Online Variance Minimization*

Manfred K. Warmuth and Dima Kuzmin

Computer Science Department
University of California, Santa Cruz
{manfred, dima}@cse.ucsc.edu

Abstract. We design algorithms for two online variance minimization problems. Specifically, in every trial t our algorithms get a covariance matrix $\boldsymbol{\mathcal{C}}_t$ and try to select a parameter vector $\boldsymbol{w}_t$ such that the total variance over a sequence of trials $\sum_t \boldsymbol{w}_t^\top \boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t$ is not much larger than the total variance of the best parameter vector $\boldsymbol{u}$ chosen in hindsight. Two parameter spaces 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. For the first parameter space we apply the Exponentiated Gradient algorithm which is motivated with a relative entropy. In the second case the algorithm maintains a mixture of unit vectors which is represented as 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 case we prove bounds on the additional total variance incurred by the online algorithm over the best offline parameter.

1 Introduction

In one of the simplest settings of learning with expert advice [FS97], the learner has to commit to 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 whose total loss over a sequence of trials is close to loss of the best expert in all trials, i.e. the total loss of the online algorithm $\sum_t \boldsymbol{w}_t \cdot \boldsymbol{l}_t$ should be close to the total loss of the best expert chosen in hindsight, which is $\inf_i \sum_t l_{t,i}$, 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 $\boldsymbol{\mathcal{C}}$ of a random loss vector $\boldsymbol{l}$, where $\boldsymbol{\mathcal{C}}(i,j)$ 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 $\boldsymbol{\mathcal{C}}$ can be expressed as $\boldsymbol{w}^\top \boldsymbol{\mathcal{C}} \boldsymbol{w} = \mathbf{Var}(\boldsymbol{w} \cdot \boldsymbol{l})$. Our goal

* Supported by NSF grant CCR 9821087. Some of this work was done while visiting National ICT Australia in Canberra.

G. Lugosi and H.U. Simon (Eds.): COLT 2006, LNAI 4005, pp. 514–528, 2006.

is to minimize the total variance over a sequence of trials: $\sum_t \boldsymbol{w}_t^\top \boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t$. More precisely, we want algorithms whose total variance is close to the total variance of the best probability vector $\boldsymbol{u}$ chosen in hindsight, i.e. the total variance of the algorithm should be close to $\inf_{\boldsymbol{u}} \boldsymbol{u}^\top (\sum_t \boldsymbol{\mathcal{C}}_t) \boldsymbol{u}$ (where the minimization is over the probability simplex).

In a more general setting one actually might want to optimize trade-offs between first-order and second order terms: $\boldsymbol{w} \cdot \boldsymbol{l} + \gamma \boldsymbol{w}^\top \boldsymbol{\mathcal{C}} \boldsymbol{w}$, where $\gamma \geq 0$ is a risk-aversion parameter. Such problems arise in Markowitz portfolio optimization (See e.g. discussion in [BV04], Section 4.4). For the sake of simplicity, in this paper we focus on minimizing the variance by itself.

We develop an algorithm for the above online variance minimization problem. The parameter space is the probability simplex. 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 [KW97]. The bounds we obtain are similar to the bounds of the Exponentiated Gradient algorithm when applied to linear regression.

In the second part of the paper we focus on the same online variance minimization problem, but now the parameter space that we compare against is the unit sphere of direction vectors instead of the probability simplex and the total loss of the algorithm is to be close to $\inf_{\boldsymbol{u}} \boldsymbol{u}^\top (\sum_t \boldsymbol{\mathcal{C}}_t) \boldsymbol{u}$, 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 $\sum_t \boldsymbol{\mathcal{C}}_t$.

Note that the variance $\boldsymbol{u}^\top \boldsymbol{\mathcal{C}} \boldsymbol{u}$ can be rewritten using the trace operator: $\boldsymbol{u}^\top \boldsymbol{\mathcal{C}} \boldsymbol{u} = \mathrm{tr}(\boldsymbol{u}^\top \boldsymbol{\mathcal{C}} \boldsymbol{u}) = \mathrm{tr}(\boldsymbol{u}\boldsymbol{u}^\top \boldsymbol{\mathcal{C}})$. 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: $\inf_{\boldsymbol{u}} \mathrm{tr}(\boldsymbol{u}\boldsymbol{u}^\top (\sum_t \boldsymbol{\mathcal{C}}_t))$ (where $\boldsymbol{u}$ is unit length).[1]

In the original experts setting, the offline problem involved a minimum over experts. Now its a minimum over dyads and the best dyad corresponds to an eigenvector with minimum eigenvalue. The algorithm for the original expert setting 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. This algorithm is the Continuous Weighted Majority (WMC) [LW94] (which was reformulated as the Hedge algorithm in [FS97]). It uses exponentially decaying weights $w_{t,i} = \dfrac{e^{-\eta \sum_{q=1}^{t-1} l_{q,i}}}{Z_t}$, where Z_t is a normalization factor.

In the generalized setting we need to maintain uncertainty over 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). Note that the vector of eigenvalues of such matrices is a probability vector. Using the methodology of [TRW05, War05] we develop a matrix version of the Weighted Majority algorithm for solving our second online variance minimization problem.

[1] In this paper we upper bound the total variance of our algorithm, whereas the generalized Bayes rule of [War05, WK06] is an algorithm for which the sum of the negative logs of the variances is upper bounded.

Now the density matrix parameter has the form $\boldsymbol{\mathcal{W}}_t = \dfrac{\mathbf{exp}(-\eta \sum_{q=1}^{t-1} \boldsymbol{\mathcal{C}}_q)}{Z_t}$, where **exp** is the matrix exponential and Z_t normalizes the trace of the parameter matrix to one. When the covariance matrices $\boldsymbol{\mathcal{C}}_q$ are the diagonal matrices $\mathrm{diag}(\boldsymbol{l}_q)$ then the matrix update becomes the original expert update. In other words the original update may be seen as a special case of the new matrix update when the eigenvectors are fixed to the standard basis vectors and are not updated.

The original weighted majority type update may be seen as a softmin calculation, because as $\eta \to \infty$, the parameter vector $\boldsymbol{w}_t$ puts all of its weight on $\arg\min_i \sum_{q=1}^{t-1} l_{q,i}$. Similarly, the generalized update is a soft eigenvector calculation for the eigenvectors with the 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 $\mathrm{tr}(\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{C}})$. If the eigendecomposition of the parameter matrix $\boldsymbol{\mathcal{W}}$ consists of the eigenvectors $\boldsymbol{w}_i$ and associated eigenvalues ω_i then this loss can be rewritten as

$$\mathrm{tr}(\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{C}}) = \mathrm{tr}((\sum \omega_i \boldsymbol{w}_i \boldsymbol{w}_i^\top)\boldsymbol{\mathcal{C}}) = \sum_i \omega_i \, \boldsymbol{w}_i^\top \boldsymbol{\mathcal{C}} \boldsymbol{w}_i$$

In other words it 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 $\boldsymbol{\mathcal{W}}$ represents a mixture state of a particle and $\boldsymbol{\mathcal{C}}$ the instrument (See [War05, WK06] for additional discussion). Again the dot product $\boldsymbol{w} \cdot \boldsymbol{l}$ is the special case when the eigenvectors are the standard basis vectors, i.e.

$$\mathrm{tr}(\mathrm{diag}(\boldsymbol{w})\,\mathrm{diag}(\boldsymbol{l})) = \mathrm{tr}((\sum w_i \boldsymbol{e}_i \boldsymbol{e}_i^\top)\ \mathrm{diag}(\boldsymbol{l})) = \sum_i w_i \ \boldsymbol{e}_i^\top\ \mathrm{diag}(\boldsymbol{l})\boldsymbol{e}_i = \sum_i w_i l_i.$$

The new update is motivated and analyzed using the quantum relative entropy (due to Umegaki, see e.g. [NC00]) instead of the standard relative entropy (also called Kullback-Leibler divergence). The analysis is a fancier version of the original online loss bound for WMC that uses the Golden-Thompson inequality and some lemmas developed in [TRW05].

2 Variance Minimization over the Probability Simplex

2.1 Definitions

In this paper we only consider symmetric matrices. Such matrices always have an eigendecomposition of the form $\boldsymbol{\mathcal{W}} = \boldsymbol{W}\boldsymbol{\omega}\boldsymbol{W}^\top$, where $\boldsymbol{W}$ is an orthogonal matrix of eigenvectors and $\boldsymbol{\omega}$ is a diagonal matrix of the corresponding eigenvalues. Alternatively, the decomposition can be written as $\boldsymbol{\mathcal{W}} = \sum_i \omega_i \boldsymbol{w}_i \boldsymbol{w}_i^\top$, 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 $\boldsymbol{\mathcal{M}}$ is called *positive semidefinite* if for all vectors $\boldsymbol{w}$ we have $\boldsymbol{w}^\top \boldsymbol{\mathcal{M}} \boldsymbol{w} \geq 0$. This is also written as a generalized inequality $\boldsymbol{\mathcal{M}} \succeq \mathbf{0}$. In eigenvalue terms this

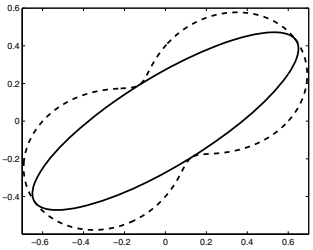

Fig. 1. An ellipse $\boldsymbol{\mathcal{C}}$ in $\mathbb{R}^2$: The eigenvectors are the directions of the axes and the eigenvalues their lengths from the origin. Ellipses are weighted combinations of the one-dimensional degenerate ellipses (dyads) corresponding to the axes. (For unit $\boldsymbol{w}$,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

means that all eigenvalues of matrix are ≥ 0. A matrix is *strictly positive definite* if all eigenvalues are > 0. In what follows we will drop the semi- prefix and call any matrix $\boldsymbol{\mathcal{M}} \succeq \mathbf{0}$ simply positive definite.

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

$$\begin{aligned}\mathbf{Var}(\boldsymbol{l}^\top\boldsymbol{w}) =& \boldsymbol{E}\left((\boldsymbol{l}^\top\boldsymbol{w} - \boldsymbol{E}(\boldsymbol{l}^\top\boldsymbol{w}))^2\right)\\ =& \boldsymbol{E}\left(((\boldsymbol{l}^\top - \boldsymbol{E}(\boldsymbol{l}^\top))\boldsymbol{w})^\top((\boldsymbol{l}^\top - \boldsymbol{E}(\boldsymbol{l}^\top))\boldsymbol{w})\right)\\ =& \boldsymbol{E}\left(\boldsymbol{w}^\top(\boldsymbol{l} - \boldsymbol{E}(\boldsymbol{l}))(\boldsymbol{l} - \boldsymbol{E}(\boldsymbol{l}))^\top)\boldsymbol{w}\right)\\ =& \boldsymbol{w}^\top\boldsymbol{\mathcal{C}}\boldsymbol{w}.\end{aligned}$$

A covariance matrix can be depicted as an ellipse $\{\boldsymbol{\mathcal{C}}\boldsymbol{w} : \|\boldsymbol{w}\|_2 = 1\}$ centered at the origin. The eigenvectors of $\boldsymbol{\mathcal{C}}$ form the axes of the ellipse and eigenvalues are the lengths of the axes from the origin (See Figure 1 taken from [War05]).

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:

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

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 both arguments.

2.2 Risk Minimization

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 $\boldsymbol{\mathcal{C}}$, then the variance of the capital change for our portfolio is $\boldsymbol{w}^\top \boldsymbol{\mathcal{C}} \boldsymbol{w}$. This variance is clearly associated with the risk of our investment. Our problem is then to "track" the performance of minimum risk portfolio over a sequence of trading periods.

2.3 Algorithm and Motivation

Let us reiterate the setup and the goal for our algorithm. On every trial t it must produce a probability vector $\boldsymbol{w}_t$. It then gets a covariance matrix $\boldsymbol{\mathcal{C}}_t$ and incurs a loss equal to the variance $\boldsymbol{w}_t^\top \boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t$. Thus for a sequence of T trials the total loss of the algorithm will be $L_{\text{alg}} = \sum_{t=1}^{T} \boldsymbol{w}_t^\top \boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t$. We want this loss to be comparable to the total variance of the best probability vector $\boldsymbol{u}$ chosen in hindsight, i.e. $L_{\boldsymbol{u}} = \min_{\boldsymbol{u}} \boldsymbol{u}^\top \left(\sum_{t=1}^{T} \boldsymbol{\mathcal{C}}_t \right) \boldsymbol{u}$, where $\boldsymbol{u}$ lies in the probability simplex. 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.

The natural choice for an online algorithm for this problem is the Exponentiated Gradient algorithm of [KW97] since it maintains a probability vector as its parameter. Recall that for a general loss function $L_t(\boldsymbol{w}_t)$, the probability vector of Exponentiated Gradient algorithm is updated as

$$w_{t+1,i} = \frac{w_{t,i} e^{-\eta (\nabla L_t(\boldsymbol{w}_t))_i}}{\sum_i w_{t,i} e^{-\eta (\nabla L_t(\boldsymbol{w}_t))_i}}.$$

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:

$$\boldsymbol{w}_{t+1} \approx \underset{\boldsymbol{w} \text{ prob.vec.}}{\arg\min} \; \mathrm{d}(\boldsymbol{w}, \boldsymbol{w}_t) + \eta L_t(\boldsymbol{w}),$$

where $\approx$ comes from the fact that the gradient at $\boldsymbol{w}_{t+1}$ that should appear in the exponent is approximated by the gradient at $\boldsymbol{w}_t$ (See [KW97] for more discussion). In our application, $L_t(\boldsymbol{w}_t) = \frac{1}{2} \boldsymbol{w}_t^\top \boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t$ and $\nabla L_t(\boldsymbol{w}_t) = \boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t$, leading to the following update:

$$w_{t+1,i} = \frac{w_{t,i} e^{-\eta (\boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t)_i}}{\sum_{i=1}^{n} w_{t,i} e^{-\eta (\boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t)_i}}.$$

2.4 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.

Lemma 1. *Let $\boldsymbol{w}_t$ be the weight vector of the algorithm before trial t and let $\boldsymbol{u}$ be an arbitrary comparison probability vector. Also, let r be the bound on the range of elements in covariance matrix $\boldsymbol{\mathcal{C}}_t$, specifically let $\max_{i,j}|\boldsymbol{\mathcal{C}}_t(i,j)| \leq \frac{r}{2}$. 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:*

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

Proof. The proof given in Appendix A follows the same outline as Lemma 5.8 of [KW97] which gives an inequality for the Exponentiated Gradient algorithm when applied to linear regression. □

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

$$L_{alg} \;\leq\; (1+c)\, L_{\boldsymbol{u}} + \left(1 + \frac{1}{c}\right) r\ \mathrm{d}(\boldsymbol{u}, \boldsymbol{w}_1).$$

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 1 and obtain:

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

Summing over the trials t results in:

$$\frac{c}{c+1} L_{\mathrm{alg}} - c L_{\boldsymbol{u}} \leq r(\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}_1) - \mathrm{d}(\boldsymbol{u}, \boldsymbol{w}_{t+1})) \leq r\, \mathrm{d}(\boldsymbol{u}, \boldsymbol{w}_1).$$

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 1. *Let $\boldsymbol{\mathcal{C}}_1, \ldots, \boldsymbol{\mathcal{C}}_T$ be an arbitrary sequence of covariance matrices such that $\max_{i,j}|\boldsymbol{\mathcal{C}}_t(i,j)| \leq \frac{r}{2}$ and assume that $\boldsymbol{u}^\top \sum_{t=1}^{T} \boldsymbol{\mathcal{C}}_t \boldsymbol{u} \leq L$. Then running our algorithm with uniform start vector $\boldsymbol{w}_1 = (\frac{1}{n}, \ldots, \frac{1}{n})$ and learning rate $\eta = \frac{2\sqrt{L \log n}}{r\sqrt{\log n} + \sqrt{rL}}$ leads to the following bound:*

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

Proof. By Lemma 2 and since $\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}_1) \leq \log n$:

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

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. □

3 Variance Minimization over the Unit Sphere

3.1 Definitions

The *trace* $\mathrm{tr}(\boldsymbol{\mathcal{A}})$ of a square matrix $\boldsymbol{\mathcal{A}}$ 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. $\mathrm{tr}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{B}}) = \mathrm{tr}(\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{A}}) = \sum_{i,j} \boldsymbol{\mathcal{A}}(i,j)\boldsymbol{\mathcal{B}}(i,j)$. The trace is also a linear operator, that is $\mathrm{tr}(a\boldsymbol{\mathcal{A}} + b\boldsymbol{\mathcal{B}}) = a\,\mathrm{tr}(\boldsymbol{\mathcal{A}}) + b\,\mathrm{tr}(\boldsymbol{\mathcal{B}})$. Another useful property of the trace is its cycling invariance, i.e. $\mathrm{tr}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{C}}) = \mathrm{tr}(\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{A}}) = \mathrm{tr}(\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{B}})$. A particular instance of this is the following manipulation: $\boldsymbol{u}^\top \boldsymbol{\mathcal{A}}\boldsymbol{u} = \mathrm{tr}(\boldsymbol{u}^\top \boldsymbol{\mathcal{A}}\boldsymbol{u}) = \mathrm{tr}(\boldsymbol{\mathcal{A}}\boldsymbol{u}\boldsymbol{u}^\top)$.

Dyads have trace one because $\mathrm{tr}(\boldsymbol{u}\boldsymbol{u}^\top) = \boldsymbol{u}^\top \boldsymbol{u} = 1$. We generalize mixtures or probability vectors to *density matrices*. Such matrices are mixtures of any number of dyads, i.e. $\boldsymbol{\mathcal{W}} = \sum_i \alpha_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$ where $\alpha_j \geq 0$ and $\sum_i \alpha_i = 1$. Equivalently, density matrices are arbitrary symmetric positive definite matrices of trace one. Any density matrix $\boldsymbol{\mathcal{W}}$ can be decomposed into a sum of exactly n dyads corresponding to the orthogonal set of its eigenvectors $\boldsymbol{w}_i$, i.e. $\boldsymbol{\mathcal{W}} = \sum_{i=1}^n \omega_i \boldsymbol{w}_i \boldsymbol{w}_i^\top$ where the vector $\boldsymbol{\omega}$ of the n eigenvalues must be a probability vector. In quantum physics density matrices over the field of complex numbers represent the mixed state of a physical system.

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

$$\mathbf{exp}(\boldsymbol{\mathcal{A}}) = \sum_i e^{\alpha_i}\, \boldsymbol{a}_i \boldsymbol{a}_i^\top, \quad \mathbf{log}(\boldsymbol{\mathcal{A}}) = \sum_i \log \alpha_i\, \boldsymbol{a}_i \boldsymbol{a}_i^\top$$

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: $\mathbf{exp}(\boldsymbol{\mathcal{A}} + \boldsymbol{\mathcal{B}}) = \mathbf{exp}(\boldsymbol{\mathcal{A}})\,\mathbf{exp}(\boldsymbol{\mathcal{B}})$. However this is only true when the symmetric matrices $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ commute, i.e. $\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{B}} = \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{A}}$, 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:

$$\mathrm{tr}(\mathbf{exp}(\boldsymbol{\mathcal{A}} + \boldsymbol{\mathcal{B}})) \leq \mathrm{tr}(\mathbf{exp}(\boldsymbol{\mathcal{A}})\,\mathbf{exp}(\boldsymbol{\mathcal{B}})).$$

The following quantum relative entropy is a generalization of the classical relative entropy to density matrices due to Umegaki (see e.g. [NC00]):

$$\Delta(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{W}}) = \mathrm{tr}(\boldsymbol{\mathcal{U}}(\log \boldsymbol{\mathcal{U}} - \log \boldsymbol{\mathcal{W}})).$$

We will also use generalized inequalities for the cone of positive definite matrices: $\boldsymbol{\mathcal{A}} \preceq \boldsymbol{\mathcal{B}}$ if $\boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{A}}$ positive definite.

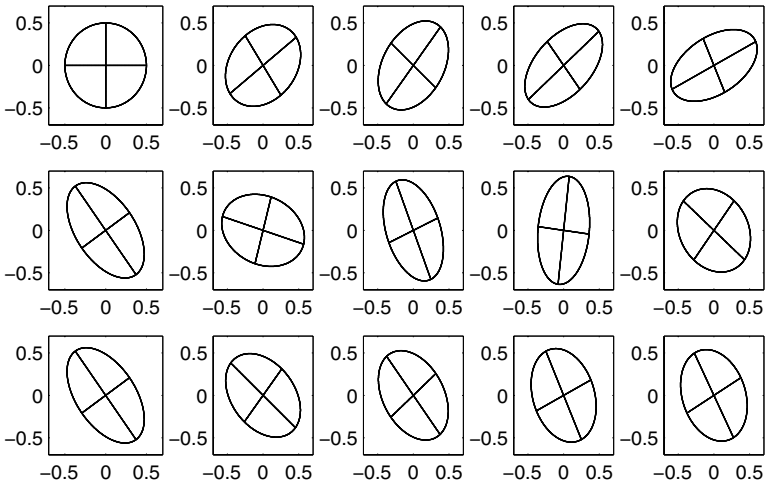

Fig. 2. The figure depicts a sequence of updates for the density matrix algorithm when the dimension is 2. All 2-by-2 matrices are represented as ellipses. The top row shows the density matrices $\boldsymbol{\mathcal{W}}_t$ chosen by the algorithm. The middle row shows the covariance matrix $\boldsymbol{\mathcal{C}}_t$ received in that trial. Finally, the bottom row is the average $\boldsymbol{\mathcal{C}}_{\leq t} = \frac{\sum_{q=1}^{t} \boldsymbol{\mathcal{C}}_t}{t}$ of all covariance matrices so far. By the update (1), $\boldsymbol{\mathcal{W}}_{t+1} = \frac{\exp(-\eta t \boldsymbol{\mathcal{C}}_{\leq t})}{Z_t}$, where Z_t is a normalization. Therefore, $\boldsymbol{\mathcal{C}}_{\leq t}$ in the third row has the same eigensystem as the density matrix $\boldsymbol{\mathcal{W}}_{t+1}$ in the next column of 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.

3.2 Applications

We develop online algorithms that perform as well as the eigenvector associated with a minimum (or maximum) eigenvalue. It seems that online versions of principal component analysis and other spectral methods can also be developed using the methodology of this paper. For instance, spectral clustering methods of [CSTK01] use a similar form of loss.

3.3 Algorithm and Motivation

As before we briefly review our setup. On each trial t our algorithm chooses a density matrix $\boldsymbol{\mathcal{W}}_t$ described as a mixture $\sum_i \omega_{t,i} \boldsymbol{w}_{t,i} \boldsymbol{w}_{t,i}^\top$. It then receives a covariance matrix $\boldsymbol{\mathcal{C}}_t$ and incurs a loss equal to the expected variance of its mixture:

$$\mathrm{tr}(\boldsymbol{\mathcal{W}}_t \boldsymbol{\mathcal{C}}_t) = \mathrm{tr}((\sum_i \omega_{t,i} \boldsymbol{w}_{t,i} \boldsymbol{w}_{t,i}^\top) \boldsymbol{\mathcal{C}}_t) = \sum_i \omega_{t,i} \boldsymbol{w}_{t,i}^\top \boldsymbol{\mathcal{C}}_t \boldsymbol{w}_{t,i}.$$

On a sequence of T trials the total loss of the algorithm will be $L_{\mathrm{alg}} = \sum_{t=1}^{T} \mathrm{tr}(\boldsymbol{\mathcal{W}}_t \boldsymbol{\mathcal{C}}_t)$. 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. $L_{\boldsymbol{u}} = \mathrm{tr}(\boldsymbol{u}\boldsymbol{u}^\top \sum_t \boldsymbol{\mathcal{C}}_t)$ $= \boldsymbol{u}^\top(\sum_t \boldsymbol{\mathcal{C}}_t)\boldsymbol{u}$. 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:

$$\min_{\boldsymbol{\mathcal{U}} \text{ dens.mat.}} \mathrm{tr}(\boldsymbol{\mathcal{U}}\boldsymbol{\mathcal{C}}) = \min_{\boldsymbol{u}\,:\,\|\boldsymbol{u}\|_2=1} \boldsymbol{u}^\top \boldsymbol{\mathcal{C}}\boldsymbol{u}$$

Curiously enough our, loss $\mathrm{tr}(\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{C}})$ has interpretation in quantum mechanics as the expected outcome of measuring a physical system in mixture state $\boldsymbol{\mathcal{W}}$ with instrument $\boldsymbol{\mathcal{C}}$. Let $\boldsymbol{\mathcal{C}}$ 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 $\mathrm{tr}(\boldsymbol{u}\boldsymbol{u}^\top\boldsymbol{\mathcal{C}}) = \sum_i (\boldsymbol{u}\cdot\boldsymbol{c}_i)^2\gamma_i$. For a mixed state $\boldsymbol{\mathcal{W}}$ we have the following double expectation:

$$\mathrm{tr}(\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{C}}) = \mathrm{tr}\left((\sum_i \omega_i\, \boldsymbol{w}_i\boldsymbol{w}_i^\top)(\sum_j \gamma_j\, \boldsymbol{c}_j\boldsymbol{c}_j^\top)\right) = \sum_{i,j} (\boldsymbol{w}_i\cdot\boldsymbol{c}_j)^2\, \gamma_i\omega_j,$$

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 $\boldsymbol{\mathcal{C}}$ does not have to be positive definite, but only symmetric. The algorithm and the proof of bounds in fact work fine for this case, but the meaning of the algorithm when $\boldsymbol{\mathcal{C}}$ is not a covariance matrix is less clear, since despite all these connections 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 [War05, WK06] 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:

$$\boldsymbol{\mathcal{W}}_{t+1} = \operatorname*{arg\,min}_{\boldsymbol{\mathcal{W}} \text{ dens.mat.}} \Delta(\boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{W}}_t) + \eta\, \mathrm{tr}(\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{C}}_t)$$

Using a Lagrangian that enforces the trace constraint [TRW05], it is easy to solve this constrained minimization problem:

$$\boldsymbol{\mathcal{W}}_{t+1} = \frac{\exp(\log \boldsymbol{\mathcal{W}}_t - \eta\boldsymbol{\mathcal{C}}_t)}{\mathrm{tr}(\exp(\log \boldsymbol{\mathcal{W}}_t - \eta\boldsymbol{\mathcal{C}}_t))} = \frac{\exp(-\eta \sum_{q=1}^t \boldsymbol{\mathcal{C}}_q)}{\mathrm{tr}(\exp(-\eta \sum_{q=1}^t \boldsymbol{\mathcal{C}}_q))}. \tag{1}$$

Note that for the second equation we assumed that $\boldsymbol{\mathcal{W}}_1 = \frac{1}{n}\boldsymbol{I}$. The update is a special case of the Matrix Exponentiated Gradient update with the linear loss $\mathrm{tr}(\boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{C}}_t)$.

3.4 Proof Methodology

For the sake of clarity, we begin by recalling the proof of the worst-case loss bound for the Continuous Weighted Majority (WMC)/Hedge algorithm in the expert advice setting [LW94]. In doing so we clarify the dependence of the algorithm on the range of the losses. The update of that algorithm is given by:

$$w_{t+1,i} = \frac{w_{t,i}e^{-\eta l_{t,i}}}{\sum_i w_{t,i}e^{-\eta l_{t,i}}} \tag{2}$$

The proof always starts by considering the progress made during the update towards any comparison vector/parameter $\boldsymbol{u}$ in terms of the motivating divergence for the algorithm, which in this case is the relative entropy:

$$\mathrm{d}(\boldsymbol{u},\boldsymbol{w}_t) - \mathrm{d}(\boldsymbol{u},\boldsymbol{w}_{t+1}) = \sum_i u_i \log\frac{w_{t+1,i}}{w_{t,i}} = -\eta\,\boldsymbol{u}\cdot\boldsymbol{l}_t - \log\sum_i w_{t,i}e^{-\eta l_{t,i}}.$$

We assume that $l_{t,i} \in [0,r]$, for $r > 0$, and use the inequality $\beta^x \leq 1-(1-\beta^r)\frac{x}{r}$, for $x \in [0,r]$, with $\beta = e^{-\eta}$:

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

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

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

and rewrite the above to

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

Here $\boldsymbol{w}_t\cdot\boldsymbol{l}_t$ 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 $l_{t,i} \in [0,r]$. However, it suffices to assume that $\max_i l_{t,i} - \min_i l_{t,i} \leq r$. In other words, the individual losses can be positive or negative, as long as their range is bounded by r. For further discussion pertaining to the issues with losses having different signs see [CBMS05]. 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 $\tilde{l}_{t,i} := l_{t,i} - \min_i l_{t,i}$. The update remains unchanged when the shifted losses $\tilde{l}_{t,i}$ are used in place of the original losses $l_{t,i}$ and we immediately get the inequality

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

Summing over t and dropping the $\mathrm{d}(\boldsymbol{u},\boldsymbol{w}_{t+1}) \geq 0$ term results in a bound that holds for any $\boldsymbol{u}$ and thus for the best $\boldsymbol{u}$ as well:

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

We can now tune the learning rate following [FS97]: if $\sum_t \boldsymbol{u} \cdot \tilde{\boldsymbol{l}}_t \leq \tilde{L}$ and $\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}_1) \leq D \leq \ln n$, then with $\eta = \frac{\log(1+\sqrt{2D/\tilde{L}})}{r}$ we get the bound

$$\sum_t \boldsymbol{w}_t \cdot \tilde{\boldsymbol{l}}_t \leq \sum_t \boldsymbol{u} \cdot \tilde{\boldsymbol{l}}_t + \sqrt{2r\tilde{L}D} + r\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}_1),$$

which is equivalent to

$$\underbrace{\sum_t \boldsymbol{w}_t \cdot \boldsymbol{l}_t}_{L_{\mathrm{alg}}} \leq \underbrace{\sum_t \boldsymbol{u} \cdot \boldsymbol{l}_t}_{L_{\boldsymbol{u}}} + \sqrt{2r\tilde{L}D} + r\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}_1).$$

Note that $\tilde{L}$ is defined wrt the tilde versions of the losses and the update as well as the above bound is invariant under shifting the loss vectors 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.

3.5 Proof of Relative Loss Bounds

In addition to the Golden-Thompson inequality we will need lemmas 2.1 and 2.2 from [TRW05]:

Lemma 3. *For any symmetric $\boldsymbol{\mathcal{A}}$, such that $\mathbf{0} \preceq \boldsymbol{\mathcal{A}} \preceq \boldsymbol{I}$ and any $\rho_1, \rho_2 \in \mathbb{R}$ the following holds:*

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

Lemma 4. *For any positive semidefinite $\boldsymbol{\mathcal{A}}$ and any symmetric $\boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}}$, $\boldsymbol{\mathcal{B}} \preceq \boldsymbol{\mathcal{C}}$ implies* $\mathrm{tr}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{B}}) \leq \mathrm{tr}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{C}})$.

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

Theorem 2. *For any sequence of covariance matrices $\boldsymbol{\mathcal{C}}_1, \ldots, \boldsymbol{\mathcal{C}}_T$ such that $\mathbf{0} \preceq \boldsymbol{\mathcal{C}}_t \preceq r\boldsymbol{I}$ and for any learning rate η, the following bound holds for arbitrary density matrix $\boldsymbol{\mathcal{U}}$:*

$$\mathrm{tr}(\boldsymbol{\mathcal{W}}_t\boldsymbol{\mathcal{C}}_t) \leq \frac{r(\Delta(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{W}}_t) - \Delta(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{W}}_{t+1})) + \eta r\,\mathrm{tr}(\boldsymbol{\mathcal{U}}\boldsymbol{\mathcal{C}}_t)}{1 - e^{-r\eta}}.$$

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

$$\begin{aligned}
\Delta(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{W}}_t) - \Delta(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{W}}_{t+1}) &= \mathrm{tr}(\boldsymbol{\mathcal{U}}(\log \boldsymbol{\mathcal{U}} - \log \boldsymbol{\mathcal{W}}_t)) - \mathrm{tr}(\boldsymbol{\mathcal{U}}(\log \boldsymbol{\mathcal{U}} - \log \boldsymbol{\mathcal{W}}_{t+1})) \\
&= -\mathrm{tr}\left(\boldsymbol{\mathcal{U}}\left(\log \boldsymbol{\mathcal{W}}_t + \log \frac{\mathbf{exp}(\log \boldsymbol{\mathcal{W}}_t - \eta\boldsymbol{\mathcal{C}}_t)}{\mathrm{tr}(\mathbf{exp}(\log \boldsymbol{\mathcal{W}}_t - \eta\boldsymbol{\mathcal{C}}_t))}\right)\right) \\
&= -\eta\,\mathrm{tr}(\boldsymbol{\mathcal{U}}\boldsymbol{\mathcal{C}}_t) - \log(\mathrm{tr}(\mathbf{exp}(\log \boldsymbol{\mathcal{W}}_t - \eta\boldsymbol{\mathcal{C}}_t))).
\end{aligned} \tag{3}$$

We will now bound the log of trace term. First, the following holds via the Golden-Thompson inequality:

$$\operatorname{tr}(\mathbf{exp}(\log \boldsymbol{\mathcal{W}}_t - \eta \boldsymbol{\mathcal{C}}_t)) \leq \operatorname{tr}(\boldsymbol{\mathcal{W}}_t \mathbf{exp}(-\eta \boldsymbol{\mathcal{C}}_t)). \tag{4}$$

Since $\mathbf{0} \preceq \frac{\boldsymbol{\mathcal{C}}_t}{r} \preceq \boldsymbol{I}$, we can use Lemma 3 with $\rho_1 = -\eta r$, $\rho_2 = 0$:

$$\mathbf{exp}(-\eta \boldsymbol{\mathcal{C}}_t) \preceq \boldsymbol{I} - \frac{\boldsymbol{\mathcal{C}}_t}{r}(1 - e^{-\eta r}).$$

Now multiply both sides on the left with $\boldsymbol{\mathcal{W}}_t$ and take a trace. The inequality is preserved according to Lemma 4:

$$\operatorname{tr}(\boldsymbol{\mathcal{W}}_t \mathbf{exp}(-\eta \boldsymbol{\mathcal{C}}_t)) \leq \left(1 - \frac{\operatorname{tr}(\boldsymbol{\mathcal{W}}_t \boldsymbol{\mathcal{C}}_t)}{r}(1 - e^{-r\eta})\right).$$

Taking logs of both sides we have:

$$\log(\operatorname{tr}(\boldsymbol{\mathcal{W}}_t \mathbf{exp}(-\eta \boldsymbol{\mathcal{C}}_t))) \leq \log\left(1 - \frac{\operatorname{tr}(\boldsymbol{\mathcal{W}}_t \boldsymbol{\mathcal{C}}_t)}{r}(1 - e^{-\eta r})\right). \tag{5}$$

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

$$\log\left(1 - \frac{\operatorname{tr}(\boldsymbol{\mathcal{W}}_t \boldsymbol{\mathcal{C}}_t)}{r}(1 - e^{-r\eta})\right) \leq -\frac{\operatorname{tr}(\boldsymbol{\mathcal{W}}_t \boldsymbol{\mathcal{C}}_t)}{r}(1 - e^{-r\eta}). \tag{6}$$

By combining inequalities (4-6), we obtain the following bound on the log trace term:

$$-\log(\operatorname{tr}(\mathbf{exp}(\log \boldsymbol{\mathcal{W}}_t - \eta \boldsymbol{\mathcal{C}}_t))) \geq \frac{\operatorname{tr}(\boldsymbol{\mathcal{W}}_t \boldsymbol{\mathcal{C}}_t)}{r}(1 - e^{-r\eta}).$$

Plugging this into equation (3) we obtain

$$r(\Delta(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{W}}_t) - \Delta(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{W}}_{t+1})) + \eta r \operatorname{tr}(\boldsymbol{\mathcal{U}} \boldsymbol{\mathcal{C}}_t) \geq \operatorname{tr}(\boldsymbol{\mathcal{W}}_t \boldsymbol{\mathcal{C}}_t)(1 - e^{-r\eta}),$$

which is the inequality of the theorem. □

Note the our density matrix update (1) is invariant wrt the variable change $\widetilde{\boldsymbol{\mathcal{C}}_t} = \boldsymbol{\mathcal{C}}_t - \lambda_{\min}(\boldsymbol{\mathcal{C}}_t)\boldsymbol{I}$. Therefore by the above theorem, the following inequality holds whenever $\lambda_{\max}(\boldsymbol{\mathcal{C}}_t) - \lambda_{\min}(\boldsymbol{\mathcal{C}}_t) \leq r$:

$$\operatorname{tr}(\boldsymbol{\mathcal{W}}_t \widetilde{\boldsymbol{\mathcal{C}}_t}) \leq \frac{r(\Delta(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{W}}_t) - \Delta(\boldsymbol{\mathcal{U}}, \boldsymbol{\mathcal{W}}_{t+1})) + \eta r \operatorname{tr}(\boldsymbol{\mathcal{U}} \widetilde{\boldsymbol{\mathcal{C}}_t})}{1 - e^{-r\eta}}.$$

We can now sum over trials and tune the learning rate as done at the end of Section 3.4. If $\sum_t \operatorname{tr}(\boldsymbol{\mathcal{U}} \widetilde{\boldsymbol{\mathcal{C}}_t}) \leq \tilde{L}$ and $\Delta(\boldsymbol{\mathcal{U}}, \boldsymbol{W}_1) \leq D$, with $\eta = \frac{\log(1 + \sqrt{\frac{2D}{\tilde{L}}})}{r}$ we get the bound:

$$\underbrace{\sum_t \operatorname{tr}(\boldsymbol{\mathcal{W}}_t \boldsymbol{\mathcal{C}}_t)}_{L_{\text{alg}}} \leq \underbrace{\sum_t \operatorname{tr}(\boldsymbol{\mathcal{U}} \boldsymbol{\mathcal{C}}_t)}_{L_{\boldsymbol{\mathcal{U}}}} + \sqrt{2r\tilde{L}D} + r\Delta(\boldsymbol{\mathcal{U}}, \boldsymbol{W}_1).$$

4 Conclusions

We presented two algorithms for online variance minimization problems. For the first problem, the variance was measured along a probability vector. It would be interesting to combine this work with the online algorithms considered in [HSSW98, Cov91] that maximize the return of the portfolio. It should be possible to design online algorithms that minimize a trade off between the return of the portfolio (first order information) and the variance/risk. 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 [KW97].

In our second problem the variance was measured along an arbitrary direction. We gave a natural generalization of the WMC/Hedge algorithm to the case when the parameters are density matrices. Note that in this paper we upper bounded the sum of the expected variances over trials, whereas in [War05, WK06] a Bayes rule for density matrices was given for which a lower bound was provided on the product of the expected variances over trials.[2]

Much work has been done on exponential weight updates for the experts. In particular, algorithms have been developed for shifting experts by combining the exponential updates with an additive "sharing update"[HW98]. In preliminary work we showed that these techniques easily carry over to the density matrix setting. This includes the more recent work on the "sharing to the past average" update, which introduces a long-term memory [BW02].

Appendix A

Proof of Lemma 1

Begin by analyzing the progress towards the comparison vector $\boldsymbol{u}$:

$$\begin{aligned}
\mathrm{d}(\boldsymbol{u}, \boldsymbol{w}_t) - \mathrm{d}(\boldsymbol{u}, \boldsymbol{w}_{t+1}) &= \sum u_i \log \frac{u_i}{w_{t,i}} - \sum u_i \log \frac{u_i}{w_{t+1,i}} \\
&= \sum u_i \log w_{t+1,i} - \sum u_i \log w_{t,i} \\
&= \sum u_i \log \frac{w_{t,i} e^{-\eta(\boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t)_i}}{\sum w_{t,i} e^{-\eta(\boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t)_i}} - \sum u_i \log w_{t,i} \\
&= \sum u_i \log w_{t,i} - \eta \sum u_i (\boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t)_i - \\
&\quad - \log \left(\sum w_{t,i} e^{-\eta(\boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t)_i} \right) - \sum u_i \log w_{t,i} \\
&= -\eta \sum u_i (\boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t)_i - \log \left(\sum w_{t,i} e^{-\eta(\boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t)_i} \right)
\end{aligned}$$

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

$$F = a \boldsymbol{w}_t^\top \boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t - b \boldsymbol{u}^\top \boldsymbol{\mathcal{C}} \boldsymbol{u} + \eta \boldsymbol{u}^\top \boldsymbol{\mathcal{C}} \boldsymbol{w}_t + \log \left(\sum w_{t,i} e^{-\eta(\boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t)_i} \right)$$

[2] This amounts to an upper bound on the sum of the negative logarithms of the expected variances.

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

$$e^{-\eta(\boldsymbol{\mathcal{C}}_t\boldsymbol{w}_t)_i} = e^{-\eta l}(e^{-\eta r})^{\frac{(\boldsymbol{\mathcal{C}}_t\boldsymbol{w}_t)_i - l}{r}} \leq e^{-\eta b}\left(1 - \frac{(\boldsymbol{\mathcal{C}}_t\boldsymbol{w}_t)_i - l}{r}(1 - e^{-\eta r})\right)$$

Using this inequality we obtain:

$$\log\left(\sum w_{t,i} e^{-\eta(\boldsymbol{\mathcal{C}}_t\boldsymbol{w}_t)_i}\right) \leq -\eta l + \log\left(1 - \frac{\boldsymbol{w}_t^\top \boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t - l}{r}(1 - e^{-\eta r})\right)$$

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

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

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

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

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

$$\frac{\partial G}{\partial \boldsymbol{z}} = -2b\boldsymbol{z} + \eta\sqrt{\boldsymbol{\mathcal{C}}_t}\boldsymbol{w}_t = 0, \quad \boldsymbol{z} = \frac{\eta}{2b}\sqrt{\boldsymbol{\mathcal{C}}_t}\boldsymbol{w}_t$$

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

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

Since $l \leq (\boldsymbol{\mathcal{C}}_t\boldsymbol{w}_t)_i \leq l + r$, then obviously so is $\boldsymbol{w}_t^\top \boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t$, since 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 $q \in \mathbb{R}$:

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

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

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

By our assumptions $\boldsymbol{w}_t^\top \boldsymbol{\mathcal{C}}_t \boldsymbol{w}_t \leq \frac{r}{2}$, 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:

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

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

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

This concludes the proof. □

References

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[BW02] O. Bousquet and M. K. Warmuth. Tracking a small set of experts by mixing past posteriors. *J. of Machine Learning Research*, 3(Nov):363–396, 2002.

[CBMS05] Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. In *Proceedings of the 18th Annual Conference on Learning Theory (COLT 05)*, pages 217–232. Springer, June 2005.

[Cov91] T. M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991.

[CSTK01] Nello Cristianini, John Shawe-Taylor, and Jaz Kandola. Spectral kernel methods for clustering. In *Advances in Neural Information Processing Systems 14*, pages 649–655. MIT Press, December 2001.

[FS97] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[HSSW98] D. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

[HW98] M. Herbster and M. K. Warmuth. Tracking the best expert. *Journal of Machine Learning*, 32(2):151–178, August 1998.

[KW97] J. Kivinen and M. K. Warmuth. Additive versus exponentiated gradient updates for linear prediction. *Information and Computation*, 132(1):1–64, January 1997.

[LW94] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[NC00] M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

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

[War05] Manred K. Warmuth. Bayes rule for density matrices. In *Advances in Neural Information Processing Systems 18 (NIPS 05)*. MIT Press, December 2005.

[WK06] Manfred K. Warmuth and Dima Kuzmin. A Bayesian probability calculus for density matrices. Unpublished manuscript, March 2006.