Daniel Dadush, László A. Végh, and Giacomo Zambelli

Rescaled coordinate descent methods for linear programming

Book section

Original citation:
Originally published in Dadush, Daniel and Végh, László A. and Zambelli, Giacomo (2016) *Rescaled coordinate descent methods for linear programming.* In: Louveaux, Quentin and Skutella, Martin, (eds.) Integer Programming and Combinatorial Optimization. Lecture Notes in Computer Science, 9682. Springer, Cham, Switzerland, pp. 26-37. ISBN 9783319334608

Rescaled coordinate descent methods for Linear Programming

Daniel Dadush, László A. Végh, and Giacomo Zambelli

Centrum Wiskunde & Informatica, dadush@cwi.nl,
London School of Economics, {l.vegh,g.zambelli}@lse.ac.uk,

Abstract. We propose two simple polynomial-time algorithms to find a positive solution to $Ax = 0$. Both algorithms iterate between coordinate descent steps similar to von Neumann's algorithm, and rescaling steps. In both cases, either the updating step leads to a substantial decrease in the norm, or we can infer that the condition measure is small and rescale in order to improve the geometry. We also show how the algorithms can be extended to find a solution of maximum support for the system $Ax = 0$, $x \geq 0$.

1 Introduction

Let $A = [a_1, \ldots, a_n]$ be an integral $m \times n$ matrix with rank m, and let L denote the encoding size of A. We propose two simple polynomial algorithms for the linear feasibility problem, that is, to find a solution to systems of the form

$$\begin{aligned} Ax &= 0 \\ x &> 0. \end{aligned} \tag{1}$$

Our main contributions are: *(i)* a new simple iterative method for (1) with guaranteed finite convergence, *(ii)* a new geometric potential for these systems together with a rescaling method for improving it.
Additionally, we show that our algorithms can be adapted to solve the more general problem of finding a solution to $Ax = 0, x \geq 0$, having *maximum support*; that is, where the set of positive coordinates of x is inclusion-wise maximum. To motivate this last problem, we note that while general LP feasibility (and thus LP optimization) can be reduced to (1) via standard perturbation methods (see for example [18]), this is not desirable for numerical stability. On the other hand, any algorithm for the maximum support problem can be used directly to test feasibility of a system of the form $Ax = b$, $x \geq 0$. Indeed, given a maximum support solution $(\bar{x}, \bar{x}_0)$ to the homogenous system $Ax - bx_0 = 0$, $(x, x_0) \geq 0$, if $\bar{x}_0 > 0$ then the point $\tilde{x} = \bar{x}/\bar{x}_0$ is a solution to the original problem, otherwise we conclude that $Ax = b$, $x \geq 0$ is infeasible.

The algorithms we propose fit into a line of research developed over the past 10 years [3,8,4,6,5,15,2,20,16], where simple iterative updates, such as variants of perceptron [17] or of the relaxation method [1,11], are combined with some form of rescaling in order to get polynomial time algorithms for linear programming.

While these methods are slower than current interior point methods, they nevertheless yield important insights into the structure of linear programs. In particular, rescaling methods provide geometric potentials associated with a linear system which quantify how "well-conditioned" the system is, together with rescaling procedures for improving these potentials. Importantly, these potentials often provide more fine grained measures of the complexity of solving the linear system than the encoding length of the data, and help identify interesting subclasses of LPs that can be solved in strongly polynomial time (see for example [5]). We note that it is an open problem to devise any polynomial method for solving the maximum support problem that does not depend directly on the bit complexity L, but only on purely geometric parameters.

Preliminaries. Throughout the paper, we denote $\mathcal{L} := \{x \in \mathbb{R}^n : Ax = 0\}$, $\mathcal{L}_+ := \mathcal{L} \cap \mathbb{R}^n_+$, $\mathcal{L}_> := \mathcal{L} \cap \mathbb{R}^n_>$. We will also let $\mathcal{L}^\perp$ denote the orthogonal complement of $\mathcal{L}$; clearly, $\mathcal{L}^\perp = \{z \in \mathbb{R}^n : \exists y \in \mathbb{R}^m,\ z = A^\mathsf{T} y\}$. Let $\mathcal{L}^\perp_+ := \mathcal{L}^\perp \cap \mathbb{R}^n_+$ and $\mathcal{L}^\perp_> := \mathcal{L}^\perp \cap \mathbb{R}^n_>$. Therefore (1) is the problem of finding a point in $\mathcal{L}_>$. By strong duality, (1) is feasible if and only if $\mathcal{L}^\perp_+ = \{0\}$, that is,

$$A^\mathsf{T} y \geq 0, \tag{2}$$

has no solution other than $y = 0$.
Denote by $\mathrm{supp}(\mathcal{L}_+) \subseteq [n]$ the maximum support of a point in $\mathcal{L}_+$. Obviously $\mathrm{supp}(\mathcal{L}_+) \cap \mathrm{supp}(\mathcal{L}^\perp_+) = \emptyset$, whereas the strong duality theorem implies that $\mathrm{supp}(\mathcal{L}_+) \cup \mathrm{supp}(\mathcal{L}^\perp_+) = [n]$.
For any vector $v \in \mathbb{R}^m$ we denote by $\hat{v}$ the normal vector in the direction of v, that is $\hat{v} := v/\|v\|$. We let $\hat{A} := [\hat{a}_1, \ldots, \hat{a}_n]$. Note that, given $v, w \in \mathbb{R}^m$, $\hat{v}^\mathsf{T}\hat{w}$ is the cosine of the angle between them. Let $B(c, r)$ denote the m-dimensional Euclidean ball with center $c \in \mathbb{R}^m$ and radius $r \in \mathbb{R}_+$. Let $\boldsymbol{e}_j$ denote the jth unit vector an $\boldsymbol{e}$ denote the all-ones vector of appropriate dimension (depending on the context).

Coordinate descent algorithms. Various coordinate descent methods are known for finding non-zero points in $\mathcal{L}_+$ or $\mathcal{L}^\perp_+$. Most algorithms address either the $\mathrm{supp}(\mathcal{L}_+) = [n]$ or the $\mathrm{supp}(\mathcal{L}^\perp_+) = [n]$ case; here we outline the common update steps.
At every iteration, maintain a non-negative, non-zero vector $x \in \mathbb{R}^n$, and let $y = Ax$. If $y = 0$, then x is a non-zero point in $\mathcal{L}_+$. If $A^\mathsf{T} y > 0$, then $A^\mathsf{T} y \in \mathcal{L}^\perp_>$. Otherwise, choose an index $k \in [n]$ such that $a_k^\mathsf{T} y \leq 0$, and update x and y as follows:

$$y' := \alpha y + \beta \hat{a}_k; \quad x' := \alpha x + \frac{\beta}{\|a_k\|} \boldsymbol{e}_k, \tag{3}$$

where $\alpha, \beta > 0$ depend on the specific algorithm. Below we discuss various possible update choices. These can be seen as coordinate descent methods for minimizing $\|y\|^2$ subject to $y = Ax, x \geq 0$, and some further constraint is added, e.g. $\boldsymbol{e}^\mathsf{T} x = 1$ in the von Neumann algorithm.
An important quantity in the convergence analysis of the algorithms we will describe is the condition measure introduced by Goffin [10]:

$$\rho_A := \max_{\|y\|=1, y \in \mathbb{R}^m} \min_{j \in [n]} \hat{a}_j^\mathsf{T} y \tag{4}$$

Geometrically, $|\rho_A|$ is the distance of the origin from the boundary of $\mathrm{conv}(\hat{A})$, where $\rho_A > 0$ if and only if $\mathrm{supp}(\mathcal{L}_+^\perp) = [n]$ (in which case the origin is outside $\mathrm{conv}(\hat{A})$), $\rho_A < 0$ if and only if $\mathrm{supp}(\mathcal{L}_+) = [n]$ (in which case the origin is in the interior $\mathrm{conv}(\hat{A})$), and $\rho_A = 0$ otherwise. In particular, if $\rho_A < 0$, then $-\rho_A$ is the radius of the largest ball in $\mathbb{R}^n$ inscribed in $\mathrm{conv}(\hat{A})$ and centered at the origin. If $\rho_A > 0$, then ρ_A is the width of the *dual cone* $\{y \in \mathbb{R}^m : A^\mathsf{T} y > 0\}$, that is, the radius of the largest ball in $\mathbb{R}^m$ inscribed in the dual cone and centered at a point at distance one from the origin.

von Neumann's algorithm maintains at every iteration the condition that y is a convex combination of $\hat{a}_1, \ldots, \hat{a}_n$. The parameters $\alpha, \beta > 0$ are chosen so that $\alpha + \beta = 1$ and $\|y'\|$ is smallest possible. That is, y' is the point of minimum norm on the line segment joining y and $\hat{a}_k$. If we denote by y^t the vector at iteration t, a simple argument shows that $\|y^t\| \leq 1/\sqrt{t}$ (see Dantzig [7]). If 0 is contained in the interior of the convex hull, that is $\rho_A < 0$, Epelman and Freund [9] showed that $\|y^t\|$ decreases by a factor of $\sqrt{1-\rho_A^2}$ in every iteration. Though the norm of y converges exponentially to 0, we note that this method may not actually terminate in finite time. If 0 is outside the convex hull however, that is, $\rho_A > 0$, then the algorithm terminates after at most $1/\rho_A^2$ iterations.
Betke [3] gave a polynomial time algorithm, based on a combinatorial variant of von Neumann's update, for the case $\mathrm{supp}(\mathcal{L}_+^\perp) = [n]$. Chubanov uses von Neumann's update on the columns of the projection matrix to $\mathcal{L}$, and is able to solve the maximum support problem in time $O(n^4 L)$.[1]

Perceptron chooses $\alpha = \beta = 1$ at every iteration. If $\rho_A > 0$, then, similarly to the von Neumann algorithm, the perceptron algorithm terminates with a solution to the system $A^\mathsf{T} y > 0$ after at most $1/\rho_A^2$ iterations (see Novikoff [13]). Peña and Soheili gave a smoothed variant of the perceptron update which guarantees termination in time $O(\sqrt{\log n}/\rho_A)$ [14], and showed how this gives rise to a polynomial-time algorithm [15] using the rescaling introduced by Betke in [3]. The same running time $O(\sqrt{\log n}/\rho_A)$ was achieved by Wei Yu et al. [21] by adapting the Mirror-Prox algorithm of Nemirovski [12].

Dunagan-Vempala [8] choose $\alpha = 1$ and $\beta = -(\hat{a}_k^\mathsf{T} y)$. The choice of β is the one that makes $\|y'\|$ the smallest possible when $\alpha = 1$. It can be readily computed that

$$\|y'\| = \|y\|\sqrt{1 - (\hat{a}_k^\mathsf{T} \hat{y})^2}. \tag{5}$$

In particular, the norm of y' decreases at every iteration, and the larger is the angle between a_k and y, the larger the decrease. If $\rho_A < 0$, then $|\hat{a}_k^\mathsf{T} \hat{y}| \geq |\rho_A|$, therefore this guarantees a decrease in the norm of at least $\sqrt{1-\rho_A^2}$.

[1] It had been suggested by Prof. Cornelis Roos that Chubanov's algorithm could be further improved to $O(n^{3.5}L)$, but the paper was subsequently withdrawn due to a gap in the argument.

Our Algorithms. Both our algorithms use Dunagan-Vempala updates: Algorithm 1 on the columns of A, and Algorithm 2 on the orthogonal projection matrix Π to the space $\mathcal{L}^\perp$. These iterations are performed as long as we obtain a substantial decrease in $\|y\|$. Otherwise, a rescaling is performed in order to improve a geometric potential which serves as a proxy to the condition measure $|\rho_A|$. The rescaling in Algorithm 1 is the same as in Dunagan-Vempala [8], even though they solve the dual problem of finding a point in $\mathcal{L}^\perp_>$. We will describe the differences after the description of the algorithm.
Our Algorithm 2 is inspired by the work of Chubanov [6], and it uses the same rescaling. Our algorithms are in some sense dual to each other however: Chubanov uses von Neumann updates on the projection matrix to $\mathcal{L}^\perp$ whereas we use Dunagan-Vempala on the projection Π to $\mathcal{L}$. For the same algorithm, we provide two entirely different analyses, one similar to Chubanov's, and another volumetric one, as for Algorithm 1. Thus, while the rescaling is seemingly very different from the one used in 1, there is indeed a similar underlying geometry. We compare our algorithm to Chubanov's at the end of Section 3.
The running time of our Algorithm 1 is $O(m^3n + n^2m)L$, whereas Algorithm 2 runs in $O(mn^4L)$ time. Although the second running time bound is worse, this algorithm can be extended to solve the full support problem with the same running time estimation. Algorithm 1 can be modified to solve the maximum support problem as well (see Appendix B), but it comes at the expense of substantially increasing the running time.

2 Algorithm 1

Algorithm 1, described below, solves (1) (that is, finding a point in $\mathcal{L}_>$), using the Dunagan-Vempala update. It uses the parameters

$$\varepsilon := \frac{1}{11m}, \quad N := 6mL, \quad \delta := \min_{j\in[n]} \frac{1}{\|(AA^\mathsf{T})^{-1}a_j\|}. \tag{6}$$

It follows from (5) that, if in a given iteration there exists $k \in [n]$ such that $\hat{a}_k^\mathsf{T}\hat{y} \le -\varepsilon$, then we obtain a substantial decrease in the norm, namely

$$\|y'\| \le \|y\|\sqrt{1-\varepsilon^2}. \tag{7}$$

On the other hand, if $\hat{a}_j^\mathsf{T}\hat{y} \ge -\varepsilon$ for all $j \in [n]$, then it follows that $|\rho_A| < \varepsilon$, that is, the condition measure is small. Our aim is to perform a geometric rescaling that improves the condition measure. As a proxy for $|\rho_A|$, we use the volume of the polytope P_A defined by

$$P_A := \mathrm{conv}(\hat{A}) \cap (-\mathrm{conv}(\hat{A})). \tag{8}$$

Note that $|\rho_A|$ is the radius of the largest ball around the origin inscribed in P_A.
If $\hat{a}_j^\mathsf{T}\hat{y} \ge -\varepsilon$, then P_A is contained in a "narrow strip" of width 2ε, namely $P_A \subseteq \{z \in \mathbb{R}^m : -\varepsilon \le \hat{y}^\mathsf{T}z \le \varepsilon\}$. If we replace A with the matrix $A' := (I + \hat{y}\hat{y}^\mathsf{T})A$, Lemma 2.2 shows that the volume of $P_{A'}$ is at least 3/2 times the volume of P_A. Geometrically, A' is obtained by

Algorithm 1

Input: *A matrix $A \in \mathbb{Z}^{m\times n}$ with rank m.*

Output: *Either a solution to the system (1) or the statement that (1) is infeasible.*

Set $x_j := 1$ for all $j \in [n]$ and $y := Ax$. Set $t := 0$.

While $\|y\| \geq \delta$ *and* $t \leq N$, *do*
  If $A^\mathsf{T} y \geq 0$, **then** *STOP because (1) is infeasible.;*
  Else, *let* $k := \arg\min_{j\in[n]} \hat{a}_j^\mathsf{T} \hat{y}$;
    If $\hat{a}_k^\mathsf{T} \hat{y} < -\varepsilon$, **then update**
      $y := y - (\hat{a}_k^\mathsf{T} y)\hat{a}_k; \quad x := x - \dfrac{a_k^\mathsf{T} y}{\|a_k\|^2} \boldsymbol{e}_k$
    Else, rescale:
      $A := \left(I + \hat{y}\hat{y}^\mathsf{T}\right) A;$
      $y := 2y;$
      $t := t + 1;$
Endwhile;
If $\|y\| < \delta$, *output the feasible solution* $x := x - A^\mathsf{T}(AA^\mathsf{T})^{-1}y;$
Else *(1) is infeasible.*

applying to the columns of A the linear transformation that "stretches" them by a factor of two in the direction of $\hat{y}$.
Thus, at every iteration we either have a substantial decrease in the length of the current y, or we have a constant factor increase in the volume of P_A. Since the volume of P_A is bounded by the volume of the unit ball in $\mathbb{R}^m$, it follows that the algorithm cannot perform too many rescalings, unless (1) is infeasible.
After a polynomial number of iterations we either conclude that (1) is infeasible or we achieve a vector y of norm less than δ. In the latter case, we show that the orthogonal projection of the current x onto the null-space of A is a feasible solution to (1). Our main result is the following.

Theorem 2.1. *For any input matrix $A \in \mathbb{Z}^{m\times n}$, Algorithm 1 returns a feasible solution x for (1) if and only if (1) is feasible. The total number of iterations of the while cycle is $O(m^3 L)$, and the total number of arithmetic operations performed is $O\left((m^3 n + mn^2)L\right)$.*

Relation to previous work. Even though our update step and rescaling are the same as the one used by Dunagan and Vempala [8], the algorithm and analysis are substantially different. In fact [8] assumes that $\mathrm{supp}(\mathcal{L}_+^\perp) = [n]$, and shows that the dual cone width ρ_A increases with a high probability. Their algorithm makes use of both perceptron as well as the Dunagan-Vempala coordinate descent steps. The latter is always restarted from a random point y in the unit sphere (so in their algorithm

y is not a conic combination of the a_i's). Our algorithm uses the coordinate descent method in a more natural and direct way for the primal full dimensional case $\mathrm{supp}(\mathcal{L}_+) = [n]$.
An earlier volumetric rescaling was introduced by Betke [3]. In his rescaling, given any $y = Ax$, $\|y\| \leq 1/(\sqrt{mn})$, x a convex combination, Betke shrinks each column of A in the direction of the a_i that has the largest coefficient x_i, i.e. $a_j \leftarrow a_j - 1/2(\hat{a}_i^\mathsf{T} a_j)\hat{a}_i$. This has the effect of increasing the volume of the intersection of the cone $A^\mathsf{T} z > 0$ with the unit Euclidean ball, which can be interpreted as a smooth proxy for ρ_A. Here, one can view our potential as the natural primal counterpart to Betke's.

2.1 Analysis

The crucial part of the analysis is to bound the volume increase of P_A at every rescaling iteration; the proof is deferred to Appendix A.

Lemma 2.2. *Assume* (1) *is feasible. For some* $0 < \varepsilon < 1/(11m)$*, let* $v \in \mathbb{R}^m$*,* $\|v\| = 1$*, such that* $\hat{a}_j^\mathsf{T} v \geq -\varepsilon \ \forall j \in [n]$*. Let* $A' = (I + vv^\mathsf{T})A$*. Then* $\mathrm{vol}(P_{A'}) \geq \frac{3}{2}\mathrm{vol}(P_A)$.

To analyse the running time of Algorithm 1 we need to estimate some of the parameters in terms of the encoding size of A. The proofs are also deferred to Appendix A.

Proposition 2.3. $\delta \geq 2^{-3L}$.

Proposition 2.4. *If* $\mathrm{conv}(A)$ *contains the origin in its interior, then* $\mathrm{conv}(A) \supseteq B(0, 2^{-2L})$ *and* $|\rho_A| \geq 2^{-3L}$.

Proof of Theorem 2.1 *Correctness.* If the algorithm terminates because it found a $y \neq 0$ such that $A^\mathsf{T} y \geq 0$, then (1) is indeed infeasible (note that $y \neq 0$ because $\|y\| > \delta$). Assume the algorithm terminates because it performed N rescalings, and suppose by contradiction that (1) is feasible. Then $\mathrm{conv}(A)$ would contain the origin in the interior, so by Proposition 2.4 at the beginning P_A would contain a ball of radius at least 2^{-3L}. In particular, at the beginning $\mathrm{vol}(P_A) \geq V_m 2^{-3mL}$, where V_m denotes the volume of the unit m-ball. By Lemma 2.2, after N iterations $\mathrm{vol}(P_A) \geq (3/2)^N 2^{-3mL} V_m > V_m$, which is impossible since P_A is contained in the unit m-ball.
Assume then that the algorithm terminates with vectors $\bar{x}$, $\bar{y}$ such that $\|\bar{y}\| \leq \delta$. Observe that at every iteration we maintain the invariant $y = Ax$ and $x_j \geq 1$ for all $j \in [n]$. Now, consider A to be the initial matrix and let $\bar{A}$ be the current matrix in the last iteration of the algorithm, so that $\bar{y} = \bar{A}\bar{x}$. Let x' be the solution returned by the algorithm, that is $x' = \bar{x} - \bar{A}^\mathsf{T}(\bar{A}\bar{A}^\mathsf{T})^{-1}\bar{y}$. One can readily verify that $\bar{A}x' = \bar{A}\bar{x} - \bar{y} = 0$ (indeed, x' is the orthogonal projection of $\bar{x}$ onto the subspace $\{x : \bar{A}x = 0\}$).
We need to check that $Ax' = 0$ and $x' > 0$. Note that $\bar{A}$ is obtained by a sequence of rescalings of A, therefore it is of the form $\bar{A} = TA$, where

$T = (I+v_k v_k^\mathsf{T}) \cdots (I+v_1 v_1^\mathsf{T})$ for some sequence of vectors $v_1, \ldots, v_k \in \mathbb{R}^m$ with norm 1, therefore $Ax' = T^{-1}\bar{A}x' = 0$. We need to prove $x' > 0$. Define $z := A\bar{x}$. In particular, $z = T^{-1}\bar{y}$. Note that, for any vector $v \in \mathbb{R}^m$ of norm 1, $(I + vv^\mathsf{T})^{-1} = I - \frac{1}{2}vv^\mathsf{T}$, therefore $T^{-1} = (I - \frac{1}{2}v_1 v_1^\mathsf{T}) \cdots (I - \frac{1}{2}v_k v_k^\mathsf{T})$.
Observe that, for any vector $y \in \mathbb{R}^m$ and any vector $v \in \mathbb{R}^m$ with unit norm, $\|(I - \frac{1}{2}vv^\mathsf{T})y\|^2 = \|y\|^2 - \frac{3}{4}(v^\mathsf{T}y)^2 \leq \|y\|^2$. This implies that $\|z\| \leq \|\bar{y}\| < \delta$. Now $x' = \bar{x} - \bar{A}^\mathsf{T}(\bar{A}\bar{A}^\mathsf{T})^{-1}\bar{y} = \bar{x} - A^\mathsf{T}(AA^\mathsf{T})^{-1}T^{-1}\bar{y} = \bar{x} - A^\mathsf{T}(AA^\mathsf{T})^{-1}z > 0$ where the last inequality follows from the fact that $\bar{x}_j \geq 1$, $j \in [n]$, and by $|a_j^\mathsf{T}(AA^\mathsf{T})^{-1}z| \leq \|(AA^\mathsf{T})^{-1}a_j\|\|z\| < \frac{1}{\delta}\delta = 1$ for all $j \in [n]$.
Termination. By (7), $\|y\|^2$ decreases by a factor of $(1-\varepsilon^2)$ every time we perform an update. Every time we perform a rescaling, $\|y\|^2$ increases by a factor of 4. Initially, $y = Ae$, thus at the beginning $\|y\|^2 \leq 2^{4L}$. It follows that the number κ of updates performed by the algorithm satisfies $\delta^2 \leq \|Ae\|^2 4^N (1-\varepsilon^2)^\kappa \leq 2^{4L+2N} e^{-k\varepsilon^2}$. Since $\delta \geq 2^{-2L}$ by Proposition 2.3, it follows that $\kappa \leq \varepsilon^{-2}(8L + 2N) = 121m^2(8 + 12m)L$. Therefore the total number of iterations is at most $N + \kappa = O(m^3 L)$.
Finally observe that, whenever we perform an update the computation of $A^\mathsf{T}y$ can be performed in $O(n)$ arithmetic operations, provided that we pre-compute the matrix $A^\mathsf{T}A$ every time we perform a rescaling. The number of rescalings is $O(mL)$, computing $(I + \hat{y}\hat{y}^\mathsf{T})A$ requires $O(nm)$ operations, while computing $A^\mathsf{T}(I + \hat{y}\hat{y}^\mathsf{T})(I + \hat{y}\hat{y}^\mathsf{T})A$ requires $O(n^2)$ arithmetic operations, provided that we had previously computed $A^\mathsf{T}A$. Therefore the total number of arithmetic operations is $O((m^3 n + mn^2)L)$.

3 Algorithm 2: A dual Chubanov algorithm

Let $\Pi = A^\mathsf{T}(AA^\mathsf{T})^{-1}A$ denote the orthogonal projection matrix to $\mathcal{L}^\perp$ (i.e., the space spanned by the rows of A), and let $\pi_1, \ldots, \pi_n$ denote the columns of Π and π_{ij} $(i, j \in [n])$ denote the (i, j) entry of Π. We recall the following well known properties of the projection matrix Π.

Proposition 3.1. *Let $A \in \mathbb{R}^{m\times n}$ and let $\Pi = A^\mathsf{T}(AA^\mathsf{T})^{-1}A$. The following hold (i) For all $x, z \in \mathbb{R}^n$, $\Pi x = 0$ if and only if $x \in \mathcal{L}$, and $\Pi z = z$ if and only if $z \in \mathcal{L}^\perp$; (ii) $\Pi^2 = \Pi$; (iii) For every $w \in \mathbb{R}^n$, $\|\Pi w\| \leq \|w\|$; (iv) For all $j \in [n]$, $\pi_j = \Pi e_j$, thus $\|\pi_j\| \leq 1$; (v) $\pi_{jj} = \|\pi_j\|^2$ for all $j \in [n]$; (vi) $\operatorname{trace}(\Pi) = \sum_{j=1}^n \|\pi_j\|^2 = m$.*

In Algorithm 2 below, we set $\varepsilon := \frac{1}{16n\sqrt{3m}}$. Throughout this section, for every $I \subseteq [n]$ we denote by D_I the diagonal matrix with $d_{jj} = 1/2$ if $j \in I$, $d_{jj} = 1$ if $j \notin I$. Thus $D_I = I - (1/2)\sum_{j\in I} e_j e_j^\mathsf{T}$.
Note that, since $z_j = \pi_j^\mathsf{T} z$ for all $j \in [n]$, the update step is just the Dunagan-Vempala update applied to the matrix Π instead of on A. Thus, at each update the norm of the current z decreases by at least a multiplicative factor $\sqrt{1-\varepsilon^2}$.
Observe also that at every iteration $w_j \geq 1$ for all $j \in [n]$, so in particular $\|z\| < 1$ immediately implies $w - z > 0$, thus the algorithm terminates with the solution $x := w - z$ if $\|z\| \leq 1$.

Algorithm 2

Input: A matrix $A \in \mathbb{Z}^{m\times n}$ with rank m.

Output: Either a solution $x \in \mathcal{L}_>$, or a set $R \subseteq [n]$ disjoint from the support of $\mathcal{L}_+$.

Compute $\Pi = A^\mathsf{T}(AA^\mathsf{T})^{-1}A$.
Set $w_j := 1$ for all $j \in [n]$, $z := \Pi w$, $\mathrm{count}_j := 0$ for all $j \in [n]$.
While $\mathrm{count}_j < L$ for all $j \in [n]$ do
  If $w - z > 0$, output $x := w - z$ and STOP;
  If $z \geq 0$, return $R := \{j \in [n] : z_j \neq 0\}$ and STOP;
  Else, let $i := \arg\min_{j\in[n]} \frac{z_j}{\|z\|\|\pi_j\|}$;
    If $\frac{z_i}{\|z\|\|\pi_i\|} < -\varepsilon$, **then update**
      $z := z - \frac{z_i\pi_i}{\|\pi_i\|^2}$; $w := w - \frac{z_i \boldsymbol{e}_i}{\|\pi_i\|^2}$;
    else, rescale
      let $I := \{j \in [n] : \frac{z_j}{\|z\|} > \frac{1}{\sqrt{3n}}\}$;
      $A := AD_I$;
      Recompute $\Pi = A^\mathsf{T}(AA^\mathsf{T})^{-1}A$;
      Set $w_j := 1$ for all $j \in [n]$, $z := \Pi w$;
      $\mathrm{count}_j := \mathrm{count}_j + 1$ for all $j \in I$;
Endwhile;
Output $R := \{j : \mathrm{count}_j = L\}$.

We give a proof of correctness of the algorithm. Afterwards, we provide a different analysis, reminiscent of Lemma 2.2, which relates the rescaling step to the change of a certain geometric quantity related the condition measure of Π.

3.1 Correctness of the algorithm

For any $a \in \mathbb{R}$, we let $a^+ := \max\{0, a\}$ and $a^- = (-a)^+$. The correctness of the algorithm is based on the following simple bound due to Roos [16]]

Lemma 3.2 (Roos 2014). *Let $z \in \mathcal{L}^\perp$ and let $k \in [n]$ such that $z_k > 0$. Then, for every $x \in \mathcal{L} \cap [0,1]^n$.*

$$x_k \leq \frac{\sum_{j=1}^n z_j^-}{z_k}. \tag{9}$$

Proof. For any $x \in \mathcal{L}$, $z^\mathsf{T}x = 0$, therefore $x_k = \frac{\sum_{j\in[n]\setminus\{k\}} -z_j x_j}{z_k}$. The statement follows from the fact that, for every $x \in [0,1]^n$, $\sum_{j\in[n]\setminus\{k\}} -z_j x_j \leq \sum_{j=1}^n z_j^-$. □

Lemma 3.3. *Let A be the current matrix at a given iteration of Algorithm 2. Suppose that the current $z = \Pi w$ satisfies $z_j \geq -\varepsilon\|z\|\|\pi_j\|$. Then the set $I = \{j \in [n] : \frac{z_j}{\|z\|} > \frac{1}{\sqrt{3n}}\}$ is nonempty. Furthermore, every $x \in \mathcal{L} \cap [0,1]^n$ satisfies $x_k \leq \frac{1}{2}$ for all $k \in I$.*

Proof. Note first that

$$\sum_{j=1}^{n}\left(\frac{z_j^+}{\|z\|}\right)^2 = 1-\sum_{j=1}^{n}\left(\frac{z_j^-}{\|z\|}\right)^2 \geq 1-\varepsilon^2\sum_{j=1}^{n}\|\pi_j\|^2 = 1-m\varepsilon^2 > \frac{1}{3},$$

which implies that there exists $k\in[n]$ such that $\frac{z_k}{\|z\|} > \frac{1}{\sqrt{3n}}$. Given $k\in I$, Lemma 3.2 implies that, for every $x\in\mathcal{L}\cap[0,1]^m$,

$$x_k \leq \frac{\sum_{j=1}^n z_j^-}{z_k} \leq \varepsilon\frac{\|z\|}{z_k}\sum_{j=1}^{n}\|\pi_j\| \leq \varepsilon\sqrt{3n}\sqrt{n}(\sum_{j=1}^{n}\|\pi_j\|^2)^{1/2} = \varepsilon n\sqrt{3}\sqrt{m} < \frac{1}{2}.$$

□

Observe that rescaling has the effect of replacing the null space $\mathcal{L}$ of A with $D_I^{-1}\mathcal{L}$, that is, multiplying by 2 the components indexed by I of all vectors in $\mathcal{L}$. Let $\mathcal{L}^0$ be the null space of the input matrix A (i.e. before any rescaling). Lemmas 3.2 and 3.3 show that, at any iteration of the algorithm, $\mathcal{L}^0\cap[0,1]\subseteq\{x\in\mathbb{R}^n : x_j < 2^{-\mathrm{count}_j}\}$. It is well know (see for example Schrijver [18]) that, if $j\in[n]$ is in the support $Ax=0$, $x\geq 0$, then there exists a solution with $x_j\geq 2^{-L}$. This shows that, whenever $\mathrm{count}_j = L$ for some $j\in[n]$, j cannot be in the support.

Running time At the beginning of the algorithm and after each rescaling, $z=\Pi\boldsymbol{e}$, therefore $\|z\|\leq\|\boldsymbol{e}\|=\sqrt{n}$. Every Dunagan-Vempala update decreases $\|z\|^2$ by a factor $1-\varepsilon^2$, and the algorithm terminates with $x:=w-z>0$ when $\|z\|<1$. This shows that the number κ of updates between any two rescalings satisfies $n(1-\varepsilon^2)^\kappa\geq 1$, therefore $\kappa\leq\ln(n)\varepsilon^{-2}=O(n^2m\log(n))$. Since the algorithm performs at most L rescaling for every variable, it follows that the algorithm performs at most $O(n^3m\log(n)L)$ updates. Each update requires $O(n)$ operations, therefore the running-time of the algorithm is $O(n^4m\log(n)L)$. (It should be noted here that the recomputation of the matrix Π at every rescaling can be performed in $O(|I|n^2)$ arithmetic operations using the Sherman-Morrison formula [19], therefore the total number of arithmetic operations performed during the rescalings is $O(n^3L)$).
Finally, the $\log(n)$ factor appearing in the running time can be eliminated by slightly modifying the algorithm, choosing the next w after each rescaling more carefully, as shown by the following lemma (proved in Appendix A).

Lemma 3.4. *Let $A\in\mathbb{R}^{m\times n}$, $\Pi=A^\mathsf{T}(AA^\mathsf{T})^{-1}A$. Given $I\subseteq[n]$, let $\Pi'=D_IA^\mathsf{T}(AD_I^2A^\mathsf{T})^{-1}AD_I$. Given $z=\pi w$ for some $w\in\mathbb{R}^n$, if we let $w'=D_I^{-1}w$ and $z'=\Pi'w'$, then $\|z'\|\leq 2^{|I|}\|z\|$.*

By the above lemma, we can ensure that, throughout the entire execution of the algorithm, rescaling increases the norm of z by a factor of at most 2^{nL}. This implies that the total number κ of updates performed by the algorithm must satisfy $n(1-\varepsilon^2)^\kappa 4^{nL}\geq 1$, which implies $\kappa\leq(\ln n + nL\ln 4)\varepsilon^{-2}=O(n^3mL)$. It follows that the running time is $O(n^4mL)$.

The maximum support problem Algorithm 2 can be used to identify the support of $Ax = 0$, $x \geq 0$: whenever the algorithm returns a set R of indices not in the support, we set $x_j := 0$ for all $j \in R$, remove the columns of A indexed by R, and repeat. If the algorithm terminates with a feasible solution $x > 0$ for the current system, this defines a maximal support solution for the original problem. In the worst case, we need to run Algorithm 2 n times, giving a naïve running time estimate of $O(n^5 mL)$. However, observe that whenever Algorithm 2 terminates with a set R of indices, at the subsequent call to the algorithm we can initialize count_j, $j \notin R$, to the values computed at the end of the last call. Therefore, the total number of arithmetic operations needed to compute a maximum support solution is $O(n^4 mL)$, the same as the worst-case running time of Algorithm 2.

3.2 Analysis based on a geometric potential

Let $Q_\Pi := \text{conv}(\Pi) \cap \text{conv}(-\Pi)$. Throughout this section, we denote by $\widehat{\text{vol}}(\cdot)$ the volume with respect to the measure induced on $\mathcal{L}^\perp$. We will consider as a potential $\widehat{\text{vol}}(Q_\Pi)$.

Lemma 3.5. *Let $\varepsilon' = 1/(16\sqrt{3nm})$. Let $z \in \mathcal{L}^\perp$ such that $z_j \geq -\varepsilon' \|z\| \|\pi_j\|$ for all $j \in [n]$. Let $I = \{j \in [n] : \frac{z_j}{\|z\|} > \frac{1}{\sqrt{3n}}\}$, and $\Pi' = D_I A^\mathsf{T}(AD_I^2 A^\mathsf{T})^{-1} A D_I$. Then*

$$\widehat{\text{vol}}(Q_{\Pi'}) \geq e^{1/8}\, \widehat{\text{vol}}(Q_\Pi).$$

The proof is given in Appendix A.
Since $\varepsilon \leq \varepsilon'$ because $m \leq n$, it follows that when Algorithm 2 performs a rescaling, the current point $z = \Pi w$ satisfies the hypothesis of Lemma 3.5, thus after rescaling, $\widehat{\text{vol}}(Q_\Pi)$ increases by a constant factor. Let us recall that $Q_\Pi \subseteq B(0,1) \cap \mathcal{L}$, therefore $\widehat{\text{vol}}(Q_\Pi) \leq V_0$, where V_0 is the volume of the m-dimensional unit ball in $\mathbb{R}^m$. We also have

Proposition 3.6. *Let $A \in \mathbb{Z}^{m \times n}$, let L denote the encoding size of A, and let $\Pi = A^\mathsf{T}(AA^\mathsf{T})^{-1}A$. If $\mathcal{L}_> \neq \emptyset$, then $\widehat{\text{vol}}(Q_\Pi) \geq 2^{-3mL}$.*

The proof of Proposition 3.6 is postponed to Appendix A. It follows that, if $Ax = 0$, $x > 0$ has a solution, then the algorithm cannot perform more than $(24 \ln 2)mL$ rescalings. In particular, in $O(mL)$ rescalings one can either find a solution to $Ax = 0$, $x > 0$, or argue that none exists. Since $m \leq n$, this means that typically we may be able to prove that $Ax = 0$, $x > 0$ has no solution before we are actually able to identify any index j not in the support.

Refinements Note that the two analyses we provided are somewhat "loose", in the sense that the parameters in Algorithm 2 have been chosen to ensure that both analyses hold. Here we propose a few refinements and variants.

(a) To optimize the algorithm based on the potential $\widehat{\text{vol}}(Q_\Pi)$, it is clear from Proposition 3.6 that we can use $\varepsilon' = 1/(8\sqrt{nm})$ instead of $\varepsilon = 1/(8\sqrt{mn})$. As we have seen, the maximum number of rescaling that

the algorithm can do if $Ax = 0$, $x > 0$ is feasible is $O(24\ln(2)mL)$. This guarantees that the numbers κ of updates satisfies $n(1-\varepsilon'^2)^\kappa 4^{24\ln(2)mL} \geq 1$, therefore $\kappa = O(n^2m^2L)$. This gives a total running time of $O(n^2m^3L)$.
(b) The analysis of the algorithm based on the argument in Section 3.1 can be simplified if we set $\bar{\varepsilon} = 1/(2\sqrt{mn})$, and do an update when the condition $z_i \leq -\bar{\varepsilon}\|\pi_i\|$ is satisfied by some $i \in [n]$ (rather then when $z_i \leq -\varepsilon\|z\|\|\pi_i\|$). This implies that the norm of $z' := z - (z_i/\|\pi_i\|^2)\pi_i$ satisfies $\|z'\|^2 \leq \|z\|^2(1-(\bar{\varepsilon}/\|z\|)^2) = \|z\|^2 - 1/(4mn)$. Since after each rescaling $\|z\| \leq \sqrt{n}$, this ensures that between every two rescalings there are at most $4mn^2$ updates (without the need of resorting to Lemma 3.4). When $z_j \geq -\bar{\varepsilon}\|\pi_j\|$ for every $j \in [n]$, it follows that there must be at least one $k \in [n]$ such that the bound in (9) is at most $1/2$. Indeed, for any k such that $z_k \geq 1$ (one such k must exist because $w - z \not> 0$ and $w_j \geq 1$ for all $j \in [n]$) we have $(\sum_{j=1}^n z_j^-)/z_k \leq \varepsilon\sum_{j=1}^n \|\pi_j\| \leq \varepsilon\sqrt{nm} = 1/2$.
(c) A variant of the algorithm that gives the same running time but could potentially be more efficient in practice is the following. Define $\tilde{\varepsilon} = 1/(2\sqrt{n})$. At each iteration, let $N(z) := \{j : z_j < 0\}$, and compute $q := \sum_{j\in N(z)} \pi_j$. Note that $\|q\| \leq \sqrt{|N(z)|}$, since q is the projection onto $\mathcal{L}^\perp$ of the incidence vector of $N(z)$.
Instead of checking if there exists $i \in [n]$ such that $z_i \leq -\varepsilon\|z\|\|\pi_i\|$, check if $q^\mathsf{T} z \leq -\tilde{\varepsilon}\|q\|$. If such an index exists, then update as follows

$$z' := z - q\frac{q^\mathsf{T} z}{\|q\|^2}; \quad w' := w - \frac{q^\mathsf{T} z}{\|q\|^2}\sum_{j\in N(z)} \boldsymbol{e}_j.$$

It follows that $\|z'\|^2 \leq \|z\|^2 - 1/(4n)$, hence the maximum number of updates between rescalings is $4n^2$. If instead $q^\mathsf{T} z > -\tilde{\varepsilon}\|q\|$, then for every $k \in [n]$ such that $z_k \geq 1$, we have $(\sum_{j=1}^n z_j^-)/z_k = (-q^\mathsf{T} z)/z_k \leq \tilde{\varepsilon}\|q\| \leq \tilde{\varepsilon}\sqrt{n} = 1/2$.
Note that the total number of updates performed by the algorithm is $O(n^3L)$, which is better than $O(mn^3L)$ updates performed by Algorithm 2. However, the number of arithmetic operations needed to compute q is, in the worst case, $O(n^2)$, therefore the total number of arithmetic operations is still $O(n^5L)$. Nevertheless, this variant may be better in practice because it provides faster convergence.

Comparison with Chubanov's algorithm Chubanov's algorithm works on the projection matrix $\bar{\Pi} = [\bar{\pi}_1, \ldots, \bar{\pi}_n]$ to the null space $\mathcal{L}$ of A, that is, $\bar{\Pi} = I - \Pi$. At every iteration, Chubanov maintains a vector $v \in \mathbb{R}^n_+$ such that $\boldsymbol{e}^\mathsf{T} v = 1$, starting from $y = \bar{\pi}_j$ for some $j \in [n]$, and computes $y = \bar{\Pi}v$. If $y > 0$, then Chubanov's algorithm terminates with $y \in \mathcal{L}_>$, else it selects an index $i \in [n]$ with $y_i \leq 0$ and performs a von Neumann step $y' = \lambda y + (1-\lambda)\bar{\pi}_i$. By Dantzig's analysis of von Neumann's algorithm [7], $\|y'\|^{-2} \geq \|y\|^{-2} + 1$, hence after at most $4n^3$ operations $\|y\| \leq 1/(2n\sqrt{n})$. Now, if $k = \arg\max_{j\in[n]} v_j$, then $v_k \geq 1/n$, therefore we have that for every $x \in \mathcal{L}_+ \cap [0,1]^n$, $x_k \leq (v^\mathsf{T} x)/v_k = (y^\mathsf{T} x)/v_k \leq (\|x\|\|y\|)/v_k \leq \sqrt{n}\|y\|/v_k \leq 1/2$. Thus, after at most $O(n^3)$ steps, Chubanov's algorithm performs the same rescaling as Algorithm 2 using $I := \{j \in [n] : \|y\|/v_k \leq 1/(2\sqrt{n})\}$.

Note that, while the rescaling used by Algorithm 2 and Chubanov's algorithm are the same, and both algorithm ultimately produce a point in $\mathcal{L}_>$ if one exists, the updating steps work in the opposite direction. Indeed, both algorithms maintain a nonnegative vector in $\mathbb{R}^n$, but every von Neumann step in Chubanov's algorithm decreases the norm of the orthogonal projection of the nonnegative vector onto $\mathcal{L}$, whereas every Dunagan-Vempala update of Algorithm 2 decreases the norm of the orthogonal projection z onto $\mathcal{L}^\perp$. Also, Chubanov's iterations guarantee a fixed increase in $\|y\|^{-2}$, and rescaling occurs when $\|y\|$ is small enough, whereas Algorithm 2 terminates when $\|z\|$ is small enough (that is, when $\|z\| \leq 1$), and rescaling occurs when the updating step would not produce a sufficient decrease in $\|z\|$.
We note that Chubanov's algorithm solves the maximum support problem in $O(n^4 L)$, and hence is faster than ours. His speedup is based on an amortized analysis that we do not currently know how to reproduce with Dunagan-Vempala updates, though we imagine that this should be possible.

References

1. S. Agmon, The relaxation method for linear inequalities, *Canadian Journal of Mathematics 6* (1954) 382-392.
2. A. Basu, J. De Loera, M. Junod, On Chubanov's method for Linear Programming, (2012) to appear on *INFORMS Journal on Computing*. `arXiv:1204.2031v1`
3. U. Betke, Relaxation, new combinatorial and polynomial algorithms for the linear feasibility problem, *Discrete & Computational Geometry 32* (2004) 317-338.
4. S. Chubanov, A strongly polynomial algorithm for linear systems having a binary solution, *Mathematical Programmming 134* (2012), 533-570.
5. S. Chubanov, A polynomial algorithm for linear optimization which is strongly polynomial under certain conditions on optimal solutions, `http://www.optimization-online.org/DB_FILE/2014/12/4710.pdf` (2015).
6. S. Chubanov, A polynomial projection algorithm for linear programming, *Mathematical Programming 153* (2015) 687-713.
7. G. B. Dantzig, An ε-precise feasible solution to a linear program with a convexity constraint in $1/\varepsilon^2$ iterations independent of problem size, Report SOL 92-5, Stanford University (1992).
8. J. Dunagan, S. Vempala, A simple polynomial-time rescaling algorithm for solvign linear programs, *Mathematical Programming 114* (2006) 101-114.
9. M. Epelman and R. M. Freund. Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system. *Mathematical Programming, 88(3)* (2000) 451-485.
10. J. Goffin. The relaxation method for solving systems of linear inequalities. *Math. Oper. Res., 5* (1980) 388-414.
11. T. Motzkin, I.J. Schoenberg, The relaxation method for linear inequalities, *Canadian Journal of Mathematics 6* (1954) 393-404.

12. A. Nemirovski, Prox-method with rate of convergence $o(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convexconcave saddle point problems, *SIAM Journal on Optimization 15* (2004) 229-251.
13. A.B.J Novikoff, On convergence proofs for perceptrons, *Proceedings of the Symposium on the Mathematical Theory of Autoomata XII* (1962) 615-622.
14. N. Soheili and J. Peña, A smooth perceptron algorithm, *SIAM Journal on Optimization 22* (2012) 728-737.
15. J. Peña, N. Soheili, A deterministic rescaled perceptron algorithm, *Mathematical Programming* (2015)
16. K. Roos, On Chubanovs Method for Solving a Homogeneous Inequality System, *Numerical Analysis and Optimization 134* (2015) 319-338.
17. F. Rosenblatt, The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, *Cornell Aeronautical Laboratory, Psychological Review 65* (1958) 386-408
18. A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, New York (1986).
19. J. Sherman and W. J. Morrison, Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix, *Annals of Mathematical Statistics 21* (1949) 124-127.
20. L.A. Végh, G. Zambelli, A polynomial projection-type algorithm for linear programming, *Operations Research Letters 42* (2014), 91-96.
21. A. Wei Yu, F. Kılınç-Karzan, J. Carbonell, Saddle Points and Accelerated Perceptron Algorithms, *Proceedings of The 31st International Conference on Machine Learning – Journal of Machine Learning Research 32* (2014) 1827-1835.

A Proofs of technical lemmas

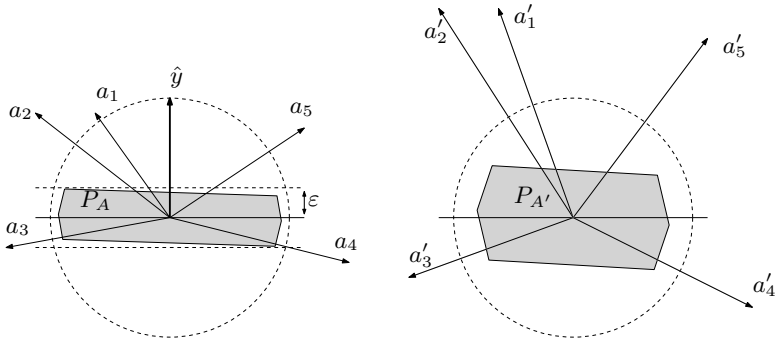

Fig. 1. Effect of rescaling. The dashed circle represent the set of points of norm 1. The shaded areas are P_A and $P_{A'}$.

The following easy technical claim will be needed in the proof of Lemma 2.2.

Lemma A.1. *Let $X \in \mathbb{R}$ be a random variable supported on the interval $[-\varepsilon, \eta]$, where $0 \le \varepsilon \le \eta$, satisfying $\mathbb{E}[X] = \mu$. Then for $c \ge 0$, we have that*

$$\mathbb{E}[\sqrt{1+cX^2}] \le \sqrt{1+c\eta(\varepsilon+|\mu|)}$$

Proof. Let $l(x) = \frac{\eta-x}{\eta+\varepsilon}\sqrt{1+c\varepsilon^2} + \frac{x+\varepsilon}{\eta+\varepsilon}\sqrt{1+c\eta^2}$ denote the unique affine interpolation of $\sqrt{1+cx^2}$ through the points $\{-\varepsilon, \eta\}$. By convexity of $\sqrt{1+cx^2}$, we have that $l(x) \ge \sqrt{1+cx^2}$ for all $x \in [-\varepsilon, \eta]$. Hence, we see that

$$\begin{aligned}\mathbb{E}[\sqrt{1+cX^2}] &\le \mathbb{E}[l(X)] \quad (\text{since } X \text{ is supported on } [-\varepsilon,\eta])\\ &= l(\mathbb{E}[X]) = l(\mu) \quad (\text{since } l \text{ is affine})\,.\end{aligned}$$

From here, we get that

$$\begin{aligned}l(\mu) &= \frac{\eta-\mu}{\eta+\varepsilon}\sqrt{1+c\varepsilon^2} + \frac{\mu+\varepsilon}{\eta+\varepsilon}\sqrt{1+c\eta^2}\\ &\le \sqrt{1+c\left(\frac{\eta-\mu}{\eta+\varepsilon}\varepsilon^2 + \frac{\mu+\varepsilon}{\eta+\varepsilon}\eta^2\right)} \quad \left(\text{by concavity of } \sqrt{x}\right)\\ &= \sqrt{1+c\left(\eta\varepsilon + (\eta-\varepsilon)\mu\right)} \le \sqrt{1+c\eta(\varepsilon+|\mu|)} \quad (\text{since } \varepsilon \le \eta)\,,\end{aligned}$$

as needed. □

Lemma 2.2. *Assume (1) is feasible. For some $0 < \varepsilon < 1/(11m)$, let $v \in \mathbb{R}^m$, $\|v\| = 1$, such that $\hat{a}_j^\mathsf{T} v \ge -\varepsilon \ \forall j \in [n]$. Let $A' = (I + vv^\mathsf{T})A$. Then $\mathrm{vol}(P_{A'}) \ge \frac{3}{2}\mathrm{vol}(P_A)$.*

Proof. Let $T := (I + vv^\mathsf{T})$. We shall prove $TP_A \subseteq (1+3\varepsilon)P_{A'}$. The claim then follows easily, since $\mathrm{vol}(TP_A) = 2\mathrm{vol}(P_A)$ as $\det(T) = 2$. Thus we obtain $\mathrm{vol}(P'_A) \geq 2\mathrm{vol}(P_A)/(1+3\varepsilon)^m \geq \frac{3}{2}\mathrm{vol}(P_A)$, since $\ln(1+3\varepsilon)^m \leq \ln(1+3/(11m))^m \leq \frac{3}{11} \leq \ln\frac{4}{3}$.
To show $TP_A \subseteq (1+3\varepsilon)P_{A'}$, let us consider an arbitrary point $z \in P_A$. By symmetry, it suffices to show $Tz \in (1+3\varepsilon)\mathrm{conv}(\hat{A}')$. By definition, there exists $\lambda \in \mathbb{R}^n_+$ such that $\sum_{j=1}^n \lambda_j = 1$ and $z = \sum_{j=1}^n \lambda_j \hat{a}_j$. Note that

$$Tz = \sum_{j=1}^n \lambda_j T\hat{a}_j = \sum_{j=1}^n (\lambda_j \|T\hat{a}_j\|)\hat{a}'_j = \sum_{j=1}^n \lambda_j \sqrt{1+3(v^\mathsf{T}\hat{a}_j)^2}\,\hat{a}'_j. \quad (10)$$

Thus, since $0 \in \mathrm{conv}(\hat{A}')$, it suffices to show that $\sum_{j=1}^n \lambda_j \sqrt{1+3(v^\mathsf{T}\hat{a}_j)^2} \leq 1+3\varepsilon$.
The above is of the form $\mathbb{E}[\sqrt{1+3X^2}]$ where X is a random variable supported on $[-\varepsilon, 1]$ and $|\mathbb{E}[X]| = |\sum_{j=1}^n \lambda_j v^\mathsf{T}\hat{a}_j| = |v^\mathsf{T}z|$. Note that $|v^\mathsf{T}z| \leq \varepsilon$ because both z and $-z$ are in P_A. Hence, by Lemma A.1, $\sum_{j=1}^n \lambda_j \sqrt{1+3(v^\mathsf{T}\hat{a}_j)^2} \leq \sqrt{1+3(2\varepsilon)} \leq 1+3\varepsilon$. □

For the proofs of the next two propositions (Propositions 2.3 and 2.4), we recall that, for every square submatrix B of A, $|\det(B)| \leq 2^L$.

Proposition 2.3. $\delta \geq 2^{-3L}$.

Proof. Given a matrix B and subsets P, Q of the row and column indices, respectively, we denote by $B_{P,Q}$ the submatrix of B defined by the rows indexed by P and by the columns indexed by Q. Let $M := AA^\mathsf{T}$. By the Cauchy-Binet formula, for every $1 \leq k \leq m$ and every choice of $P, Q \subseteq [m]$ such that $|P| = |Q| = k$

$$\det(M_{P,Q})^2 = \sum_{\substack{U\subseteq[m]\\|U|=k}} \det(A_{P,U})\det(A_{Q,U}) \leq \binom{n}{k} 2^{2L} \leq 2^{3L}, \quad (11)$$

since $\binom{n}{k} \leq n^k$ and $L \geq m\log n$.
It follows that the absolute value of each entry of $\mathrm{adj}(M)$, the adjugate matrix of M, is at most $2^{\frac{3}{2}L}$, therefore the absolute value of each entry of the matrix $\mathrm{adj}(M)a_j$ is at most $m2^{\frac{5}{2}L}$, $j \in [n]$. In particular, $\|\mathrm{adj}(M)a_j\|^2 \leq m^3 2^{5L} \leq 2^{6L}$. Thus $\delta \geq \frac{\det M}{2^{3L}} \geq 2^{-3L}$. □

Proposition 2.4. *If* $\mathrm{conv}(A)$ *contains the origin in its interior, then* $\mathrm{conv}(A) \supseteq B(0, 2^{-2L})$ *and* $|\rho_A| \geq 2^{-3L}$.

Proof. To prove the first part of the statement, let p be the point of minimum norm on the boundary of $\mathrm{conv}(A)$. Since p is contained in some facet of $\mathrm{conv}(\hat{A})$, there exist a nonsingular $m \times m$ submatrix B of A such that p is the orthogonal projection of the origin onto the hyperplane $H := \{y \in \mathbb{R}^m : \exists x \text{ s.t. } y = Bx, \boldsymbol{e}^\mathsf{T}x = 1\}$. If we let $\gamma = (B^{-1})^\mathsf{T}\boldsymbol{e}$, then $H = \{y \in \mathbb{R}^m : \gamma^\mathsf{T}y = 1\}$, thus $p = -\gamma/\|\gamma\|^2$, so $\|p\| = \|\gamma\|^{-1}$. Since the absolute value of each entry of $\mathrm{adj}(B)$ is at most 2^L, the absolute value

of each entry of γ is at most $\frac{m2^L}{\det(B)}$. Thus $\|\gamma\| \leq \frac{m^{3/2}2^L}{\det(B)} \leq 2^{2L}$, where the last inequality follows from the fact that $2^L \geq L \geq mn \geq m^{3/2}$ and that $\det(B) \geq 1$ since A is an integral matrix.
For the second part, let $\alpha = \min_{j\in[n]} \|a_j\|$. Then $\text{conv}(\hat{A}) \subseteq \alpha^{-1}\text{conv}(A)$, thus $|\rho_A| \geq \alpha^{-1}\|p\|$. Note that $\alpha \leq 2^L$. This shows that $|\rho_A| \geq (\alpha\|\gamma\|)^{-1} \geq 2^{-3L}$. □

Lemma 3.4. *Let $A \in \mathbb{R}^{m\times n}$, $\Pi = A^\mathsf{T}(AA^\mathsf{T})^{-1}A$. Given $I \subseteq [n]$, let $\Pi' = D_I A^\mathsf{T}(AD_I^2A^\mathsf{T})^{-1}AD_I$. Given $z = \pi w$ for some $w \in \mathbb{R}^n$, if we let $w' = D_I^{-1}w$ and $z' = \Pi' w'$, then $\|z'\| \leq 2^{|I|}\|z\|$.*

Proof. We only need to prove the lemma for the case where $|I| = 1$, say $I = \{k\}$. In particular, $AD_I^2A^\mathsf{T} = AA^\mathsf{T} - (3/4)a_k a_k^\mathsf{T}$. By the Sherman-Morrison formula [19])(i.e. $(B + uv^\mathsf{T})^{-1} = B^{-1} - \dfrac{B^{-1}uv^\mathsf{T}B^{-1}}{1 + v^\mathsf{T}B^{-1}u}$ for every non-singular square matrix B and vectors u, v),

$$(AD_I^2(A)^\mathsf{T})^{-1} = (AA^\mathsf{T})^{-1} + \frac{(AA^\mathsf{T})^{-1}a_k a_k^\mathsf{T}(AA^\mathsf{T})^{-1}}{4/3 + \|\pi_k\|^2}.$$

(since $a_k^\mathsf{T}(AA^\mathsf{T})^{-1}a_k = \pi_{kk} = \|\pi_k\|^2$). It follows that

$$\Pi' = D_I\left(P + \frac{3\pi_k\pi_k^\mathsf{T}}{4 - 3\|\pi_k\|^2}\right)D_I,$$

therefore

$$\|z'\|^2 = w'^\mathsf{T}\Pi' w' = \|z\|^2 + \frac{3z_k^2}{4 - 3\|\pi_k\|^2} \leq 4\|z\|^2,$$

where the last inequality follows from $z_k^2 \leq \|z\|^2$ and $\|\pi_k\| \leq 1$. □

Lemma 3.5. *Let $\varepsilon' = 1/(16\sqrt{3nm})$. Let $z \in \mathcal{L}^\perp$ such that $z_j \geq -\varepsilon'\|z\|\|\pi_j\|$ for all $j \in [n]$. Let $I = \{j \in [n] : \frac{z_j}{\|z\|} > \frac{1}{\sqrt{3n}}\}$, and $\Pi' = D_I A^\mathsf{T}(AD_I^2A^\mathsf{T})^{-1}AD_I$. Then*

$$\widehat{\text{vol}}(Q_{\Pi'}) \geq e^{1/8}\,\widehat{\text{vol}}(Q_\Pi).$$

Proof. We assume $\widehat{\text{vol}}(Q_\Pi) > 0$, otherwise the statement is trivial. Observe that, if we define $H = A^\mathsf{T}(AA^\mathsf{T})^{-1}$ and $Q_A = \text{conv}(A) \cap \text{conv}(-A)$, then $Q_\Pi = HQ_A$, thus

$$\widehat{\text{vol}}(Q_\Pi) = \text{vol}(Q_A)\det(H^\mathsf{T}H)^{1/2} = \text{vol}(Q_A)\det(AA^\mathsf{T})^{-1/2}. \tag{12}$$

The statement will follow by proving the next two inequalities.

$$\ln(\det(AD_I^2A^\mathsf{T})) \leq \ln(\det(AA^\mathsf{T})) - \frac{1}{2}; \tag{13}$$

$$\text{vol}(Q_A) \leq (1 + 1/(8m))^m\,\text{vol}(Q_{AD_I}). \tag{14}$$

We prove (13). Note that $D_I^2 = I - 3/4\sum_{i\in I}\boldsymbol{e}_i\boldsymbol{e}_i^\mathsf{T}$, thus $AD_I^2A^\mathsf{T} = AA^\mathsf{T} - 3/4\sum_{i\in I}a_i a_i^\mathsf{T}$. Recalling that the Jacobian of $\ln\det(X)$ is X^{-1} and that $\ln\det(X)$ is concave, we have that

$$\begin{aligned}\ln(\det(AD_I{}^2A^\mathsf{T})) &\leq \ln(\det(AA^\mathsf{T})) - (3/4)\text{trace}((AA^\mathsf{T})^{-1}(\sum_{i\in I}a_i a_i^\mathsf{T})) \\ &= \ln(\det(AA^\mathsf{T})) - (3/4)\sum_{i\in I}\|\pi_i\|^2.\end{aligned}$$

Hence, to prove (13) it suffices to show that $\sum_{i\in I}\|\pi_i\|^2 \geq 2/3$. Note that $(z_i/\|z\|)^2 < 1/(3n)$ for all $i \notin I$, therefore

$$\sum_{i\in I}\|\pi_i\|^2 \geq \sum_{i\in I}\frac{(\pi_i^{\mathsf{T}} z_i)^2}{\|z\|^2} = \sum_{i\in I}\frac{z_i^2}{\|z\|^2} = 1 - \sum_{i\in[n]\setminus I}\frac{z_i^2}{\|z\|^2} \geq 1 - n(1/\sqrt{3n})^2 = \frac{2}{3},$$

where the first inequality follows from Cauchy-Schwartz.

We prove (14). Since $\pi_i z = z_i$ and $\|\pi_i\| \leq 1$ for all $i \in [n]$, the assumption implies $\pi_i^{\mathsf{T}}\hat{z} \geq -\varepsilon'$ $\forall i \in [n]$. This implies that, for all $v \in \mathrm{conv}(\Pi)$, $\hat{z}^{\mathsf{T}} v \geq -\varepsilon'$.
Consider now $y \in Q_A$, and let $v = Hy$. It follows that $v \in Q_\Pi$, thus $-v \in \mathrm{conv}(\Pi)$. The previous argument implies $\hat{z}^{\mathsf{T}} v \leq \varepsilon'$. We may write $y = \sum_{i=1}^n \lambda_i a_i$ where $\lambda \geq 0$, $\boldsymbol{e}^{\mathsf{T}}\lambda = 1$. We then have $\varepsilon' \geq \hat{z}^{\mathsf{T}} v = \hat{z} HAv = \hat{z}^{\mathsf{T}}\Pi\lambda = \hat{z}^{\mathsf{T}}\lambda$. Now note that

$$y = \sum_{i\in[n]}\lambda_i a_i = \sum_{i\in I} 2\lambda_i(a_i/2) + \sum_{i\in[n]\setminus I}\lambda_i a_i,$$

hence to prove the statement, since $0 \in \mathrm{conv}(AD_I)$, it suffices to show that $\sum_{i\in I}\lambda_i \leq 1/(8m)$. Assume not, then we see that

$$\hat{z}^{\mathsf{T}}\lambda = \sum_{i\in I}\lambda_i\frac{z_i}{\|z\|} + \sum_{i\in[n]\setminus I}\lambda_i\frac{z_i}{\|z\|} > \frac{1}{8m}\frac{1}{\sqrt{3n}} - \varepsilon' = 2\varepsilon' - \varepsilon' = \varepsilon',$$

a contradiction.
Finally, by equations (12)(13)(14), since

$$\begin{aligned}\ln\widehat{\mathrm{vol}}(Q_{\Pi'}) &= \ln\left(\mathrm{vol}(P_{A'})\det(AD^2A^{\mathsf{T}})^{-1/2}\right)\\ &\geq \ln\left(\mathrm{vol}(P_A)\det(AA^{\mathsf{T}})^{-1/2}\right) + 1/4 - m\ln(1+1/(8m))\\ &\geq \ln\widehat{\mathrm{vol}}(Q_\Pi) + 1/8\end{aligned}$$

which implies the statement. □

Proposition 2.4. *If* $\mathrm{conv}(A)$ *contains the origin in its interior, then* $\mathrm{conv}(A) \supseteq B(0, 2^{-2L})$ *and* $|\rho_A| \geq 2^{-3L}$.

Proof. By Proposition 2.4, $\mathrm{vol}(Q_A) \geq 2^{-2mL}V_0$, while by Proposition (11), $\det(AA^{\mathsf{T}})^2 \leq 2^{3L}$. It follows that $\widehat{\mathrm{vol}}(Q_\Pi) = \mathrm{vol}(Q_A)\det(AA^{\mathsf{T}})^{-1} \geq 2^{-2mL}2^{-3/2L} \geq 2^{-3mL}$. □

B Finding a maximal support solution with Algorithm 1

We now turn to the general case of finding the maximum support a solution to

$$\begin{aligned} Ax &= 0\\ x &\geq 0. \end{aligned} \tag{15}$$

Throughout this section, we denote $S := \mathrm{supp}(\mathcal{L}_+)$. Algorithm 1 addressed the case $S = [n]$ only. The reason we need this assumption is that if $S \neq [n]$ then P_A is lower dimensional and hence its volume is 0. However, we show that running the same algorithm, with a different choice of ε and N, can provide further information that enables solving the maximum support problem. We define the parameters

$$\varepsilon := \frac{1}{12m^{3/2}}, \quad N := 12m^2L, \quad K = 2^{3mL}e^{N/4m} \tag{16}$$

and keep the same δ as in (6).
In the remainder, for $H \subseteq [n]$, A_H denotes the submatrix of A corresponding to the column set H. Let $\alpha_j = \|a_j\|$ denote the length of the j'th column of the original matrix. We avoid normalizing the columns to maintain integrality of the matrix.

Algorithm 3

Input: *A matrix $A \in \mathbb{Z}^{m\times n}$ with rank m, and $\|a_j\| = \alpha_j$.*

Output: *A maximum support solution to* (15).

Set $H = [n]$.
While *$H \neq \emptyset$, do*
Run Algorithm 1 with input matrix A_H and parameters ε, N as in (16).
If *a solution x is returned,* **then** *STOP*
Output $S = H$ and the solution x to (15)*, extended by $x_i = 0$ for $i \in [n] \setminus H$.*
If *a vector $y \neq 0$, $A_H^\mathsf{T} y \geq 0$ is returned,* **then**
Set $H = H \setminus \{i : a_i^\mathsf{T} y > 0\}$ and repeat.
If *$\|y\| > \delta$ at termination* **then**
Set $H = \{i \in H : \|a_i\| < K\alpha_i\}$ and repeat.
Endwhile;
Output $S = \emptyset$.

The overall algorithm (Algorithm 3) runs Algorithm 1 in the previous section multiple times, for a subset of columns of the matrix A. We start with the entire matrix A, and removes columns that turn out not to be in the set S; the current column set $H \subseteq [n]$ will provably contain the entire S.
If Algorithm 1 finds a full support solution for H, then we conclude $S = H$ and return the maximum support solution. If a nonzero vector y is found with $A^\mathsf{T} y \geq 0$, then we may remove all indices i with $a_i^\mathsf{T} y > 0$ from H as they cannot be in S. If the algorithm does not terminate with either output within N rescalings, then we examine the length of the vectors after all the rescalings. We remove every i from S whose lengths

increased by at least a factor K as compared to the original input. The correctness of the algorithm is based on the following Theorem.

Theorem B.1. *Assume Algorithm 1 does not terminate within N rescaling steps, where the parameters in* (16) *are used. Then $\|a_i\| > K\alpha_i$ implies $i \notin S$. Further,* $\mathrm{rk}\{a_i : \|a_i\| < K\alpha_i\} < m$.

The theorem implies that $S \subseteq H$ is maintained in every step of the algorithm. Furthermore, the dimension of H reduces by at least one in every iteration, and hence Algorithm 1 will terminate within m iterations. (Note that the dimension of H also decreases in the case when a y is found with $A^\mathsf{T} y \geq 0$.)
One can easily modify the argument in the proof of Theorem 2.1 to see that the number of iterations in Algorithm 1 with the modified values of ε and N is $O(m^5 L)$. Hence the overall running time of Algorithm 3 is $O(m^6 nL + m^2 n^2 L)$.
The intuition behind Theorem B.1 is the following. The polytope P_A might be contained in a subspace X and hence have volume 0. If the rescaling vector v falls into this subspace or has a large projection to X, we can argue that the relative volume of P_A increases significantly. If v is almost orthogonal to X, then P_A may even decrease; however, only by a small amount. The length of a vector a_i becoming very large during the N rescalings is equivalent to saying that on average the rescaling vectors had a large projection to the direction of a_i. If $i \in S$, then this means that the rescaling vector on average had large projection to X and hence the relative volume of P_A must have increased beyond the volume of the unit ball, a contradiction. Therefore we can conclude $i \notin S$.
For the second part, assume there is a full dimensional set of vectors a_i which remained shorter than K. In this case, we use a volume argument not on P_A, but on the full dimensional parallelepiped defined by these vectors $\hat{a}_i$. Since the rescaling vector on average had a small projection on them, one can derive a contradiction by giving a lower bound on the volume increase over the sequence of rescalings.

B.1 Analyzing the relative volume

For any set $D \subseteq \mathbb{R}^m$, we let $\mathrm{span}(D)$ denote the smallest linear subspace containing D.
Let $X = \mathrm{span}(P_A)$ denote the subspace containing P_A. We will analyze the relative volume of P_A in X, which we denote by $\widehat{\mathrm{vol}}(P_A)$. We have already used that if $S = [n]$ then P_A is full-dimensional, that is, $X = \mathbb{R}^m$. The following Lemma extends this observation to the general case.

Lemma B.2. $\mathrm{span}(P_A) = \mathrm{span}\{a_i : i \in S\}$, *and* $P_A = P_{A_S}$.

Proof. Without loss of generality we can assume $\|a_i\| = 1$, hence $\hat{a}_i = a_i$. The first claim is equivalent to showing that for every $i \in [n]$, there exists an $\alpha > 0$ such that $\alpha a_i \in P_A$ if and only if $i \in S$.
Consider first an index $i \notin S$. For a contradiction, assume that $\alpha a_i \in P_A \subseteq \mathrm{conv}(-A)$ for some $\alpha > 0$. That is, $\alpha a_i = -\sum_j \lambda_j a_j$ for some

coefficients $\lambda \geq 0$. Setting $x_j = \lambda_j$ if $j \neq i$ and $x_i = \alpha + \lambda_i$ gives a solution to (15) with $x_i > 0$, a contradiction to $i \notin S$.
Let us now take an index $i \in S$. Let x be a maximum support solution to (15) with $\sum_i x_i = 1$; in particular, $x_i > 0$. We observe that $x_i a_i \in P_A$. Indeed, $x_i a_i \in \mathrm{conv}(A)$, and $x_i a_i = -\sum_{j \neq i} x_j a_j \in -\mathrm{conv}(A)$. (Here we use that $i \in S$ implies $S \neq \emptyset$ and therefore $|S| \geq 2$, and $0 \in \mathrm{conv}(A)$).
For the second claim, consider a vector $z \in P_A$. This is equivalent to the existence of convex combinations λ and μ such that $z = \sum \lambda_i a_i = -\sum \mu_j a_j$. The claim follows by showing that λ_i or μ_i can be positive only if $i \in S$. This holds since $x = \lambda + \mu$ is a solution to (15), with $S \supseteq \mathrm{supp}(x) = \mathrm{supp}(\lambda) \cup \mathrm{supp}(\mu)$. □

The following Lemma naturally extends Lemma 2.2. Let $d = |S|$; according to the previous Lemma, $\dim(X) = d$.

Lemma B.3. *Assume $S \neq \emptyset$, or equivalently, $0 \in \mathrm{conv}(A)$ and let $\varepsilon > 0$. Let $v \in \mathbb{R}^m$ be such that $\|v\| = 1$ and $\hat{a}_j^\mathsf{T} v \geq -\varepsilon$, for all $j \in S$. Let v_X denote the orthogonal projection of v onto X, and let $\bar{\varepsilon} = \min\{\varepsilon, \|v_X\|\}$. Let $A' = (I + vv^\mathsf{T})A$. Then*

$$\widehat{\mathrm{vol}}(P_{A'}) \geq \widehat{\mathrm{vol}}(P_A)\sqrt{\frac{1 + 3\|v_X\|^2}{(1 + 6\bar{\varepsilon}\|v_X\|)^d}}.$$

Proof. The proof follows the same lines as Lemma 2.2. Let $T = (I + vv^\mathsf{T})A$.

Claim. $TP_A \subseteq \sqrt{1 + 6\bar{\varepsilon}\|v_X\|}P_{A'}$.

Proof. Consider an arbitrary point $z \in P_A$. We must show $\pm Tz \in \sqrt{1 + 6\bar{\varepsilon}\|v_X\|}\mathrm{conv}(\hat{A}')$; by symmetry of P_A, we may restrict to proving this containment for Tz.
Using $P_A = P_{A_S}$ shown in Lemma B.2, we may write $z = \sum_{i \in S} \lambda_i \hat{a}_i$ for a convex combination λ. From this, we see that

$$|v^\mathsf{T}\hat{a}_i| = |v_X^\mathsf{T}\hat{a}_i| \leq \|v_X\|\|\hat{a}_i\| = \|v_X\| \ \forall i \in S.$$

Since λ is a convex combination we have that $v^\mathsf{T} z \geq -\min\{\varepsilon, \|v_X\|\} = -\bar{\varepsilon}$. Furthermore, by symmetry, we also have that $v^\mathsf{T} z \leq \bar{\varepsilon}$. The same computation as in (10) gives

$$Tz = \sum_{i \in S} \lambda_i \sqrt{1 + 3(v^\mathsf{T}\hat{a}_i)^2}\hat{a}_i'.$$

Since $0 \in \mathrm{conv}(\hat{A}')$, to show that $z \in \sqrt{1 + 6\bar{\varepsilon}\|v_X\|}\mathrm{conv}(\hat{A}')$ it suffices to show that

$$\sum_{i \in S} \lambda_i \sqrt{1 + 3(v^\mathsf{T}\hat{a}_i)^2} \leq \sqrt{1 + 3\|v_X\|(2\bar{\varepsilon})} = \sqrt{1 + 6\|v_X\|\bar{\varepsilon}}. \tag{17}$$

This is a consequence of Lemma A.1, since the above is of the form $\mathbb{E}[\sqrt{1 + 3X^2}]$ where X is a random variable supported on $[-\bar{\varepsilon}, \|v_X\|]$ and $|\mathbb{E}[X]| = |\sum_{i \in S} \lambda_i (v^\mathsf{T}\hat{a}_i)| = |v^\mathsf{T} z| \leq \bar{\varepsilon}$ (note that z and $-z$ are in P_A). □

Using this claim, and that P_A and $P_{A'}$ have dimension d, we see that

$$\widehat{\mathrm{vol}}(P_{A'}) \geq \widehat{\mathrm{vol}}\left(\frac{TP_A}{\sqrt{1+6\bar{\varepsilon}\|v_X\|}}\right) = \frac{\widehat{\mathrm{vol}}(TP_A)}{(1+6\bar{\varepsilon}\|v_X\|)^{d/2}}.$$

It remains to show that $\widehat{\mathrm{vol}}(TP_A) = \widehat{\mathrm{vol}}(P_A)\sqrt{1+3\|v_X\|^2}$. Let U be a matrix whose columns define an orthonormal basis of X. From here, note that $v_X = UU^\mathsf{T}v$ and $\|v_X\| = \|U^\mathsf{T}v\|$. Now the amount by which T dilates volume when restricted to X is exactly equal to the volume of the parallelepiped generated by the columns of TU, which is

$$\sqrt{\det((TU)^\mathsf{T}(TU))} = \sqrt{\det(I+3U^\mathsf{T}vv^\mathsf{T}U)} = \sqrt{1+3\|U^\mathsf{T}v\|^2} = \sqrt{1+3\|v_X\|^2}.$$

□

We now analyze the sequence of rescalings during Algorithm 1 with ε and N as in (16). Let $A^t = [a_1^t, \ldots, a_n^t]$ be the current matrix A after t rescalings (so at the beginning $A = [a_1^0, \ldots, a_n^0]$), and let y^t be the vector used to determine the tth rescaling, so that $a_j^t = (I + \hat{y}^t(\hat{y}^t)^\mathsf{T})a_j^{t-1}$ for $j \in [n]$. According to the rescaling condition in the algorithm, for every $t \in [N]$ and $j \in [n]$, we have $(\hat{a}_j^t)^\mathsf{T}y^t \geq -\varepsilon$. Let $X^t = \mathrm{span}(P_{A^t})$ and let η_t denote the length of the orthogonal projection of $\hat{y}^t$ onto X^t.
If we let $w_{jt} := (\hat{y}^t)^\mathsf{T}\hat{a}_j^t$, a simple calculation shows that

$$\|a_j^t\| = \|a^{t-1}\|\sqrt{1+3w_{jt}^2}. \tag{18}$$

Let $r := \max\{\tau : B(0,\tau) \cap X^0 \subseteq \mathrm{conv}(A^0)\}$. Recall that, if $S = [n]$, then $r = |\rho_A|$, where ρ_A is the condition measure of the original matrix A^0 defined in (4), and that in this case $r \geq 2^{-3L}$ by Proposition 2.4. The proof of Proposition 2.4 can be easily modified to prove that the bound $r \geq 2^{-3L}$ holds even in the case $\emptyset \subsetneq S \subsetneq [n]$. Recall also that $\alpha_i = \|a_i^0\|$.

Lemma B.4. *Assume $S \neq \emptyset$, and for $\kappa \in [N]$, let $\bar{\eta} = \sum_{t=1}^{\kappa}\eta_t/\kappa$. Then the following hold:*

(i) If $\bar{\eta} \geq 12\varepsilon d$, then $\widehat{\mathrm{vol}}(P_{A_\kappa}) \geq \widehat{\mathrm{vol}}(P_{A^0})e^{\kappa\bar{\eta}^2/4}$.

(ii) If $\kappa \geq 4d\frac{\ln(1/r)}{\gamma^2}$, then $\bar{\eta} \leq \max\{\gamma, 12\varepsilon d\}$. In particular, for $\kappa = N$, we have $\bar{\eta} \leq 1/\sqrt{m}$.

(iii) $\|a_i^\kappa\| \leq \alpha_i e^{\kappa 3d\varepsilon\bar{\eta}}/r^d, \quad \forall i \in S$.

Proof.

(i) By Lemma B.3, we have that

$$\widehat{\mathrm{vol}}(P_{A^\kappa}) \geq \widehat{\mathrm{vol}}(P_{A^0})\prod_{t=1}^{\kappa}\sqrt{\frac{1+3\eta_i^2}{(1+6\varepsilon\eta_t)^d}}\ .$$

Using the inequalities $1+x \leq e^x\ \forall x \in \mathbb{R}$, and $1+x \geq e^{x/3}$, $\forall x \in [0,3]$, we see that

$$\prod_{t=1}^{\kappa}\sqrt{\frac{1+3\eta_t^2}{(1+6\varepsilon\eta_t)^d}} \geq e^{\sum_{t=1}^{\kappa}\eta_t^2/2-3d\varepsilon\eta_t}\ .$$

Since $\sum_{t=1}^{\kappa} \eta_t^2$ subject to $\sum_{t=1}^{\kappa} \eta_t/\kappa = \bar{\eta}$ is minimized when $\eta_t = \bar{\eta}$, $\forall t \in [\kappa]$, we see that

$$e^{\sum_{t=1}^{\kappa} \eta_t^2/2 - 3d\varepsilon\eta_i} \geq e^{\kappa(\bar{\eta}^2/2 - 3d\varepsilon\bar{\eta})} = e^{\kappa\bar{\eta}(\bar{\eta}/2 - 3d\varepsilon)}.$$

The statement follows by noting that if $\bar{\eta} \geq 12\varepsilon d$, then $\bar{\eta}(\bar{\eta}/2 - 3d\varepsilon) \geq \bar{\eta}^2/4$.

(ii) Noting that P_A^{κ} is contained inside a d-dimensional unit ball and that P_{A^0} contains a d-dimensional ball of radius r, we see that

$$\frac{\widehat{\mathrm{vol}}(P_{A^\kappa})}{\widehat{\mathrm{vol}}(P_{A^0})} \leq \frac{1}{r^d}. \tag{19}$$

Consider now $\kappa \geq 4d\ln(1/r)/\gamma^2$ and assume for a contradiction that $\bar{\eta} > \max\{\gamma, 12\varepsilon d\}$. Since then $\bar{\eta} \geq 12\varepsilon d$, by the first part of the Lemma

$$\widehat{\mathrm{vol}}(P_{A^N}) \geq \widehat{\mathrm{vol}}(P_{A^0}) e^{\kappa(\bar{\eta}^2/4)} > \widehat{\mathrm{vol}}(P_{A^0}) e^{(4d\ln(1/r)/\gamma^2)(\gamma^2/4)} = \frac{1}{r^d}\widehat{\mathrm{vol}}(P_{A^0}) \ ,$$

a contradiction to (19). Hence $\bar{\eta} \leq \max\{\gamma, 12\varepsilon d\}$ as needed.
If $\kappa = N$, since $r \geq 2^{-3L}$ and $m \geq d$, we have

$$N = 12m^2L \geq \frac{4d\ln(1/r)}{(1/\sqrt{m})^2}.$$

The result now follows from the first part and the fact that $1/\sqrt{m} \geq 12\varepsilon d$.

(iii) Select any $t \in [\kappa]$ and $j \in S$. Note that $w_{jt} \leq \eta_t$ holds. Using Lemma B.3 and (18), we obtain

$$\begin{aligned}\frac{\|a_j^t\|}{\widehat{\mathrm{vol}}(P_{A^t})} &= \frac{\|a_j^{t-1}\|\sqrt{1+3w_{jt}^2}}{\widehat{\mathrm{vol}}(P_{A^t})} \leq \frac{\|a_j^{t-1}\|}{\widehat{\mathrm{vol}}(P_{A^{t-1}})}\sqrt{\frac{1+3w_{jt}^2}{1+3\eta_i^2}}(1+6\varepsilon\eta_t)^{d/2} \\ &\leq \frac{\|a_j^{t-1}\|}{\widehat{\mathrm{vol}}(P_{A^{t-1}})}(1+6\varepsilon\eta_t)^{d/2} \leq \frac{\|a_j^{t-1}\|}{\widehat{\mathrm{vol}}(P_{A^{t-1}})}e^{3d\varepsilon\eta_t}.\end{aligned}$$

Applying the above iteratively, and using (19), we get that

$$\|a_j^\kappa\| \leq \|a_j^0\|\frac{\widehat{\mathrm{vol}}(P_{A^\kappa})}{\widehat{\mathrm{vol}}(P_{A^0})}\prod_{i=1}^{\kappa} e^{3\varepsilon\eta_i} \leq \alpha_j \frac{e^{\kappa 3d\varepsilon\bar{\eta}}}{r^d} \ ,$$

completing the proof. □

The next simple claim gives the change of the volume of a full dimensional parallelepiped during a rescaling step.

Claim. Let $B \subseteq [n]$ be a basis of A and let A_B be the corresponding submatrix. Given $v \in \mathbb{R}^m$, such that $\|v\| = 1$, let $T = (I + vv^{\mathsf{T}})$, and let $A' = TA$. Then

$$\det(\hat{A}'_B) = \det(\hat{A}_B)\frac{2}{\prod_{j\in B}\sqrt{1+3(v^{\mathsf{T}}\hat{a}_j)^2}}.$$

We are ready to prove Theorem B.1.

Proof (Proof of Theorem B.1). The first part is straightforward using parts (ii) and (iii) of Lemma B.4. Indeed, we have $\bar{\eta} \leq 1/\sqrt{m}$, $r \geq 2^{-3L}$, $d \leq m$. Therefore the bound in (iii) is at most K.
Let us now turn to the second part. For a contradiction, assume that there exists a basis $B \subseteq [n]$ for A^N such that $\|a_j^N\| \leq K\alpha_j$ for all $j \in B$. Applying Claim B.1 iteratively for the sequence of N rescalings we obtain

$$\det(\hat{A}_B^N) = \frac{2^N \det(\hat{A}_B^0)}{\prod_{t=1}^N \prod_{j\in B} \sqrt{1+3w_{jt}^2}}. \tag{20}$$

It follows that

$$2^N |\det(\hat{A}_B^0)| \leq \prod_{t=1}^N \prod_{j\in B} \sqrt{1+3w_{jt}^2} = \prod_{j\in B} \frac{\|a_j^N\|}{\alpha_j} \leq K^m = 2^{3m^2L} e^{N/4} , \tag{21}$$

where the first inequality follows from (20) and the fact that $|\det(\hat{A}_B^N)| \leq 1$, the first equality follows from (18), and the second inequality follows from the choice of B.
Since A^0 is an integral matrix, it follows that $|\det(\hat{A}_B^0)| \geq 2^{-L}$. Since $e^{1/4} < 4/3$, (21) implies that $N \leq \frac{1+3m^2}{\log_2(3/2)} L$, a contradiction. □