Journal of Machine Learning Research 26 (2025) 1-35 Submitted 9/23; Revised 12/24; Published 4/25

Optimization Over a Probability Simplex

James Chok[1,2] JAMES.CHOK@ED.AC.UK
Geoffrey M. Vasil[1] GVASIL@ED.AC.UK
[1] *School of Mathematics and Maxwell Institute for Mathematical Sciences, The University of Edinburgh, Edinburgh EH9 3FD, United Kingdom*
[2] *Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Wilberforce Road, Cambridge CB3 0WA, United Kingdom*

Editor: Silvia Villa

Abstract

We propose a new iteration scheme, the Cauchy-Simplex, to optimize convex problems over the probability simplex $\{w \in \mathbb{R}^n \mid \sum_i w_i = 1 \text{ and } w_i \geq 0\}$. Specifically, we map the simplex to the positive quadrant of a unit sphere, envisage gradient descent in latent variables, and map the result back in a way that only depends on the simplex variable. Moreover, proving rigorous convergence results in this formulation leads inherently to tools from information theory (e.g., cross-entropy and KL divergence). Each iteration of the Cauchy-Simplex consists of simple operations, making it well-suited for high-dimensional problems. In continuous time, we prove that $f(x_T) - f(x^*) = \mathcal{O}(1/T)$ for differentiable real-valued convex functions, where T is the number of time steps and w^* is the optimal solution. Numerical experiments of projection onto convex hulls show faster convergence than similar algorithms. Finally, we apply our algorithm to online learning problems and prove the convergence of the average regret for (1) Prediction with expert advice and (2) Universal Portfolios.

Keywords: Constrained Optimization, Convex Hull, Simplex, Orthogonal Matrix, Gradient Flow

1. Introduction

Optimization over the probability simplex, (*i.e.*, unit simplex) occurs in many subject areas, including portfolio management (Helmbold et al., 1998; Kalai and Vempala, 2003; Das and Banerjee, 2011; Canyakmaz et al., 2023), machine learning (Collins et al., 2008; Rakotomamonjy et al., 2008; Duan, 2020; Floto et al., 2023), population dynamics (Schuster and Sigmund, 1983; Bomze, 2002), and multiple others including statistics and chemistry (Kuznetsova and Strekalovsky, 2001; de Klerk et al., 2008; Amsler et al., 2018; Candelieri et al., 2025). In this paper, we look at minimizing differentiable real-valued convex functions $f(w)$ with $w \in \mathbb{R}^n$ within the probability simplex,

$$\min_{w \in \Delta^n} f(w), \quad \text{where } \Delta^n \;=\; \{w \in \mathbb{R}^n \mid \sum_i w_i = 1 \text{ and } w_i \geq 0\}. \tag{1}$$

While quadratic programs can produce an exact solution under certain restrictions on f, they tend to be computationally expensive when n is large (Nocedal and Wright, 2006). Here, we provide an overview of iterative algorithms to solve this problem approximately

and introduce a new algorithm for differentiable real-valued convex functions f. We also show how our method can be applied to other constraints, particularly orthogonal matrices.

Previous works (e.g., projected and exponentiated gradient descent) are equivalent to the discretizations of an underlying continuous-time gradient flow method. However, in each case, the gradient flow only enforces either the positivity or the unit-sum condition. Alternatively, the Frank-Wolfe algorithm is not derived from a gradient flow method and is inherently a discrete-time method.

We propose a continuous-time gradient flow that is able to satisfy both constraints. Being a continuous-time gradient flow allows us to prove convergence in a continuous-time setting, and we also prove convergence of its forward Euler discretization scheme. Moreover, we see the ideas of projected and exponentiated gradient descent in our algorithm.

The remainder of this paper is structured as follows. In Section 2, we provided an overview of methods used to solve (1). In Section 3, we present our proposed algorithm, show how it can be derived using simple calculus, and reveal connections to previous works. Theoretical analysis of the algorithm is shown in Section 4, proving a convergence rate of $\mathcal{O}(1/T)$. Section 5 shows how to extend the method presented in Section 3 to constraints over orthogonal matrices. Finally, Section 6 provides theoretical applications in game theory and finance, as well as numerical applications in geometry and finding how to weight survey questions to follow a desired distribution.

2. Previous Works

Two common classes of methods currently exist to find the minimum value of a smooth, convex function within a non-empty, convex, and compact $R \subset \mathbb{R}^n$: (i) Projection-Based Methods, adjust guesses to stay in the set; and (ii) Frank-Wolfe Methods which reduce the function's value while explicitly staying in the set.

Projection-based methods use a projection to satisfy the constraints of R.

This paper uses two definitions of a projection. Classically in the optimization literature (Galántai, 2004), projection of a point $y \in \mathbb{R}^n$ refers to the optimization procedure

$$\text{proj}_R(y) \;=\; \min_{x \in R} \|x - y\|. \tag{2}$$

Since R is convex and compact, a unique solution exists.

Any linear idempotent map generalizes the classical notion of projection (Halmos, 1998), *i.e.*, $g : D \mapsto D$ for D a non-empty set with $g(g(x)) = g(x), \forall x \in D$. Defining the set $g(D) = \{g(x) | x \in D\} \subseteq D$, g is said to project D onto $g(D)$. This generalization of projection still encompasses the optimization procedure as if we take $g(y) = \text{proj}_R(y)$, then $g(g(y)) = \text{proj}_R(g(y)) = g(y)$.

Projected gradient descent (PGD) is a simple method to solve differentiable real-valued convex problems over R. The iteration scheme follows

$$w^{t+1} \;=\; \text{proj}_R(w^t - \alpha_t \nabla_w f(w^t)), \qquad \text{where} \qquad \text{proj}_R(y) \;=\; \min_{x \in R} \|x - y\|, \tag{3}$$

and learning rate $\alpha_t > 0$. In general, $w^t - \alpha_t \nabla_w f(w^t)$ neither satisfies the positivity or unit-sum constraint. Projection of this vector into the unit simplex can be performed in $O(mn)$ operations (Michelot, 1986; Chen and Ye, 2011; Wang and Carreira-Perpinán, 2013;

Kyrillidis et al., 2013), where n is the dimension of w and $1 \leq m \leq n$ is the number of iterations of the algorithm. While this added cost is often negligible, it can become significant for large n. More recent applications increasingly require large dimensions, e.g. with $n \geq 1000$ (Markowitz et al., 1994).

This formulation of PGD can be simplified with linear constraints $Aw = b$, where A is an $k \times n$ matrix and $b \in \mathbb{R}^k$. Luenberger (1997) projects $\nabla_w f(w^t)$ into the nullspace of A and descends the result along the projected direction. For the unit-sum constraint, this algorithm requires solving the constrained optimization problem

$$\arg\min_x \frac{1}{2}\|\nabla_w f(w^t) - x\|^2, \quad \text{with} \quad \sum_i x_i \;=\; 0.$$

This problem yields to the method of Lagrange multipliers, giving the solution

$$w_i^{t+1} \;=\; w_i^t - \alpha_t \left(\nabla_{w_i} f(w^t) - \frac{1}{n} \sum_i \nabla_{w_i} f(w^t) \right).$$

While this scheme satisfies the unit-sum constraint similarly to (3), it does not satisfy the positivity constraint. This requires the same projection used in PGD, thus costing $\mathcal{O}(mn)$ operations (Nocedal and Wright, 2006; Yousefzadeh, 2021).

Exponentiated Gradient Descent (EGD) first presented by Nemirovsky and Yudin (1983), and later by Kivinen and Warmuth (1997), is a specific case of mirror descent. They consider gradient flow in the mirror space $(\Delta^n)^*$, and one maps between Δ^n and $(\Delta^n)^*$ via the functions $\nabla h : \Delta^n \mapsto (\Delta^n)^*$ and $(\nabla h)^{-1} : (\Delta^n)^* \mapsto \Delta^n$, where $h(w) = \sum_i (w_i \log(w_i) - w_i)$ (Vishnoi, 2021). This yields the gradient flow in the mirror space

$$\frac{d\theta}{dt} \;=\; -\alpha \nabla_x f\big((\nabla h)^{-1}(\theta)\big),$$

for continuous learning rate $\alpha > 0$. Mapping back into Δ^n yields the gradient flow

$$\frac{d}{dt}\log(w_i) \;=\; -\alpha \nabla_{w_i} f(w), \tag{4}$$

While this preserves positivity, the unit-sum constraint is not preserved. This can be seen as the differential equation can be rewritten as $dw_i/dt = -\alpha w_i \nabla_{w_i} f(w)$, and $\sum_i dw_i/dt$ is not zero in general.

A forward Euler discretization of (4) yields $w_i^{t+1} = w_i^t \exp(-\alpha_t \nabla_{w_i} f(w^t))$, thus mapping $\mathbb{R}^n_{\geq 0} \mapsto \mathbb{R}^n_{\geq 0}$ where $\mathbb{R}^n_{\geq 0} = \{x \in \mathbb{R}^n - \{0\} | x_i \geq 0\}$. To enforce the unit-sum constraint, one can use the projection defined in (2) by taking $R = \Delta^n$; however, as suggested by Kivinen and Warmuth (1997), a simple idempotent map to project $\mathbb{R}^n_{\geq 0}$ onto Δ^n can be used, given by

$$w_i^{t+1} \;=\; \frac{w_i^t \exp(-\alpha_t \nabla_{w_i} f(w^t))}{\sum_j w_j^t \exp(-\alpha_t \nabla_{w_j} f(w^t))},$$

with discrete learning-rate $\alpha_t > 0$. Thus the projection, given by the normalization factor, takes $\mathcal{O}(n)$ operations. Moreover, discretization is necessary for the algorithm to satisfy the constraint.

Frank-Wolfe-based methods exploit the convexity of the domain R, eliminating the need of a projection.

The Frank-Wolfe method is a classic scheme (Frank and Wolfe, 1956) that is experiencing a recent surge popularity (Lacoste-Julien et al., 2013; Bellet et al., 2015; Mu et al., 2016; Tajima et al., 2021). The method skips projection by assuming R is convex. That is,

$$\begin{aligned} w^{t+1} &= (1-\gamma_t)w^t + \gamma_t s^t, \\ \text{where} \quad s^t &= \arg\min_{s\in R} \; s \cdot \nabla_w f(w^t) \quad \text{and} \quad 0 \;\leq\; \gamma_t \;\leq\; 1. \end{aligned} \tag{5}$$

Since R is convex, $w^{t+1} \in R$ automatically. For the simplex, the subproblem (5) is known in closed form. Frank-Wolfe-based methods tend to be fast for sparse solutions but display oscillatory behavior near the solution, resulting in slow convergence (Lacoste-Julien and Jaggi, 2015; Bomze and Zeffiro, 2021).

The Pairwise Frank-Wolfe (PFW) method improves upon the original by introducing an 'away-step' to prevent oscillations allowing faster convergence (Guélat and Marcotte, 1986; Jaggi, 2013; Lacoste-Julien and Jaggi, 2015).

While PFW, EGD, and PGD have guaranteed convergence under constant and decreasing step sizes (Vishnoi, 2021; Jaggi, 2013), a line search is often used in practice to improve run-time (Nocedal and Wright, 2006). Taking f in (1) to be quadratic (Bomze, 1998, 2002; Selvi et al., 2023), a line search has analytical solutions for the Frank-Wolfe and PFW methods but not for EGD and PGD. EGD and PGD require approximate methods (e.g., backtracking method (Nocedal and Wright, 2006)), adding extra run time per iteration.

3. The Main Algorithm

For convex problems over a probability simplex, we propose what we named the Cauchy-Simplex (CS). For an initial vector $w^0 \in \text{relint}(\Delta^n) = \{w \in \mathbb{R}^n | \sum_i w_i = 1, w_i > 0\}$, we propose the following iteration scheme

$$\begin{aligned} w_i^{t+1} &= w_i^t - \eta^t d_i^t, \\ \text{with } d_i^t &= w_i^t(\nabla_{w_i} f(w^t) - w^t \cdot \nabla_w f(w^t)), \qquad 0 < \eta^t < \eta^{t,\max}, \\ \text{and } \eta^{t,\max} &= \frac{1}{\max_i(\nabla_{w_i} f(w^t) - w^t \cdot \nabla_w f(w^t))}. \end{aligned} \tag{6}$$

Remark 1 *Rewriting*

$$\max_i \left(\nabla_{w_i} f(w^t) - w^t \cdot \nabla_w f(w^t)\right) = (e_j - w^t) \cdot \nabla f(w^t), \tag{7}$$

where $e_j \in \mathbb{R}^n$ is the standard unit vector with $j = \arg\max_i \nabla_{w_i} f(w^t)$. For finite iterations $T > 0$, $w^T \in$ relint(Δ^n) and thus $e_j - w^T \neq 0$. Therefore (7) is zero when $\nabla f(w^T) = 0$, i.e., the optimal solution has been reached.

When $\max_i(\nabla_{w_i} f(w^t) - w^t \cdot \nabla_w f(w^t)) = 0$, then $\eta^{t,\max} = \infty$ and η^t is set to any finite positive constant. As seen in Remark 1, this condition being met implies $\nabla f(w^t) = 0$, and $w^{t+1} = w^t$ for finite η^t.

The upper bound on the learning rate, η_t, ensures that w_i^{t+1} is positive for all i. Summing over the indices of d^t

$$\sum_i w_i^t \left(\nabla_{w_i} f(w^t) \; - \; w^t \cdot \nabla_w f(w^t) \right) \; = \; (w^t \cdot \nabla_w f(w^t)) \Big(1 \; - \; \sum_i w_i^t \Big).$$

Defining the linear operator $\mathcal{L}(v) = \sum_i v_i$, if $\mathcal{L}(w^t) = 1$, then $\mathcal{L}(d^t) = 0$. This makes w^{t+1} also satisfy the unit-sum constraint as $\mathcal{L}(w^{t+1}) = \mathcal{L}(w^t) = 1$.

However, it is important to note that once an index is set to zero, it will stay zero for future iterations. Thus, having the initial condition in the relative interior of Δ^n ensures flow for each of the indices. Another important implication of this is that taking $\eta^t = \eta^{t,\max}$ may cause an index to be incorrectly set to zero. As such, step sizes should be taken such that $\eta^t < \eta^{t,\max}$ to prevent this.

3.1 Motivating Derivation

Our derivation begins by modeling $w \in \Delta^n$ through a latent variable, $\psi \in \mathbb{R}^n$,

$$w_i \; = \; w_i(\psi) \; = \; \frac{\psi_i^2}{\sum_j \psi_j^2},$$

which automatically satisfies positivity and unit probability. Thus, the optimization problem over the probability simplex can be considered as an unconstrained optimization problem

$$\min_{\psi \in \mathbb{R}^n} F(\psi), \qquad \text{where} \qquad F(\psi) \; = \; f(w(\psi)).$$

Now consider the continuous-time gradient descent on ψ,

$$\frac{d\psi_j}{dt} \; = \; -\alpha \frac{\partial f}{\partial \psi_j}, \tag{8}$$

for continuous learning rate $\alpha > 0$. This then induces a gradient flow in w. To find this, first note that

$$\frac{\partial w_i}{\partial \psi_j} \; = \; \frac{2\psi_i \|\psi\|^2 \delta_{ij} \; - \; \psi_i^2 \psi_j}{\|\psi\|^4} \; = \; \frac{2}{\|\psi\|^2} \left(\psi_i \delta_{ij} \; - \; 2 w_i \psi_j \right),$$

where $\delta_{ij} = 1$ if $i = j$ and zero otherwise. Using the notation $\dot{\psi} = d\psi/dt$, the chain rule gives the gradient flow

$$\frac{dw_i}{dt} \; = \; \sum_j \frac{\partial w_i}{\partial \psi_j} \frac{d\psi_j}{dt} \; = \; \frac{2}{\|\psi\|^2} \Big(\psi_i \dot{\psi}_i \; - \; w_i \, \psi \cdot \dot{\psi} \Big).$$

By equation (8),

$$\psi \cdot \dot{\psi} \; = \; -\alpha \sum_{i,j} \psi_j \frac{\partial w_i}{\partial \psi_j} \frac{\partial f}{\partial w_i} \; = \; -2\alpha (w \cdot \nabla_w f) \left(1 - \sum_j w_j \right) \; = \; 0.$$

Thus, the gradient flow in w can be simplified to

$$\frac{dw_i}{dt} \;=\; \frac{2}{\|\psi\|^2}\,\psi_i\dot{\psi}_i \;=\; -\beta w_i(\nabla_{w_i} f(w) - w\cdot\nabla_w f(w)) \quad \text{where} \quad \beta \;=\; \frac{4\alpha}{\|\psi\|^2}. \tag{9}$$

Thus giving the iterative scheme

$$w_i^{t+1} \;=\; w_i^t - \eta^t d_i^t \quad \text{where} \quad d_i^t \;=\; w_i^t\Big(\nabla_{w_i} f(w^t) - w^t\cdot\nabla_w f(w^t)\Big). \tag{10}$$

While w is initially modeled through a latent variable, the resulting gradient flow is only in terms of the constrained variable w. This allows the iteration scheme to only reference vectors in Δ^n, and we no longer need to consider the latent domain $\mathbb{R}^n$. This derivation is to make a connection to recent work using a latent-variable approach (Lezcano-Casado and Martínez-Rubio, 2019; Bucci et al., 2022; Li et al., 2023).

3.2 On the Learning Rate

The proof of convergence in Section 4 assumes that $\eta^t < \eta^{t,\max}$; thus, all weights stay strictly positive but may be arbitrarily close to zero. However, accumulated rounding errors may result in some weights becoming zero or negative. As such, in our numerical implementation, a weight is set to zero once it is lower than some threshold (we choose 1e-10).

Once an index is set to zero, it will remain zero and can be ignored. This gives an altered maximum learning rate

$$\tilde{\eta}^{t,\max} \;=\; \frac{1}{\max_{i\in S}(\nabla_{w_i} f(w^t) - w^t\cdot\nabla_w f(w^t))},$$

where $S = \{i \mid w_i > 0\}$. It follows that $\eta^{t,\max} \leq \tilde{\eta}_{t,\max}$, allowing for larger step-sizes to be taken.

The requirement of a maximum learning rate is not unique to the CS and is shared with PGD and EGD using fixed learning rates (Lu et al., 2018). However, this maximum learning rate is based on the Lipschitz constant of ∇f, which is typically unknown. While numerical methods to approximate the Lipschitz constant exist, they can cause a failure of convergence (Hansen et al., 1992).

More generally, PGD and EGD converge with a line search bounded by an arbitrarily large positive constant (Xiu et al., 2007; Li and Cevher, 2018), and becomes an extra parameter in the method. In contrast, each iteration of the CS has an easily computable maximum possible learning rate, $\eta^{t,\max}$, to be used in a line search.

3.3 Connections to Previous Methods

There are two ways of writing the gradient flow for the Cauchy-Simplex. In terms of the flow in w:

$$\frac{dw}{dt} \;=\; -W\Pi_w\nabla_w f(w), \quad \text{where} \quad \Pi_w \;=\; \mathrm{I} - \mathbb{1}\otimes w, \tag{11}$$

and W is a diagonal matrix filled with w, I is the identity matrix, and $\mathbb{1}$ is a vector full of ones.

In terms of the flow in $\log(w)$:

$$\frac{d}{dt}\log(w) \;=\; -\Pi_w \nabla_w f(w), \tag{12}$$

giving an alternative exponential iteration scheme

$$w^{t+1} \;=\; w^t \exp(-\eta^t \Pi_w \nabla_w f(w^t)). \tag{13}$$

Claim 2 *Π_w projects $\mathbb{R}^n$ on the null space of the operator $\mathcal{L}_w(v) = \sum_i w_i v_i$, provided that $\sum_i w_i = 1$.*

Proof To see that Π_w is an idempotent map,

$$\begin{aligned}\Pi_w^2 \;&=\; \mathrm{I}^2 - 2(\mathbb{1}\otimes w) + (\mathbb{1}\otimes w)(\mathbb{1}\otimes w)\\ &=\; \mathrm{I} - 2(\mathbb{1}\otimes w) + \left(\sum_i w_i\right)(\mathbb{1}\otimes w) \;=\; \Pi_w,\end{aligned}$$

and is, therefore, a projection.

To see that $\Pi_w(\mathbb{R}^n)$ is the null space of $\mathcal{L}_w$, it is easy to see that for any $u \in \mathbb{R}^n$, $\mathcal{L}_w(\Pi_w u) = 0$. For the converse, let v be in the null space of $\mathcal{L}_w$. By definition of the null space, $\mathcal{L}_w(v) = \sum_i w_i v_i = 0$, hence

$$\Pi_w(v) \;=\; (\mathrm{I} - 1\otimes w)v \;=\; v.$$

Thus Π_w is surjective, mapping $\mathbb{R}^n \mapsto \mathrm{Nul}(\mathcal{L}_w)$. ■

Remark 3 *While Π_w is a projection, it takes $\mathcal{O}(n)$ operations.*

The formulations (11) and (12) draw a direct parallel to both PGD and EGD, as summarized in Table 1.

PGD can be written in continuous form as

$$\frac{dw}{dt} \;=\; -\Pi_{1/n}\nabla_w f(w) \quad \text{where} \quad \Pi_{1/n} \;=\; \mathrm{I} - \frac{1}{n}\mathbb{1}\otimes\mathbb{1}.$$

The projector helps PGD satisfy the unit-sum constraint, but not perfectly for general w. However, the multiplication with the matrix W slows the gradient flow for a given index w_i as it nears the boundary, with zero flow once it hits the boundary. Thus preserving positivity.

EGD, similarly, can be written in continuous form as

$$\frac{d}{dt}\log(w) \;=\; -\nabla_w f(w).$$

Performing the descent through $\log(w)$ helps EGD preserve positivity. Introducing the projector helps the resulting exponential iteration scheme (13) to agree with its linear iteration scheme (10) up to $\mathcal{O}((\eta^t)^2)$ terms. Thus helping preserve the unit-sum constraint.

Table 1: Comparison of gradient flow for different optimization methods and constraints.

	$\sum_i w_i \neq 1$	$\sum_i w_i = 1$
$w_i \not\geq 0$	GD: $\frac{dw}{dt} = -\nabla_w f(w)$	(Luenberger) PGD: $\frac{dw}{dt} = -\Pi_{1/n} \nabla_w f(w)$
$w_i \geq 0$	EGD: $\frac{d}{dt} \log(w) = -\nabla_w f(w)$	CS: $\frac{d}{dt} \log(w) = -\Pi_w \nabla_w f(w)$

Claim 4 *The Cauchy-Simplex exponentiated iteration scheme (13) agrees up to $\mathcal{O}((\eta^t)^2)$ with it's linear iteration scheme (6). This can be seen by Taylor expanding (13) w.r.t. η^t around zero.*

Remark 5 *Combining PGD and EGD as*

$$\frac{d}{dt} \log(w) \;=\; -\Pi_{1/n} \nabla_w f(w)$$

preserves positivity but not the unit-sum constraint. This can be seen as the differential equation can be rewritten as $dw_i/dt = -w_i \Pi_{1/n} \nabla_{w_i} f(w)$. In general, $\sum_i dw_i/dt$ is non-zero and thus not an invariant property of the gradient flow.

Unlike both PGD and EGD, the continuous-time dynamics of the CS are enough to enforce the probability-simplex constraint. This allows us to use the gradient flow of CS, *i.e.* (9), to prove convergence when optimizing convex functions (seen in Section 4). This contrasts with PGD and EGD, in which the continuous dynamics only satisfy one constraint. The discretization of these schemes is necessary to allow an additional projection step, thus satisfying both constraints.

3.4 The Algorithm

The pseudo-code of our method can be seen in Algorithm 1.

4. Convergence Proof

We prove the convergence of the Cauchy-Simplex via its gradient flow. We also state the theorems for convergence of the discrete linear scheme but leave the proof in the appendix.

Theorem 6 *Let f be convex with Lipschitz continuous gradient, real-valued and continuously differentiable w.r.t. w^t and w^t continuously differentiable w.r.t. t. For the Cauchy-Simplex gradient flow (9) with initial condition $w^0 \in relint(\Delta^n)$, $f(w^t)$ is a strictly decreasing function and stationary at the optimal solution for all $t \in [0, T]$ and finite $T > 0$.*

Proof Notice that dw^t/dt can be rewritten as

$$\frac{d}{dt} \log(w_i^t) \;=\; -\Pi_{w^t} \nabla_w f(w^t).$$

Algorithm 1 Our proposed algorithm

Require: $\varepsilon \leftarrow 10^{-10}$ (Tolerance for the zero set)
  $w^0 \leftarrow (1/n, \cdots, 1/n)$
  while termination conditions not met **do**
    $S \leftarrow \{i = 1, \cdots, n \mid w_i > \varepsilon\}$
    $Q \leftarrow \{i = 1, \cdots, n \mid w_i \leq \varepsilon\}$

    Choose $\eta^t > 0$
    $\eta^{\max} \leftarrow \dfrac{1}{\max_{i \in S}(\nabla_{w_i} f(w^t)) - w^t \cdot \nabla_w f(w^t)}$
    $\eta^t \leftarrow \min(\eta^t, \eta^{\max})$

    $\hat{w}_i^{t+1} \leftarrow w_i^t - \eta^t w_i^t (\nabla_{w_i} f(w^t) - w^t \cdot \nabla_w f(w^t))$

    $\hat{w}_j^{t+1} \leftarrow 0, \quad \forall j \in Q$
    $w_i^{t+1} \leftarrow \hat{w}_i^{t+1} / \sum_j \hat{w}_j^{t+1}$ (Normalizing for numerical stability)
  end while

Since ∇f is Lipschitz, and Δ^n is a compact subset of $\mathbb{R}^n$, $\|\nabla f(w^t)\|$ is bounded for all $w^t \in \Delta^n$. Thus, if $w^0 \in \text{relint}(\Delta^n)$, strict positivity of w_i^t is preserved for $t \in [0, T]$. Furthermore, since $\sum_i dw_i/dt = \sum_i (W \Pi_w \nabla_w f(w))_i = 0$, $\sum_i w_i$ is an invariant quantity of the gradient flow. Therefore, if $w^0 \in \text{relint}(\Delta^n)$ then $w^t \in \text{relint}(\Delta^n)$ for $t \in [0, T]$, and finite $T > 0$.

By direction computation

$$\begin{aligned}
\frac{df}{dt} &= \frac{\partial f}{\partial w^t} \cdot \frac{dw^t}{dt} = -\sum_i \nabla_{w_i} f(w^t) \Big(w_i^t (\nabla_{w_i} f(w^t) - w^t \cdot \nabla_w f(w^t)) \Big) \\
&= -\Big(\sum_i w_i^t \left(\nabla_{w_i} f(w^t) \right)^2 - \left(w^t \cdot \nabla_w f(w^t) \right)^2 \Big) \\
&:= -\text{Var}[\nabla_w f(w^t) \,|\, w^t],
\end{aligned}$$

where $\text{Var}[v \,|\, w]$ is the variance of a vector $v \in \mathbb{R}^n$ with respect to the discrete measure $w \in \Delta^n$. The variance can be rewritten as

$$\text{Var}[v \,|\, w] = \sum_i w_i (v_i - v \cdot w)^2 = \sum_i w_i (\Pi_w v)_i^2,$$

for all $v \in \mathbb{R}^n$, and thus is non-negative. Since $w^t \in \text{relint}(\Delta^n)$, it follows that $\frac{df}{dt} \leq 0$ and that f is decreasing in time.

As $w^t \in \text{relint}(\Delta^n)$, $df/dt = 0$ only when $\Pi_{w_t} \nabla_w f(w^t) = 0$. However, this occurs only if w^t is an optimal solution to (1), which can be verified by checking the KKT conditions with the constraints of the simplex, *i.e.*, $\sum_i w_i = 1$, and $w_i \geq 0$, shown in Appendix C.

Thus f is strictly decreasing in time, and stationary only at the optimal solution. ■

Theorem 7 *Let f be real-valued, convex and continuously differentiable w.r.t. w^t, w^t continuously differentiable w.r.t. t, and $w^* \in \Delta^n$ be a solution to (1). Under the Cauchy-Simplex gradient flow (9), for all $t \in [0, T]$ and finite $T > 0$, the relative entropy*

$$D(w^*|w^t) \;=\; \sum_{w_i^* \neq 0} w_i^* \log\left(\frac{w_i^*}{w_i^t}\right)$$

is a decreasing function in time for $w^t \in$ relint(Δ^n).

Proof We rewrite the relative entropy into

$$D(w^*|w^t) \;=\; \sum_{w_i^* \neq 0} w_i^* \log(w_i^*) - \sum_i w_i^* \log(w_i^t).$$

By direction computation

$$\frac{d}{dt} D(w^*|w^t) \;=\; -\sum_i \frac{w_i^*}{w_i^t}\frac{dw_i^t}{dt} \;=\; \sum_i w_i^* (\Pi_{w^t} \nabla_{w_i} f(w^t)) \;=\; \nabla_w f(w^t) \cdot (w^* - w^t).$$

Since f is convex, and w^* is a minimum of f in the simplex,

$$\frac{d}{dt} D(w^*|w^t) \;\leq\; f(w^*) - f(w^t) \leq 0. \tag{14}$$

■

Theorem 8 *Let f be real-valued, convex and continuously differentiable w.r.t. w^t, w^t continuously differentiable w.r.t. t, and $w^* \in \Delta^n$ a solution to (1). For $w^0 = (1/n, \ldots, 1/n)$ and finite $T > 0$, the Cauchy-Simplex gradient flow (9) gives the bound*

$$f(w^T) - f(w^*) \;\leq\; \frac{\log(n)}{T}.$$

Proof Taking $u = w^*$ and integrating (14) w.r.t. t gives

$$\int_0^T \Big(f(w^t) - f(w^*)\Big)\, dt \;\leq\; -\int_0^T \frac{d}{dt} D(w^*|w^t)\, dt \;=\; D(w^*|w^0) - D(w^*|w^T).$$

By Jensen's inequality, the relative entropy can be bounded from below by

$$D(u|v) \;=\; -\sum_{u_i \neq 0} u_i \log(v_i/u_i) \;\geq\; -\log\left(\sum_{u_i \neq 0} v_i\right) \;\geq\; 0,$$

for $u \in \Delta^n$ and $v \in$ relint(Δ^n). This can also be shown using the Bergman divergence (Chou et al., 2023).

Thus, relative entropy is non-negative (Jaynes, 2003; Gibbs, 2010), yielding

$$\int_0^T \Big(f(w^t) - f(w^*)\Big)\, dt \;\leq\; D(w^*|w^0).$$

Completing the integral on the left side of the inequality and dividing by T gives

$$\frac{1}{T}\int_0^T f(w^t)\,dt - f(w^*) \;\leq\; \frac{1}{T}D(w^*|w^0).$$

Using Theorem 6, $f(w^t)$ is a decreasing function. Thus

$$f(w^T) \;=\; \frac{1}{T}\int_0^T f(w^T)\,dt \;\leq\; \frac{1}{T}\int_0^T f(w^t)\,dt.$$

Let $w^0 = (1/n, \ldots, 1/n)$, then the relative entropy can be bounded by [1]

$$D(u|w^0) \;=\; \sum_{u_i \neq 0} u_i \log(u_i) \;+\; \log(n) \;\leq\; \log(n), \qquad \text{for all} \qquad u \in \Delta^n.$$

This gives the required bound

$$f(w^T) - f(w^*) \;\leq\; \frac{D(w^*|w^0)}{T} \leq \frac{\log(n)}{T}.$$

■

Theorem 9 (Convergence of Linear Scheme) *Let f be a differentiable convex function that obtains a minimum at w^* with ∇f L-Lipschitz continuous. Let $w^0 = (1/n, \ldots, 1/n)$ and $\{\eta^t\}_{t=0}^{T-1}$ be a sequence that satisfies $0 < \eta^t < \min\{1/L, \eta^{t,\max}\}$ and* [2]

$$C_{\gamma_t} \;\leq\; \frac{w^t \cdot (\Pi^t \nabla_w f(w_t))^2}{2\max_i (\Pi^t \nabla_w f(w_t))_i^2}, \tag{15}$$

with $\gamma_t = \eta^t/\eta^{t,\max}$, $C_{\gamma_t} = \gamma_t^{-2}\log(e^{-\gamma_t}/(1-\gamma_t))$ and $\eta^{t,\max}$ defined in (6). Then the linear Cauchy-Simplex scheme (6) produces iterates $\{w^t\}_{t=0}^{T-1}$ such that, for finite integer $T > 0$,

$$f(w^T) - f(w^*) \;\leq\; \frac{\log(n)}{\sum_{t=0}^{T-1}\eta^t}.$$

The proof can be found in Appendix B.4.

In practice, finding η_t, and by extension γ_t, to satisfy the assumptions of Theorem 9 can be hard. Instead, we show asymptotic convergence of the linear Cauchy-Simplex scheme under a line search.

Lemma 10 (Asymptotic Convergence of Linear Scheme) *Let f be a differentiable convex function with ∇f Lipschitz continuous. The linear Cauchy-Simplex (6) has asymptotic convergence when η_t is chosen through a line search, i.e.,*

$$f(w^{t+1}) \;=\; \min_{\eta^t \in [0, \eta^{t,\max} - \varepsilon]} f(w^t - \eta^t d^t),$$

for some $0 < \varepsilon \ll 1$ and $d_i^t = w_t^t(\Pi^t \nabla f^t)_i$. That is $f(w^t) \to f(w^*)$ as $t \to \infty$.

The proof can be found in Appendix B.3.

1. Intuitively, this can be seen by computing the relative entropy between a state with maximal entropy (the uniform distribution) and minimal entropy (the Dirac mass).
2. Note that C_{γ_t} is an increasing function of γ_t, with $C_{\gamma_t} \geq 0$ for $\gamma_t \in [0, 1]$. Thus $\gamma_t = \eta^t/\eta^{t,\max}$ can also be chosen to satisfy (15) and that $\{\eta^t\}_t$ is a sequence with $0 < \eta^t < \min\{1/L, \eta^{t,\max}\}$.

5. Extension: Optimization over Orthogonal Matrices

Another often explored constraint is the orthogonal matrix constraint

$$\min_{Q \in \mathcal{S}^n} f(Q), \quad \text{where} \quad \mathcal{S}^n \;=\; \{Q \in \mathbb{R}^{n \times n} \mid QQ^T \;=\; I\},$$

with $\mathcal{S}^n$ also known as the Stiefel manifold. To name a few, this problem occurs in blind source separation (Särelä and Valpola, 2005; Omlor and Giese, 2011), principle component analysis (Bertsimas and Kitane, 2023), and neural networks (Fiori, 2005; Achour et al., 2022; Woodworth et al., 2020; Chou et al., 2024).

Various iterative methods have been suggested to solve this problem, with them split up between Riemannian optimization, which uses expensive iterations that remain inside the Stiefel manifold (Wen and Yin, 2012; Jiang and Dai, 2014), and landing methods (Ablin and Peyré, 2022), which use cheap iterations that are not in the Stiefel manifold but over time will be arbitrarily close to the manifold.

Using a similar method as in Section 3, we can also derive an explicit scheme that preserves orthogonality up to an arbitrary accuracy.

Let $X \in \mathbb{R}^{n \times n}$ be an anti-symmteric matrix ($X = -X^T$). Then the Cayley transform

$$Q \;=\; (I - X)(I + X)^{-1} \;=\; 2(I - X)^{-1} - I,$$

parameterizes orthogonal matrices with $\mathrm{Det}(Q) = 1$, where I is the identity matrix. Performing similar calculations as above yields the gradient flow

$$\frac{dQ}{dt} \;=\; -\eta\, \Omega(\Lambda\Omega^T - \Omega\Lambda^T)\Omega$$

where $\Omega = Q + I$, and $\Lambda = \Omega^T\, \partial_Q f$ (full derivation can be found in Appendix A).

However, an Euler discretization of this differential equation does not produce a scheme that preserves orthogonality. Instead, we consider a corrected iteration scheme

$$Q_{t+1} \;=\; (I + C)(Q_t - \eta dQ) \quad \text{where} \quad dQ \;=\; \Omega(\Lambda\Omega^T - \Omega\Lambda^T)\Omega,$$

for some matrix C. An iteration scheme that preserves orthogonality up to arbitrary accuracy can then be made by looking at the coefficients for powers of η in the expansion of $Q_{t+1}Q_{t+1}^T$ and solving for C. In particular, a $\mathcal{O}(\eta^2)$ correct scheme can be written as

$$Q_{t+1} \;=\; (I - \frac{\eta^2}{2} dQ\, dQ^T)(Q_t - \eta dQ),$$

and a $\mathcal{O}(\eta^4)$ correct scheme is

$$Q_{t+1} \;=\; (I - \frac{\eta^2}{2} dQ\, dQ^T + \frac{3\eta^4}{8} dQ\, dQ^T\, dQ\, dQ^T)(Q_t - \eta dQ).$$

As such, parameterizing the manifold by a latent Euclidean space yields a gradient flow that remains explicitly inside the Stiefel manifold. Moreover, this gradient flow is explicitly in terms of the orthogonal matrix. This contrasts with Riemannian optimization methods, which use Riemannian gradient flow on the manifold. Moreover, since the tangent space

at each point of the Stiefel manifold does not lie in the manifold itself, once the scheme is discretized, an exponential map (or retraction) must be used to remain on the manifold. These often involve costly matrix operations like exponentials, square roots, and inversions (Jiang and Dai, 2014; Chen et al., 2021; Lezcano-Casado and Martínez-Rubio, 2019). In contrast, working in Euclidean space allows the discretized scheme to remain on the manifold using cheaper addition and multiplication matrix operations.

6. Applications

As noted in Section 3.2, approximations of the Lipschitz constant L can cause failure of convergence. As such, our experiments use a line search outlined in Lemma 10.

6.1 Projection onto the Convex Hull

Projection onto a convex hull arises in many areas like machine learning (Mizutani, 2014; Nandan et al., 2014; Grünewälder, 2018), collision detection (Wang et al., 2020) and imaging (Jung et al., 2009; Jenatton et al., 2011). It involves finding a point in the convex hull of a set of points $\{x_i\}_{1\leq i\leq n}$, with $x_i \in \mathbb{R}^d$, that is closest to an arbitrary point $y \in \mathbb{R}^d$, *i.e.*,

$$\min_w \|wX - y\|^2 \qquad \text{where} \qquad \sum_i w_i \;=\; 1 \text{ and } w_i \;\geq\; 0,$$

and $X = [x_1, \cdots, x_n]^T$ is a $\mathbb{R}^{n\times d}$ matrix. This is also known as simplex-constrained regression.

Experimental Details: We look at a convex hull sampled from the unit hypercube $[0, 1]^d$ for $d \in [10, 15, \ldots, 50]$. For each hypercube, we sample 50 points uniformly on each of its surfaces, giving a convex hull X with $n = 100d$ data points.

Once X is sampled, 50 y's are created outside the hypercube perpendicular to a surface and unit length away from it. This is done by considering the 50 points in X lying on a randomly selected surface of the hypercube. A point y_{true} is created as a random convex combination of these points. The point y can then be created perpendicular to this surface and a unit length away from y_{true}, and thus also from the convex hull of X.

Each y is then projected onto X using CS, EGD, and PFW. These algorithms are ran until a solution, $\hat{y}$, is found such that $\|\hat{y} - y_{\text{true}}\| \leq 1e^{-5}$ or 10 000 iterations have been made. We do not implement PGD and Frank-Wolfe due to their inefficiency in practice.

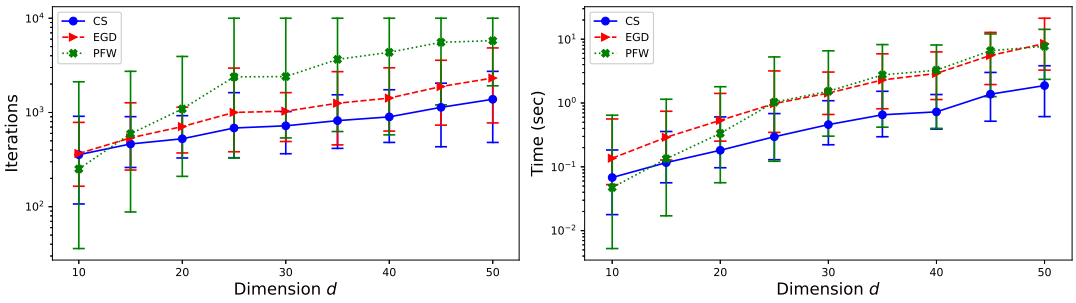
Implementation Details:

The learning rate for EGD, PFW, and CS is found through a line search. In the case of the PFW and CS algorithms, an explicit solution can be found and used. At the same time, EGD implements a back-tracking linear search with Armijo conditions (Bonnans et al., 2006) to find an approximate solution.

Experiments were written in Python and ran on Google Colab. The code can be found on GitHub[3]. The random data is seeded for reproducibility.

Results: The results can be seen in Fig. 1. For $d = 10$, we see that PFW outperforms both CS and EGD in terms of the number of iterations required and the time taken. But for $d > 10$, on average, CS converges with the least iterations and time taken.

3. https://github.com/infamoussoap/ConvexHull



(a) Average steps taken in log-scale

(b) Average time taken in log-scale

Figure 1: Number of steps and time required for PFW, EGD, and CS to project 50 randomly sampled points onto the d-hypercube. The bars indicate the minimum and maximum values.

6.2 Optimal Question Weighting

It is often desirable that the distribution of exam marks matches a target distribution, but this rarely happens. Modern standardized tests (e.g. IQ exams) solve this problem by transforming the distribution of the raw score of a given age group so it fits a normal distribution (Bartholomew, 2004; Gottfredson, 2009; Mackintosh, 2011).

While IQ exams have many criticisms (Richardson, 2002; Shuttleworth-Edwards et al., 2004; Shuttleworth-Edwards, 2016), we are interested in the raw score. As noted by Gottfredson (2009), the raw score has no intrinsic meaning as it can be boosted by adding easier questions to the test. We also argue it is hard to predict the difficulty of a question relative to an age group and, thus, even harder to give it the correct weight. Hence making the raw score a bad reflection of a person's performance.

Here we propose a framework to find an optimum weighting of questions such that the weighted scores will fit a target distribution. A demonstration can be seen in Fig. 2.

Consider d students taking an exam with n true or false questions. For simplicity, assume that person j getting question i correct can be modeled as a random variable $\mathcal{X}_{i,j}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, d$. Consider the discrete distribution of $X_j = \sum_i w_i \mathcal{X}_{i,j}$ for some $w \in \Delta^n$, the weighted mark of person j. This distribution can be approximated as a continuous distribution,

$$\rho_\varepsilon(z) \;=\; \frac{1}{d}\sum_{j=1}^{d} \mu_\varepsilon(z - X_j) \quad \text{with} \quad \mu_\varepsilon(x) \;=\; \frac{1}{\varepsilon}\mu(x/\varepsilon),$$

$\varepsilon > 0$ and μ is a continuous probability distribution, i.e. $\mu \geq 0$ and $\int \mu(x)dx = 1$. This is also known as kernel density estimation (KDE).

We want to minimize the distance between ρ_ε and some target distribution f. A natural choice is the relative entropy,

$$\min_{w\in\Delta^n} D(\rho_\varepsilon | f) \;=\; \min_{w\in\Delta^n} \int \rho_\varepsilon(x) \log\left(\frac{\rho_\varepsilon(x)}{f(x)}\right) dx,$$

of which we take its Riemann approximation,

$$\min_{w\in\Delta^n} \hat{D}(\rho_\varepsilon|f) \;=\; \min_{w\in\Delta^n} \sum_{k=1}^{M} \rho_\varepsilon(x_k)\log\left(\frac{\rho_\varepsilon(x_k)}{f(x_k)}\right)(x_k - x_{k-1}), \tag{16}$$

where $\{x_k\}_{0\le k\le M}$ is a partition of a finite interval $[a,b]$.

We remark that this problem is not convex, as μ cannot be chosen to be convex w.r.t. w and be a probability distribution.

This is similar to robust location parameter estimation (Huber, 1964; Maronna et al., 2006), which considers data sampled from a contaminated distribution (*i.e.* an uncertain mixture of two known distributions), for instance, when a predicting apparatus failures from two populations. Specifically for some contamination level $0 < \alpha \ll 1$, data was sampled at a rate of $1-\alpha$ from a distribution, G (*i.e.* easy questions), and α from an alternate distribution, H (*i.e.* hard questions). Thus, data is sampled from the distribution

$$F \;=\; (1-\alpha)G + \alpha H.$$

Under this contaminated distribution, robust parameter estimation finds point estimates (e.g., mean and variance) that are robust when α is small. In this sense, robust parameter estimation is interested in finding weights, w, that make the KDE, $\rho_\varepsilon(z)$, robust to contamination levels α. Instead, we wish to make the KDE match a target distribution.

Experiment Details: We consider 25 randomly generated exam marks, each having $d = 200$ students taking an exam with $n = 75$ true or false questions. For simplicity, we assume that $\mathcal{X}_{i,j} \sim \text{Bernoulli}(q_i s_j)$ where $0 < q_i < 1$ is the difficulty of question i and $0 < s_j < 1$ the j-th student's smartness.

For each scenario, $q_i = 7/8$ for $1 \le i \le 60$ and $q_i = 1/5$ for $60 < i \le 75$, while $s_j = 7/10$ for $1 \le j \le 120$ and $s_j = 1/2$ for $120 < j \le 200$. $\mathcal{X}_{i,j} \sim \text{Bernoulli}(q_i s_j)$ are then sampled. This setup results in a bimodal distribution, with an expected average of 0.532 and an expected standard deviation of 0.1206, as shown in Figure 2.

For the kernel density estimate, $\mu(x)$ is chosen as a unit normal distribution truncated to $[0,1]$, with smoothing parameter $\varepsilon = 0.05$. Similarly, f is a normal distribution with mean 0.5 and variance 0.1, truncated to $[0,1]$. We take the partition $\{k/400\}_{0\le k\le 400}$ for the Riemann approximation (16).

The algorithms CS, EGD, and PFW are ran for 150 iterations.

Implementation Details: The learning rate for EGD, PFW, and CS is found through a line search. However, explicit solutions are not used. Instead, a back-tracking line search with Armijo conditions is used to find an approximate solution.

Experiments were written in Python and ran on Google Colab and can be found on GitHub[4]. The random data is seeded for reproducibility.

Results: A table with the results can be seen in Table 2. In summary, of the 25 scenarios, PFW always produces solutions with the smallest relative entropy, with CS producing the largest relative entropy 13 times and EGD 12 times. For the time taken to make the 150 steps, PFW is the quickest 15 times, EGD 7 times, and CS 3 times. At the same time, EGD is the slowest 13 times, CS 7 times, and PFW 5 times.

4. https://github.com/infamoussoap/ConvexHull

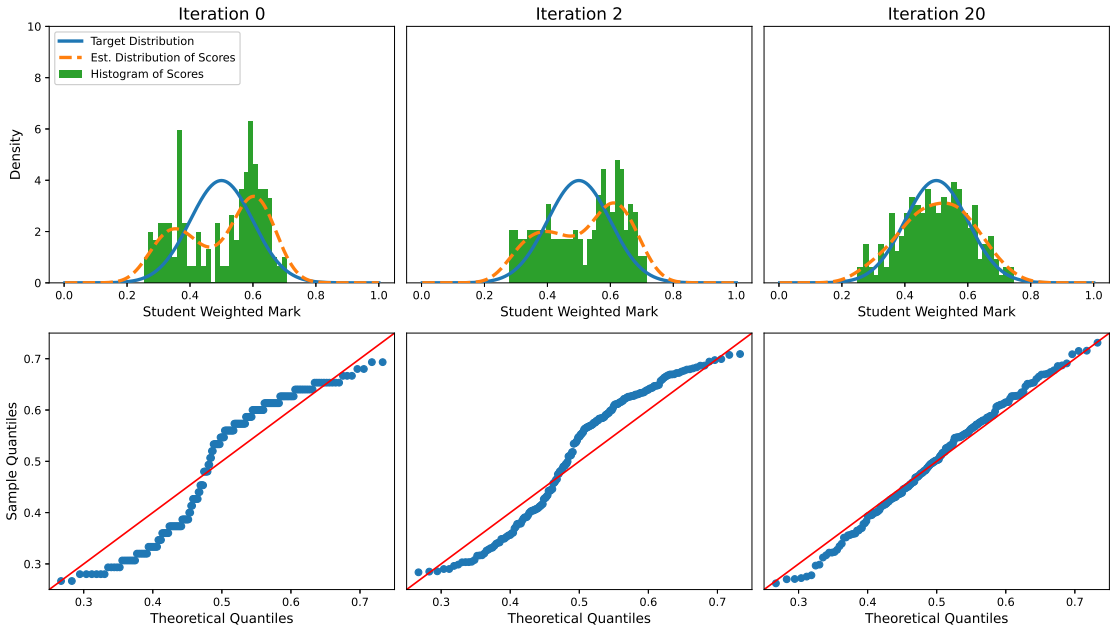


Figure 2: Optimal question weighting for (randomly generated) exam scores with 200 students and 75 questions. The setup follows the experimental details in Section 6.2. The kernel density estimate uses a truncated unit normal distribution with $\varepsilon = 0.05$, and the target distribution is a truncated normal distribution with a mean of 0.5 and a standard deviation of 0.1. We take $w^0 = (0.01, \ldots, 0.01)$, and the Cauchy-Simplex is applied, with each step using a backtracking line search. The resulting weighted histogram and kernel density estimate is shown.

The distribution of the weighted marks is shown on the top row, and its QQ plots against a normal distribution of mean 0.5 and a standard deviation of 0.1 are shown on the bottom row.

At iterations 0, 5, and 20, the weighted scores have a mean of 0.499, 0.514, and 0.501, with standard deviations of 0.128, 0.124, and 0.109, respectively.

Table 2: Results for optimal question weighting after running CS, EGD, and PFW for 150 iterations. The final distance (relative entropy) and the time taken in seconds are shown. Bold represents the minimum (best) value, and underline represents the maximum (worst) value.

	CS		EGD		PFW	
Trial	Distance	Time	Distance	Time	Distance	Time
1	0.032432	8.14	0.032426	10.98	**0.032114**	**6.02**
2	0.010349	5.95	0.010535	5.08	**0.010101**	**4.46**
3	0.016186	5.74	0.016252	**4.93**	**0.015848**	5.50
4	0.025684	4.62	0.025726	6.19	**0.025309**	**4.51**
5	0.020561	4.63	0.020486	6.03	**0.020213**	**4.50**
6	0.016559	5.58	0.016514	5.00	**0.016287**	**4.52**
7	0.025957	5.42	0.025867	**4.87**	**0.025757**	5.38
8	0.014506	4.77	0.014343	6.14	**0.013504**	**4.28**
9	0.032221	4.68	0.032412	6.02	**0.032028**	**4.55**
10	0.023523	5.59	0.023528	**4.92**	**0.023232**	5.34
11	0.016153	**4.93**	0.016231	5.01	**0.015792**	5.63
12	0.035734	4.53	0.035738	6.07	**0.035212**	**4.22**
13	0.030205	4.53	0.030234	6.13	**0.029859**	**4.37**
14	0.021725	5.80	0.021598	**4.99**	**0.021282**	5.85
15	0.030026	**4.44**	0.029982	4.99	**0.029751**	5.64
16	0.009212	4.75	0.009182	5.94	**0.008931**	**4.27**
17	0.015573	5.22	0.015661	5.46	**0.015188**	**4.48**
18	0.017681	5.69	0.017618	**4.92**	**0.017321**	5.48
19	0.017888	**4.64**	0.017874	5.34	**0.017283**	5.11
20	0.013597	4.55	0.013719	6.04	**0.013075**	**4.42**
21	0.016933	5.76	0.016780	4.84	**0.016687**	**4.77**
22	0.032185	5.80	0.032141	**5.03**	**0.032039**	6.03
23	0.018377	4.69	0.018250	6.06	**0.018084**	**4.54**
24	0.031167	4.59	0.031211	6.10	**0.030820**	**4.55**
25	0.035608	5.46	0.035674	**4.98**	**0.035408**	5.98
Average		5.22		5.68		4.97

It is perhaps expected that PFW outperforms both EGD and CS due to the low dimensionality of this problem. However, the CS produces similar relative entropies to EGD while maintaining a lower average run time of 5.22 seconds compared to the average run time of 5.68 sec for EGD.

6.3 Prediction from Expert Advice

Consider N 'experts' (e.g., Twitter) who give daily advice for $1 \leq t \leq T$ days. By taking advice from expert i on day t, we incur a loss $l_i^t \in [0, 1]$. However, this loss is not known

beforehand, only once the advice has been taken. Since the wisdom of the crowd is typically better than any given expert (Landemore and Elster, 2012), we'd like to take the weighted average of our experts, giving a daily loss of $w^t \cdot l^t$, where $w^t \in \Delta^N$ are the weights given to the experts. This problem is also known as the multi-armed bandit problem.

A simple goal is to generate a sequence of weight vectors $\{w^t\}_t$ to minimize the averaged expected loss. This goal is, however, a bit too ambitious as the loss vectors l^t are not known beforehand. An easier problem is to find a sequence, $\{w^t\}_t$, such that its averaged expected loss approaches the average loss of the best expert as $T \to \infty$, that is

$$\frac{1}{T} R_T \to 0, \quad \text{where} \quad R_T \;=\; \sum_{t=1}^{T} w^t \cdot l^t - \min_i \sum_{t=1}^{T} l_i^t$$

as $T \to \infty$. R_T is commonly known as the regret of the strategy $\{w^t\}_t$.

Previous works (Littlestone and Warmuth, 1994; Cesa-Bianchi et al., 1997; Freund and Schapire, 1997; Arora et al., 2012) all yield $\mathcal{O}(\sqrt{T \log N})$ convergence rate for the regret.

Theorem 11 *Consider a sequence of adversary loss vectors $l^t \in [0,1]^N$. For any $u \in \Delta^N$, the regret generated by the Cauchy-Simplex scheme*

$$w^{t+1} \;=\; w^t(1 - \eta^t \Pi_{w^t} \nabla f^t) \quad \textit{where} \quad \nabla f^t \;=\; l^t,$$

is bounded by

$$\sum_{t=1}^{T} w^t \cdot l^t - \sum_{t=1}^{T} u \cdot l^t \;\leq\; \frac{D(u|w^1)}{\eta} + \frac{T\eta}{2(1-\eta)},$$

for a fixed learning rate $\eta^t = \eta < 1$.

In particular, taking $w^1 = (1/N, \ldots, 1/N)$ and $\eta = \frac{\sqrt{2\log(N)}}{\sqrt{2\log(N)}+\sqrt{T}}$ gives the bound

$$\sum_{t=1}^{T} w^t \cdot l^t - \sum_{t=1}^{T} u \cdot l^t \;\leq\; \sqrt{2T\log(N)} + \log(N).$$

Moreover, this holds when $u = e_j$, where j is the best expert and e_j is the standard basis vector.

The proof can be found in Appendix B.5.

6.4 Universal Portfolio

Consider an investor with a fixed-time trading horizon, T, managing a portfolio of N assets. Define the price relative for the i-th stock at time t as $x_i^t = C_i^t / C_i^{t-1}$, where C_i^t is the closing price at time t for the i-th stock. So today's closing price of asset i equals x_i^t times yesterday's closing price, *i.e.* today's price relative to yesterday's.

A portfolio at day t can be described as $w^t \in \Delta^N$, where w_i^t is the proportion of an investor's total wealth in asset i at the beginning of the trading day. Then the wealth of

the portfolio at the beginning of day $t+1$ is $w^t \cdot x^t$ times the wealth of the portfolio at day t.

Consider the average log-return of the portfolio

$$\frac{1}{T}\log\left(\prod_{t=1}^{T} w^t \cdot x^t\right) \;=\; \frac{1}{T}\sum_{t=1}^{T}\log(w^t \cdot x^t).$$

Similarly to predicting with expert advice, it is too ambitious to find a sequence of portfolio vectors $\{w^t\}_t$ that maximizes the average log-return. Instead, we wish to find such a sequence that approaches the best fixed-weight portfolio, *i.e.*

$$\frac{1}{T}LR_T \to 0, \quad \text{where} \quad LR_T \;=\; \sum_{t=1}^{T}\log(u \cdot x^t) - \sum_{t=1}^{T}\log(w^t \cdot x^t)$$

as $T \to \infty$, for some $u \in \Delta^N$. If such a sequence can be found, $\{w^t\}_t$ is a universal portfolio. LR_T is commonly known as the log-regret.

Two standard assumptions are made when proving universal portfolios: (1) For every day, all assets have a bounded price relative, and at least one is non-zero, *i.e.* $0 < \max_i x_i^t < \infty$ for all t, and (2) No stock goes bankrupt during the trading period, *i.e.* $a := \min_{i,t} x_i^t > 0$, where a known as the market variability parameter. This is also known as the no-junk-bond assumption.

Over the years, various bounds on the log-regret have been proven under both assumptions. Some examples include Cover (1991), in his seminal paper, with $\mathcal{O}(\log T)$, Helmbold et al. (1998) with $\mathcal{O}(\sqrt{T\log N})$, Agarwal et al. (2006) with $\mathcal{O}(N^{1.5}\log(NT))$, Hazan and Kale (2015) with $\mathcal{O}(N\log(T+N))$, and Gaivoronski and Stella (2000) with $\mathcal{O}(C^2\log(T))$ where $C = \sup_{x\in\Delta^N}\|\nabla_b \log(b\cdot x)\|$ and adding an extra assumption on independent price relatives. Each with varying levels of computational complexity.

Remark 12 *Let $x^t \in [a,b]^N$ be a bounded sequence of price relative vectors for $0 \le a \le b < \infty$. Since the log-regret is invariant under re-scalings of x^t, w.l.o.g. we can look at the log-regret for the re-scaled return vectors $x^t \in [\tilde{a},1]^N$ for $0 \le \tilde{a} \le 1$.*

Theorem 13 *Consider a bounded sequence of price relative vectors $x^t \in [a,1]^N$ for some positive constant $0 < a \le 1$ (no-junk-bond), and $\max_i x_i^t = 1$ for all t. Then the log-regret generated by the Cauchy-Simplex*

$$w^{t+1} \;=\; w^t(1 - \eta\Pi_{w^t}\nabla f^t), \quad \textit{where} \quad \nabla f^t \;=\; -\frac{x^t}{w^t \cdot x^t},$$

is bounded by

$$\sum_{t=1}^{T}\log(u\cdot x^t) - \sum_{t=1}^{T}\log(w^t\cdot x^t) \;\le\; \frac{D(u|w^1)}{\eta} + \frac{T\eta}{2a^2(1-\eta)},$$

for any $u \in \Delta^N$ and $0 < \eta \le 1$.

In particular, taking $w^1 = (1/N,\ldots,1/N)$ and $\eta = \frac{a\sqrt{2\log(N)}}{a\sqrt{2\log(N)}+\sqrt{T}}$ gives the bound

$$\sum_{t=1}^{T}\log(u\cdot l^t) - \sum_{t=1}^{T}\log(w^t\cdot l^t) \;\le\; \frac{\sqrt{2T\log(N)}}{a} + \log(N).$$

The proof can be found in Appendix B.6.

Experimental Details: We look at the performance of our algorithm on four standard datasets used to study the performance of universal portfolios: (1) NYSE is a collection of 36 stocks traded on the New York Stock Exchange from July 3, 1962, to Dec 31, 1984, (2) DJIA is a collection of 30 stocks tracked by the Dow Jones Industrial Average from Jan 14, 2009, to Jan 14, 2003, (3) SP500 is a collection of the 25 largest market cap stocks tracked by the Standard & Poor's 500 Index from Jan 2, 1988, to Jan 31, 2003, and (4) TSE is a collection of 88 stocks traded on the Toronto Stock Exchange from Jan 4, 1994, to Dec 31, 1998.[5]

Two other portfolio strategies are considered: (1) Helmbold *et. al.* Helmbold et al. (1998) (EGD), who uses the EGD scheme $w^{t+1} = w^t \exp(\eta \nabla f^t) / \sum_i w_i^t \exp(\eta \nabla_i f^t)$, with $\nabla f^t = x^t / w^t \cdot x^t$, and (2) Buy and Hold (B&H) strategy, where one starts with an equally weighted portfolio and the portfolio is left to its own devices.

Two metrics are used to evaluate the performance of the portfolio strategies: (1) The Annualized Percentage Yield: $\text{APY} = \text{R}^{1/y} - 1$, where R is the total return over the full trading period, and $y = T/252$, where 252 is the average number of annual trading days, and (2) The Sharpe Ratio: $\text{SR} = (\text{APY} - R_f)/\sigma$, where σ^2 is the variance of the daily returns of the portfolio, and R_f is the risk-free interest rate. Intuitively, the Sharpe ratio measures the performance of the portfolio relative to a risk-free investment while also factoring in the portfolio's volatility. Following Li et al. (2012), we take $R_f = 4\%$.

We take the learning rate as the one used to prove that CS and EGD are universal portfolios. In particular, $\eta^{\text{CS}} = \frac{a\sqrt{2\log N}}{a\sqrt{2\log N}+\sqrt{T}}$ and $\eta^{\text{EGD}} = 2a\sqrt{2\log(N)/T}$, respectively, where a is the market variability parameter. We assume that the market variability parameter is given for each dataset.

Experiments were written in Python and can be found on GitHub[6].

Results: A table with the results can be seen in Table 3. For the NYSE, DJIA, and SP500 datasets, CS slightly outperforms EGD in both the APY and Sharpe ratio, with EGD having a slight edge on the APY for the NYSE dataset. But curiously, the B&H strategy outperforms both CS and EGD on the TSE.

We remark that this experiment does not reflect real-world performance, as the market variability parameter is assumed to be known, transaction costs are not factored into our analysis, and the no-junk-bond assumption tends to overestimate performance (Gilbert and Strugnell, 2010; Bailey et al., 2014; Bailey and de Prado, 2014). However, this is outside of the scope of this paper. It is only shown as a proof of concept.

7. Conclusion

This paper presents a new iterative algorithm, the Cauchy-Simplex, to solve convex problems over a probability simplex. Within this algorithm, we find ideas from previous works which only capture a portion of the simplex constraint. Combining these ideas, the Cauchy-Simplex provides a numerically efficient framework with nice theoretical properties.

5. The datasets were original found on `http://www.cs.technion.ac.il/~rani/portfolios/`, but is now unavailable. It was retrieved using the WayBack Machine `https://web.archive.org/web/20220111131743/http://www.cs.technion.ac.il/~rani/portfolios/`.
6. `https://github.com/infamoussoap/UniversalPortfolio`

Table 3: Performance of different portfolio strategies on different datasets. Bold represents the maximum (best) value, and underline represents the minimum (worst) value.

	CS		EGD		B&H	
Dataset	APY	Sharpe	APY	Sharpe	APY	Sharpe
NYSE	0.162	**14.360**	**0.162**	14.310	<u>0.129</u>	<u>9.529</u>
DJIA	**-0.099**	**-8.714**	-0.101	-8.848	<u>-0.126</u>	<u>-10.812</u>
SP500	**0.104**	**4.595**	0.101	4.395	<u>0.061</u>	<u>1.347</u>
TSE	0.124	10.225	<u>0.123</u>	<u>10.204</u>	**0.127**	**10.629**

The Cauchy-Simplex maintains the linear form of Projected Gradient Descent, allowing one to find analytical solutions to a line search for certain convex problems. But unlike projected gradient descent, this analytical solution will remain in the probability simplex. A backtracking line search can be used when an analytical solution cannot be found. However, this requires an extra parameter, the maximum candidate step size. The Cauchy-Simplex provides a natural answer as a maximum learning rate is required to enforce positivity, rather than the exponentials used in Exponentiated Gradient Descent.

Since the Cauchy-Simplex satisfies both constraints of the probability simplex in its iteration scheme, its gradient flow can be used to prove convergence for differentiable and convex functions. This implies the convergence of its discrete linear scheme. This is in contrast to EGD, PFW, and PGD, in which its discrete nature is crucial in satisfying both constraints of the probability simplex. More surprisingly, we find that in the proofs, formulas natural to probability, *i.e.*, variance, and relative entropy, are necessary when proving convergence.

We believe that the strong numerical results and simplicity seen through its motivating derivation, gradient flow, and iteration scheme make it a strong choice for solving problems with a probability simplex constraint.

Acknowledgments

We thank Prof. Johannes Ruf for the helpful discussion and his suggestion for potential applications in the multi-armed bandit problem, which ultimately helped the proof for universal portfolios. We also thank the anonymous reviewers for their helpful suggestions about improving the presentation and readability of this paper. The authors declare no competing interests.

Appendix A. Gradient Flow for Orthogonal Matrix Constraint

Consider the optimization problem

$$\min_{Q\in\mathcal{S}^n} f(Q), \quad \text{where} \quad \mathcal{S}^n = \{Q \in \mathbb{R}^{n\times n} \,|\, QQ^T = I\}.$$

Orthogonal matrices with $\det(Q) = 1$ can be parameterized using the Cayley Transform $Q = 2(I - X)^{-1} - I$, where $X \in \mathbb{R}^{n\times n}$ is an anti-symmetric matrix, and let $F(X) = f(Q)$. The anti-symmetric matrix X can be parameterized as

$$X \;=\; \sum_{j>k} X_{jk}\, E^{jk},$$

for $X_{jk} \in \mathbb{R}$ and $E^{jk}_{pq} = \delta_{pj}\delta_{qk} - \delta_{pk}\delta_{qj}$.

Consider the gradient flow

$$\begin{aligned}\frac{dQ_{pq}}{dt} &= -\sum_{jk} \frac{\partial Q_{pq}}{\partial X_{jk}} \frac{dX_{jk}}{dt} \\ &= -\sum_{jk} \frac{\partial Q_{pq}}{\partial X_{jk}} \sum_{ab} \frac{\partial f}{\partial Q_{ab}} \frac{\partial Q_{ab}}{\partial X_{jk}}.\end{aligned}$$

Under the Cayley Transform,

$$\begin{aligned}\frac{\partial Q}{\partial X_{jk}} &= 2(I - X)^{-1} \frac{\partial X}{\partial X_{jk}} (I - X)^{-1} \\ &= 2\Omega\, E^{jk}\, \Omega.\end{aligned}$$

Thus

$$\begin{aligned}\frac{\partial Q_{ab}}{\partial X_{jk}} &= 2\sum_{lm} \Omega_{al}\, E^{jk}_{lm}\, \Omega_{mb} \\ &= 2(\Omega_{aj}\Omega_{kb} - \Omega_{ak}\Omega_{jb})\end{aligned}$$

The gradient flow for the latent variables X_{jk} is therefore

$$\begin{aligned}\frac{dX_{jk}}{dt} &= 2\sum_{ab} \frac{\partial f}{\partial Q_{ab}} (\Omega_{aj}\Omega_{kb} - \Omega_{ak}\Omega_{jb}) \\ &= 2\left(\Omega^T \frac{\partial f}{\partial Q_{ab}} \Omega^T - \Omega \left(\frac{\partial f}{\partial Q_{ab}}\right)^T \Omega\right) \\ &= 2(\Lambda\, \Omega^T - \Omega\, \Lambda^T)_{jk},\end{aligned}$$

where $\Lambda = \Omega^T \frac{\partial f}{\partial Q}$. Converting this to the gradient flow for the orthogonal matrix Q yields

$$\begin{aligned}\frac{dQ_{pq}}{dt} &= -2\sum_{jk} (\Omega_{pj}\Omega_{kq} - \Omega_{pk}\Omega_{jq})(\Lambda\Omega^T - \Omega\Lambda^T)_{jk} \\ &= -2[\Omega(\Lambda\Omega^T - \Omega\Lambda^T)\Omega - \Omega(\Lambda\Omega^T - \Omega\Lambda^T)^T\Omega] \\ &= -4\Omega(\Lambda\Omega^T - \Omega\Lambda^T)\Omega,\end{aligned}$$

as required.

Appendix B. Convergence Proofs

We will use the notation $\Pi^t = \Pi_{w_t}$ and $\nabla f^t = \nabla_w f(w^t)$ for the remaining section.

B.1 Decreasing Relative Entropy

Theorem 14 *For any $u \in \Delta^N$, each iteration of the linear Cauchy-Simplex (6) satisfies the bound*

$$D(u|w^{t+1}) - D(u|w^t) \;\le\; \eta^t u \cdot (\Pi^t \nabla f^t) + C_{\gamma_t} (\eta^t)^2 u \cdot (\Pi^t \nabla f^t)^2,$$

where $C_{\gamma_t} = \gamma_t^{-2} \log(e^{-\gamma_t}/(1-\gamma_t))$ and $\eta^t = \gamma_t \eta^{t,\max}$ with $\gamma_t \in (0,1)$.

Proof By the CS update scheme (6)

$$\begin{aligned} D(u|w^{t+1}) - D(u|w^t) \;&=\; \sum_i u_i \log\left(w_i^{t+1}/w_i^t\right) \\ &=\; \sum_i u_i \log\left(\frac{1}{1-\eta^t \Pi^t \nabla_i f^t}\right) \\ &=\; \sum_i u_i \log\left(\frac{e^{-\eta^t \Pi^t \nabla_i f^t}}{1-\eta^t \Pi^t \nabla_i f^t} e^{\eta^t \Pi^t \nabla_i f^t}\right) \\ &=\; \eta^t u \cdot (\Pi^t \nabla f^t) + \sum_i u_i \log\left(\frac{e^{-\eta^t \Pi^t \nabla_i f^t}}{1-\eta^t \Pi^t \nabla_i f^t}\right). \end{aligned}$$

Since the learning rate can be written as $\eta^t = \gamma_t \eta^{t,\max}$, with $\gamma_t \in (0,1)$,

$$\eta^t \Pi^t \nabla_i f^t \;\le\; (\gamma_t \eta^{t,\max}) \max_i \Pi^t \nabla_i f^t \;=\; \gamma_t \;<\; 1.$$

It can be shown that $x^{-2}\log(e^{-x}/(1-x)) > 0$ for $x \in (0,1)$ and is an increasing function on $(0,1)$ with $x^{-2}\log(e^{-x}/(1-x)) \to \infty$ as $x \to 1$. Thus, in the interval $x \in (0, \gamma_t]$ with $\gamma_t < 1$, this can be upper-bounded by

$$0 \;<\; x^{-2}\log[e^{-x}/(1-x)] \;\le\; \gamma_t^{-2}\log[e^{-\gamma_t}/(1-\gamma_t)] \;=\; C_{\gamma_t}, \qquad \text{for} \qquad x \in (0, \gamma_t].$$

This yields the inequality $0 < \log(e^{-x}/(1-x)) \le C_{\gamma_t}\, x^2$ for $0 < x \le \gamma_t$.

Since $C_{\gamma_t} > 0$ for $\gamma_t \in (0,1)$ and $C_{\gamma_t} \to \infty$ as $\gamma_t \to 1$, we have the bound

$$0 \;\le\; \log\left(\frac{e^{-\eta^t \Pi^t \nabla_i f^t}}{1-\eta^t \Pi^t \nabla_i f^t}\right) \;\le\; C_{\gamma_t} (\eta^t)^2 (\Pi^t \nabla_i f^t)^2.$$

Giving the required inequality

$$D(u|w^{t+1}) - D(u|w^t) \;\le\; \eta^t u \cdot (\Pi^t \nabla f^t) + C_{\gamma_t} (\eta^t)^2 u \cdot (\Pi^t \nabla f^t)^2.$$

■

B.2 Progress Bound

Theorem 15 *Let f be convex, differentiable, and ∇f is L-Lipschitz continuous. Then each iteration of the linear Cauchy-Simplex (6) guarantees*

$$f(w^{t+1}) \;\leq\; f(w^t) - \frac{\eta^t}{2}\, Var[\nabla f^t \,|\, w^t], \quad \textit{for} \quad 0 < \eta^t < \min\left\{\frac{1}{L}, \eta^{t,\max}\right\},$$

where $\eta^{t,\max}$ is defined in (6).

Proof Since f is convex with ∇f Lipschitz continuous, the descent lemma yields (Bertsekas, 2016)

$$f(w^{t+1}) \;\leq\; f(w^t) + \nabla f^t \cdot (w^{t+1} - w^t) + \frac{L}{2}\|w^{t+1} - w^t\|^2. \tag{17}$$

Our iteration scheme gives that

$$\|w^{t+1} - w^t\|^2 \;=\; (\eta^t)^2 \sum_i \left(w_i^t\, \Pi^t \nabla_i f^t\right)^2 \;\leq\; (\eta^t)^2 \sum_i w_i^t (\Pi^t \nabla_i f^t)^2,$$

since $0 \leq w_i^t \leq 1$. Hence

$$f(w^{t+1}) \;\leq\; f(w^t) - \frac{\mathrm{Var}[\nabla f^t | w_t]}{2L}\left(2z - z^2\right),$$

where $z = \eta^t L$. However, $-(2z - z^2) \leq -z$ for $0 \leq z \leq 1$. Therefore,

$$f(w^{t+1}) \;\leq\; f(w^t) - \frac{\eta^t}{2}\mathrm{Var}[\nabla f^t \mid w^t], \quad \text{for} \quad 0 < \eta^t \leq \min\left\{\frac{1}{L}, \eta^{t,\max}\right\}.$$

■

B.3 Proof of Lemma 10

Proof Notice that for $\eta^t \in (0, \eta^{t,\max} - \varepsilon]$, the bound (17) still holds, and that a line search minimizes the left-hand side of (17). Bounding the right-hand side yields

$$f(w^{t+1}) \;\leq\; f(w^t) - \frac{\eta^t}{2}\mathrm{Var}[\nabla f^t \,|\, w^t], \quad \text{for} \qquad \eta^t = \min\left\{\frac{1}{L}, \eta^{t,\max} - \varepsilon\right\}.$$

As shown in Theorem 6, $\mathrm{Var}[\nabla f^t | w^t] = 0$ only when $w^t = w^*$. Thus w.l.o.g., for finite $T > 0$, assume the sequence of vectors $\{w_t\}_{t=0}^T$ generated by the line search satisfies $\Pi^t \nabla f^t \neq 0$, *i.e.*, the optimality condition has not been reached. Then $\{f(w^t)\}_{t=0}^T$ is a strictly decreasing sequence. Since f is convex and Δ is compact, it is bounded from below on Δ. Hence $f(w^T) \to f(w^*)$ as $T \to \infty$. ■

B.4 Proof of Theorem 9

Proof W.l.o.g, we assume that the sequence $\{\eta^t\}_{t=0}^T$, which satisfies the assumptions of Theorem 9, generates the sequence $\{w^t\}_{t=0}^T$ such that $\Pi^t \nabla f^t \neq 0$, *i.e.*, the optimality condition has not been reached.

By Theorem 14,

$$D(w^*|w^{t+1}) - D(w^*|w^t) \;\leq\; \eta^t \nabla f^t \cdot (w^* - w^t) + C_{\gamma_t} (\eta^t)^2 w^* \cdot (\Pi^t \nabla f^t)^2.$$

By convexity of f, rearranging gives

$$\eta^t \big(f(w^t) - f(w^*)\big) \;\leq\; D(w^*|w^t) - D(w^*|w^{t+1}) + C_{\gamma_t} (\eta^t)^2 w^* \cdot (\Pi^t \nabla f^t)^2. \tag{18}$$

By Theorem 15, we have that

$$f(w^t) \;\geq\; f(w^{t+1}) + \frac{\eta^t}{2} w^t \cdot (\Pi^t \nabla f^t)^2.$$

Repeatedly applying this inequality gives

$$f(w^T) + \frac{1}{2} \sum_{k=t}^{T-1} \eta^k w^k \cdot (\Pi^k \nabla f^k)^2 \;\leq\; f(w^t),$$

for $t \leq T-1$. Thus, (18) gives the bound

$$\begin{aligned} \eta^t (f(w^T) - f(w^*)) \;\leq\; & D(w^*|w^t) - D(w^*|w^{t+1}) + C_{\gamma_t} (\eta^t)^2 w^* \cdot (\Pi^t \nabla f^t)^2 \\ & - \frac{1}{2} \eta^t \sum_{k=t}^{T-1} \eta^k w^k \cdot (\Pi^k \nabla f^k)^2. \end{aligned}$$

Summing over time and collapsing the sum gives

$$\begin{aligned} (f(w^T) - f(w^*)) \sum_{t=0}^{T-1} \eta^t \;\leq\; & D(w^*|w^0) - D(w^*|w^T) \\ & + \sum_{t=0}^{T-1} C_{\gamma_t} (\eta^t)^2 w^* \cdot (\Pi^t \nabla f^t)^2 \\ & - \frac{1}{2} \sum_{t=0}^{T-1} \sum_{k=t}^{T-1} \eta^k \eta^t w^k \cdot (\Pi^k \nabla f^k)^2. \end{aligned} \tag{19}$$

We can rewrite the last term as

$$\sum_{t=0}^{T-1} \sum_{k=t}^{T-1} \eta^k \eta^t w^k \cdot (\Pi^k \nabla f^k)^2 \;=\; \sum_{t=0}^{T-1} \sum_{k=0}^{t} \eta^k \eta^t w^t \cdot (\Pi^t \nabla f^t)^2.$$

Thus, the last two terms of (19) can be bounded by

$$\begin{aligned} S \;&:=\; \sum_{t=0}^{T-1} \eta^t \left(C_{\gamma_t} \eta^t w^* \cdot (\Pi^t \nabla f^t)^2 - \frac{w^t \cdot (\Pi^t \nabla f^t)^2}{2} \sum_{k=0}^{t} \eta^k \right) \\ &\leq\; \sum_{t=0}^{T-1} \eta^t \left(C_{\gamma_t} \eta^t \max_i (\Pi^t \nabla f^t)_i^2 - \frac{w^t \cdot (\Pi^t \nabla f^t)^2}{2} \sum_{k=0}^{t} \eta^k \right), \end{aligned}$$

as $w^* \in \Delta^n$. By assumption on C_{γ_t},

$$C_{\gamma_t}\eta^t \max_i (\Pi^t \nabla f^t)_i^2 - \frac{w^t \cdot (\Pi^t \nabla f^t)^2}{2} \sum_{k=0}^{t} \eta^k \;\leq\; \frac{w^t \cdot (\Pi^t \nabla f^t)^2}{2} \left(\eta^t - \sum_{k=0}^{t} \eta^k \right) \;\leq\; 0.$$

It follows that $S \leq 0$. Thus (19) gives

$$f(w^T) - f(w^*) \;\leq\; \frac{D(w^*|w^0) - D(w^*|w^T)}{\sum_{t=0}^{T-1} \eta^t}.$$

Taking $w^0 = (1/n, \ldots, 1/n)$, then $D(u|w^0) \leq \log(n)$ for all $u \in \Delta^n$. Since relative entropy is non-negative,

$$f(w^T) - f(w^*) \;\leq\; \frac{\log(n)}{\sum_{t=0}^{T-1} \eta^t}.$$

■

B.5 Proof of Theorem 11

Proof Rearranging Theorem 14 gives

$$-\eta^t u \cdot (\Pi^t \nabla f^t) \;\leq\; D(u|w^t) - D(u|w^{t+1}) + C_{\gamma_t}(\eta^t)^2 u \cdot (\Pi^t \nabla f^t)^2. \tag{20}$$

Since $\nabla f^t = l^t$, we have the inequality

$$-1 \;\leq\; \Pi^t \nabla_i f^t \;=\; l_i^t - w^t \cdot l^t \;\leq\; 1,$$

as $l_i^t \in [0, 1]$. Thus dividing (20) by η^t gives

$$w^t \cdot l^t - u \cdot l^t \;\leq\; \frac{D(u|w^t) - D(u|w^{t+1})}{\eta^t} + C_{\gamma_t}\eta^t,$$

where $\eta^t = \gamma_t \eta^{t,\max}$, for some $\gamma_t \in (0, 1)$.

Since the maximum learning rate has the lower bound

$$\eta^{t,\max} \;=\; \frac{1}{\max_i l_i^t - w^t \cdot l^t} \;\geq\; \frac{1}{\max_i l_i^t} \;\geq\; 1,$$

we can take a fixed learning rate $\eta^t = \eta \in (0, 1)$. Moreover, $\gamma_t = \eta/\eta^{t,\max} \leq \eta$. Since C_{γ_t} is an increasing function of γ_t, $C_{\gamma_t} \leq C_\eta$, thus giving the bound

$$w^t \cdot l^t - u \cdot l^t \;\leq\; \frac{D(u|w^t) - D(u|w^{t+1})}{\eta} + C_\eta \eta.$$

Summing over time and collapsing the sum gives the bound

$$\begin{aligned}\sum_{t=1}^{T} w^t \cdot l^t - \sum_{t=1}^{T} u \cdot l^t &\leq \frac{D(u|w^1) - D(u|w^{T+1})}{\eta} + TC_\eta\eta \\ &\leq \frac{D(u|w^1)}{\eta} + TC_\eta\eta \\ &= \frac{D(u|w^1)}{\eta} + \frac{T\log(e^{-\eta}/(1-\eta))}{\eta},\end{aligned}$$

by definition of C_η. Using the inequality $\log(e^{-x}/(1-x))/x \leq x/(2(1-x))$ for $0 \leq x \leq 1$,

$$\sum_{t=1}^{T} w^t \cdot l^t - \sum_{t=1}^{T} u \cdot l^t \leq \frac{D(u|w^1)}{\eta} + \frac{T\eta}{2(1-\eta)}.$$

Let $w^1 = (1/N, \ldots, 1/N)$, then $D(u|w^1) \leq \log(N)$ for all $u \in \Delta^N$. Thus giving the desired bound

$$\sum_{t=1}^{T} w^t \cdot l^t - \sum_{t=1}^{T} u \cdot l^t \leq \frac{\log(N)}{\eta} + \frac{T\eta}{2(1-\eta)}.$$

The right side of this inequality is minimized when $\eta = \frac{\sqrt{2\log(N)}}{\sqrt{2\log(N)+\sqrt{T}}} < 1$. Upon substitution gives the bound

$$\sum_{t=1}^{T} w^t \cdot l^t - \sum_{t=1}^{T} u \cdot l^t \leq \sqrt{2T\log(N)} + \log(N).$$

■

B.6 Proof of Theorem 13

Proof Rearranging Theorem 14 gives

$$-\eta^t u \cdot (\Pi^t \nabla f^t) \leq D(u|w^t) - D(u|w^{t+1}) + C_{\gamma_t}(\eta^t)^2 u \cdot (\Pi^t \nabla f^t)^2.$$

Since $\nabla f = -x^t/(w^t \cdot x^t)$, we have the inequality

$$-\frac{1}{a} \leq \Pi\nabla_i f = 1 - x_i^t/(w^t \cdot x^t) \leq 1,$$

as $x_i^t \in [a, 1]$. Thus diving by η^t gives the bound

$$\frac{u \cdot x^t}{w^t \cdot x^t} - 1 \leq \frac{D(u|w^t) - D(u|w^{t+1})}{\eta^t} + \frac{C_\gamma \eta^t}{a^2}.$$

Using the inequality $e^x - 1 \geq x$ for all x gives

$$\log\left(\frac{u \cdot x^t}{w^t \cdot x^t}\right) \;\leq\; \frac{D(u|w^t) - D(u|w^{t+1})}{\eta^t} + \frac{C_\gamma \eta^t}{a^2}.$$

Since the maximum learning rate has the lower bound

$$\eta^{t,\max} \;=\; \frac{1}{\max_i(1 - x_i^t/w^t \cdot x^t)} \;=\; \frac{1}{1 - \min_i(x_i^t/w^t \cdot x^t)} \;\geq\; 1,$$

we can take a fixed learning rate $\eta^t = \eta$.

Following the steps from Theorem 11 gives the bound

$$\sum_{t=1}^{T} \log(u \cdot l^t) - \sum_{t=1}^{T} \log(w^t \cdot l^t) \;\leq\; \frac{D(u|w^1)}{\eta} + \frac{T\eta}{2a^2(1-\eta)}.$$

Taking $w^1 = (1/N, \ldots, 1/N)$ and minimizing the right-hand side of the inequality w.r.t. η gives $\eta = \frac{a\sqrt{2\log(N)}}{a\sqrt{2\log(N)}+\sqrt{T}}$. Thus giving the bound

$$\sum_{t=1}^{T} \log(u \cdot l^t) - \sum_{t=1}^{T} \log(w^t \cdot l^t) \;\leq\; \frac{\sqrt{2T\log(N)}}{a} + \log(N).$$

■

Appendix C. Karush-Kuhn-Tucker Conditions

The Karush-Kuhn-Tucker (KKT) Conditions are first-order conditions that are necessary but insufficient for optimality in constrained optimization problems. For convex problems, it becomes a sufficient condition for optimality. We give a brief overview of the KKT conditions here, and for a full treatment of the subject, we suggest Kochenderfer and Wheeler (2019).

Consider a general constrained optimization problem

$$\min_w f(w) \qquad \text{s.t.} \qquad g_i(w) \;\leq\; 0 \quad \text{and} \quad h_j(w) \;=\; 0, \tag{21}$$

for $i = 1, ..., n$ and $j = 1, ..., m$. The (primal) Lagrangian is defined as

$$\mathcal{L}(w, \alpha, \beta) \;=\; f(w) + \sum_i \alpha_i g_i(w) + \sum_j \beta_j h_j(w).$$

Consider the new optimization problem

$$\theta(w) \;=\; \max_{\alpha,\beta;\ \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta).$$

Note that

$$\theta(w) \;=\; \begin{cases} f(w) & \text{if } w \text{ satisfies the constraints} \\ \infty & \text{otherwise.} \end{cases}$$

Hence, α_i and β_j are slack variables that render a given Lagrangian variation equation irrelevant when violated.

To solve (21), we can instead consider the new optimization problem

$$\min_w \theta(w) \;=\; \min_w \max_{\alpha,\beta;\ \alpha_i \geq 0} \mathcal{L}(w, \alpha, \beta).$$

Assume f and g_i are convex, h_j is affine, and the constraints are feasible. A solution (w^*, α^*, β^*) is an optimal solution to (21) if the following conditions, known as the KKT conditions, are satisfied:

$$\begin{aligned}
\frac{\partial}{\partial w_i}\mathcal{L}(w^*, \alpha^*, \beta^*) &= 0 \qquad \text{(Stationarity)} \\
\alpha_i^* g_i(w^*) &= 0 \qquad \text{(Complementary Slackness)} \\
g_i(w^*) \leq 0,\ h_j(w) &= 0 \qquad \text{(Primal Feasibility)} \\
\alpha_i^* &\geq 0 \qquad \text{(Dual Feasibility)},
\end{aligned}$$

for all i and j.

When the constraint is a simplex, the Lagrangian becomes

$$\mathcal{L}(w, \alpha, \beta) \;=\; f(w) - \sum_i \alpha_i w_i + \beta \left(\sum_i w_i - 1 \right).$$

Thus stationarity gives

$$\frac{\partial}{\partial w_i}\mathcal{L} \;=\; \nabla_i f - \alpha_i + \beta = 0.$$

Let $Q = \{i : w_i = 0\}$ be the active set and $S = \{i : w_i > 0\}$ be the support. The complementary slackness requires $\alpha_i = 0$ for $i \in S$, so stationarity gives $\beta = \nabla_i f$, *i.e.* constant on the support. The active set's dual feasibility and stationarity conditions thus require $\alpha_i = \beta + \nabla_i f \geq 0$.

References

P. Ablin and G. Peyré. Fast and accurate optimization on the orthogonal manifold without retraction. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.

E. M. Achour, F. Malgouyres, and F. Mamalet. Existence, stability and scalability of orthogonal convolutional neural networks. *Journal of Machine Learning Research*, 23(1), jan 2022.

A. Agarwal, E. Hazan, S. Kale, and R. E. Schapire. Algorithms for portfolio management based on the Newton method. In *Proceedings of the 23rd international conference on Machine learning*. ACM Press, 2006.

M. Amsler, V. I. Hegde, S. D. Jacobsen, and C. Wolverton. Exploring the high-pressure materials genome. *Physical Review X*, 2018.

S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: A meta-algorithm and applications. *Theory of Computing*, 2012.

D. H. Bailey and M. L. de Prado. The deflated Sharpe ratio: Correcting for selection bias, backtest overfitting and non-normality. *SSRN Electronic Journal*, 2014.

D. H. Bailey, J. M. Borwein, M. L. de Prado, and Q. J. Zhu. Pseudomathematics and financial charlatanism: The effects of backtest over fitting on out-of-sample performance. *Notices of the AMS*, 2014.

D. J. Bartholomew. *Measuring intelligence*. Cambridge University Press, Cambridge, England, August 2004.

A. Bellet, Y. Liang, A. B. Garakani, M. F. Balcan, and F. Sha. *A Distributed Frank–Wolfe Algorithm for Communication-Efficient Sparse Learning*. Society for Industrial and Applied Mathematics, 2015.

D. P. Bertsekas. *Nonlinear programming*, page 667. Athena Scientific, 3rd edition, 2016.

D. Bertsimas and D. L. Kitane. Sparse PCA: A geometric approach. *Journal of Machine Learning Research*, 2023.

F. Bomze, I. M.and Rinaldi and D. Zeffiro. Frank–Wolfe and friends: A journey into projection-free first-order optimization methods, 2021.

I. M. Bomze. On standard quadratic optimization problems. *Journal of Global Optimization*, 1998.

I. M. Bomze. Regularity versus degeneracy in dynamics, games, and optimization: A unified approach to different aspects. *SIAM Review*, 2002.

J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer-Verlag, 2006.

A. Bucci, L. Ippoliti, and P. Valentini. Comparing unconstrained parametrization methods for return covariance matrix prediction. *Statistics and Computing*, 2022.

A. Candelieri, A. Ponti, and F. Archetti. Bayesian optimization over the probability simplex. *Annals of Mathematics and Artificial Intelligence*, 2025.

I. Canyakmaz, W. Lin, G. Piliouras, and A. Varvitsiotis. Multiplicative updates for online convex optimization over symmetric cones. *arXiv preprint arXiv:2307.03136*, 2023.

N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M.K. Warmuth. How to use expert advice. *Journal of the ACM*, 1997.

S. Chen, A. Garcia, M. Hong, and S. Shahrampour. Decentralized Riemannian gradient descent on the Stiefel manifold. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021.

Yunmei Chen and Xiaojing Ye. Projection onto a simplex. *arXiv preprint arXiv:1101.6081*, 2011.

H.-H. Chou, J. Maly, and H. Rauhut. More is less: Inducing sparsity via overparameterization. *Information and Inference: A Journal of the IMA*, 2023.

Hung-Hsu Chou, Holger Rauhut, and Rachel Ward. Robust implicit regularization via weight normalization, 2024.

M. Collins, A. Globerson, T. Koo, X. Carreras, and P. L. Bartlett. Exponentiated gradient algorithms for conditional random fields and max-margin Markov networks. *Journal of Machine Learning Research*, 2008.

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

P. Das and A. Banerjee. Meta optimization and its application to portfolio selection. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2011.

E. de Klerk, D. den Hertog, and G. Elabwabi. On the complexity of optimization over the standard simplex. *European Journal of Operational Research*, 2008.

L. L. Duan. Latent simplex position model: High dimensional multi-view clustering with uncertainty quantification. *Journal of Machine Learning Research*, 2020.

S. Fiori. Quasi-geodesic neural learning algorithms over the orthogonal group: A tutorial. *Journal of Machine Learning Research*, 2005.

G. Floto, T. Jonsson, M. Nica, S. Sanner, and E. Z. Zhu. Diffusion on the probability simplex. *arXiv preprint arXiv:2309.02530*, 2023.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 1956.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997.

A. A. Gaivoronski and F. Stella. Stochastic nonstationary optimization for finding universal portfolios. *Annals of Operations Research*, 2000.

A. Galántai. *Projectors and Projection Methods*. Springer US, 2004.

J. W. Gibbs. *Elementary Principles in Statistical Mechanics*. Cambridge University Press, September 2010.

E. Gilbert and D. Strugnell. Does survivorship bias really matter? An empirical investigation into its effects on the mean reversion of share returns on the JSE (1984–2007). *Investment Analysts Journal*, 2010.

L. S. Gottfredson. Logical fallacies used to dismiss the evidence on intelligence testing. In *Correcting fallacies about educational and psychological testing*. American Psychological Association, 2009.

S. Grünewälder. Compact convex projections. *Journal of Machine Learning Research*, 2018.

J. Guélat and P. Marcotte. Some comments on Wolfe's 'away step'. *Mathematical Programming*, 1986.

P. R. Halmos. *Introduction to Hilbert Space and the Theory of Spectral Multiplicity: Second Edition*. Dover Books on Mathematics. Dover Publications, 1998.

P. Hansen, B. Jaumard, and S. H. Lu. On using estimates of Lipschitz constants in global optimization. *Journal of Optimization Theory and Applications*, 1992.

E. Hazan and S. Kale. An online portfolio selection algorithm with regret logarithmic in price variation. *Mathematical Finance*, 2015.

D. P. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, October 1998.

P. J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 1964.

M. Jaggi. Revisiting Frank–Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*. PMLR, 2013.

E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.

R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, 2011.

B. Jiang and Y.-H. Dai. A framework of constraint preserving update schemes for optimization on Stiefel manifold. *Mathematical Programming*, 2014.

S. W. Jung, T. H. Kim, and S. J. Ko. A novel multiple image deblurring technique using fuzzy projection onto convex sets. *IEEE Signal Processing Letters*, 2009.

A. Kalai and S. Vempala. Efficient algorithms for universal portfolios. *Journal of Machine Learning Research*, 3:423–440, mar 2003.

J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 1997.

M. J. Kochenderfer and T. A. Wheeler. *Algorithms for optimization*. The MIT Press, 2019.

A. Kuznetsova and A. Strekalovsky. On solving the maximum clique problem, 2001.

A. Kyrillidis, S. Becker, V. Cevher, and C. Koch. Sparse projections onto the simplex. In *Proceedings of the 30th International Conference on Machine Learning*. PMLR, 2013.

S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank–Wolfe optimization variants, 2015.

S. Lacoste-Julien, M. Jaggi, M. Schmidt, and P. Pletscher. Block-coordinate Frank–Wolfe optimization for structural SVMs. In *Proceedings of the 30th International Conference on Machine Learning*. PMLR, 2013.

H. Landemore and J. Elster. *Collective Wisdom: Principles and Mechanisms*. Cambridge University Press, 2012.

M. Lezcano-Casado and D. Martínez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019.

B. Li, P. Zhao, S. C. H. Hoi, and V. Gopalkrishnan. PAMR: Passive aggressive mean reversion strategy for portfolio selection. *Machine Learning*, 2012.

Q. Li, D. McKenzie, and W. Yin. From the simplex to the sphere: Faster constrained optimization using the hadamard parametrization. *Information and Inference: A Journal of the IMA*, 2023.

Y.-H. Li and V. Cevher. Convergence of the exponentiated gradient method with Armijo line search. *Journal of Optimization Theory and Applications*, 2018.

N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 1994.

H. Lu, R. M. Freund, and Y. Nesterov. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 2018.

D. G. Luenberger. *Optimization by Vector Space Methods*. John Wiley and Sons, Inc., USA, 1st edition, 1997.

N. Mackintosh. *IQ and Human Intelligence*. Oxford University Press, London, England, 2nd edition, 2011.

H. M. Markowitz, R. Lacey, J. Plymen, M. A. H. Dempster, and R. G. Tompkins. The general mean-variance portfolio selection problem [and discussion]. *Philosophical Transactions: Physical Sciences and Engineering*, 1994.

R. A. Maronna, R. D. Martin, and V. J. Yohai. *Robust Statistics*. Wiley, 2006.

C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of $\mathbb{R}^n$. *Journal of Optimization Theory and Applications*, 1986.

T. Mizutani. Ellipsoidal rounding for nonnegative matrix factorization under noisy separability. *Journal of Machine Learning Research*, 2014.

C. Mu, Y. Zhang, J. Wright, and D. Goldfarb. Scalable robust matrix recovery: Frank–Wolfe meets proximal methods. *SIAM Journal on Scientific Computing*, 2016.

M. Nandan, P. P. Khargonekar, and S. S. Talathi. Fast SVM training using approximate extreme points. *Journal of Machine Learning Research*, 2014.

A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. A Wiley-Interscience publication. Wiley, 1983.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

L. Omlor and M. A. Giese. Anechoic blind source separation using Wigner marginals. *Journal of Machine Learning Research*, 2011.

A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 2008.

K. Richardson. What IQ tests test. *Theory & Psychology*, 2002.

J. Särelä and H. Valpola. Denoising source separation. *Journal of Machine Learning Research*, 2005.

P. Schuster and K. Sigmund. Replicator dynamics. *Journal of Theoretical Biology*, 100(3): 533–538, 1983.

A. Selvi, D. den Hertog, and W. Wiesemann. A reformulation-linearization technique for optimization over simplices. *Mathematical Programming*, 2023.

A. B. Shuttleworth-Edwards. Generally representative is representative of none: commentary on the pitfalls of IQ test standardization in multicultural settings. *The Clinical Neuropsychologist*, 2016.

A. B. Shuttleworth-Edwards, R. D. Kemp, A. L. Rust, J. G. L. Muirhead, N. P. Hartman, and S. E. Radloff. Cross-cultural effects on IQ test performance: A review and preliminary normative indications on WAIS-III test performance. *Journal of Clinical and Experimental Neuropsychology*, 2004.

K. Tajima, Y. Hirohashi, E. R. R. Zara, and T. Kato. *Frank–Wolfe algorithm for learning SVM-type multi-category classifiers*. Society for Industrial and Applied Mathematics, 2021.

N. K. Vishnoi. *Mirror Descent and the Multiplicative Weights Update*, page 108–142. Cambridge University Press, 2021.

W. Wang and M. A Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013.

X. Wang, J. Zhang, and W. Zhang. The distance between convex sets with Minkowski sum structure: Application to collision detection. *Computational Optimization and Applications*, 2020.

Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 2012.

B. Woodworth, S. Gunasekar, J. D. Lee, E. Moroshko, P. Savarese, I. Golan, D. Soudry, and N. Srebro. Kernel and rich regimes in overparametrized models. In *Proceedings of Thirty Third Conference on Learning Theory*. PMLR, 2020.

N. Xiu, C. Wang, and L. Kong. A note on the gradient projection method with exact stepsize rule. *Journal of Computational Mathematics*, 2007.

R. Yousefzadeh. A sketching method for finding the closest point on a convex hull. *arXiv preprint arXiv:2102.10502*, 2021.