Sparse projections onto the simplex

Anastasios Kyrillidis, Stephen Becker and Volkan Cevher *
Laboratory for Information and Inference Systems
École Polytechnique Fédérale de Lausanne
{anastasios.kyrillidis, volkan.cevher}@epfl.ch
Laboratory Jacques Louis-Lions
Université Pierre et Marie Curie
stephen.becker@upmc.fr

June 15, 2012

Abstract

The past decade has seen the rise of ℓ_1-relaxation methods to promote sparsity for better interpretability and generalization of learning results. However, there are several important learning applications, such as Markowitz portolio selection and sparse mixture density estimation, that feature simplex constraints, which disallow the application of the standard ℓ_1-penalty. In this setting, we show how to efficiently obtain sparse projections onto the positive and general simplex with sparsity constraints. We provide an exact sparse projector for the positive simplex constraints, and derive a novel approach with online optimality and approximation guarantees for sparse projections onto the general simplex constraints. Even for small sized problems, this new approach is three orders of magnitude faster than the alternative, state-of-the-art branch-and-bound based CPLEX solver with no sacrifice in solution quality. We also empirically demonstrate that our projectors provide substantial benefits in portfolio selection and density estimation.

1 Introduction

Learning applications typically boil down to optimization of a loss function $f(\beta)$ in order to obtain a (model or feature) vector $\beta \in \mathbb{R}^p$. In this setting, we prefer *sparse* solutions even if there exist other solutions that might obtain better loss values. By sparse, we mean that β has at most s nonzero coefficients where $s \ll p$. This choice is well-justified as sparsity-based learning not only shows great empirical success with improved interpretability, but also is backed up with rigorous theoretical analysis. For instance, sparsity provably avoids over-fitting for better generalization of learning algorithms, and provably circumvents the ill-posed nature of regression problems [1, 2].

However, sparsity inherently introduces non-convexity into learning problems, which is undesirable according to conventional wisdom. For instance, to obtain a minimizer of $f(\beta)$ with a specific sparsity s, we have to deal with a non-convex constraint: $\|\beta\|_0 \leq s$, where $\|\cdot\|_0$ counts the sparsity of β. In turn, the resulting minimization is typically NP-Hard (even if $f(\beta)$ is convex) if we seek a global minimizer (cf., [3]). Surprisingly, it is possible to obtain *sparse critical points* of general loss functions with the projected gradient algorithm [4]. This algorithm iteratively uses the Euclidean projection onto the constraint $\|\beta\|_0 \leq s$, which is efficiently obtained via hard-thresholding.

Luckily, we can almost always achieve approximately sparse solutions via a simple convexification. For instance, we simply replace the $\|\beta\|_0 \leq s$ constraint with an ℓ_1-norm constraint as $\|\beta\|_1 \leq \lambda$, where $\lambda \in \mathbb{R}^+$. The Euclidean projection onto the convex constraint $\|\beta\|_1 \leq \lambda$ can be efficiently obtained via soft-thresholding [5, 6], which automatically sparsifies solutions. Moreover, if $f(\beta)$ is convex, then we can leverage decades of research in convex optimization, which not only provide computationally efficient algorithms to obtain accurate solutions, but also strong analytical tools to establish consistency and prediction efficiency of the solutions as λ varies [2].

*This work was supported in part by the European Commission under Grant MIRG-268398, ERC Future Proof, and DARPA KeCoM program #11-DARPA-1055. VC also would like to acknowledge Rice University for his Faculty Fellowship.

While the ℓ_1-norm is now a *de facto standard* in sparsity related problems, several important learning problems involve *simplex* constraints that mar its application in further sparsifying the solutions. Let us first recall the definitions of the two simplex variants. The *standard simplex* in $\mathbb{R}^p$ with parameter λ is given by $\Delta_\lambda^+ = \{\beta \in \mathbb{R}^p : \beta_i \geq 0,\ \sum_i \beta_i = \lambda\}$. The *general simplex* in $\mathbb{R}^p$ with parameter λ is given by $\Delta_\lambda = \{\beta \in \mathbb{R}^p :\ \sum_i \beta_i = \lambda\}$. It is clear that direct applications of the ℓ_1-norm (regularization, constraint, or otherwise) cannot achieve further sparsification beyond what the standard simplex constraint obtains. Unfortunately, many applications require further sparsification.

As a stylized example, consider the Markowitz portfolio selection, where we try to learn a design vector β that minimizes a return-adjusted empirical risk objective [7, 8]. Here, we immediately have simplex constraints, i.e., Δ_1^+ with no-short positions and Δ_1 with short positions due to limited capital resources. Moreover, we typically need solutions sparser than the ones with the simplex constraint due to two reasons. The first reason is robustness: since empirical covariance and return estimates are rather noisy, sparser portfolios generalize better [7, 8]. The second reason is cost: transactions fees increase with sparsity. Other example learning problems include sparse mixture/kernel density estimation [9, 10], and boosting/leveraging weak classifiers [11].

Contributions: Within this context, our contributions can be summarized as follows:

1. We prove a new result that states greedy sparse selection followed by standard simplex projections provides efficient and optimal sparse Euclidean projections onto Δ_λ^+.
2. We introduce a new projection algorithm for efficiently obtaining approximate sparse Euclidean projections onto Δ_λ. This new scheme is based on a principled, iterative selection method that searches over relaxations of the general simplex constraint to obtain the best approximate projection. We establish the exact convergence of the scheme, and show that we can certify the optimality of our projections or obtain approximation guarantees *online*.
3. We apply our projections in iterative optimization algorithms to real-life and synthetic examples. In Markowitz portfolio selection problems, we can reduce transaction costs while still improving average return. In mixture density learning, our results naturally reveal the underlying model order. Finally, in general simplex projection problems, our approach outperforms the state-of-the-art branch-and-bound approaches by three orders of magnitude in efficiency even in small size problems, while obtaining exactly the same solutions.

To the best of our knowledge, sparse Euclidean projections onto the simplex constraints have not been considered before. The closest works to ours are the papers [12, 13], where the authors recover a vector β in regression setting, where β is already sparse and within a convex norm-ball constraint. In this paper, we consider the problem of projecting an *arbitrary* given vector w onto convex-based and sparse constraints, *jointly*. While the results [12, 13] only apply to linear regression problems with the restricted isometry property, our projectors can be used in general minimization problems.

Notation: Given a set $\mathcal{S} \subseteq \mathcal{N} = \{1, \ldots, p\}$, the complement $\mathcal{S}^c$ is defined with respect to $\mathcal{N}$, and the cardinality is written $|\mathcal{S}|$. The support set of w is defined as $\mathrm{supp}(w) = \{i : w_i \neq 0\}$. Given a vector $w \in \mathbb{R}^p$, $w_\mathcal{S}$ is either the projection (in $\mathbb{R}^p$) of w onto S, i.e. $(w_\mathcal{S})_{\mathcal{S}^c} = 0$, or a vector in $\mathbb{R}^{|\mathcal{S}|}$ depending on context. The all-ones column vector is $\mathbb{1}$, with dimensions apparent from the context. We define Σ_s as the set of all s-sparse subsets of $\mathcal{N}$, and we sometimes write $\beta \in \Sigma_s$ to mean $\mathrm{supp}(\beta) \in \Sigma_s$.

2 Preliminaries

Problem statement: In this paper, we mainly study the following problem and its variant:

Problem 1. *Given a vector $w \in \mathbb{R}^p$, find a Euclidean projection of w onto the intersection of s-sparse signals and the standard simplex Δ_λ^+:*

$$\beta^\star \in \operatorname*{arg\,min}_{\beta:\mathrm{supp}(\beta)\in\Sigma_s, \beta\in\Delta_\lambda^+} \|\beta - w\|_2^2. \tag{1}$$

Problem 2. *Same as Problem 1 but use the general simplex Δ_λ instead of Δ_λ^+.*

Remark 1. *Problems 1–2 are special instances of mixed-integer programming and are in general combinatorially hard. However, there exist well-built solvers relying on good branch-and-bound heuristics, such as CPLEX [14]. In the sequel, we prove that Problem 1 can be solved optimally. Therefore, we use CPLEX as a benchmark against our relaxed solution to Problem 2.*

Projections: We will write $\mathcal{P}_\lambda$ for the projector onto either the standard or general simplex (it will be clear from context which we mean). Without loss of generality, let w be sorted in descending order so that w_1 is the largest element, and write $[w_i]_+$ to mean $\max(w_i, 0)$.

Definition 1 (Greedy basis algorithm $\mathcal{P}_{\Sigma_s}$)**.** *The greedy basis algorithm calculates the optimal projection $\mathcal{P}_{\Sigma_s}$ onto Σ_s itself in $\mathcal{O}(p \max(s, \log(p)))$-time, by picking the s-largest entries of w. The optimality of this algorithm follows from the matroid structure of cardinality constraints [12, 15]. This particular operation is also known as hard thresholding.*

Definition 2 (Projection onto the standard simplex $\mathcal{P}_\lambda$)**.** *The projector onto the standard simplex is given by*

$$(\mathcal{P}_\lambda(w))_i = [w_i - \tau]_+ \text{ where } \tau = \tau_\rho,\ \tau_j := \frac{1}{j}\left(\sum_{i=1}^{j} w_i - \lambda\right) \text{ and } \rho := \max\{j : w_j > \tau_j\}. \tag{2}$$

The projector onto the convex hull of the standard simplex (that is, require $\sum \beta_i \leq \lambda$ rather than $\sum \beta_i = \lambda$) is written $\bar{\mathcal{P}}_\lambda$ and is identical except we set replace τ with $[\tau]_+$.

Definition 3 (Projection onto the general simplex $\mathcal{P}_\lambda$)**.** *The projector onto the general simplex is [16]*

$$(\mathcal{P}_\lambda(w))_i = w_i + \tau \quad \text{where} \quad \tau = \frac{1}{p}(\lambda - \sum_{i=1}^{p} w_i). \tag{3}$$

A reformulation: The following observation is convenient in the sequel:

Remark 2. *The Problem 1 statement* (1) *can be equivalently transformed into the following nested minimization problem:*

$$\{\mathcal{S}^\star, \beta^\star_{\mathcal{S}^\star}\} = \operatorname*{arg\,min}_{\mathcal{S}:\mathcal{S}\in\Sigma_s} \Big[\min_{\substack{\beta:\beta_\mathcal{S}\in\Delta_\lambda^+,\\ \beta_{\mathcal{S}^c}=0}} \|(\beta - w)_\mathcal{S}\|_2^2 + \|(w)_{\mathcal{S}^c}\|_2^2 \Big]. \tag{4}$$

where $\operatorname{supp}(\beta^\star) = \mathcal{S}^\star$ *and* $\beta^\star \in \Delta_\lambda^+$.

3 Efficient sparse projections onto the simplex

Let $\beta^\star$ be a projection of w onto $\Sigma_s \cap \mathcal{K}_\lambda$, where $\mathcal{K}_\lambda$ is either Δ_λ^+ or Δ_λ. Let $C(\beta)$ be a mapping that encodes the constraint $\mathcal{K}_\lambda$. For instance, $C(\beta) = \sum_i \beta_i$ so $C(\beta) = \lambda$ evaluates the simplex constraint.

We first make an elementary observation based on Remark 2. Given $\mathcal{S}^\star = \operatorname{supp}(\beta^\star)$, we can find $\beta^\star$ by projecting $w_{\mathcal{S}^\star}$ onto $\mathcal{K}_\lambda$ within the s-dimensional space. Thus, the difficulty is finding $\mathcal{S}^\star$.

Algorithm 1 suggests an obvious greedy approach. We select the set $\mathcal{S}^\star$ by naively projecting w onto Σ_s. Remarkably, this gives the correct support set for Problem 1, as we prove in §4.1. After finding the support, we project (restricted to the support) back to $\mathcal{K}_\lambda$. We call this algorithm the greedy selector and simplex projector (GSSP).

Unfortunately, the greedy approach fails for Problem 2. Here we propose Algorithm 2 which is non-obvious. The algorithm first projects w onto $\mathcal{K}_\tau$, which we call a relaxed version of $\mathcal{K}_\lambda$ since it is possible that $\tau \neq \lambda$. Projecting once more onto Σ_s gives

$$\hat{\beta}_\tau := \mathcal{P}_{\Sigma_s}(\mathcal{P}_\tau(w)).$$

Algorithm 1: Greedy selector and simplex projector (GSSP)

1: $\mathcal{S}^\star = \operatorname{supp}(\mathcal{P}_{\Sigma_s}(w))$ {Select support}
2: $(\beta)_{\mathcal{S}^\star} = \mathcal{P}_\lambda(w_{\mathcal{S}^\star}),\ (\beta)_{\mathcal{S}^{\star,c}} = 0$ {Final projection}

Let $C(\tau) := C(\hat{\beta}_\tau)$. If $C(\tau) = \lambda$ then $\hat{\beta}_\tau$ is feasible.

In §4.2, we will prove that if τ is chosen such that $C(\tau) = \lambda$, then remarkably $\operatorname{supp}(\hat{\beta}_\tau) = \operatorname{supp}(\beta^\star)$ for some optimal solution $\beta^\star$. Finding this correct value of τ can be efficiently done with a bisection search, leading to an algorithm with $\mathcal{O}(ps \log s)$ complexity. When there is no τ such that $C(\tau) = \lambda$, the algorithm still provides an estimate which has a provable bound on its sub-optimality. Due to overwhelming numerical evidence, we also conjecture that $\operatorname{supp}(\hat{\beta}_\tau)$ is optimal in this case as well: in low-dimensional tests, the algorithm has always returned the exact global solution.

Algorithm 2: Relaxed selector and simplex projector (RSSP)

1: $t^\star = \arg\min_t |\lambda - C(\hat{\beta}_t)|$ {One-dimensional optimization using relaxed constraints}
2: $\mathcal{S}^\star = \operatorname{supp}(\hat{\beta}_{t^\star})$ {Select support}
3: $(\beta)_{\mathcal{S}^\star} = \mathcal{P}_\lambda(w_{\mathcal{S}^\star}),\ (\beta)_{\mathcal{S}^{\star,c}} = 0$ {Final projection}

4 Main results

Remark 3. *When the symbol $\mathcal{S}$ is used as $\mathcal{S} = \operatorname{supp}(\bar{\beta})$ where $\bar{\beta} = \mathcal{P}_{\Sigma_s}(\bar{w})$ for any $\bar{w}$, then if $|\mathcal{S}| < s$, we enlarge $\mathcal{S}$ until it has s elements by taking the first $s - |\mathcal{S}|$ elements that are not already in $\mathcal{S}$ and defining $\bar{\beta} = 0$ on these elements. The lexicographic approach is used to break ties when there are more than one solution to Problem 1.*

4.1 Projection onto the standard simplex with cardinality constraints

Theorem 1. *Algorithm 1 exactly solves Problem 1.*

Proof. By Remark 2, we split the problem into the task of finding the support and then finding the values on the support. Given any support $\mathcal{S}$, the unique corresponding estimator is $\hat{\beta}_{\mathcal{S}} = \mathcal{P}_\lambda(w_{\mathcal{S}})$.

We may conclude that $\beta^\star$ satisfies $(\beta^\star)_{\mathcal{S}^\star} = (\mathcal{P}_\lambda(w))_{\mathcal{S}^\star}$ and $(\beta^\star)_{(\mathcal{S}^\star)^c} = 0$, where

$$\begin{aligned}
\mathcal{S}^\star \in \operatorname*{arg\,min}_{\mathcal{S}:\mathcal{S}\in\Sigma_s} \|(\mathcal{P}_\lambda(w))_{\mathcal{S}} - w\|_2 &= \operatorname*{arg\,max}_{\mathcal{S}:\mathcal{S}\in\Sigma_s} \left[\|(w)_{\mathcal{S}}\|_2^2 - \|(\mathcal{P}_\lambda(w) - w)_{\mathcal{S}}\|_2^2\right] \\
&= \operatorname*{arg\,max}_{\mathcal{S}:\mathcal{S}\in\Sigma_s} \sum_{i\in\mathcal{S}} \left(w_i^2 - ((\mathcal{P}_\lambda(w))_i - w_i)^2\right) \\
&= \operatorname*{arg\,max}_{\mathcal{S}:\mathcal{S}\in\Sigma_s} F(\mathcal{S}) \qquad (5)
\end{aligned}$$

with $F(\mathcal{S}) := \sum_{i\in\mathcal{S}} \left(w_i^2 - ((\mathcal{P}_\lambda(w))_i - w_i)^2\right)$. We now introduce a simple yet powerful lemma.

Lemma 1. *Let $\beta = \mathcal{P}_\lambda(w)$ where $\beta_i = [w_i - \tau]_+$. Then $w_i \geq \tau$ for all $i \in \mathcal{S} = \operatorname{supp}(\beta)$. Furthermore, $\tau = \frac{1}{|\mathcal{S}|}\left(\sum_{i\in\mathcal{S}} w_i - \lambda\right)$.*

Proof. Directly from the definition of τ_j in (2). The intuition is quite simple: the "threshold" τ should be smaller than the smallest entry in the selected support. Otherwise, we unnecessarily shrink the coefficients that are larger without introducing any new support to the solution. □

Let $\mathcal{S}$ be any support set, and let $|\mathcal{S}| = l \leq s$. If $\mathcal{S}$ is not chosen greedily, then there exists some $w_k \notin \mathcal{S}$ and some $w_j \in \mathcal{S}$ such that $w_k > w_j$. Define $\hat{\mathcal{S}} = (\mathcal{S} \setminus \{j\}) \cup \{k\}$. We will show that $F(\hat{\mathcal{S}}) \geq F(\mathcal{S})$, and therefore it is never harmful to keep adding the largest elements of w, and hence the greedy set is at least as good as all other possible sets. For convenience, define $r = \sum_{i \in \mathcal{S}} w_i$ and $\hat{r} = \sum_{i \in \hat{\mathcal{S}}} w_i$. First, we note that

$$F(\mathcal{S}) = \sum_{i \in \mathcal{S}} (w_i^2 - \tau^2) = \left(\sum_{i \in \mathcal{S}} w_i^2 \right) - \frac{1}{l}(r - \lambda)^2$$

which follows from Lemma 1. Now,

$$\begin{aligned} F(\hat{\mathcal{S}}) - F(\mathcal{S}) &= (w_k^2 - w_j^2) - \frac{1}{l}\left((\hat{r} - \lambda)^2 - (r - \lambda)^2 \right) \\ &= (w_k - w_j)(w_k + w_j) - \frac{1}{l}(\hat{r} - r)(\hat{r} + r - 2\lambda) \\ &\geq (w_k - w_j)\left[(w_k - (\hat{r} - \lambda)/l) + (w_j - (r - \lambda)/l) \right] \\ &\geq 0 \end{aligned}$$

since $w_k - w_j > 0$ and $w_k \geq (\hat{r} - \lambda)/l$ and $w_j \geq (r - \lambda)/l$ by Lemma 1. □

Remark 4. *The same algorithm works for projecting onto the convex hull of the standard simplex, since forcing $\tau \geq 0$ does not change the proof.*

4.2 Projection onto the general simplex with cardinality constraints

We are now in position to specialize Algorithm 2 to project onto $\Delta_\lambda \cap \Sigma_s$.

Theorem 2. *For all λ except on a set Λ of finite measure, Algorithm 2 computes the projection of w onto $\Delta_\lambda \cap \Sigma_s$ exactly. If $\lambda \in \Lambda$, the algorithm returns a solution with online optimality guarantees.*

Empirically, the algorithm is exact even when $\lambda \in \Lambda$, but we defer this proof for later work. The rest of this subsection proves the theorem.

Consider the nested form of the problem, as in (4). Given a support set $\mathcal{S}$, the inner minimization in (4) has an analytical solution as a function of β using (3):

$$\beta_{\mathcal{S}} = w_{\mathcal{S}} + \tau(\mathcal{S}) \cdot \mathbb{1}_{\mathcal{S}}, \tag{6}$$

with $\tau(\mathcal{S}) := \frac{(\lambda - \mathbb{1}_{\mathcal{S}}^T w_{\mathcal{S}})}{|\mathcal{S}|} \in \mathbb{R}$. We sometimes write τ when the set $\mathcal{S}$ is clear from context. Replacing (6) in (4), we obtain the following subset selection problem:

$$\begin{aligned} \mathcal{S}^\star \in \operatorname*{arg\,min}_{\mathcal{S}:|\mathcal{S}| \leq s} \left[\|\tau \cdot \mathbb{1}_{\mathcal{S}}\|_2^2 + \|(w)_{\mathcal{S}^c}\|_2^2 \right] &= \operatorname*{arg\,max}_{\mathcal{S}:|\mathcal{S}| \leq s} \left[\|w\|_2^2 - \left(\|\tau \cdot \mathbb{1}_{\mathcal{S}}\|_2^2 + \|(w)_{\mathcal{S}^c}\|_2^2 \right) \right] \\ &= \operatorname*{arg\,max}_{\mathcal{S}:|\mathcal{S}| \leq s} \sum_{i \in \mathcal{S}} \left(w_i^2 - \tau^2 \right). \end{aligned} \tag{7}$$

Write $F(\mathcal{S}) = \sum_{i \in \mathcal{S}} \left(w_i^2 - \tau^2 \right)$, and thus we seek $\mathcal{S} \in \Sigma_s$ that maximizes $F(\mathcal{S})$.

Write $\beta_\tau = w + \tau$, so that $\beta_\tau = \mathcal{P}_{\mathcal{K}_{\lambda_\tau}}(w)$ for some λ_τ, depending on τ. Then, let $\widehat{\beta}_\tau = \mathcal{P}_{\Sigma_s}(\beta_\tau)$ be the s-sparse projection of β_τ, and define $C(\tau) = \mathbb{1}^T \widehat{\beta}_\tau$. Algorithm 2 suggests that finding the minimizer of $|C(\tau) - \lambda|$ will define the correct support. We start with a lemma about $C(\tau)$.

Lemma 2. *The map $C(\tau)$ is monotonically increasing in τ, and is piecewise linear apart from s "turning points" $\tau_k,\ k = 1, \ldots, s$, where it is multi-valued.*

Proof. Fix τ, and let $\sigma(j)$ be a permutation such that $|w_{\sigma(1)}+\tau| \geq |w_{\sigma(2)}+\tau| \geq \ldots \geq |w_{\sigma(n)}+\tau|$. Let $\mathcal{S} = \{\sigma(1),\ldots,\sigma(s)\}$, so $\mathcal{P}_{\mathcal{S}}(\beta_\tau)$ is a valid projection of β_τ onto Σ_s. Thus a value for $C(\tau)$ is $C(\tau) = \sum_{i\in\mathcal{S}}(w_i+\tau)$. The set $\mathcal{S}$ need not be unique; if $w_{\sigma(s)} = w_{\sigma(s+1)}$, then there there are other valid $\mathcal{S}$ but it will result in the same value of $C(\tau)$, and if $w_{\sigma(s)} \neq w_{\sigma(s+1)}$ but $|w_{\sigma(s)}+\tau| = |w_{\sigma(s+1)}+\tau|$ then there is another valid $\mathcal{S}$ that gives a different value of $C(\tau)$. This latter case only happens at s-many points which we call "turning points."

Now examine $C(\tau+\Delta\tau)$ for any $\Delta\tau > 0$. If there is no turning point $\tau_k \in [\tau, \tau+\Delta\tau]$, then $C(\tau+\Delta\tau) - C(\tau) = s\Delta\tau$. If there is a turning point, assume $\Delta\tau$ is sufficiently small so that only one turning point, τ_k, is in $[\tau, \tau+\Delta\tau]$. Let $\mathcal{S}$ be the index set corresponding to τ; then the index set for $\tau+\Delta\tau$ is the same, except it has an additional index $w_{(\text{new})}$ and it lacks some index $w_{(\text{old})}$. Then $C(\tau+\Delta\tau) - C(\tau) = s\Delta\tau + w_{(\text{new})} - w_{(\text{old})}$. Since $w_{(\text{new})}$ has been added, this means $|w_{(\text{new})}+\tau+\Delta\tau| > |w_{(\text{new})}+\tau|$ and thus $w_{(\text{new})} > 0$. Likewise, $w_{(\text{old})} < 0$, and hence $C(\tau+\Delta\tau) - C(\tau) > 0$, so C is a monotonically increasing function. □

The s turning points are easily found by sorting w (so assume $w_1 \geq w_2 \geq \ldots$) and finding the points τ such that $|w_k+\tau| = |w_{p-s+k}+\tau|$ for $k = 1,\ldots,s$, i.e. $\tau_k = (w_{(p-s+k)}^2 - w_k^2)/(2(w_k - w_{p-s+k}))$.

Now, we approach the problem from the combinatorial side. Let $\widehat{\mathcal{S}}$ be the unique support defined by any $\hat{\tau}(\widehat{\mathcal{S}})$ that is not a turning point (for simplicity, we exclude the possibility that w has identical entries). The support set does not change as $\hat{\tau}(\widehat{\mathcal{S}})$ changes except when $\hat{\tau}(\widehat{\mathcal{S}})$ crosses a turning point. If $\widehat{\mathcal{S}}$ is fixed, then consider projecting $w_{\widehat{\mathcal{S}}}$ onto $\Delta_{\lambda_{\hat{\tau}(\widehat{\mathcal{S}})}}$ where $\lambda_{\hat{\tau}(\widehat{\mathcal{S}})} = s\hat{\tau}(\widehat{\mathcal{S}}) + \mathbb{1}_{\widehat{\mathcal{S}}}^T w_{\widehat{\mathcal{S}}}$. Here, we use Remark 3, where $|\widehat{\mathcal{S}}| = s$. The nonzero entries of β over $\widehat{\mathcal{S}}$ are given by (6) for

$$\hat{\tau}(\widehat{\mathcal{S}}) = \frac{\lambda_{\hat{\tau}(\widehat{\mathcal{S}})} - \mathbb{1}_{\widehat{\mathcal{S}}}^T w_{\widehat{\mathcal{S}}}}{s}.$$

The algorithm proceeds by picking $\tau \in (\tau_k, \tau_{k+1})$, with the convention that $\tau_0 = -\infty$ and $\tau_{s+1} = +\infty$, which defines a support $\mathcal{S}_\tau$ and hence $\lambda_\tau := C(\tau)$. We search for $\arg\min_\tau |\lambda - \lambda_\tau|$. Because $C(\tau)$ is monotonic, we can compute bounds on λ_τ using the bisection method. Let τ_l and τ_u be values of τ such that $\lambda_l = \lambda_{\tau_l} < \lambda < \lambda_{\tau_u} = \lambda_u$.

Using (7), the optimal s-sparse set can be found by maximizing $F(\mathcal{S})$. The final step in our proof shows that $\lambda_\tau \simeq \lambda$ implies $F(\mathcal{S}_\tau) \simeq F(\mathcal{S}^\star)$.

Lemma 3. *If $C(\tau) := C(\hat{\beta}_\tau) = \lambda$, then $\hat{\beta}_\tau = \beta^\star$.*

Proof. Since $\hat{\beta}_\tau$ is defined as a projection of $w+\tau$, we have

$$\begin{aligned}
\hat{\beta}_\tau &= \operatorname*{arg\,min}_{\beta\in\Sigma_s} \|\beta - (w+\tau\mathbb{1})\|_2^2 \\
&= \operatorname*{arg\,min}_{\beta\in\Sigma_s, C(\beta)=\lambda_\tau} \|\beta - (w+\tau\mathbb{1})\|_2^2 \quad \text{since } C(\hat{\beta}_\tau) = \lambda_\tau \\
&= \operatorname*{arg\,min}_{\beta\in\Sigma_s, C(\beta)=\lambda_\tau} \|\beta - w\|_2^2 + \underbrace{\|\tau\mathbb{1}\|^2 - 2\tau\mathbb{1}^T(\beta - w)}_{c}.
\end{aligned}$$

Because of the constraint $C(\beta) = \lambda_\tau$, then $\mathbb{1}^T\beta$, and hence c, is a constant over the feasible set, and so this is the same as Problem 2 using λ_τ. Thus if $\lambda_\tau = \lambda$, we have solved the original problem. □

Lemma 4. *Let $\widehat{\mathcal{S}} = \operatorname{supp}(\hat{\beta}_{t^\star})$ be the solution index set of $t^\star = \arg\min_\tau |\lambda - \lambda_\tau|$, and $\widehat{\lambda} = \lambda_\tau$. Define $\epsilon_{\widehat{\mathcal{S}}} = G(\lambda,\widehat{\lambda})/F(\widehat{\mathcal{S}})$ and $G(\lambda,\widehat{\lambda}) = \frac{\lambda^2-\widehat{\lambda}^2}{s} + \frac{2(\widehat{\lambda}-\lambda)}{s}\sum_{i\in\mathcal{S}^\star} w_i$. Then, we have the following approximation guarantee*

$$F(\widehat{\mathcal{S}}) \geq \left(1 - \epsilon_{\widehat{\mathcal{S}}}\right) F(\mathcal{S}^\star). \tag{8}$$

Proof. We observe the following:

$$
\begin{aligned}
F(\widehat{\mathcal{S}}, w) &= \sum_{i \in \widehat{\mathcal{S}}} [w]_i^2 - \frac{1}{s} \left(\widehat{\lambda} - \sum_{i \in \mathcal{S}_l} [w]_i \right)^2 \\
&\geq \sum_{i \in \mathcal{S}^\star} [w]_i^2 - \frac{1}{s} \left(\widehat{\lambda} - \sum_{i \in \mathcal{S}^\star} [w]_i \right)^2 \\
&= \sum_{i \in \mathcal{S}^\star} [w]_i^2 - \frac{1}{s} \left(\widehat{\lambda} + \lambda - \lambda - \sum_{i \in \mathcal{S}^\star} [w]_i \right)^2 \\
&= F(\mathcal{S}^\star, w) - \frac{2\lambda(\widehat{\lambda} - \lambda)}{s} + \frac{2(\widehat{\lambda} - \lambda)}{s} \sum_{i \in \mathcal{S}^\star} [w]_i - \frac{(\widehat{\lambda} - \lambda)^2}{s} \\
&= F(\mathcal{S}^\star, w) + \frac{\lambda^2 - \widehat{\lambda}^2}{s} + \frac{2(\widehat{\lambda} - \lambda)}{s} \sum_{i \in \mathcal{S}^\star} [w]_i,
\end{aligned}
$$

where we simply use the fact that given $\widehat{\lambda}$, the support $\widehat{\mathcal{S}}$ obtains the maximum objective. □

We obtain $\epsilon_{\widehat{\mathcal{S}}}$ values to be much smaller than 1 after the termination of RSSP (e.g., typical values for the simplex projection experiments in Section 5.3 vary between 10^{-4} to 10^{-7}). Given this epsilon value for RSSP, it is possible to obtain an online approximation guarantee for iterative optimization algorithms that can use RSSP (cf., [12]).

Remark 5. *The time complexity of the algorithm is $\mathcal{O}(ps \log_2(s))$ because we need only test τ that define unique support, and there are $s+1$ of these. Because C is monotonic, we can test just $\log_2(s+1)$ supports by using the bisection method. For each support, we compute a projection onto Σ_s which takes $\mathcal{O}(ps)$ complexity.*

Remark 6. *We conjecture that it is possible to relax the constraint from $\sum \beta_i = \lambda$ (which is non-convex) to $\sum \beta_i \leq \lambda$ (which is convex), but we leave a proof for future work.*

5 Experiments

5.1 Portfolio optimization

Given a sample covariance matrix Σ and mean μ, the Markowitz mean-variance (MV) framework selects a portfolio $\beta^\star = \mathbb{R}^p$ (without short positions) via $\beta^\star \in \arg\min_{\beta \in \Delta_1^+} \left[\beta^T \Sigma \beta - \alpha \mu^T \beta \right]$, where Δ_1^+ encodes the normalized capital constraint, and α trades off risk and return [7, 8]. The solution $\beta^\star \in \Delta_1^+$ indicates the fractional investments over the p available assets. In mathematical terms, this leads to the following quadratic optimization problem:[1]

$$
\beta^\star = \underset{\beta \in \Delta_1^+}{\arg\min} \left[\beta^T \Sigma \beta - \alpha \mu^T \beta \right], \tag{9}
$$

for a given regularization parameter $\alpha \geq 0$.

In practice, the preferences of the investor may lead to further constraints in the optimization problem. Additional fees for asset trading (transaction costs) and costs of monitoring and portfolio re-weighting naturally lead to cardinality constraints in the optimization procedure. This additional flavor leads to mixed integer quadratic programming formulation which is difficult to solve by standard optimization techniques. Numerous approaches have been proposed in the literature to solve this problem [18, 19] and references therein. Most of the investigations on cardinality

[1]There is an extensive list of works in the literature where the mean-variance efficient portfolios are compared with global minimum variance portfolios. The primary reason is that estimating the expected return from past data is a difficult task where large outliers lead to imprecise estimations. We refer the reader to [7, 17] for an excellent discussion.

constrained porfolio optimization focus on finding local solutions using greedy techniques, simulated annealing and evolution methods, genetic algorithms, and branch and bound ideas ([18, 20, 21] and references therein).

Here, we are interested in the MV optimization with the added twist that the solution satisfies $\beta^\star \in \Sigma_s$:

$$\beta^\star \in \underset{\beta \in \Delta_1^+ \cap \Sigma_s}{\arg\min} \left[\beta^T \Sigma \beta - \alpha \mu^T \beta\right], \tag{10}$$

for a given level of sparsity s.

Out-of-sample performance: We use a publicly available dataset compiled by Farma and French[2]. In this dataset, we monitor 49 diverse industry assets and consider both monthly and daily recordings.

Procedure: We evaluate the out-of-sample performances of the estimated portfolios over various time periods. For instance, during each year from 1971 to 2011, we estimate expected monthly returns of the stocks and their covariance values using the available data from the preceding 5 years. Finally, we evaluate the estimated portfolio $\beta^\star$ by computing the monthly returns and risks each year keeping $\beta^\star$ fixed.

We compare the following approaches: (i) the constrained quadratic optimization as described in (10) without the cardinality constraint Σ_s using `quadprog` in MATLAB [22], (ii) the cardinality-constrained projected gradient descent algorithm that solves (10) using GSSP for $s = \{4, 10\}$ and, (iii) the naive $1/p$-strategy where we use the same weight over the portfolio, i.e. $\beta_i = 1/p$ for all i.

Results: We provide some typical return evaluations with $\alpha = 1$ in Table 1 (as α varies, the results qualitatively remain the same). Our approach with GSSP performs quite well, especially for smaller active portfolio sizes as constrained by s. We observe that as s decreases, the expected return $\hat{\mu}$ as well as the standard deviation $\hat{\sigma}$ of the returns increase. Surprisingly, the GSSP solutions exhibit competitive Sharpe ratios $\hat{\mu}/\hat{\sigma}$, which measures the risk adjusted return, as compared to the MV portfolio, and with much lower transactions costs. Overall, the quadratic programming approach has a median sparsity level of 14 and a mean sparsity level of 14.78. The $1/p$ baseline strategy has the worst returns and worst Sharpe ratios for most years.

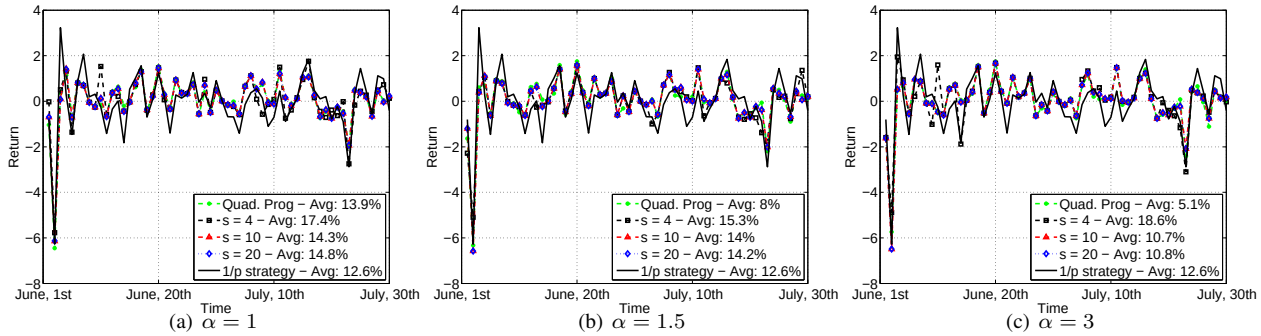
Interestingly, the naive baseline strategy does well in recession years like '81 to '86 and '06 to '11. In these years, presumably the model (Σ, μ) is less accurate, and hence the quadratic programming solution does much worse than the naive strategy. The sparsity-constrained solution does better than the quadratic programming solution, suggesting that sparsity helps buffer against inaccurate models.

Period	Quad. Prog.			GSSP, $s = 4$			GSSP, $s = 10$			$1/p$-strategy		
	$\hat{\mu}$ (%)	$\hat{\sigma}$	$\hat{\mu}/\hat{\sigma}$	$\hat{\mu}$ (%)	$\hat{\sigma}$	$\hat{\mu}/\hat{\sigma}$	$\hat{\mu}$ (%)	$\hat{\sigma}$	$\hat{\mu}/\hat{\sigma}$	μ (%)	$\hat{\sigma}$	$\hat{\mu}/\hat{\sigma}$
'71 - '11	11.82	0.423	**0.28**	**13.19**	0.489	0.27	11.78	0.418	**0.28**	10.1	0.49	0.19
'71 - '76	15.14	0.445	**0.39**	**15.58**	0.468	0.27	15.07	0.447	0.33	1.61	0.552	0.03
'76 - '81	7.61	0.473	**0.16**	**8.05**	0.543	0.15	7.83	0.467	**0.16**	2.49	0.608	0.04
'81 - '86	7.64	0.377	0.20	8.80	0.421	0.21	7.68	0.369	0.21	**11.48**	0.513	**0.22**
'86 - '91	21.25	0.449	0.47	**26.67**	0.538	0.42	25.13	0.475	0.53	20.93	0.477	**0.55**
'91 - '96	**11.15**	0.440	**0.24**	10.71	0.513	0.22	10.7	0.453	0.23	8.41	0.562	0.15
'96 - '01	19.42	0.329	**0.59**	**20.22**	0.402	0.48	17.85	0.321	0.55	14.22	0.274	0.52
'01 - '06	5.53	0.430	0.12	**6.71**	0.518	0.12	3.50	0.443	0.07	6.46	0.470	**0.14**
'06 - '11	6.42	0.329	0.19	8.83	0.329	0.26	6.36	0.315	0.2	**11.65**	0.356	**0.32**

Table 1: *Portfolio performance evaluation with no-short positions for $\alpha = 1$. In the table, $\hat{\mu}$ denotes the average monthly returns, $\hat{\sigma}$ is the standard deviation of the returns and $\hat{\mu}/\hat{\sigma}$ is the Sharpe ratio. Numbers in red are the best for that row among all methods.*

Efficient frontier with cardinality constraints: To provide further numerical evidence, we generate random expected returns and covariance quantities of $p = 100$ assets. In Figure 2, we depict the *unconstrained* Pareto frontier by solving the optimization problem (9). Without any cardinality constraints, the MV framework suggests dense portfolio solutions for low risk investments (additional selections lower the risk) while sparser solutions can be obtained

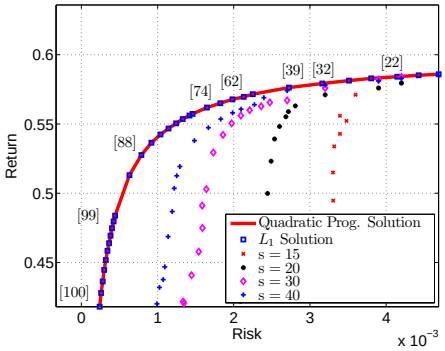
[2] `http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`



(a) $\alpha = 1$ (b) $\alpha = 1.5$ (c) $\alpha = 3$

Figure 1: *Evaluation of out-of-sample portfolios on a daily basis.*

for riskier investments. In practice, dense portfolios are difficult to administrate and have higher transactions costs. To this end, we propose sparser portfolio strategies using a projected gradient descent solver for (10) where we use GSSP. The corresponding frontiers are depicted in Figure 2 for various s.



p	s	α	Obj.	$\|\beta^\star\|_0$	Time (sec)
50	10	0.01	0.0027/0.0035	10/10	4.3/**0.29**
50	20	1	0.2807/0.2814	11/12	1.25/**0.03**
50	30	1	0.2817/0.283	12/12	1.29/**0.04**
70	10	1	0.2855/0.286	10/10	1.71/**0.05**
70	20	1	0.2866/0.286	12/12	1.72/**0.05**
100	20	0.01	0.054/0.055	20/20	0.62/**0.38**
100	40	1	0.2874/0.2874	13/12	1.8/**0.06**

Figure 2: *Left: Pareto efficient frontier for $p = 100$ assets over a range of α values. We perform 500 Monte-Carlo iterations and depict the median values. Red solid curve denotes the quadratic programming solution as obtained by (9) and blue squares represent a variation of ℓ_1-norm regularized solver in [8]. We present the solutions of our approach in (10) for $s = 15, 20, 30, 40$. Right: additional numerical comparisons using synthetic data. The table depicts the median values over 100 Monte-Carlo realizations.*

Additional comparisons: We perform a comparative study between the commercial toolbox CPLEX Studio V12.3 [14] using YALMIP [23] and our positive simplex, cardinality constrained, projected gradient method. The table in Figure 2 depicts a summary of the results. Even within this relatively small-scale configuration, our projected gradient descent algorithm admits fast performance (due to the first-order gradient statistics) and almost equivalent estimation performance compared to the state-of-the-art CPLEX algorithm.

5.2 Sparse kernel density estimation

Let $x^{(1)}, x^{(2)}, \ldots, x^{(n)} \in \mathbb{R}^p$ be a n-size corpus of p-dimensional samples, drawn from an unknown probability density function (pdf) $f(x)$. Given this dataset, we are interested in estimating $f(x)$. We focus on kernel estimation techniques that employ the Integrated Squared Error (ISE) criterion, ISE $= E\|\hat{f}(x) - f(x)\|_2^2$, to design a weight vector $\beta^\star \in \Delta_1^+$ such that $\hat{f}(x) := \sum_{i=1}^n \beta_i^\star \kappa_\sigma(x, x^{(i)})$ and $\hat{f}(x)$ minimizes the ISE. The resulting problem can be written as follows [10, 24]

$$\beta^\star \in \underset{\beta \in \Delta_1^+}{\arg\min} \left[\beta^T \Sigma \beta - c^T \beta\right], \tag{11}$$

where $\kappa_\sigma(x, y)$ is a Gaussian kernel, $\Sigma \in \mathbb{R}^{n\times n}$ with $\Sigma_{ij} = \kappa_{\sqrt{2}\sigma}(x^{(i)}, x^{(j)})$ and $c \in \mathbf{R}^{n\times 1}$ with

$$c_i = \frac{1}{n-1}\sum_{j\neq i} \kappa_\sigma(x^{(i)}, x^{(j)}),\ \forall i, j \tag{12}$$

While the term $-c^T\beta$ induces sparsity in the solution for $\beta_i \geq 0, \forall i$, in several cases $\beta^\star$ is not sparse enough even if $f(x)$ is a linear combination of a few components; even sparser weight solutions are required for instance to avoid overfitting or obtain interpretable results. In this context, we extend (11) to include cardinality constraints, i.e. $\beta^\star \in \Delta_1^+ \cap \Sigma_s$.

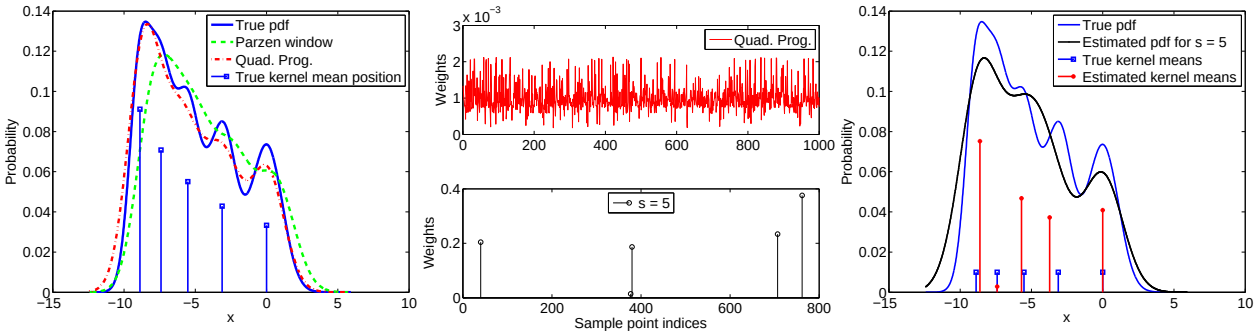
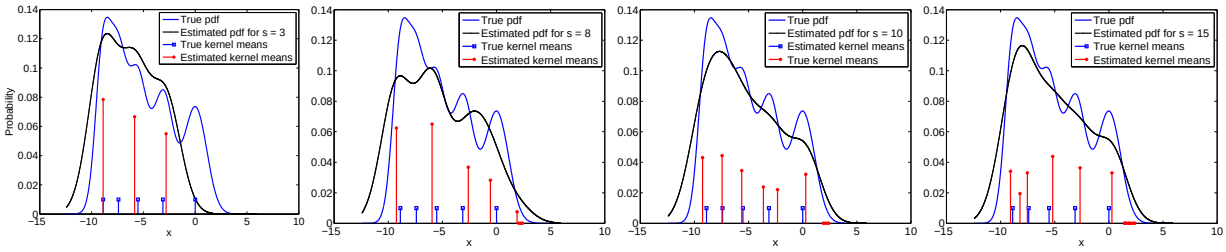


Figure 3: *Density estimation results using the Parzen window method, the quadratic programming in* (11) *and our approach for* $s = 5$.



(a) $s = 3$ (b) $s = 8$ (c) $s = 10$ (d) $s = 15$

Figure 4: *Density estimation results for various sparsity levels* s. *Red spikes depict the estimated kernel means as well as the their relative contribution to the Gaussian mixture. As* s *increases, the additional nonzero coefficients in* $\beta^\star$ *tend to have small weights.*

Sparse estimation of Gaussian mixtures: We consider the following one-dimensional mixture of 5 Gaussians: $f(x) = \frac{1}{5}\sum_{i=1}^{5} \kappa_{\sigma_i}(\mu_i, x)$ where $\sigma_i = (7/9)^i$ and $\mu_i = 14(\sigma_i - 1)$. A sample of 1000 points is drawn from $f(x)$. We compare the density estimation performance of: (i) the Parzen window method [25], (ii) the quadratic programming formulation in (11), and (iii) our cardinality-constrained version of (11) using GSSP. While $f(x)$ is constructed by kernels with various widths, we assume a constant width during the kernel estimation. In practice, the width is not known *a priori* but can be found using cross-validation techniques, i.e., leave-one-out technique; for simplicity, we assume kernels with width $\sigma = 1$.

Figure 3(a) depicts the true pdf and the estimated densities using the Parzen window method and the quadratic programming approach. Moreover, the figure also include a scaled plot of $1/\sigma_i$, indicating the height of the individual Gaussian mixtures. By default, the Parzen window method estimation interpolates 1000 Gaussian kernels with centers around the sampled points to compute the estimate $\hat{f}(x)$; unfortunately, neither the quadratic programming approach (as Figure 3(b) illustrates) nor the Parzen window estimator results are easily *interpretable* while both approaches provide a good approximation of the true pdf.

In stark contrast, using our cardinality-constrained approach, we can significantly enhance the interpretability. This is because the sparsity-constrained approach approximately reveals the number of Gaussian components. To see this,

we first show the coefficient profile of the proposed approach for $s = 5$ in Figure 3(b) (middle-bottom). Moreover, Figure 3(c) shows the estimated pdf for $s = 5$ along with the positions of weight coefficients obtained by our sparsity enforcing approach. Note that most of the weights obtained concentrate around the true means, providing some prior information about the ingredients of $f(x)$—this happens with rather high frequency in the experiments we conducted. Figure 4 illustrates further estimated pdf's using our approach for various s. From the plot, it is also clear that if $s > 5$, the resulting solutions are still approximately 5-sparse as the over estimated coefficients exhibit extremely small values.

p	s	Error $\|\beta^\star - w\|_2$ GSSP / PAMP / CPLEX / RSSP	Time (sec) GSSP / PAMP / CPLEX / RSSP
20	4	$2.96 / 2.86 / \mathbf{2.81} / \mathbf{2.81}$	$9.65 \cdot 10^{-5} / 0.006 / 1.0 / 0.04$
30	5	$3.63 / 3.50 / \mathbf{3.42} / \mathbf{3.42}$	$8.8 \cdot 10^{-5} / 0.007 / 39.26 / 0.04$
10^4	10	$99.43 / 99.37 / - / \mathbf{99.37}$	$8.16 \cdot 10^{-4} / 1.11 / - / 0.09$
10^4	100	$95.75 / 95.74 / - / \mathbf{95.68}$	$8.07 \cdot 10^{-4} / 1.13 / - / 0.09$
10^4	300	$89.74 / 89.74 / - / \mathbf{89.71}$	$8.16 \cdot 10^{-4} / 0.41 / - / 0.09$
10^4	30	$97.59 / 97.59 / - / \mathbf{97.54}$	$8.11 \cdot 10^{-4} / 1.12 / - / 0.09$
10^4	100	$95.75 / 95.74 / - / \mathbf{95.68}$	$8.03 \cdot 10^{-4} / 1.12 / - / 0.09$
10^4	200	$92.52 / 92.51 / - / \mathbf{92.49}$	$8.14 \cdot 10^{-4} / 0.60 / - / 0.09$

Table 2: *The table depicts median values over* 20 *Monte-Carlo iterations for* $p \in \{20, 30\}$ *(also shown in the figure on the right for* $p = 20$*) and* 100 *Monte-Carlo iterations for* $p = 10^4$*. Results are listed as GSSP / Proximal Alternating projection / CPLEX / RSSP. Bold numbers highlight the best projection. "–" denotes that the corresponding algorithm is not applicable due to slow convergence.*

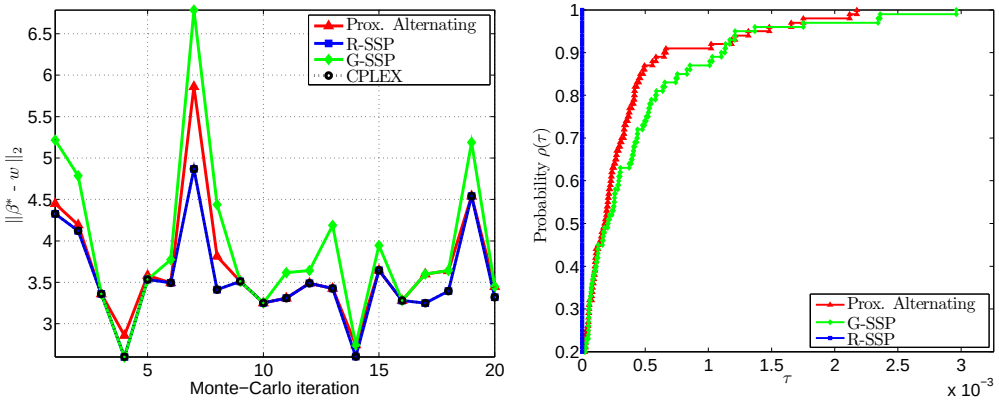


Figure 5: *Left:* $p = 30,\ s = 5,\ \lambda = 2$. *Right: Performance evaluation plot for* $p = 10^4,\ s = 300,\ \lambda = 10$.

5.3 Sparse projections onto the general simplex

While the general simplex constraints also have applications with Markowitz portfolio selection, it is better to use synthetic data to numerically benchmark the quality and efficiency of our projection onto the general simplex Δ_λ.

Here, we generate random vector realizations $w \in \mathbb{R}^p$ with $p = \{10^2, 10^4\}$ and compare the projection performance of the following approaches over various s and λ: (i) the GSSP on $\Sigma_s \cap \Delta_\lambda$, (ii) the proximal alternating minimization projection (PAMP) [26], (iii) the commercial mixed integer quadratic programming toolbox CPLEX Studio V12.3 [14], and (iv) the RSSP on $\Sigma_s \cap \Delta_\lambda$. The PAMP approach is based on a general proximal scheme that tries to solve a coupled problem and requires turning parameters. For a fair comparison, we tuned PAMP parameters for best performance; we note that deviations from this fine tuning leads to performance degradation.

Table 2 depicts the efficiency of our approaches where both GSSP and RSSP efficiently decrease the distance to the original vector w compared to state-of-the-art algorithms with total computational costs that scale well as p increases. Note that while RSSP requires more computation, its solution quality is consistently (and strictly, in our experiments) better, where the difference can be quite large. Some illustrative instance is depicted in Figure 5. In Figure 5(b), let $\beta_i^\star,\ i \in \mathcal{I} = \{\text{GSSP}, \text{RSSP}, \text{Prox.Alter.}\}$ be the solutions obtained by the algorithms in comparison and let $e_{i,j}$ denote the distance $\|\beta_i^\star - w\|_2,\ i \in \mathcal{I}$ at the j-th MC iteration. Then, we use the performance profile notion [27] where $\rho(\alpha)$ is the probability for a solver that the performance ratio $\frac{e_{i,j}}{\min\{e_{i,j}: i \in \mathcal{I}\}}$ is within a factor $\alpha \in \mathbb{R}$ of the best possible ratio. Note that RSSP performs exactly the same as CPLEX within numerical precision for small size problems while being three orders of magnitude faster. The PAMP approach is quite competitive but it requires some level of tuning.

6 Conclusions

While non-convexity in learning algorithms is undesirable according to conventional wisdom, avoiding it might be difficult in many problems. In particular, sparse learning problems with simplex constraints disallow the application of the ℓ_1-relaxation for sparsification of the solution. In this setting, we show how to efficiently obtain sparse projections onto positive and general simplex and sparsity constraints. To this end, we provide an exact sparse projector for the positive simplex constraints and a relaxation based approach for approximate sparse projections onto the general simplex constraints. Interestingly, the latter approach can feature online optimality guarantees. We also empirically demonstrate that our projectors provide substantial benefits in generalization and interpretability in sparse portfolio selection and density estimation applications.

Acknowledgments

This work was supported in part by the European Commission under Grant MIRG-268398, ERC Future Proof, SNF 200021_132548, and DARPA KeCoM program #11-DARPA-1055. VC also would like to acknowledge Rice University for his Faculty Fellowship, and SB would like to acknowledge the Fondation Sciences Mathématiques de Paris for his fellowship.

References

[1] Y. Freund and R. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[2] P. Bühlmann and S. Van De Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer-Verlag New York Inc, 2011.

[3] B.K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, 1995.

[4] H. Attouch, J. Bolte, and B.F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss-Seidel methods. *Mathematical Programming*, pages 1–39, 2011.

[5] P. Brucker. An O(n) algorithm for quadratic knapsack problems. *Operations Research Letters*, 3(3):163–166, 1984.

[6] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *ICML*, 2008.

[7] V. DeMiguel, L. Garlappi, F.J. Nogales, and R. Uppal. A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management Science*, 55(5):798–812, 2009.

[8] J. Brodie, I. Daubechies, C. De Mol, D. Giannone, and I. Loris. Sparse and stable Markowitz portfolios. *Proceedings of the National Academy of Sciences*, 106(30):12267–12272, 2009.

[9] R.M. Willett and R.D. Nowak. Multiscale poisson intensity and density estimation. *Information Theory, IEEE Transactions on*, 53(9):3171–3187, 2007.

[10] F. Bunea, A.B. Tsybakov, M.H. Wegkamp, and A. Barbu. SPADES and mixture models. *The Annals of Statistics*, 38(4):2525–2558, 2010.

[11] G. Ratsch, B. Scholkopf, A.J. Smola, S. Mika, T. Onoda, and K.R. Muller. Robust ensemble learning. *Advances in Neural Information Processing Systems*, pages 207–220, 1999.

[12] A. Kyrillidis and V. Cevher. Combinatorial selection and least absolute shrinkage via the CLASH algorithm. *Arxiv preprint arXiv:1203.2936v2*, 2011.

[13] A. Kyrillidis G. Puy and V. Cevher. Hard thresholding with norm constraints. In *IEEE ICASSP*, 2012.

[14] I. Cplex Studio. 12.3 user's manual. *IBM ILOG*, 2011.

[15] G.L. Nemhauser and L.A. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.

[16] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[17] R. Jagannathan and T. Ma. Risk reduction in large portfolios: Why imposing the wrong constraints helps. *The Journal of Finance*, 58(4):1651–1684, 2003.

[18] D. Bertsimas and R. Shioda. Algorithm for cardinality-constrained quadratic optimization. *Computational Optimization and Applications*, 43(1):1–22, 2009.

[19] F. Cesarone, A. Scozzari, and F. Tardella. Portfolio selection problems in practice: a comparison between linear and quadratic optimization models. *Arxiv preprint arXiv:1105.3594*, 2011.

[20] T.J. Chang, N. Meade, J.E. Beasley, and Y.M. Sharaiha. Heuristics for cardinality constrained portfolio optimisation. *Computers and Operations Research*, 27(13):1271–1302, 2000.

[21] M.A. Gomez, C.X. Flores, and M.A. Osorio. Hybrid search for cardinality constrained portfolio optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1865–1866. ACM, 2006.

[22] MATLAB. *version 7.13.0.564 (R2011b)*. The MathWorks Inc., Natick, Massachusetts, 2011.

[23] J. Lofberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2004.

[24] D. Kim. *Least squares mixture decomposition estimation*. PhD thesis, 1995.

[25] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[26] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Alternating minimization and projection methods for nonconvex problems. *Arxiv preprint arXiv:0801.1780*, 2008.

[27] E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.