big(\boldsymbol{u},\boldsymbol{w}^{t-1}\big) - \mathrm{d}\big(\boldsymbol{u},\boldsymbol{w}^t\big)\big).$$

Summing over the trials t results in:

$$\frac{c}{c+1}L_{\text{alg}} - cL_{\boldsymbol{u}} \leq r\big(\mathrm{d}\big(\boldsymbol{u},\boldsymbol{w}^0\big) - \mathrm{d}\big(\boldsymbol{u},\boldsymbol{w}^t\big)\big) \leq r\,\mathrm{d}\big(\boldsymbol{u},\boldsymbol{w}^0\big).$$

Now the statement of the lemma immediately follows. □

The following theorem describes how to choose the learning rate for the purpose of minimizing the upper bound:

Theorem 2.1 *Let* $\boldsymbol{C}^1,\ldots,\boldsymbol{C}^T$ *be an arbitrary sequence of covariance matrices such that* $\max_{i,j}|\boldsymbol{C}^t(i,j)| \leq \frac{r}{2}$ *and assume that* $\boldsymbol{u}^{\top}\sum_{t=1}^{T}\boldsymbol{C}^t\boldsymbol{u} \leq L$ *for some probability vector* $\boldsymbol{u}$. *Then the Exponentiated Gradient algorithm on the* n*-dimensional probability simplex with uniform start vector* $\boldsymbol{w}^0 = (\frac{1}{n},\ldots,\frac{1}{n})$ *and learning rate* $\eta = \frac{2\sqrt{L\log n}}{r\sqrt{\log n}+\sqrt{rL}}$ *has the following bound*:

$$L_{\text{alg}} \leq L_{\boldsymbol{u}} + 2\sqrt{rL\log n} + r\log n.$$

Proof By Lemma 2.2 and since $\mathrm{d}(\boldsymbol{u},\boldsymbol{w}^0) \leq \log n$:

$$L_{\text{alg}} \leq L_{\boldsymbol{u}} + cL + \frac{r\log n}{c} + r\log n.$$

By differentiating we see that $c = \sqrt{\frac{r\log n}{L}}$ minimizes the r.h.s., and substituting this choice of c gives the bound of the theorem. □

Since $\max_{i,j}|\boldsymbol{C}^t(i,j)| \leq \frac{r}{2}$, the total loss $\boldsymbol{u}^{\top}\sum_{t=1}^{T}\boldsymbol{C}^t\boldsymbol{u}$ is always upper bounded by $\frac{1}{2}rT$. Using this choice of L, the bound of the theorem becomes

$$L_{\text{alg}} \leq L_{\boldsymbol{u}} + \frac{r}{2}\sqrt{T\log n} + r\log n.$$

2.4 Bounds for variance-loss tradeoff case

In each trial the algorithm now receives both a vector of losses $\boldsymbol{l}^t$ and a covariance matrix $\boldsymbol{C}^t$. The loss of probability vector $\boldsymbol{w}^{t-1}$ becomes an additive tradeoff between first-order and

second-order components, i.e. we want to make our loss small, but we want to control the variance as well ($\gamma \geq 0$ is a tradeoff parameter—the bigger it is, the less attention we pay to the variance):

$$L^{t,\gamma}\left(\boldsymbol{w}^{t-1}\right) = \gamma\, \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t + \left(\boldsymbol{w}^{t-1}\right)^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1}. \tag{2.4}$$

Similarly, we define L^γ_{alg} and $L^\gamma_{\boldsymbol{u}}$ by summing this loss over a sequence of trials. Using the EG strategy for this loss results in the following update:

$$w_i^t = \frac{w_i^{t-1} e^{-\eta(\gamma l_i^t + (\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i)}}{\sum_{i=1}^n w_i^{t-1} e^{-\eta(\gamma l_i^t + (\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i)}}. \tag{2.5}$$

The proof follows the same outline as before. Here we only state the corresponding lemmas and theorem. The actual proofs are given in Appendix A.

Lemma 2.3 *Let $\boldsymbol{w}^t$ be the weight vector of the Exponentiated Gradient algorithm computed at trail t and let $\boldsymbol{u}$ be an arbitrary probability vector that serves as a comparator. Also, let r be the bound on the range of elements of covariance matrix $\boldsymbol{C}^t$, that is $\max_{i,j} |\boldsymbol{C}^t(i,j)| \leq \frac{r}{2}$ and let the loss vector $\boldsymbol{l}^t$ be bounded as: $\boldsymbol{l}_i^t \in [0 \ldots q]$. For $\gamma \geq 0$, any constants a and b such that $a \leq \frac{b(r-2\gamma q)^2}{(r+b(r+\gamma q)^2)(r+2\gamma q)}$ and learning rate $\eta = \frac{2b(r-2\gamma q)}{r+b(r+\gamma q)^2}$ the following inequality holds:*

$$a\left(\gamma\, \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t + \left(\boldsymbol{w}^{t-1}\right)^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1}\right) - b\left(\gamma\, \boldsymbol{u} \cdot \boldsymbol{l}^t + \boldsymbol{u}^\top \boldsymbol{C}^t \boldsymbol{u}\right) \leq d\left(\boldsymbol{u}, \boldsymbol{w}^{t-1}\right) - d\left(\boldsymbol{u}, \boldsymbol{w}^t\right).$$

Lemma 2.4 *Let $\max_{i,j} |\boldsymbol{C}^t(i,j)| \leq \frac{r}{2}$ and $\boldsymbol{l}_i^t \in [0 \ldots q]$ as before. Lets use R and Q to denote the following quantities:*

$$R = \frac{(r-2\gamma q)^2}{r(r+2\gamma q)} \quad \text{and} \quad Q = \frac{(r+\gamma q)^2}{r}.$$

Then for arbitrary positive b and learning rate $\eta = \frac{2b(r-2\gamma q)}{r+b(r+\gamma q)^2}$, the following bound holds:

$$L^\gamma_{\text{alg}} \leq \left(\frac{1}{R} + \frac{bQ}{R}\right) L^\gamma_{\boldsymbol{u}} + \frac{1}{bR} d\left(\boldsymbol{u}, \boldsymbol{w}^0\right) + \frac{Q}{R} d\left(\boldsymbol{u}, \boldsymbol{w}^0\right).$$

Theorem 2.2 *Let $\boldsymbol{C}^1, \ldots, \boldsymbol{C}^T$ be an arbitrary sequence of covariance matrices such that $\max_{i,j} |\boldsymbol{C}^t(i,j)| \leq \frac{r}{2}$ and $\boldsymbol{l}^1, \ldots, \boldsymbol{l}^T$ be an arbitrary sequence of loss vectors such that $\boldsymbol{l}_i^t \in [0 \ldots q]$. Additionally assume that $L^\gamma_{\boldsymbol{u}} = \gamma\, \boldsymbol{u} \cdot \sum_{t=1}^T \boldsymbol{l}^t + \boldsymbol{u}^\top \sum_{t=1}^T \boldsymbol{C}^t \boldsymbol{u} \leq L$ for some probability vector $\boldsymbol{u}$. Then the Exponentiated Gradient algorithm on the n-dimensional probability simplex with uniform start vector $\boldsymbol{w}^0 = (\frac{1}{n}, \ldots, \frac{1}{n})$ and learning rate $\eta = \frac{2\sqrt{\log n / QL}(r-2\gamma q)}{r+\sqrt{\log n / QL}(r+\gamma q)^2}$ has the following bound:*

$$L^\gamma_{\text{alg}} \leq L^\gamma_{\boldsymbol{u}} + \frac{2}{R}\sqrt{QL \log n} + \frac{Q}{R} \log n + PL,$$

where Q and R are as in Lemma 2.4 and

$$P = \frac{2\gamma q(3r - 2\gamma q)}{(r-2\gamma q)^2}.$$

Note that when $\gamma = 0$, then the "tradeoff" loss (2.4) used in this section becomes the "covariance loss" used in the previous section, and the bound in the above theorem coincides with the bound of Theorem 2.1 obtained for the covariance loss. When $\gamma \to \infty$ then the covariance term in the tradeoff loss becomes negligible and after dividing by γ the tradeoff loss becomes $\boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t$. Relative loss bound for this commonly analyzed "dot loss" will be reviewed in Sect. 3.3. Unfortunately the known bounds for that loss are stronger than the bounds obtainable from the above theorem by letting γ approach ∞. Thus for large γ the loss bounds of this section are not competitive.

3 Variance minimization over the unit sphere

We now describe and analyze the new Matrix Hedge algorithm that minimizes the variance over the unit sphere. As discussed in the introduction, we represent unit vectors as dyads and use convex combinations of dyads (density matrices) as our parameter space. We begin with some definitions and then motivate the Matrix Hedge algorithm using the quantum relative entropy as a regularization. We then recall how Bregman divergence based proof methods were used to prove regret bounds for the original Hedge algorithm. The divergence employed in the original setting is the relative entropy. Analogous methods lead to regret bounds for Matrix Hedge by using the quantum relative entropy. In an appendix we show how the alternate proof methods commonly used in the expert setting that are based on potential functions also lead to an analysis of Matrix Hedge. For the original Hedge algorithm a range of update factors can be used. We show that an analogous phenomenon happens for Matrix Hedge and we discuss these generalizations in Appendix B.

3.1 Definitions

The *trace* $\operatorname{tr}(\boldsymbol{A})$ of a square matrix $\boldsymbol{A}$ is the sum of its diagonal elements. It is invariant under a change of basis transformation and thus it is also equal to the sum of eigenvalues of the matrix. The trace generalizes the normal dot product between vectors to the space of matrices, i.e. $\operatorname{tr}(\boldsymbol{A}\,\boldsymbol{B}) = \operatorname{tr}(\boldsymbol{B}\boldsymbol{A}) = \sum_{i,j} \boldsymbol{A}(i,j)\boldsymbol{B}(i,j)$. The trace is also a linear operator, that is $\operatorname{tr}(a\boldsymbol{A} + b\boldsymbol{B}) = a\operatorname{tr}(\boldsymbol{A}) + b\operatorname{tr}(\boldsymbol{B})$. Another useful property of the trace is its invariance with respect to cycling, i.e. $\operatorname{tr}(\boldsymbol{A}\boldsymbol{B}\boldsymbol{C}) = \operatorname{tr}(\boldsymbol{B}\boldsymbol{C}\boldsymbol{A}) = \operatorname{tr}(\boldsymbol{C}\boldsymbol{A}\boldsymbol{B})$. A particular instance of this is the following manipulation: $\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{u} = \operatorname{tr}(\boldsymbol{u}^\top \boldsymbol{A}\boldsymbol{u}) = \operatorname{tr}(\boldsymbol{A}\boldsymbol{u}\boldsymbol{u}^\top)$.

Dyads have trace one because $\operatorname{tr}(\boldsymbol{u}\boldsymbol{u}^\top) = \boldsymbol{u}^\top \boldsymbol{u} = 1$. We generalize mixtures or probability vectors to *density matrices*. Such matrices are mixtures of any number of dyads, i.e. $\boldsymbol{W} = \sum_i \lambda_i\, \boldsymbol{u}_i \boldsymbol{u}_i^\top$, where the $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$. Equivalently, density matrices are arbitrary symmetric positive definite matrices of trace one. Any density matrix $\boldsymbol{W}$ can be decomposed into a sum of exactly n dyads corresponding to the orthogonal set of its eigenvectors $\boldsymbol{w}_i$, i.e. $\boldsymbol{W} = \sum_{i=1}^n \omega_i\, \boldsymbol{w}_i \boldsymbol{w}_i^\top$, where the vector $\boldsymbol{\omega}$ of the n eigenvalues is a probability vector. In quantum physics density matrices over the field of complex numbers (instead of the reals) represent the mixed state of a physical system. Throughout the paper we denote matrices with bold roman capital letters, the eigenvalues with same lower case greek letters, and the corresponding eigenvectors with bold lower case roman letters.

We also need the matrix generalizations of the exponential and logarithm operations. Given the eigendecomposition of a symmetric matrix $\boldsymbol{A} = \sum_i \alpha_i\, \boldsymbol{a}_i \boldsymbol{a}_i^\top$, the matrix exponential and logarithm denoted as exp and log are computed as follows:

$$\exp(\boldsymbol{A}) = \sum_i e^{\alpha_i} \boldsymbol{a}_i \boldsymbol{a}_i^\top, \qquad \log(\boldsymbol{A}) = \sum_i (\log \alpha_i) \boldsymbol{a}_i \boldsymbol{a}_i^\top.$$

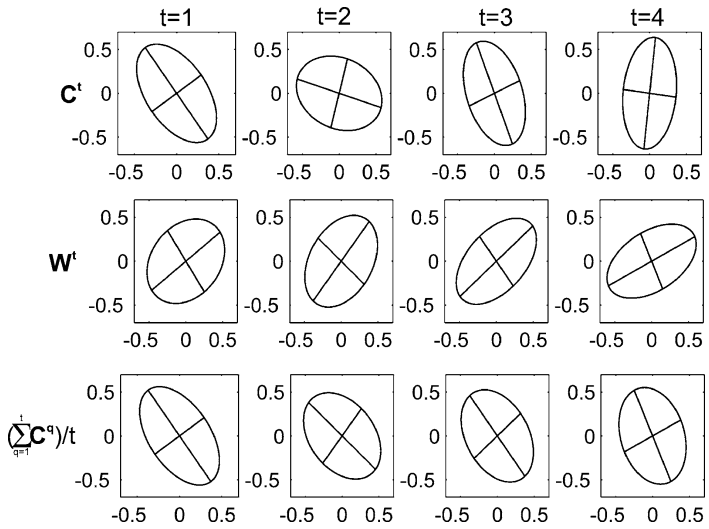


Fig. 2 The figure depicts a sequence of updates for the density matrix algorithm when the dimension is 2 with $\eta = 1$. All 2-by-2 matrices are represented as ellipses. The *top row* shows the covariance matrices $\boldsymbol{C}^t$ received in that trial. The *middle row* shows the density matrices $\boldsymbol{W}^t$ chosen by the algorithm. Finally, the *bottom row* is the average of all covariance matrices so far, i.e. $\frac{\boldsymbol{C}^{\leq t}}{t}$, where $\boldsymbol{C}^{\leq t} = \sum_{q=1}^{t} \boldsymbol{C}^t$. By the update (3.3), $\boldsymbol{W}^t = \frac{\exp(-\eta \boldsymbol{C}^{\leq t})}{Z^t}$, where Z^t is a normalization. Therefore, $\frac{\boldsymbol{C}^{\leq t}}{t}$ in the *third row* has the same eigensystem as the density matrix $\boldsymbol{W}^t$ in the *first row*. Note the tendency of the algorithm to try to place more weight on the minimal eigenvalue of the covariance average. Since the algorithm is not sure about the future, it does not place the full weight onto that eigenvalue but hedges its bets instead and places some weight onto the other eigenvalues as well. This can be seen as a soft minimum calculation and as $\eta \to \infty$ it becomes the normal minimum

In other words, the exponential and the logarithm are applied to the eigenvalues and the eigenvectors remain unchanged. Obviously, the matrix logarithm is only defined when the matrix is strictly positive definite. In analogy with the exponential for numbers, one would expect the following equality to hold for symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$: $\exp(\boldsymbol{A} + \boldsymbol{B}) = \exp(\boldsymbol{A}) \exp(\boldsymbol{B})$. However this is only true when the symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ commute, i.e. $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{B}\boldsymbol{A}$, which occurs iff both matrices share the same eigensystem. On the other hand, the following trace inequality, called the Golden-Thompson inequality, holds for arbitrary symmetric matrices (Bhatia 1997):

$$\operatorname{tr}\bigl(\exp(\boldsymbol{A} + \boldsymbol{B})\bigr) \leq \operatorname{tr}\bigl(\exp(\boldsymbol{A}) \exp(\boldsymbol{B})\bigr). \tag{3.1}$$

The following *quantum relative entropy* is a generalization of the classical relative entropy (2.1) to density matrices due to Umegaki (see e.g. Nielsen and Chuang 2000): For two density matrices $\boldsymbol{U}$ and $\boldsymbol{W}$,

$$\Delta(\boldsymbol{U}, \boldsymbol{W}) := \operatorname{tr}\bigl(\boldsymbol{U}(\log \boldsymbol{U} - \log \boldsymbol{W})\bigr).$$

We will also use generalized inequalities for the cone of positive definite matrices: For two symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $\boldsymbol{A} \preceq \boldsymbol{B}$ if $\boldsymbol{B} - \boldsymbol{A}$ is positive definite.

3.2 Algorithm and motivation

As before we briefly review our setup. On each trial t our algorithm chooses a density matrix $\boldsymbol{W}^{t-1}$ described as a mixture $\sum_i \omega_i^{t-1} \boldsymbol{w}_i^{t-1} (\boldsymbol{w}_i^{t-1})^\top$. It then receives a covariance matrix $\boldsymbol{C}^t$

and incurs a loss equal to the expected variance of its mixture:

$$\operatorname{tr}\bigl(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\bigr) = \operatorname{tr}\biggl(\Bigl(\sum_i \omega_i^{t-1}\,\boldsymbol{w}_i^{t-1}\bigl(\boldsymbol{w}_i^{t-1}\bigr)^\top\Bigr)\boldsymbol{C}^t\biggr) = \sum_i \omega_i^{t-1}\,\bigl(\boldsymbol{w}_i^{t-1}\bigr)^\top \boldsymbol{C}^t\,\boldsymbol{w}_i^{t-1}.$$

On a sequence of T trials the total loss of the algorithm will be $L_{\mathrm{alg}} = \sum_{t=1}^{T} \operatorname{tr}(\boldsymbol{W}^{t-1}\boldsymbol{C}^t)$. We want this loss to be not too much larger than the total variance of best unit vector $\boldsymbol{u}$ chosen in hindsight, i.e. $L_{\boldsymbol{u}} = \operatorname{tr}(\boldsymbol{u}\boldsymbol{u}^\top \sum_t \boldsymbol{C}^t) = \boldsymbol{u}^\top(\sum_t \boldsymbol{C}^t)\boldsymbol{u}$. The set of dyads is not a convex set. We therefore close it by using convex combinations of dyads (i.e. density matrices) as our parameter space. The best offline parameter is still a single dyad, i.e. for any symmetric matrix C,

$$\min_{\operatorname{tr}(\boldsymbol{U})=1} \operatorname{tr}(\boldsymbol{U}\boldsymbol{C}) = \min_{\boldsymbol{u}\,:\,\|\boldsymbol{u}\|_2=1} \boldsymbol{u}^\top \boldsymbol{C}\,\boldsymbol{u}.$$

Curiously enough, our loss $\operatorname{tr}(\boldsymbol{W}\boldsymbol{C})$ has the interpretation in quantum mechanics as the expected outcome of measuring a physical system in mixture state $\boldsymbol{W}$ with instrument $\boldsymbol{C}$. Let $\boldsymbol{C}$ be decomposed as $\sum_i \gamma_i \boldsymbol{c}_i \boldsymbol{c}_i^\top$. The eigenvalues γ_i are the possible numerical outcomes of measurement. When measuring a pure state specified by unit vector $\boldsymbol{u}$, the probability of obtaining outcome γ_i is given as $(\boldsymbol{u}\cdot\boldsymbol{c}_i)^2$ and the expected outcome is $\operatorname{tr}(\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{C}) = \sum_i (\boldsymbol{u}\cdot\boldsymbol{c}_i)^2\gamma_i$. For a mixed state $\boldsymbol{W}$ we have the following double expectation:

$$\operatorname{tr}(\boldsymbol{W}\boldsymbol{C}) = \operatorname{tr}\biggl(\Bigl(\sum_i \omega_i\,\boldsymbol{w}_i\boldsymbol{w}_i^\top\Bigr)\Bigl(\sum_j \gamma_j\,\boldsymbol{c}_j\boldsymbol{c}_j^\top\Bigr)\biggr) = \sum_{i,j} (\boldsymbol{w}_i\cdot\boldsymbol{c}_j)^2\,\gamma_i\omega_j,$$

where the matrix of measurement probabilities $(\boldsymbol{w}_i\cdot\boldsymbol{c}_j)^2$ is a doubly stochastic matrix. Note also, that for the measurement interpretation, the matrix $\boldsymbol{C}$ does not have to be positive definite, but only symmetric. The algorithm and the proof of bounds in fact work fine for this case, as long as we have some assumption on the range of eigenvalues of $\boldsymbol{C}$, but the meaning of the algorithm when $\boldsymbol{C}$ is not a covariance matrix is less clear, since despite all these connections to quantum physics our algorithm does not seem to have the obvious quantum-mechanical interpretation. Our update clearly is not a unitary evolution of the mixture state and a measurement does not cause a collapse of the state as is the case in quantum physics. The question of whether this type of algorithm is still doing something quantum-mechanically meaningful remains intriguing. See also Warmuth and Kuzmin (2010) for additional discussion.

To derive our algorithm we use the trace expression for expected variance as our loss and replace the relative entropy with its matrix generalization. The following optimization problem produces the update:

$$\boldsymbol{W}^t = \operatorname*{arg\,min}_{\operatorname{tr}(\boldsymbol{W})=1} \Delta\bigl(\boldsymbol{W}, \boldsymbol{W}^{t-1}\bigr) + \eta \operatorname{tr}\bigl(\boldsymbol{W}\boldsymbol{C}^t\bigr). \tag{3.2}$$

Using a Lagrangian that enforces the trace constraint (Tsuda et al. 2005), it is easy to solve this constrained minimization problem:

$$\boldsymbol{W}^t = \frac{\exp(\log \boldsymbol{W}^{t-1} - \eta\boldsymbol{C}^t)}{\operatorname{tr}(\exp(\log \boldsymbol{W}^{t-1} - \eta\boldsymbol{C}^t))} = \frac{\exp(-\eta\boldsymbol{C}^{\leq t})}{\operatorname{tr}(\exp(-\eta\boldsymbol{C}^{\leq t}))}. \tag{3.3}$$

We call this the *Matrix Hedge* update. Note that the updated matrix $\boldsymbol{W}^t$ is again symmetric and positive definite. In particular, the quantum relative entropy functions serves as a barrier

for the implicit non-negativity constraints $\boldsymbol{W} \succeq \boldsymbol{0}$. For the second form of the update above, we unraveled the first, assumed that $\boldsymbol{W}^0 = \frac{1}{n}\boldsymbol{I}$ and used the shorthand $\boldsymbol{C}^{\leq t} = \sum_{q=1}^{t} \boldsymbol{C}^q$. The above update is a special case of the Matrix Exponentiated Gradient update of Tsuda et al. (2005) with the linear loss $\operatorname{tr}(\boldsymbol{W}\boldsymbol{C}^t)$. Recall that when we minimized the variance over the probability simplex, there was a difference between the implicit and explicit version of the Exponentiated Gradient update (Sect. 2.2). However since the loss $\operatorname{tr}(\boldsymbol{W}\boldsymbol{C}^t)$ is now linear in the parameter $\boldsymbol{W}$, the implicit and explicit updates coincide. Also the following batch motivation produces the same update:

$$\boldsymbol{W}^t = \operatorname*{arg\,min}_{\boldsymbol{W}\text{ s.t. }\operatorname{tr}(\boldsymbol{W})=1} \Delta\big(\boldsymbol{W}, \boldsymbol{W}^0\big) + \eta \sum_{q=1}^{t} \operatorname{tr}\big(\boldsymbol{W}\boldsymbol{C}^q\big). \tag{3.4}$$

When inequality constraints are added to the arg min, then (3.3) and (3.4) may have different solutions (see Kuzmin and Warmuth 2007 for a discussion).

3.3 Bregman divergence based proof methodology for expert setting

For the sake of clarity, we begin by the reproving the worst-case loss bound for the *Hedge update* since the later proof for the density matrix generalization (i.e. for Matrix Hedge) will follow the same outline. In doing so we also clarify the dependence of the algorithm on the range of the losses. The update of that algorithm is given by[1]:

$$w_i^t = \frac{w_i^{t-1} e^{-\eta l_i^t}}{\sum_i w_i^{t-1} e^{-\eta l_i^t}}. \tag{3.5}$$

This type of proof (Kivinen and Warmuth 1997) always starts by considering the progress made during the update towards any probability vector $\boldsymbol{u}$ that we compare against, where the progress is measured in terms of the motivating divergence for the algorithm (here the relative entropy):

$$\mathrm{d}\big(\boldsymbol{u}, \boldsymbol{w}^{t-1}\big) - \mathrm{d}\big(\boldsymbol{u}, \boldsymbol{w}^t\big) = \sum_i u_i \log \frac{w_i^t}{w_i^{t-1}} = -\eta\, \boldsymbol{u} \cdot \boldsymbol{l}^t - \log \sum_i w_i^{t-1} e^{-\eta l_i^t}.$$

We now lower bound the $-\log \sum_i w_i^{t-1} e^{-\eta l_i^t}$ term (sometimes called the potential) as in Littlestone and Warmuth (1994), Lemma 5.2. This is done by assuming that $l_i^t \in [0, r]$, for $r > 0$, and applying the inequality $e^{-\eta x} \leq 1 - (1 - e^{-\eta})\frac{x}{r}$, for $x \in [0, r]$ and $\eta \in \mathbb{R}$:

$$\mathrm{d}\big(\boldsymbol{u}, \boldsymbol{w}^{t-1}\big) - \mathrm{d}\big(\boldsymbol{u}, \boldsymbol{w}^t\big) \geq -\eta\, \boldsymbol{u} \cdot \boldsymbol{l}^t - \log\left(1 - \frac{\boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t}{r}\big(1 - e^{-\eta r}\big)\right).$$

We now apply $\log(1 - x) \leq -x$ and obtain:

$$\mathrm{d}\big(\boldsymbol{u}, \boldsymbol{w}^{t-1}\big) - \mathrm{d}\big(\boldsymbol{u}, \boldsymbol{w}^t\big) \geq -\eta\, \boldsymbol{u} \cdot \boldsymbol{l}^t + \frac{\boldsymbol{w}^{t-1} \cdot \boldsymbol{l}}{r}\big(1 - e^{-\eta r}\big).$$

[1] In the earlier Continuous Weighted Majority algorithm the loss l_i^t is the absolute loss $|y^t - x_i^t|$ between the label y^t and the prediction of the ith expert x_i^t.

Finally we rewrite the above to

$$\boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t \leq \frac{r(\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}^{t-1}) - \mathrm{d}(\boldsymbol{u}, \boldsymbol{w}^t)) + \eta r\, \boldsymbol{u} \cdot \boldsymbol{l}^t}{1 - e^{-\eta r}}.$$

Here $\boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t$ is the loss of the algorithm at trial t and $\boldsymbol{u} \cdot \boldsymbol{l}^t$ is the loss of the probability vector $\boldsymbol{u}$ which serves as a comparator.

So far we assumed that $l_i^t \in [0, r]$. However, it suffices to assume that $\max_i l_i^t - \min_i l_i^t \leq r$. In other words, the individual losses can be positive or negative, as long as their range is bounded by r. See Cesa-Bianchi et al. (2005) for further discussion regarding the issues that arise when losses have different signs. As we shall observe below, the requirement on the range of losses will become a requirement on the range of eigenvalues of the covariance matrices.

Define $\tilde{l}_i^t := l_i^t - \min_i l_i^t$. The update remains unchanged when the shifted losses $\tilde{l}_i^t$ are used in place of the original losses l_i^t and we immediately get the inequality

$$\boldsymbol{w}^{t-1} \cdot \tilde{\boldsymbol{l}}^t \leq \frac{r(\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}^{t-1}) - \mathrm{d}(\boldsymbol{u}, \boldsymbol{w}^t)) + \eta r\, \boldsymbol{u} \cdot \tilde{\boldsymbol{l}}^t}{1 - e^{-\eta r}}.$$

Summing over t results in a bound that holds for any probability vector $\boldsymbol{u}$:

$$\sum_{t=1}^{T} \boldsymbol{w}^{t-1} \cdot \tilde{\boldsymbol{l}}^t \leq \frac{r(\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}^0 - \mathrm{d}(\boldsymbol{u}, \boldsymbol{w}^T)) + \eta r\, \boldsymbol{u} \cdot \tilde{\boldsymbol{l}}^{\leq T}}{1 - e^{-\eta r}},$$

where $\boldsymbol{l}^{\leq T}$ is shorthand for $\sum_{q=1}^{T} \boldsymbol{l}^t$. We can now simplify the bound by dropping the term $\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}^T) \geq 0$ and tune the learning rate following Freund and Schapire (1997): if $\sum_t \boldsymbol{u} \cdot \tilde{\boldsymbol{l}}^t \leq \tilde{L}$ and $\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}^0) \leq D \leq \ln n$, then with $\eta = \frac{\log(1+\sqrt{2D/\tilde{L}})}{r}$ we get the inequality

$$\sum_t \boldsymbol{w}^{t-1} \cdot \tilde{\boldsymbol{l}}^t - \sum_t \boldsymbol{u} \cdot \tilde{\boldsymbol{l}}^t \leq \sqrt{2r\tilde{L}D} + r\,\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}^0).$$

By adding $\sum_{t=1}^{T} \min_i l_i^t$ to the first two sums we obtain the regret bound

$$\underbrace{\sum_t \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t}_{L_{\mathrm{alg}}} - \underbrace{\sum_t \boldsymbol{u} \cdot \boldsymbol{l}^t}_{L_{\boldsymbol{u}}} \leq \sqrt{2r\tilde{L}D} + r\,\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}^0).$$

Note that $\tilde{L}$ is defined with respect to the tilde versions of the losses and the update as well as the above bound is invariant under shifting the loss vectors $\boldsymbol{l}^t$ by arbitrary constants. If the loss vectors $\boldsymbol{l}^t$ are replaced by gain vectors, then the minus sign in the exponent of the update becomes a plus sign. In this case the inequality above is reversed and the last two terms are subtracted instead of added.

3.4 Relative loss bounds for matrix setting

In addition to the Golden-Thompson inequality we will need Lemmas 2.1 and 2.2 from Tsuda et al. (2005):

Lemma 3.1 *For any symmetric* $\boldsymbol{A}$, *such that* $\mathbf{0} \preceq \boldsymbol{A} \preceq \boldsymbol{I}$ *and any* $\rho_1, \rho_2 \in \mathbb{R}$ *the following holds*:

$$\exp\bigl(\boldsymbol{A}\rho_1 + (\boldsymbol{I} - \boldsymbol{A})\rho_2\bigr) \preceq \boldsymbol{A}e^{\rho_1} + (\boldsymbol{I} - \boldsymbol{A})e^{\rho_2}.$$

Lemma 3.2 *For any positive semidefinite* $\boldsymbol{A}$ *and any symmetric* $\boldsymbol{B}, \boldsymbol{C}$, $\boldsymbol{B} \preceq \boldsymbol{C}$ *implies* $\operatorname{tr}(\boldsymbol{A}\boldsymbol{B}) \leq \operatorname{tr}(\boldsymbol{A}\boldsymbol{C})$.

We are now ready to generalize the Hedge bound to matrices:

Theorem 3.1 *For any sequence of covariance matrices* $\boldsymbol{C}^1, \ldots, \boldsymbol{C}^T$ *such that* $\mathbf{0} \preceq \boldsymbol{C}^t \preceq r\boldsymbol{I}$ *and for any learning rate* η, *the following bound holds for arbitrary density matrix* $\boldsymbol{U}$:

$$\sum_{t=1}^{T} \operatorname{tr}\bigl(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\bigr) \leq \frac{r(\Delta(\boldsymbol{U}, \boldsymbol{W}^0) - \Delta(\boldsymbol{U}, \boldsymbol{W}^T)) + \eta r \operatorname{tr}(\boldsymbol{U}\boldsymbol{C}^{\leq T})}{1 - e^{-r\eta}}.$$

Proof We start by analyzing the progress made towards the comparison density matrix $\boldsymbol{U}$ in terms of the quantum relative entropy:

$$\Delta\bigl(\boldsymbol{U}, \boldsymbol{W}^{t-1}\bigr) - \Delta\bigl(\boldsymbol{U}, \boldsymbol{W}^t\bigr) = \operatorname{tr}\bigl(\boldsymbol{U}\bigl(-\log \boldsymbol{W}^{t-1} + \log \boldsymbol{W}^t\bigr)\bigr).$$

By plugging in r.h.s. form of update (3.3) we get

$$\Delta\bigl(\boldsymbol{U}, \boldsymbol{W}^{t-1}\bigr) - \Delta\bigl(\boldsymbol{U}, \boldsymbol{W}^t\bigr) = -\eta \operatorname{tr}\bigl(\boldsymbol{U}\boldsymbol{C}^t\bigr) + P^t - P^{t-1}, \tag{3.6}$$

where the P^t is called the *potential* and is defined as $-\log \operatorname{tr}(\exp(-\eta \boldsymbol{C}^{\leq t}))$. We first rewrite the drop of the potential using $\boldsymbol{W}^{t-1} = \frac{\exp(-\eta \boldsymbol{C}^{\leq t-1})}{\operatorname{tr}(\exp(-\eta \boldsymbol{C}^{\leq t-1}))}$ and then apply the Golden-Thompson inequality (3.1):

$$\begin{aligned} P^t - P^{t-1} &= \log \operatorname{tr}\bigl(\exp\bigl(-\eta \boldsymbol{C}^{\leq t} - \log \operatorname{tr}\bigl(-\eta \boldsymbol{C}^{\leq t-1}\bigr)\bigr)\bigr) = \log \operatorname{tr}\bigl(\exp\bigl(\log \boldsymbol{W}^{t-1} - \eta \boldsymbol{C}^t\bigr)\bigr) \\ &\leq \log \operatorname{tr}\bigl(\boldsymbol{W}^{t-1} \exp\bigl(-\eta \boldsymbol{C}^t\bigr)\bigr). \end{aligned} \tag{3.7}$$

Since $\mathbf{0} \preceq \frac{\boldsymbol{C}^t}{r} \preceq \boldsymbol{I}$, we can use Lemma 3.1 with $\rho_1 = -\eta r$, $\rho_2 = 0$:

$$\exp\bigl(-\eta \boldsymbol{C}^t\bigr) \preceq \boldsymbol{I} - \frac{\boldsymbol{C}^t}{r}\bigl(1 - e^{-\eta r}\bigr).$$

Now multiply both sides on the left with $\boldsymbol{W}^{t-1}$ and take a trace. The inequality is preserved according to Lemma 3.2:

$$\operatorname{tr}\bigl(\boldsymbol{W}^{t-1} \exp\bigl(-\eta \boldsymbol{C}^t\bigr)\bigr) \leq \left(1 - \frac{\operatorname{tr}(\boldsymbol{W}^{t-1}\boldsymbol{C}^t)}{r}\bigl(1 - e^{-r\eta}\bigr)\right). \tag{3.8}$$

Taking logs of both sides we have:

$$\log \operatorname{tr}\bigl(\boldsymbol{W}^{t-1} \exp\bigl(-\eta \boldsymbol{C}^t\bigr)\bigr) \leq \log\left(1 - \frac{\operatorname{tr}(\boldsymbol{W}^{t-1}\boldsymbol{C}^t)}{r}\bigl(1 - e^{-\eta r}\bigr)\right). \tag{3.9}$$

To bound the log expression on the right we use inequality $\log(1-x) \leq -x$:

$$\log\left(1 - \frac{\operatorname{tr}(\boldsymbol{W}^{t-1}\boldsymbol{C}^t)}{r}\left(1-e^{-r\eta}\right)\right) \leq -\frac{\operatorname{tr}(\boldsymbol{W}^{t-1}\boldsymbol{C}^t)}{r}\left(1-e^{-r\eta}\right). \tag{3.10}$$

By combining inequalities (3.7)–(3.10), we obtain the following bound on the drop of the potential

$$P^t - P^{t-1} \geq \frac{\operatorname{tr}(\boldsymbol{W}^{t-1}\boldsymbol{C}^t)}{r}\left(1-e^{-r\eta}\right). \tag{3.11}$$

Plugging this into (3.6) we obtain

$$r\left(\Delta\left(\boldsymbol{U},\boldsymbol{W}^{t-1}\right) - \Delta\left(\boldsymbol{U},\boldsymbol{W}^t\right)\right) + \eta r \operatorname{tr}\left(\boldsymbol{U}\boldsymbol{C}^t\right) \geq \operatorname{tr}\left(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\right)\left(1-e^{-r\eta}\right).$$

Summing over trials and rearranging gives the inequality of the theorem. □

Note the our density matrix update (3.3) is invariant with respect to the variable change $\tilde{\boldsymbol{C}}^t = \boldsymbol{C}^t - \lambda_{\min}(\boldsymbol{C}^t)\boldsymbol{I}$. Therefore by the above theorem, the following inequality holds whenever $\lambda_{\max}(\boldsymbol{C}^t) - \lambda_{\min}(\boldsymbol{C}^t) \leq r$:

$$\operatorname{tr}\left(\boldsymbol{W}^{t-1}\tilde{\boldsymbol{C}}^t\right) \leq \frac{r(\Delta(\boldsymbol{U},\boldsymbol{W}^{t-1}) - \Delta(\boldsymbol{U},\boldsymbol{W}^t)) + \eta r \operatorname{tr}(\boldsymbol{U}\tilde{\boldsymbol{C}}^t)}{1-e^{-r\eta}}.$$

We can now drop the $\Delta(\boldsymbol{U},\boldsymbol{W}^T) \geq 0$ term from the bound of the theorem and tune the learning rate as done at the end of Sect. 3.3. If $\sum_t \operatorname{tr}(\boldsymbol{U}\tilde{\boldsymbol{C}}^t) \leq \tilde{L}$ and $\Delta(\boldsymbol{U},\boldsymbol{W}^0) \leq D$, then with $\eta = \frac{\log(1+\sqrt{\frac{2D}{\tilde{L}}})}{r}$ and by adding $\sum_{t=1}^T \lambda_{\min}(\boldsymbol{C}^t)\boldsymbol{I}$ to both sides of the inequality we get the regret bound:

$$\underbrace{\sum_{t=1}^T \operatorname{tr}\left(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\right)}_{L_{\text{alg}}} - \underbrace{\operatorname{tr}\left(\boldsymbol{U}\boldsymbol{C}^{\leq T}\right)}_{L_{\boldsymbol{U}}} \leq \sqrt{2r\tilde{L}D} + r\Delta\left(\boldsymbol{U},\boldsymbol{W}^0\right).$$

From the assumption $\lambda_{\max}(\boldsymbol{C}^t) - \lambda_{\min}(\boldsymbol{C}^t) \leq r$, it follows that $\sum_t \operatorname{tr}(\boldsymbol{U}\tilde{\boldsymbol{C}}^t) \leq rT$. With this choice of $\tilde{L}$ and $D = \ln n$, the regret bound of the above theorem becomes

$$L_{\text{alg}} - L_{\boldsymbol{U}} \leq r\sqrt{2T\ln n} + r\ln n.$$

The bound of the theorem can be proven in a number of different ways. For the sake of completeness we sketch a number of these alternate proof methods based on potentials and Bregman projections in Appendix B. We also show in Appendix C that the same bound holds if a range of matrix factors are used in the update.

4 Conclusions

We presented two algorithms for online variance minimization problems. For the first problem, the variance was measured along a probability vector. This problem is connected to

minimizing the risk of an investment portfolio. The losses in each trial have the form $\boldsymbol{w}^\top \boldsymbol{C}\boldsymbol{w}$, where $\boldsymbol{C}$ is a covariance matrix and $\boldsymbol{w}$ is a probability vector. This loss is quadratic in the parameter $\boldsymbol{w}$. We also analyzed an algorithm that aims to optimize a simple additive tradeoff between the first order return and the second-order variance/risk information. In this case the loss have the form $\gamma\, \boldsymbol{w} \cdot \boldsymbol{l} + \boldsymbol{w}^\top \boldsymbol{C}\boldsymbol{w}$, where $\boldsymbol{l}$ is a loss vector and γ a tradeoff parameter. Our work shows that such tradeoffs are also amenable to online analysis. In future work it would be interesting to combine this work with the online algorithms considered in Cover (1991), Helmbold et al. (1998), Agarwal et al. (2006) that maximize the log return of the portfolio. Summing the log returns is natural in this context because the returns are naturally combined by their product. Ideally we would like to analyze tradeoffs between the risk and the log return. Future work should also consider other more sophisticated metrics based on ratios between the return and the variance as done in the Sharpe ratio.

Note that it is easy to extend the portfolio vector to maintain short positions: Simply keep two weights w_i^+ and w_i^- per component as is done in the EG$^\pm$ algorithm of Kivinen and Warmuth (1997).

In our second problem the variance was measured along an arbitrary direction, i.e. now the loss is $\boldsymbol{w}^\top \boldsymbol{C}\boldsymbol{w} = \mathrm{tr}(\boldsymbol{C}\boldsymbol{w}\boldsymbol{w}^\top)$, where $\boldsymbol{w}$ is a unit vector. In this case we chose convex combinations of dyads as a parameter class which are density matrices. The resulting algorithm is a natural generalization of the Hedge algorithm to the case when the parameters are density matrices instead of probability vectors. We proved regret bounds of the total loss of this algorithm against the total loss of the best density matrix chosen in hindsight. Note that for a convex combination of dyads $\boldsymbol{W} = \sum_i \omega_i \boldsymbol{w}_i \boldsymbol{w}_i^\top$, the convex combination of losses is now linear in the parameter density matrix $\boldsymbol{W}$:

$$\sum_i \omega_i\ \boldsymbol{w}_i^\top \boldsymbol{C}\boldsymbol{w}_i = \mathrm{tr}\left(\left(\sum_i \omega_i \boldsymbol{w}_i \boldsymbol{w}_i^\top\right)\boldsymbol{C}\right) = \mathrm{tr}(\boldsymbol{W}\boldsymbol{C}).$$

In the original vector case for learning with expert advice, there is an alternate to Hedge algorithm called the *Follow the Perturbed Leader* (FPL) algorithm. This algorithm adds a random perturbation to the current total loss of each expert and predict with the expert of minimum perturbed loss (Kalai and Vempala 2005). When the perturbation are from a suitably chosen distribution, then FPL is known to simulate Hedge (Kuzmin and Warmuth 2005; Kalai 2005). An open question (posed in Hazan et al. 2010) is whether there is a matrix version of FPL that simulates the Matrix Hedge algorithm by relying on an $O(n^2)$ minimum eigendirection computation rather than requiring a decomposition of the loss matrix which costs $O(n^3)$ time.

Also note in the vector case, the optimal algorithm was developed (Abernethy et al. 2008) for the case when the loss vectors are binary and the loss of the best expert is bounded. This is done by formulating a game between a learner and an adversary. In each trial, the learner chooses a probability vector over the experts and the adversary chooses a loss vector. The learner minimizes and the adversary maximizes the loss. A key open problem is to develop the optimal algorithm for the matrix case when the variance of the best dyad is bounded. Curiously enough it was conjectured in Warmuth (2007a) that the vector case is the hardest special case of the matrix case. To resolve this conjecture in the positive, one would have to show that the optimal strategy of the adversary is to choose covariance matrices that all have the same eigensystem because in that case only the variances with respect to each of the n eigendirections are relevant and the matrix case reduces to the original vector case.

Much work has been done on using exponential update factors in the expert setting. In particular, algorithms have been developed for shifting experts by combining the exponential

updates with an additive "sharing update" (Herbster and Warmuth 1998). In preliminary work we showed that these techniques easily carry over to the density matrix updates. This includes the more recent work on the "sharing to the past average" update, which introduces a long-term memory (Bousquet and Warmuth 2002). (See Warmuth and Kuzmin 2008 for some experimental evidence that the sharing to the past average update carries over to the matrix setting.)

The first use of the exponential update factors was in the algorithm Winnow for learning disjunctions (Littlestone 1988, 1989). Curiously enough this algorithm also nicely carries over to the matrix case (Warmuth 2007b) using methods similar to those applied in the second half of this paper. We addressed the case when the covariance matrices are symmetric and comparators are symmetric dyads $\boldsymbol{u}\boldsymbol{u}^\top$. The uncertainty of the algorithm is described as a mixture of such dyads, which is a density matrix. Using the techniques developed in conjunction with the matrix version of the Winnow algorithm (Warmuth 2007b) it should be possible to generalize Matrix Hedge to the case when the instance matrices $\boldsymbol{C}$ are non-square matrices in $\mathbb{R}^{m\times n}$, the comparators are asymmetric dyads $\boldsymbol{u}\boldsymbol{v}^\top$, for $\boldsymbol{u}\in\mathbb{R}^m$ and $\boldsymbol{w}\in\mathbb{R}^n$, and the loss remains the trace $\operatorname{tr}(\boldsymbol{C}\boldsymbol{v}\boldsymbol{u}^\top)=\boldsymbol{u}^\top\boldsymbol{C}\boldsymbol{v}$. The resulting algorithm would again maintain its uncertainty over asymmetric dyads as a mixture of such dyads.

Appendix A: Proof of bound for variance-loss tradeoff

In this appendix we give all the proofs for Sect. 2.4. The first lemma gives the key inequality that upper bounds the loss of the algorithm minus the loss a comparator $\boldsymbol{u}$ i.t.o. the divergence towards $\boldsymbol{u}$. A simpler version of this inequality is proven in Lemma 2.1.

Proof of Lemma 2.3 We begin by analyzing progress towards the comparison vector $\boldsymbol{u}$:

$$\begin{aligned}
&d\big(\boldsymbol{u},\boldsymbol{w}^{t-1}\big)-d\big(\boldsymbol{u},\boldsymbol{w}^{t}\big)\\
&\quad=\sum_i u_i\log w_i^t-\sum_i u_i\log w_i^{t-1}\\
&\quad\overset{(2.5)}{=}\sum_i u_i\log\frac{w_i^{t-1}e^{-\eta(\gamma l_i^t+(\boldsymbol{C}^t\boldsymbol{w}^{t-1})_i)}}{\sum_i w_i^{t-1}e^{-\eta(\gamma l_i^t+(\boldsymbol{C}^t\boldsymbol{w}^{t-1})_i)}}-\sum_i u_i\log w_i^{t-1}\\
&\quad=-\eta\sum_i u_i\big(\gamma l_i^t+\big(\boldsymbol{C}^t\boldsymbol{w}^{t-1}\big)_i\big)-\log\Big(\sum_i w_i^{t-1}e^{-\eta(\gamma l_i^t+(\boldsymbol{C}^t\boldsymbol{w}^{t-1})_i)}\Big).
\end{aligned}$$

Thus, our bound is equivalent to showing $F\le 0$ with F given as:

$$\begin{aligned}
F=\;&a\big(\gamma\boldsymbol{w}^{t-1}\cdot\boldsymbol{l}^t+\big(\boldsymbol{w}^{t-1}\big)^\top\boldsymbol{C}^t\boldsymbol{w}^{t-1}\big)-b\big(\gamma\boldsymbol{u}\cdot\boldsymbol{l}^t+\boldsymbol{u}^\top\boldsymbol{C}^t\boldsymbol{u}\big)\\
&+\eta\sum_i u_i\big(\gamma l_i^t+\big(\boldsymbol{C}^t\boldsymbol{w}^{t-1}\big)_i\big)+\log\Big(\sum_i w_i^{t-1}e^{-\eta(\gamma l_i^t+(\boldsymbol{C}^t\boldsymbol{w}^{t-1})_i)}\Big).
\end{aligned}$$

We proceed by bounding the log term. As before, let p be a lower bound on $(\boldsymbol{C}^t\boldsymbol{w}^{t-1})_i$. Now, from the assumption on the range of elements in $\boldsymbol{C}^t$ we get that $p\le(\boldsymbol{C}^t\boldsymbol{w}^{t-1})_i\le p+r$. Adding the γ and the assumptions on the range of losses we get: $p\le\gamma l_i^t+(\boldsymbol{C}^t\boldsymbol{w}^{t-1})_i\le p+r+\gamma q$, or equivalently:

$$0\le\frac{\gamma l_i^t+(\boldsymbol{C}^t\boldsymbol{w}^{t-1})_i-p}{r+\gamma q}\le 1.$$

Now we apply the inequality $a^x \leq 1 - x(1-a)$ for $a \geq 0$ and $0 \leq x \leq 1$. Let $a = e^{-\eta(r+\gamma q)}$:

$$e^{-\eta(\gamma l_i^t + (\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i)} = e^{-\eta p}\left(e^{-\eta(r+\gamma q)}\right)^{\frac{\gamma l_i^t + (\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i - p}{r+\gamma q}}$$
$$\leq e^{-\eta p}\left(1 - \frac{\gamma l_i^t + (\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i - p}{r+\gamma q}\left(1 - e^{-\eta(r+\gamma q)}\right)\right).$$

Using this inequality we obtain:

$$\log\left(\sum_i w_i^{t-1} e^{-\eta(\gamma l_i^t + (\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i)}\right)$$
$$\leq -\eta p + \log\left(1 - \frac{\gamma \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t + (\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - p}{r+\gamma q}\left(1 - e^{-\eta(r+\gamma q)}\right)\right).$$

This gives us $F \leq G$ with G given as:

$$G = a\left(\gamma \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t + \left(\boldsymbol{w}^{t-1}\right)^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1}\right)$$
$$- b\left(\gamma \boldsymbol{u} \cdot \boldsymbol{l}^t + \boldsymbol{u}^\top \boldsymbol{C}^t \boldsymbol{u}\right) + \eta \sum_i u_i \left(\gamma l_i^t + \left(\boldsymbol{C}^t \boldsymbol{w}^{t-1}\right)_i\right)$$
$$- \eta p + \log\left(1 - \frac{\gamma \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t + (\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - p}{r+\gamma q}\left(1 - e^{-\eta(r+\gamma q)}\right)\right).$$

We will split G into two parts $G_1 + G_2$ as follows:

$$G_1 = a\left(\boldsymbol{w}^{t-1}\right)^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - b\boldsymbol{u}^\top \boldsymbol{C}^t \boldsymbol{u} + \eta \boldsymbol{u}^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - \eta p$$
$$+ \log\left(1 - \frac{\gamma \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t + (\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - p}{r+\gamma q}\left(1 - e^{-\eta(r+\gamma q)}\right)\right),$$
$$G_2 = a\gamma \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t - b\gamma \boldsymbol{u} \cdot \boldsymbol{l}^t + \eta\gamma \boldsymbol{u} \cdot \boldsymbol{l}^t.$$

For G_1 we use variable substitution $\boldsymbol{z} = \sqrt{\boldsymbol{C}^t}\boldsymbol{u}$:

$$G_1(\boldsymbol{z}) = -b\boldsymbol{z}^\top \boldsymbol{z} + \eta \boldsymbol{z}^\top \sqrt{\boldsymbol{C}^t}\boldsymbol{w}^{t-1} + \text{constant}.$$

The function $G_1(\boldsymbol{z})$ is concave quadratic and is maximized at:

$$\frac{\partial G_1}{\partial \boldsymbol{z}} = -2b\boldsymbol{z} + \eta\sqrt{\boldsymbol{C}^t}\boldsymbol{w}^{t-1} = 0, \qquad z = \frac{\eta}{2b}\sqrt{\boldsymbol{C}^t}\boldsymbol{w}^{t-1}.$$

We substitute this value of $\boldsymbol{z}$ into G_1 and get $G_1 \leq H$, where H is given by:

$$H = a\left(\boldsymbol{w}^{t-1}\right)^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} + \frac{\eta^2}{4b}\left(\boldsymbol{w}^{t-1}\right)^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - \eta p +$$
$$+ \log\left(1 - \frac{\gamma \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t + (\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - p}{r+\gamma q}\left(1 - e^{-\eta(r+\gamma q)}\right)\right).$$

Since $p \leq \gamma l_i^t + (\boldsymbol{C}^t \boldsymbol{w}^{t-1})_i \leq p + r + \gamma q$, then so is $\gamma \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t + (\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1}$, since a convex combination stays within the bounds. We now apply the inequality $\log(1 - x(1 -$

$e^y)) \leq xy + \frac{y^2}{8}$ for $0 \leq x \leq 1$ and $y \in \mathbb{R}$ and obtain:

$$\log\left(1 - \frac{\gamma \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t + (\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} - p}{r + \gamma q}\left(1 - e^{-\eta(r+\gamma q)}\right)\right)$$
$$\leq -\eta\gamma \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t - \eta\left(\boldsymbol{w}^{t-1}\right)^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} + \eta p + \frac{\eta^2 (r+\gamma q)^2}{8}.$$

Substituting and reshuffling the terms a little bit we obtain $G = G_1 + G_2 \leq S_1 + S_2$, where

$$S_1 = \frac{(\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1}}{4b}\left(\eta^2 + 4ba - 4b\eta\right) + \frac{\eta^2 (r+\gamma q)^2}{8},$$
$$S_2 = a\gamma \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t - \eta\gamma \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t - b\gamma \boldsymbol{u} \cdot \boldsymbol{l}^t + \eta\gamma \boldsymbol{u} \cdot \boldsymbol{l}^t.$$

By our assumptions $(\boldsymbol{w}^{t-1})^\top \boldsymbol{C}^t \boldsymbol{w}^{t-1} \leq \frac{r}{2}$ and we use this to upper bound S_1. For S_2 we drop the negative terms $-\eta\gamma \boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t - b\gamma \boldsymbol{u} \cdot \boldsymbol{l}^t$ and use the loss upper bounds $\boldsymbol{w}^{t-1} \cdot \boldsymbol{l}^t \leq q$ and $\boldsymbol{u} \cdot \boldsymbol{l}^t \leq q$. Overall we get $G \leq Q$ with Q given as:

$$Q = \eta^2\left(\frac{r}{8b} + \frac{(r+\gamma q)^2}{8}\right) - \eta\left(\frac{r}{2} - \gamma q\right) + \frac{ar}{2} + a\gamma q.$$

We want to make Q as small as possible because we need the condition $Q \leq 0$, which assures $F \leq 0$. Minimizing over η gives us:

$$2\eta\left(\frac{r}{8b} + \frac{(r+\gamma q)^2}{8}\right) - \frac{r}{2} + \gamma q = 0, \quad \eta = \frac{2rb - 4\gamma q b}{r + b(r+\gamma q)^2}.$$

We substitute this value of η into Q:

$$Q = \frac{(2b(r-2\gamma q))^2}{(r+b(r+\gamma q)^2)^2} \cdot \frac{r + b(r+\gamma q)^2}{8b} - \frac{2b(r-2\gamma q)}{r+b(r+\gamma q)^2} \cdot \frac{r-2\gamma q}{2} + \frac{ar}{2} + a\gamma q$$
$$= \frac{b(r-2\gamma q)^2}{2(r+b(r+\gamma q)^2)} - \frac{2b(r-2\gamma q)^2}{2(r+b(r+\gamma q)^2)} + a\left(\frac{r}{2} + \gamma q\right).$$

Now $Q \leq 0$, whenever

$$a \leq \frac{b(r-2\gamma q)^2}{(r+b(r+\gamma q)^2)(r+2\gamma q)}.$$

□

As in Lemma 2.2 we now tune the learning to get a refined version of the key inequality.

Proof of Lemma 2.4 We use the inequality from Lemma 2.3 and let a be equal its upper bound:

$$a = \frac{b(r-2\gamma q)^2}{(r+b(r+\gamma q)^2)(r+2\gamma q)} = \frac{b}{1 + \frac{b}{r}(r+\gamma q)^2} \cdot \frac{(r-2\gamma q)^2}{r(r+2\gamma q)} = \frac{bR}{1+bQ}.$$

Summing the per-trial inequalities and dropping the negative term $d(\boldsymbol{u}, \boldsymbol{w}^t)$ we get the bound:

$$\frac{bR}{1+bQ} L_{\text{alg}}^{\gamma} \leq bL_{\boldsymbol{u}}^{\gamma} + d\big(\boldsymbol{u}, \boldsymbol{w}^0\big).$$

Rearranging the terms we get:

$$L_{\text{alg}}^{\gamma} \leq \left(\frac{1}{R} + \frac{bQ}{R}\right) L_{\boldsymbol{u}}^{\gamma} + \frac{1}{bR} d\big(\boldsymbol{u}, \boldsymbol{w}^0\big) + \frac{Q}{R} d\big(\boldsymbol{u}, \boldsymbol{w}^0\big). \qquad \square$$

We are now ready to prove the final theorem of Sect. 2.4.

Proof of Theorem 2.2 We start with the bound obtained in Lemma 2.4 and use our assumptions $L_{\boldsymbol{u}}^{\gamma} \leq L$ and $d(\boldsymbol{u}, \boldsymbol{w}^0) \leq \log n$:

$$L_{\text{alg}}^{\gamma} \leq \frac{1}{R} L_{\boldsymbol{u}}^{\gamma} + \frac{bQ}{R} L + \frac{1}{bR} \log n + \frac{Q}{R} \log n.$$

We can now minimize the right hand side as a function of b. Setting the derivative to zero we get:

$$\frac{\log n}{R} \frac{1}{b^2} = \frac{QL}{R}, \quad b = \sqrt{\frac{\log n}{QL}}.$$

Substituting this value of b back into the bound, we get:

$$L_{\text{alg}}^{\gamma} \leq \frac{1}{R} L_{\boldsymbol{u}}^{\gamma} + \frac{2}{R} \sqrt{QL \log n} + \frac{Q}{R} \log n.$$

Finally we split the $\frac{1}{R} L_{\boldsymbol{u}}^{\gamma}$ term:

$$\begin{aligned} \frac{1}{R} - 1 &= \frac{r^2 + 2\gamma qr - r^2 + 4\gamma qr - 4\gamma^2 q^2}{r^2 - 4r\gamma q + 4\gamma^2 q^2} = \frac{6\gamma qr - 4\gamma^2 q^2}{r^2 - 4r\gamma q + 4\gamma^2 q^2} \\ &= \frac{2\gamma q(3r - 2\gamma q)}{(r - 2\gamma q)^2} = P. \end{aligned}$$

$\square$

Appendix B: Sketch of other proof techniques for variance minimization over the unit sphere

In this section we briefly discuss some slight variations of the proof of above relative loss bound. For the sake of brevity we assume $r = 1$ for the rest of the paper. The first variation stresses the potential

$$P^t := -\log \operatorname{tr}\big(\exp\big(-\eta \boldsymbol{C}^{\leq t}\big)\big),$$

which is the value of the motivating minimization problem (3.4) when plugging in the solution $\boldsymbol{W}^t$ (easily seen by using the right form of update (3.3)).[2] As part of the previous

[2]The value of the "on-line" minimization problem (3.2) at the optimum $\boldsymbol{W}^t$ is the drop of potential $P^t - P^{t-1}$.

proof we already bounded the drop of the potential (this method is the core of many papers in the expert advice setting; Littlestone and Warmuth 1994; Kivinen and Warmuth 1999; Vovk 1990):

$$P^t - P^{t-1} \overset{(3.7\text{–}3.11)}{\geq} \operatorname{tr}\bigl(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\bigr)\bigl(1-e^{-\eta}\bigr).$$

Summing over trials we get (since $P^1 = 0$):

$$P^T \geq \sum_{t=1}^{T} \operatorname{tr}\bigl(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\bigr)\bigl(1-e^{-\eta}\bigr).$$

Finally, since P^T is the value of (3.4) at $\boldsymbol{W}^T$:

$$P^T = \Delta\bigl(\boldsymbol{W}^T, \boldsymbol{W}^0\bigr) + \eta \operatorname{tr}\bigl(\boldsymbol{W}^T\boldsymbol{C}^{\leq T}\bigr) = \operatorname*{arg\,min}_{\boldsymbol{U}\text{ s.t. }\operatorname{tr}(\boldsymbol{U})=1} \Delta\bigl(\boldsymbol{U}, \boldsymbol{W}^0\bigr) - \Delta\bigl(\boldsymbol{U}, \boldsymbol{W}^T\bigr) + \eta \operatorname{tr}\bigl(\boldsymbol{U}\boldsymbol{C}^{\leq T}\bigr).$$

Piecing the above inequality and equality for P^T together, again gives the bound of the theorem.

The second variation of the proof expresses the normalization in the update as a Bregman projection and using the Generalized Pythagorean Theorem for Bregman divergences. See Helmbold and Warmuth (2009) for an analogous discussion in the context of learning permutations. For this we start with the following *unnormalized* update:

$$\tilde{\boldsymbol{W}}^t = \exp\bigl(\log \boldsymbol{W}^{t-1} - \eta \boldsymbol{C}^t\bigr),$$

which is the solution to the minimization problem

$$\boldsymbol{W}^t = \operatorname*{arg\,min}_{\boldsymbol{W}} \tilde{\Delta}\bigl(\boldsymbol{W}, \boldsymbol{W}^{t-1}\bigr) + \eta \operatorname{tr}\bigl(\boldsymbol{W}\boldsymbol{C}^t\bigr), \tag{B.1}$$

where $\tilde{\Delta}(\boldsymbol{W}, \boldsymbol{W}^{t-1})$ is now the *unnormalized* quantum relative entropy defined as $\operatorname{tr}(\boldsymbol{U}(\log \boldsymbol{U} - \log \boldsymbol{W})) + \operatorname{tr}(\boldsymbol{W}) - \operatorname{tr}(\boldsymbol{U})$. The unnormalized version coincides with the standard relative entropy when both matrices have trace one. It is easy to see that the basic inequality in the proof of Theorem 3.1 holds for this unnormalized update as well:

$$\begin{aligned}
\tilde{\Delta}\bigl(\boldsymbol{U}, \boldsymbol{W}^{t-1}\bigr) - \tilde{\Delta}\bigl(\boldsymbol{U}, \tilde{\boldsymbol{W}}^t\bigr) &= \operatorname{tr}\bigl(\boldsymbol{U}\log \tilde{\boldsymbol{W}}^t\bigr) - \operatorname{tr}\bigl(\boldsymbol{U}\log \boldsymbol{W}^{t-1}\bigr) + \operatorname{tr}\bigl(\boldsymbol{W}^{t-1}\bigr) - \operatorname{tr}\bigl(\tilde{\boldsymbol{W}}^t\bigr) \\
&= -\eta \operatorname{tr}\bigl(\boldsymbol{U}\boldsymbol{C}^t\bigr) + \operatorname{tr}\bigl(\boldsymbol{W}^{t-1}\bigr) - \operatorname{tr}\bigl(\exp\bigl(\log \boldsymbol{W}^{t-1} - \eta \boldsymbol{C}^t\bigr)\bigr) \\
&\overset{(3.1)}{\geq} -\eta \operatorname{tr}\bigl(\boldsymbol{U}\boldsymbol{C}^t\bigr) + \operatorname{tr}\bigl(\boldsymbol{W}^{t-1}\bigl(\boldsymbol{I} - \exp\bigl(-\eta \boldsymbol{C}^t\bigr)\bigr)\bigr) \\
&\geq -\eta \operatorname{tr}\bigl(\boldsymbol{U}\boldsymbol{C}^t\bigr) + \bigl(1-e^{-\eta}\bigr)\operatorname{tr}\bigl(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\bigr).
\end{aligned}$$

The last inequality follows upper bounding $\exp(-\eta \boldsymbol{C}^t)$ and applying Lemma 3.2 as done in (3.8).

Now we introduce the concept of Bregman projections. Bregman projection onto a convex set is defined as simply the "closest" point in that set in terms of the Bregman divergence of interest to us. Take the convex set to be the set of trace one matrices. It is easy to see that the normalized update (3.3) is the Bregman projection with respect to quantum relative entropy of $\tilde{\boldsymbol{W}}^t$ onto the set of trace one matrices:

$$\boldsymbol{W}^t = \inf_{\boldsymbol{W}\text{ s.t. }\operatorname{tr}(\boldsymbol{W})=1} \tilde{\Delta}\bigl(\boldsymbol{W}, \tilde{\boldsymbol{W}}^t\bigr) = \frac{\tilde{\boldsymbol{W}}^t}{\operatorname{tr}(\tilde{\boldsymbol{W}}^t)}.$$

Applying the Generalized Pythagorean Theorem for Bregman projections (see e.g. Herbster and Warmuth 2001) then gets us the following inequality:

$$\Delta\big(\boldsymbol{U}, \tilde{\boldsymbol{W}}^t\big) \geq \Delta\big(\boldsymbol{U}, \boldsymbol{W}^t\big) + \Delta\big(\boldsymbol{W}^t, \tilde{\boldsymbol{W}}^t\big).$$

Since divergences are non-negative, we can drop the term $\Delta(\boldsymbol{W}^t, \tilde{\boldsymbol{W}}^t)$ to get the inequality:

$$\Delta\big(\boldsymbol{U}, \tilde{\boldsymbol{W}}^t\big) - \Delta\big(\boldsymbol{U}, \boldsymbol{W}^t\big) \geq 0. \tag{B.2}$$

Now we add (B.2) to our inequality for unnormalized update to obtain the desired inequality for the normalized update:

$$\Delta\big(\boldsymbol{U}, \boldsymbol{W}^{t-1}\big) - \Delta\big(\boldsymbol{U}, \boldsymbol{W}^t\big) \geq -\eta \operatorname{tr}\big(\boldsymbol{U}\boldsymbol{C}^t\big) + \big(1 - e^{-\eta}\big) \operatorname{tr}\big(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\big).$$

The proof now completes as in Theorem 3.1.

In the above proof of the bound via the unnormalized update, the Bregman projection methods can also be replaced with a potential based argument. To do this we define the unnormalized potential is defined as $\tilde{P}^t := -\operatorname{tr}(\tilde{\boldsymbol{W}}^t)$. For example by plugging the optimum unnormalized solution $\tilde{\boldsymbol{W}}^t$ into (B.1) we obtain the drop of potentials $-\operatorname{tr}(\tilde{\boldsymbol{W}}^t) - (-\operatorname{tr}(\boldsymbol{W}^{t-1}))$. We leave the details to the reader.

Appendix C: Proof for range of matrix factors for variance minimization over the unit sphere

For the regular Weighted Majority update (Littlestone and Warmuth 1994) it is well-known that if the losses satisfy the range restriction $\ell_i^t \in [0, r]$, then the multiplicative factors $e^{-\eta l_i^t}$ can be replaced by a whole range of factors f satisfying $e^{-\eta l_i^t} \leq f \leq 1 - \frac{l_i^t}{r}(1 - e^{-\eta r})$. This means that if any such factors are used in place of the Hedge update (3.5) then the basic loss bound of Theorem 3.1 still holds. Here we generalize the notion of such factors to the matrix case. As we shall see, the key is to use the right "matrix product". For the sake of brevity, we assume $r = 1$ for the rest of the paper.

In Sect. 3.4 and Appendix B we proved a bound for the following version of matrix update:

$$\boldsymbol{W}^t = \frac{\exp(\log \boldsymbol{W}^{t-1} - \eta \boldsymbol{C}^t)}{\operatorname{tr}(\exp(\log \boldsymbol{W}^{t-1} - \eta \boldsymbol{C}^t))}.$$

For our generalization of the update that uses a range of matrix factors we need the following binary operation defined for symmetric strictly positive definite matrices[3]:

$$\boldsymbol{A} \odot \boldsymbol{B} = \exp(\log \boldsymbol{A} + \log \boldsymbol{B}).$$

This operation plays a crucial role in the generalized probability calculus for density matrices that we develop in Warmuth and Kuzmin (2006a). See that paper for a discussion of many properties of $\odot$ operation.

[3]This definition can be extended to all symmetric positive definite matrices via limit formula: $\boldsymbol{A} \odot \boldsymbol{B} = \lim_{n\to\infty}(\boldsymbol{A}^{1/n}\boldsymbol{B}^{1/n})^n$. See Warmuth and Kuzmin (2006a) for a discussion.

Now we can rewrite the update (3.3) of our algorithm as follows:

$$\boldsymbol{W}^t = \frac{\boldsymbol{W}^{t-1} \odot \exp(-\eta \boldsymbol{C}^t)}{\operatorname{tr}(\boldsymbol{W}^{t-1} \odot \exp(-\eta \boldsymbol{C}^t))}.$$

The generalization have the following form

$$\boldsymbol{W}^t = \frac{\boldsymbol{W}^{t-1} \odot \boldsymbol{F}^t}{\operatorname{tr}(\boldsymbol{W}^{t-1} \odot \boldsymbol{F}^t)}, \tag{C.1}$$

where the matrix factor $\boldsymbol{F}^t$ is any symmetric matrix satisfying the following generalized inequalities:

$$\exp\bigl(-\eta \boldsymbol{C}^t\bigr) \preceq \boldsymbol{F}^t \preceq \boldsymbol{I} - \boldsymbol{C}^t\bigl(1 - e^{-\eta}\bigr). \tag{C.2}$$

Note that both the lhs and rhs of the above chain of inequalities are both symmetric positive definite and lhs $\preceq$ rhs by Lemma 3.1.

We now prove that the same bound holds for update in (C.1).

Theorem C.1 *For any sequence of symmetric loss matrices $\mathbf{0} \preceq \boldsymbol{C}^t \preceq \boldsymbol{I}$, for any learning rate $\eta > 0$ and for any comparator density matrix $\boldsymbol{U}$ the following bound holds on the loss of algorithm with update* (C.1) *when the $\boldsymbol{F}^t$ satisfy the condition* (C.2):

$$\sum_{t=1}^{T} \operatorname{tr}\bigl(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\bigr) \leq \frac{\Delta(\boldsymbol{U}, \boldsymbol{W}^0) - \Delta(\boldsymbol{U}, \boldsymbol{W}^T) + \eta \operatorname{tr}(\boldsymbol{U}\boldsymbol{C}^{\leq T})}{1 - e^{-\eta}}.$$

Proof The proof starts by analyzing progress towards $\boldsymbol{U}$ in terms of quantum relative entropy. To simplify the analysis we assume that $\boldsymbol{W}^{t-1}$ and $\boldsymbol{F}^t$ are full rank matrices. In that case $\log(\boldsymbol{W}^{t-1} \odot \boldsymbol{F}^t) = \log \boldsymbol{W}^{t-1} + \log \boldsymbol{F}^t$. This is a reasonable assumption if $\boldsymbol{W}^0$ has full rank.

$$\begin{aligned}\Delta\bigl(\boldsymbol{U}, \boldsymbol{W}^{t-1}\bigr) - \Delta\bigl(\boldsymbol{U}, \boldsymbol{W}^t\bigr) &= -\operatorname{tr}\left(\boldsymbol{U}\left(\log \boldsymbol{W}^{t-1} - \log \frac{\boldsymbol{W}^{t-1} \odot \boldsymbol{F}^t}{\operatorname{tr}(\boldsymbol{W}^{t-1} \odot \boldsymbol{F}^t)}\right)\right) \\ &= \operatorname{tr}\bigl(\boldsymbol{U} \log \boldsymbol{F}^t\bigr) - \log \operatorname{tr}\bigl(\boldsymbol{W}^{t-1} \odot \boldsymbol{F}^t\bigr). \end{aligned} \tag{C.3}$$

Now we proceed to bound the last two terms. First, note that matrix log is operator monotone, i.e. if $\boldsymbol{A} \preceq \boldsymbol{B}$ then $\log \boldsymbol{A} \preceq \log \boldsymbol{B}$ (Fact 8.8.29 of Bernstein 2005). Thus the left side of condition (C.2) gives us:

$$-\eta \boldsymbol{C}^t \preceq \log \boldsymbol{F}^t.$$

Which, in turn, by application of Lemma 3.2 gives:

$$-\eta \operatorname{tr}\bigl(\boldsymbol{U}\boldsymbol{C}^t\bigr) \leq \operatorname{tr}\bigl(\boldsymbol{U} \log \boldsymbol{F}^t\bigr).$$

Now we will bound the term $-\log \operatorname{tr}(\boldsymbol{W}^{t-1} \odot \boldsymbol{F}^t)$. First, applying Golden-Thompson inequality to the $\odot$ operation gives:

$$\operatorname{tr}\bigl(\boldsymbol{W}^{t-1} \odot \boldsymbol{F}^t\bigr) \leq \operatorname{tr}\bigl(\boldsymbol{W}^{t-1}\boldsymbol{F}^t\bigr).$$

Next, we use the condition that $\boldsymbol{F}^t \preceq \boldsymbol{I} - \boldsymbol{C}^t(1 - e^{-\eta})$:

$$\operatorname{tr}\big(\boldsymbol{W}^{t-1}\boldsymbol{F}^t\big) \leq 1 - \operatorname{tr}\big(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\big)\big(1 - e^{-\eta}\big).$$

All this together gives:

$$-\log \operatorname{tr}\big(\boldsymbol{W}^{t-1} \odot \boldsymbol{F}^t\big) \geq -\log\big(1 - \operatorname{tr}\big(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\big)\big(1 - e^{-\eta}\big)\big).$$

Now, using $\log(1 - x) \leq -x$ we get:

$$-\log \operatorname{tr}\big(\boldsymbol{W}^{t-1} \odot \boldsymbol{F}^t\big) \geq \operatorname{tr}\big(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\big)\big(1 - e^{-\eta}\big).$$

Putting all the pieces into the progress equality (C.3), we arrive at:

$$\Delta\big(\boldsymbol{U}, \boldsymbol{W}^{t-1}\big) - \Delta\big(\boldsymbol{U}, \boldsymbol{W}^t\big) \geq -\eta \operatorname{tr}\big(\boldsymbol{U}\boldsymbol{C}^t\big) + \operatorname{tr}\big(\boldsymbol{W}^{t-1}\boldsymbol{C}^t\big)\big(1 - e^{-\eta}\big).$$

Summing over trials and rearranging terms gives the desired inequality. □

References

Abernethy, J., Warmuth, M. K., & Yellin, J. (2008). When random play is optimal against an adversary. In *Proceedings of the 21st annual conference on learning theory (COLT '08)* (pp. 437–445).

Agarwal, A., Hazan, E., Kale, S., & Schapire, R. E. (2006). Algorithms for portfolio management based on the Newton method. In *Proceedings of the 23rd international conference on machine learning (ICML '06)* (pp. 9–16). New York: ACM. http://doi.acm.org/10.1145/1143844.1143846.

Arora, S., & Kale, S. (2007). A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the 39th annual ACM symposium on theory of computing (STOC '07)* (pp. 227–236). New York: ACM.

Arora, S., Hazan, E., & Kale, S. (2005). Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In *46th annual symposium on foundations of computer science (FOCS '05)* (pp. 339–348).

Bernstein, D. S. (2005). *Matrix mathematics: theory, facts, and formulas with application to linear systems theory*. Princeton: Princeton University Press.

Bhatia, R. (1997). *Matrix analysis*. Berlin: Springer.

Bousquet, O., & Warmuth, M. K. (2002). Tracking a small set of experts by mixing past posteriors. *J. Mach. Learn. Res.*, *3*, 363–396.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.

Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge: Cambridge University Press.

Cesa-Bianchi, N., Mansour, Y., & Stoltz, G. (2005). Improved second-order bounds for prediction with expert advice. In *Proceedings of the 18th annual conference on learning theory (COLT '05)* (pp. 217–232). Berlin: Springer.

Cover, T. M. (1991). Universal portfolios. *Math. Finance*, *1*(1), 1–29.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, *55*(1), 119–139.

Gordon, G. J. (2006). No-regret algorithms for online convex programs. In B. Schölkopf, J. C. Platt, & T. Hoffman (Eds.), *Proceedings of 20th annual conference on neural information processing systems (NIPS '06)* (pp. 489–496). Cambridge: MIT Press.

Hazan, E., Agarwal, A., & Kale, S. (2007). Logarithmic regret algorithms for online convex optimization. *Mach. Learn.*, *69*(2–3), 169–192.

Hazan, E., Kale, S., & Warmuth, M. K. (2010). On-line variance minimization in $O(n^2)$ per trial? In *Proceedings of the 23rd annual conference on learning theory (COLT '10)* (pp. 314–315).

Helmbold, D., & Warmuth, M. K. (2007). Learning permutations with exponential weights. In *Proceedings of the 20th annual conference on learning theory (COLT '07)* (pp. 469–483). Berlin: Springer.

Helmbold, D., & Warmuth, M. K. (2009). Learning permutations with exponential weights. *J. Mach. Learn. Res.*, *10*, 1705–1736.

Helmbold, D., Schapire, R. E., Singer, Y., & Warmuth, M. K. (1997). A comparison of new and old algorithms for a mixture estimation problem. *Mach. Learn.*, *27*(1), 97–119.

Helmbold, D., Schapire, R. E., Singer, Y., & Warmuth, M. K. (1998). On-line portfolio selection using multiplicative updates. *Math. Finance*, *8*(4), 325–347.

Helmbold, D. P., Kivinen, J., & Warmuth, M. K. (1999). Relative loss bounds for single neurons. *IEEE Trans. Neural Netw.*, *10*(6), 1291–1304.

Herbster, M., & Warmuth, M. K. (1998). Tracking the best expert. *Mach. Learn.*, *32*(2), 151–178.

Herbster, M., & Warmuth, M. K. (2001). Tracking the best linear predictor. *J. Mach. Learn. Res.*, *1*, 281–309.

Jain, R., Ji, Z., Upadhyay, S., & Watrous, J. (2010). QIP = PSPACE. In *Proceedings of the 42nd ACM symposium on theory of computing (STOC '10)* (pp. 573–582).

Kakade, S. M., Shalev-Shwartz, S., & Tewari, A. (2010). *Regularizaton techniques for learning with matrices*. arXiv:0910.0610v2.

Kalai, A. (2005). *A perturbation that makes follow the leader?* Equivalent to randomized weighted majority? Private communication.

Kalai, A., & Vempala, S. (2005). Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, *71*(3), 291–307. doi:10.1016/j.jcss.2004.10.016.

Kivinen, J., & Warmuth, M. K. (1997). Additive versus exponentiated gradient updates for linear prediction. *Inf. Comput.*, *132*(1), 1–64.

Kivinen, J., & Warmuth, M. K. (1999). Averaging expert predictions. In *Lecture notes in artificial intelligence: Vol. 1572. Computational learning theory, 4th European conference (EuroCOLT '99)*, Proceedings, Nordkirchen, Germany, March 29–31, 1999 (pp. 153–167). Berlin: Springer.

Kivinen, J., & Warmuth, M. K. (2001). Relative loss bounds for multidimensional regression problems. *Mach. Learn.*, *45*(3), 301–329.

Kivinen, J., Warmuth, M. K., & Hassibi, B. (2005). The p-norm generalization of the LMS algorithm for adaptive filtering. *IEEE Trans. Signal Process.*, *54*(5), 1782–1793.

Kuzmin, D., & Warmuth, M. K. (2005). Optimum follow the leader algorithm. In *Proceedings of the 18th annual conference on learning theory (COLT '05)* (pp. 684–686). Berlin: Springer. Open problem.

Kuzmin, D., & Warmuth, M. K. (2007). Online kernel PCA with entropic matrix updates. In *ACM international conference proceedings series*, *Proceedings of the 24rd international conference on machine learning (ICML '07)* (pp. 465–471).

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Mach. Learn.*, *2*(4), 285–318.

Littlestone, N. (1989). *Mistake bounds and logarithmic linear-threshold learning algorithms*. PhD thesis, Technical Report UCSC-CRL-89-11, University of California, Santa Cruz.

Littlestone, N., & Warmuth, M. K. (1994). The weighted majority algorithm. *Inf. Comput.*, *108*(2), 212–261. Preliminary version in *Proceedings of the 30th annual symposium on foundations of computer science (FOCS '89)*.

Nielsen, M. A., & Chuang, I. L. (2000). *Quantum computation and quantum information*. Cambridge: Cambridge University Press.

Shalev-Shwartz, S., & Singer, Y. (2006). Convex repeated games and Fenchel duality. In B. Schölkopf, J. C. Platt, & T. Hoffman (Eds.), *Proceedings of 20th annual conference on neural information processing systems (NIPS '06)* (pp. 1265–1272). Cambridge: MIT Press.

Tsuda, K., Rätsch, G., & Warmuth, M. K. (2005). Matrix exponentiated gradient updates for on-line learning and Bregman projections. *J. Mach. Learn. Res.*, *6*, 995–1018.

Vovk, V. (1990). Aggregating strategies. In *Proceedings of the 3rd annual workshop on computational learning theory* (pp. 371–383). Morgan Kaufmann: San Mateo.

Warmuth, M. K. (2007a). When is there a free matrix lunch. In *Proc. of the 20th annual conference on learning theory (COLT '07)*. Berlin: Springer. Open problem.

Warmuth, M. K. (2007b). Winnowing subspaces. In *Proceedings of the 24rd international conference on machine learning (ICML '07)*. New York: ACM.

Warmuth, M. K., & Kuzmin, D. (2006a). A Bayesian probability calculus for density matrices. In *Proc. 22nd conference on uncertainty in artificial intelligence (UAI '06)* (pp. 503–511). Morgan Kaufmann: San Mateo. Journal submission: http://www.soe.ucsc.edu/~manfred/last/bayescalc.pdf.

Warmuth, M. K., & Kuzmin, D. (2006b). Online variance minimization. In *Proceedings of the 19th annual conference on learning theory (COLT '06)* (pp. 514–528). Berlin: Springer.

Warmuth, M. K., & Kuzmin, D. (2006c). Randomized PCA algorithms with regret bounds that are logarithmic in the dimension. In *Advances in neural information processing systems 19 (NIPS '06)*. Cambridge: MIT Press.

Warmuth, M. K., & Kuzmin, D. (2008). Randomized PCA algorithms with regret bounds that are logarithmic in the dimension. *J. Mach. Learn. Res.*, *9*, 2217–2250.

Warmuth, M. K., & Kuzmin, D. (2010). Bayesian generalized probability calculus for density matrices. *Mach. Learn.*, *78*(1–2), 63–101.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proc. 20th int. conference on machine learning (ICML '03)* (pp. 928–936).