List Decodable Mean Estimation in Nearly Linear Time

Yeshwanth Cherapanamjeri, Sidhanth Mohanty, Morris Yau
Electrical Engineering and Computer Science
University of California at Berkeley
Berkeley, USA
{*yeshwanth,sidhanthm,morrisyau*}*@berkeley.edu*

***Abstract*—Learning from data in the presence of outliers is a fundamental problem in statistics. Until recently, no computationally efficient algorithms were known to compute the mean of a high dimensional distribution under natural assumptions in the presence of even a small fraction of outliers. In this paper, we consider robust statistics in the presence of overwhelming outliers where the majority of the dataset is introduced adversarially. With only an $\alpha < 1/2$ fraction of "inliers" (clean data) the mean of a distribution is unidentifiable. However, in their influential work, [1] introduces a polynomial time algorithm recovering the mean of distributions with bounded covariance by outputting a succinct list of $O(1/\alpha)$ candidate solutions, one of which is guaranteed to be close to the true distributional mean; a direct analog of 'List Decoding' in the theory of error correcting codes. In this work, we develop an algorithm for list decodable mean estimation in the same setting achieving up to constants the information theoretically optimal recovery, optimal sample complexity, and in nearly linear time up to polylogarithmic factors in dimension. Our conceptual innovation is to design a descent style algorithm on a nonconvex landscape, iteratively removing minima to generate a succinct list of solutions. Our runtime bottleneck is a saddle-point optimization for which we design custom primal dual solvers for generalized packing and covering SDP's under Ky-Fan norms, which may be of independent interest. We refer the reader to [2] for the full version of this paper.**

***Keywords*-High-dimensional Statistics, Robust Estimation, Semidefinite Programming, Mathematical Optimization, Spectral Methods.**

I. Introduction

Estimating the mean of data is a cardinal scientific task. The population mean can be shifted arbitrarily by a single outlier, a problem which is compounded in high dimensions where outliers can conspire to destroy the performance of even sophisticated estimators of central tendency. Robust statistics, beginning with the works of Tukey and Huber [3], [4], endeavors to design, model, and mitigate the effect of data deviating from statistical assumptions [5].

A canonical model of data corruption is the Huber contamination model [4]. Let $I(\mu)$ be a probability distribution parameterized by μ. We say a dataset $X_1, X_2, ..., X_N$ is α-Huber contaminated for some constant $\alpha \in [0, 1]$ if it is drawn i.i.d from

$$X_1, X_2, ..., X_N \sim \alpha\mathcal{I}(\mu) + (1-\alpha)\mathcal{O}$$

where $\mathcal{O}$ is an arbitrary outlier distribution which can be adversarial and dependent on $\mathcal{I}(\mu)$. The goal is to estimate μ with an estimator $\hat{\mu}$ such that the two are close with respect to a meaningful metric. The Huber contamination model captures the setting where only an α fraction of the dataset is subject to statistical assumptions. One would hope to design estimators $\hat{\mu}$ for which α is as small as possible thereby tolerating the largest fraction of outliers–a quantity known as the breakdown point . The study of estimators with large breakdown points is the focus of a long and extensive body of work, which we do not attempt to survey here. For review see [5], [6].

A first observation, is that the breakdown point of a single estimator must be smaller than $\frac{1}{2}$. For concreteness, consider the problem of estimating the mean of a standard normal. The adversary can set up a mixture of $\frac{1}{\alpha}$ standard normals for which the means of the mixture components are far apart. This intrinsic difficulty also gives rise to a natural notion of recovery in the presence of overwhelming outliers. Instead of outputting a single estimator, consider outputting a list of candidate estimators $\mathcal{L} = \{\hat{\mu}_1, \hat{\mu}_2, ..., \hat{\mu}_{\frac{1}{\alpha}}\}$ with the guarantee that the true μ is amongst the elements of the list. This is the setting of 'List Decodable Learning' [7], [1], analogous to list decoding in the theory of error correcting codes.

In their influential work [1] introduces list decodable learning in the context of robust statistics. They consider the problem of estimating the mean μ of a d-dimensional distribution $\mathcal{I}(\mu)$ with a bounded covariance $\mathbf{Cov}(\mathcal{I}(\mu)) \preceq \sigma^2 I$ for a constant σ from $N = \frac{d}{\alpha}$ samples. Their algorithm recovers a list $\mathcal{L}$ of $O(\frac{1}{\alpha})$ candidate means with the guarantee that there exists a $\hat{\mu}^* \in \mathcal{L}$ achieving the recovery guarantee $\|\hat{\mu}^* - \mu\| \leqslant O\left(\sigma\sqrt{\frac{\log(\frac{1}{\alpha})}{\alpha}}\right)$ with high probability $1 - \frac{1}{\text{poly}(d)}$. Furthermore, their algorithm is 'efficient', running in time $\text{poly}(N, d, \frac{1}{\alpha})$ via the polynomial time solvability of ellipsoidal convex programming.

A. Results

Our first contribution is an algorithm for list decodable mean estimation of covariance bounded distributions, which outputs a list $\mathcal{L}$ of length $O(\frac{1}{\alpha})$, achieving (up to constants) the information theoretically optimal recovery $O(\frac{\sigma}{\sqrt{\alpha}})$, with

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00022

linear sample complexity $N = \frac{d}{\alpha}$, and running in nearly linear time $\widetilde{O}(Nd\text{poly}(\frac{1}{\alpha}))$ where $\widetilde{O}$ omits logarithmic factors in d. For the matching minimax $\Omega(\frac{\sigma}{\sqrt{\alpha}})$ lower bound see [8]. Formally, we state our main theorem.

Theorem I.1. *Let $\mathcal{I}(\mu)$ be a distribution in $\mathbb{R}^d$ with unknown mean $\mu \in \mathbb{R}$ and bounded covariance $\mathbf{Cov}(\mathcal{I}(\mu)) \preceq \sigma^2 I$ for a constant $\sigma \in \mathbb{R}^+$. Let $\mathcal{I} := \{x_1, x_2, ..., x_{\alpha N}\}$ be a dataset in $\mathbb{R}^d$ drawn i.i.d from $\mathcal{I}(\mu)$. An adversary then selects an arbitrary dataset in $\mathbb{R}^d$ denoted $\mathcal{O} := \{x'_1, x'_2, ..., x'_{(1-\alpha)N}\}$ which in particular, may depend on $\mathcal{I}$. The algorithm is presented with the full dataset $X := \mathcal{I} \cup \mathcal{O}$. For any $N \geqslant \frac{d}{\alpha}$, we give an algorithm outputs a list $\mathcal{L} = \{\hat{\mu}_1, \hat{\mu}_2, ..., \hat{\mu}_{O(\frac{1}{\alpha})}\}$ of length $O(\frac{1}{\alpha})$ such that there exists a $\hat{\mu}^* \in \mathcal{L}$ satisfying $\|\hat{\mu}^* - \mu\| \leqslant O(\frac{\sigma}{\sqrt{\alpha}})$ with high probability $1 - \frac{1}{d^{10}}$. Furthermore, the algorithm runs in time $\tilde{O}(Nd\text{poly}(\frac{1}{\alpha}))$.*

For precise constants and failure probability see Section 4 of the full version. At a high level, we define a nonconvex cost function for which μ is an approximate minimizer and build a 'descent style' algorithm to find μ. As with most nonconvex algorithms, our approach is susceptible to falling in suboptimal minima. Our key algorithmic insight is that our algorithm fails to descend the cost function exactly when a corresponding dual procedure succeeds in "sanitizing" the dataset by removing a large fraction of outliers — a win-win.

Applications of List Decoding: First observed in [1], the list decoding problem lends itself to applications for which our algorithm offers immediate improvements. Firstly, it is perhaps surprising that a succinct list of estimators can be procured from a dataset overwhelmed by outliers. Perhaps more surprising is that the optimal candidate mean can be isolated from the list $\mathcal{L}$ with additional access to a mere $\log(\frac{1}{\alpha})$ clean samples drawn from $\mathcal{I}(\mu)$. This "semi-supervised" learning is compelling in settings where large quantities of data are collected from unreliable providers (crowdsourcing, multiple sensors, etc.). Although it is resource intensive to ensure the cleanliness of a large dataset, it is easier to audit a small, in our case $\log(\frac{1}{\alpha})$, set of samples for cleanliness. Given access to this small set of samples as side information, our algorithm returns estimators for mean estimation with breakdown points higher than $\frac{1}{2}$ in nearly linear time.

Faster list decodable mean estimation also accelerates finding planted partitions in semirandom graphs. In particular, consider the problem where G is a directed graph where the (outgoing) neighborhoods of an α fraction of vertices S are random while the neighborhoods of the remaining vertices are arbitrary, and the goal is to output $O(1/\alpha)$ lists such that one of them is "close" to S. Our algorithm for list decodable mean estimation implies a faster algorithm for this problem as well.

Lastly, list decodable mean estimation is a superset of learning mixture models of bounded covariance distributions with minimum mixture weight α. By treating a single cluster as the inliers, one can recover the list of means comprising the mixture model. Notably, this can be done without any separation assumptions between the mixture components and is robust to outliers.

Fast Semidefinite Programming:: Rapidly computing our cost function necessitates the design of new packing/covering solvers for Positive Semidefinite Programs (SDP) over general Fantopes (the convex hull of the projection matrices). Positive SDP's have seen remarkable success in areas spanning quantum computing, spectral graph theory, and approximation algorithms (See [9], [10], [11] and the references therein). Informally, a packing SDP computes the fractional number of ellipses that can be packed into a spectral norm ball which involves optimization over the spectrahedron. A natural question is whether the packing concept can be extended to balls equipped with general norms, say the sum of the top k eigenvalues (the Ky Fan norm), where for $k = 1$ we recover the oft studied spectral norm packing. We use results from Loewner's theory of operator monotonicity and operator algebras to design fast, and as far as we know the first solvers for packing/covering positive SDP's under Ky Fan norms (see Theorem 5.13 of the full version).

B. Related Work

Robust Statistics: Robust statistics has a long history [3], [12], [4], [13]. This extensive body of work develops the theory of estimators with high breakdown points, of influence functions and sensitivity curves, and of designing robust M-estimators. See [5], [6]. However, little was understood about the computational aspects of robustness which features prominently in high dimensional settings.

Recent work in theoretical computer science [14], [15] designed the first algorithms for estimating the mean and covariance of high dimensional gaussians tolerating a constant fraction of outliers in polynomial time $poly(N, d, \frac{1}{1-\alpha})$. Since then, a flurry of work has emerged studying robust regression [16], [17], sparse robust regression [18], [19], fast algorithms for robustly estimating mean/covariance [20], [21], [22], statistical query hardness of robustness [23], worst case hardness [24], robust graphical models [25], and applications of the sum of squares algorithm to robust statistics [26]. See survey [27] for an overview.

List Decodable Learning: Despite the remarkable progress in robust statistics for large α contamination, progress on the list decoding problem has been slower. This is partially owed to the intrinsic computational hardness of the problem. Even for the natural question of list decoding the mean of a high dimensional gaussian, [8] exhibits a quasipolynomial time lower bound against Statistical Query algorithms for achieving the information theoretically optimal recovery of $\Theta(\sqrt{\log(\frac{1}{\alpha})})$. This stands in contrast to

large α robust mean estimation where nearly linear time algorithms [20] achieve optimal recovery.

In light of this hardness, a natural question is to determine whether polynomial time algorithms can at least approach the optimal recovery for list decoding the mean of a gaussian. In a series of concurrent works [28] [8], develop the first algorithms approaching the $\Theta\big(\sqrt{\log(\frac{1}{\alpha})}\big)$ recovery guarantee. At a high level, both papers achieve recovery $O(\frac{\sigma}{\alpha^{c/k}})$ for different fixed constants $c > 1$ in time $\text{poly}(\frac{d}{\alpha})^{O(k)}$ for k a positive integer greater than 2. The [8] algorithm, known as the "multi-filter", is a spectral approach reasoning about high degree polynomials of the moments of data. Furthermore, the "low degree" multi-filter achieves a suboptimal $O\big(\sqrt{\frac{\log(\frac{1}{\alpha})}{\alpha}}\big)$ recovery guarantee for list decoding the mean of subgaussian distributions, which is fast and may be of practical value. [28] develop a convex hierarchy (sum of squares) style approach, which achieve similar guarantees for more general distributional families satisfying a poincare inequality. In particular for list decoding the mean of bounded covariance distributions they achieve the optimal $O(\frac{1}{\sqrt{\alpha}})$ guarantee via the polynomial time solvability of convex concave optimization. Finally, [8], [28] and a concurrent work [29] develop tools for reasoning about the high degree moments of data to break the long-standing "single-linkage" barrier in clustering mixtures of spherical gaussians.

In other statistical settings a series of concurrent works [30], [31] demonstrate information theoretic impossibility for list decoding regression even under subguassian design. Similar barriers arise in the context of list decodable subspace recovery [32], [33] where it is information theoretically impossible to list decode a dataset for which an α fraction is drawn from a subgaussian distribution in a subspace. Indeed, since list decoding is a superset of learning mixture models, these hardness considerations stem from barriers in learning mixtures of linear regressions and subspace clustering. On the other hand, the above works also construct polynomial time, $d^{\text{poly}(\frac{1}{\alpha})}$, algorithms for regression and subspace recovery for Gaussian design and Gaussian subspaces respectively, which holds true for a larger class of "certifiably anticoncentrated" distributions.

In this backdrop of computational and statistical hardness, and given the practical value of robust statistics, it is a natural challenge to design list decoding algorithms that are both fast and statistically optimal. The current work is a step in this direction.

SDP Solvers: There has been much recent interest in designing fast algorithms for positive SDP solvers due to the ubiquity of their application in approximation algorithms. We do not attempt to survey the full breadth of these results and their applications in this section. We refer the interested reader to [11], [10], [34], [9] for more context on these developments. We will restrict ourselves to the following class of SDPs relevant to our work:

$$\begin{aligned} &\max \sum_{i=1}^{n} w_i \\ &\text{s.t} \sum_{i=1}^{n} w_i A_i \preccurlyeq I \\ &\left\| \sum_{i=1}^{n} w_i B_i \right\|_k \leqslant k \end{aligned} \tag{Gen-Pack}$$

where $A_i \in \mathbb{S}_+^l$ and $B_i \in \mathbb{S}_+^m$. While existing fast solvers [34], [10], [11] only apply to the above setting when $k = 1$, we generalize the approach of [34] to for all k with running times scaling at most polynomiall in k. In particular, we show for small values of k, Gen-Pack can be solved in nearly linear time for a broad range of settings including ours and inherits the parallel, width-independent properties of [34]. See Theorem 5.13 of the full version for the exact statement of the result. However, carrying out this generalization brings with it a host of technical challenges which are explained in more detail in Section II including a more refined analysis of the power method and a novel technique to bound errors incurred in a hard-thresholding operator due to approximate eigenvector computation.

Semirandom Graph Inference: The study of problems that are typically computationally hard in the worst case in semirandom graph models was initiated by [35] and perpetuated by [36]. A specific problem of interest to us studied by [36] for which nearly optimal algorithms were given by [37] is the *semirandom independent set* problem where the set of edges between a planted independent set and the remaining (adversarially chosen) graph come from a randomized model. In a similar vein [1] studies a *planted partition* where instead of an independent set the given graph is some other sparse random graph (albeit directed). Our results improve upon the statistical guarantees of [1] as well as give faster algorithms, however both [1] and our work fall short of capturing the results of [37] due to the directed model we work in. However, we believe the hurdle is a technical point rather than an inherent shortcoming of our approach.

Sample Complexity:: The following lemma of [1] achieves linear sample complexity which suffices for our algorithm.

Lemma I.2 ([1, Proposition 1.1])**.** *Suppose $\mathcal{I}(\mu)$ is a distribution on $\mathbb{R}^d$ with mean μ and covariance $\mathbf{Cov}(I(\mu)) \preceq \sigma^2 I$ for a constant $\sigma > 0$. Then given $n \geqslant d$ samples from $\mathcal{I}(\mu)$, with probability $1 - \exp(\frac{-n}{64})$ there exists a subset $\mathcal{I} \in [n]$ of size $|\mathcal{I}| \geqslant \frac{n}{2}$ such that $\|\frac{1}{|I|}\sum_{i\in\mathcal{I}}(x_i - \mu)(x_i - \mu)^T\| \preceq 24\sigma^2 I$*

Taking $N = O\left(\frac{d}{\alpha}\right)$, for the rest of the paper we will adjust σ by a constant and assume the inlier set I satisfies

$\|\frac{1}{|I|}\sum_{i\in\mathcal{I}}(x_i-\mu)(x_i-\mu)^T\| \preceq \sigma^2 I$.

Notation: : We will frequently use $\mathbb{S}^n_+$ to denote the set of positive semidefinite matrices with dimension n. For $A \in \mathbb{S}^n_+$, we will frequently write the ordered eigenvalue decomposition of $A = \sum_{i=1}^n \lambda u_i u_i^\top$ with $\lambda_1 \geqslant \ldots \geqslant \lambda_n$ and for any matrix, M, $\sigma_i(M)$ denotes its i^{th} singular value. The *Ky–Fan matrix k-norm* of a matrix, M, is defined as the sum of the top-k singular values of M; i.e $\|M\|_k := \sum_{i=1}^k \sigma_i(M)$. Notably, $\|\cdot\|_1$ is the operator norm and $\|\cdot\|_d$ is the trace norm. However, we will stick with $\|\cdot\|$ for operator norm and $\|\cdot\|_*$ for trace norm. Continuing along these lines, we also define the *ℓ-Fantope*, denoted by $\mathcal{F}_\ell$ and characterized as $\mathcal{F}_\ell = \{M \in \mathbb{S}^n_+ : \operatorname{Tr} M = \ell \text{ and } \|M\| \leqslant 1\}$. Finally, given $\{b_i \geqslant 0\}_{i=1}^N$, we define the set $\Phi_b(\gamma) = \{w_i \geqslant 0 : \sum_i^N w_i = \gamma \text{ and } w_i \leqslant b_i\}$ and $\Phi_b = \Phi_b(1)$. For a set of vectors, $V = [v_1, \ldots, v_k]$, we will use $\mathcal{P}_V^\perp$ to denote the projection onto the orthogonal subspace of the span of v_i.

II. Techniques

First we present an inefficient algorithm for list decodable mean estimation. Although it is inefficient, it captures the core ideas and foreshadows the difficulties encountered by our efficient algorithm. At a high level, the inefficient algorithm greedily searches through the dataset for subsets of points with small covariance with the goal of finding the subset of inliers.

Inefficient Algorithm: Our inefficient algorithm is a list decoding analogy to the nonconvex weight minimization procedure first proposed in [14][1]. Let $\mathcal{L}$ be a list initialized to be the empty set. Let b be a vector initialized to be $(\frac{2}{\alpha N}, \frac{2}{\alpha N}, ..., \frac{2}{\alpha N}) \in \mathbb{R}^N$. The algorithm iterates the following loop for $\frac{2}{\alpha}$ iterations.

1) First, solve the optimization problem
$$\hat{w} = \arg\min_{w\in\Phi_b} \left\| \sum_{i=1}^N w_i (x-\mu(w))(x-\mu(w))^\top \right\| \quad (1)$$
Where $\mu(w) = \sum_{i=1}^N w_i x_i$
2) Second, append $\hat{\mu} = \sum_{i=1}^N \hat{w}_i x_i$ to $\mathcal{L}$
3) Third, update b such that $b_i = b_i - \hat{w}_i$

We claim the algorithm outputs a list $\mathcal{L}$ of length $\frac{2}{\alpha}$ and that there exists a $\hat{\mu}^* \in L$ satisfying $\|\hat{\mu}^* - \mu\| \leqslant O(\frac{\sigma}{\sqrt{\alpha}})$. Next we outline the proof of correctness.

Proof Outline: We proceed by contradiction and assume $\|\hat{\mu} - \mu\| \geqslant \frac{10\sigma}{\sqrt{\alpha}}$ for all $\hat{\mu} \in \mathcal{L}$. Consider the first iteration. The scaled indicator of the inliers $\frac{1}{\alpha N}\mathbb{1}[i \in \mathcal{I}]$ is feasible for Eq. (1). Thus, we have $\|\sum_{i=1}^N \hat{w}_i (x-\mu(\hat{w}))(x-\mu(\hat{w}))^\top\| \leqslant 4\sigma^2$. It is a fact that given two subsets of the data that are both covariance bounded, if the means of the subsets are far apart then the subsets don't overlap substantially. This fact extends beyond subsets and holds true even for the soft weights that we are considering here (see Fact A.3 of the full version). Applying this fact we conclude $\sum_{i\in\mathcal{I}} \hat{w} \leqslant \frac{\alpha}{2}$. By assumption, subsequent iterations of the algorithm continue to output $\hat{\mu}$ far away from the true mean so a substantial fraction of the inlier weight is preserved enabling the above argument to go through repeatedly. Formally, this would be argued inductively (see Corollary 4.4 of the full version). Thus, at every iteration $\sum_{i\in\mathcal{I}} \hat{w} \leqslant \frac{\alpha}{2}$. Notice that any algorithm that removes more outlier weight than inlier weight at a ratio $\sum_{i\in O} \hat{w}_i \geqslant \frac{2}{\alpha}\sum_{i\in I} \hat{w}_i$ will eventually remove all the outlier weight leaving more than $\frac{1}{2}$ of the inlier weight intact. Since the total inlier weight is initialized to be $\sum_{i\in\mathcal{I}} b_i = 2$, we have at the second to last iteration a dataset comprised entirely of inliers which implies $\|\hat{\mu} - \mu\| \leqslant \frac{10\sigma}{\sqrt{\alpha}}$, which is a contradiction.

Sanitizing the Dataset: : Abstracting the guarantees of our inefficient algorithm, we say that an algorithm "sanitizes" a dataset if it outputs a tuple $(\hat{\mu}, \hat{w})$ where $\sum_{i=1}^N \hat{w} \geqslant \Omega(1)$ satisfying the following conditions. If $\|\hat{\mu} - \mu\| \geqslant O(\frac{\sigma}{\sqrt{\alpha}})$ then $\sum_{i\in O} \hat{w}_i \geqslant \frac{2}{\alpha}\sum_{i\in I} \hat{w}_i$. Any algorithm that sanitizes the dataset iteratively, is guaranteed to succeed as a list decoding algorithm. This is made formal in Section 4 of the full version.

Descent Style Formulation:: The optimization problem Eq. (1) is nonconvex and hard to solve directly. A novel approach to minimizing Eq. (1) is to replace $\mu(w)$ with a parameter ν and define a cost function $f(\nu)$. First introduced in [20] in the context of robust mean estimation and later in robust covariance estimation [22] consider the function $f(\nu)$ defined as follows:
$$f(\nu) := \min_{w\in\Phi_b} \left\| \sum_{i=1}^N w_i (x-\nu)(x-\nu)^\top \right\|$$
where $b_i = \frac{1}{\alpha N}$ for all $i \in [N]$. This formulation has two appealing aspects. Firstly, the cost function can be computed efficiently via convex concave optimization. Indeed, the operator norm can be replaced by the maximization over its associated fantope $\mathcal{F}_1$
$$f(\nu) := \min_{w\in\Phi_b} \max_{M\in\mathcal{F}_1} \left\langle M, \sum_{i=1}^N w_i (x_i-\nu)(x_i-\nu)^\top \right\rangle.$$
Secondly, for $\alpha > \frac{2}{3}$ (robust mean estimation), a crucial insight of [20] is that $f(\nu)$ approximates the squared distance from ν to the mean μ. Then a good estimate of the mean is the minimizer of the cost.
$$\hat{\mu} := \arg\min_{\nu\in\mathbb{R}^d} f(\nu) \approx \arg\min_{\nu\in\mathbb{R}^d} \|\nu - \mu\|^2 \quad (2)$$

[1]There they directly design a separation oracle for the objective Eq. (1) for $\alpha > \frac{2}{3}$, which yields polynomial time guarantees for robust mean estimation via the ellipsoid algorithm. It is plausible that a similar approach could yield polynomial time algorithms for list decodable mean estimation, but use of the ellipsoid algorithm would preclude the possibility of fast algorithms so we do not pursue that avenue here.

In their setting the minimization in Eq. (2) can be performed by a descent style algorithm.

Substantial challenges arise when designing such a cost function for list decodable mean estimation. Chiefly, the inliers are unidentifiable from the dataset so there is no function of the data that approximates the distance to the true mean. Our solution is to design a function that either approximates the distance to the true mean, or when the approximation is poor, prove there exists a corresponding dual procedure that sanitizes the dataset. This win-win observation can be made algorithmic and is the subject of Section 4 of the full version.

A. Our Approach

Designing Cost:: We make extensive use of the Fantope [38], the convex hull of the rank ℓ projection matrices. This set of matrices, denoted $\mathcal{F}_\ell$, is a tight relaxation for simultaneous rank and orthogonality constraints on the positive semidefinite cone. This also makes it amenable to semidefinite optimization. We define

$$\mathcal{F}_\ell = \{M \in \mathbb{R}^{d\times d} : 0 \preceq M \preceq I \text{ and } \operatorname{Tr}(M) = \ell\}.$$

Optimization over the Fantope provides a variational characterization of the principal subspace of a symmetric matrix $B \in \mathbb{R}^{d\times d}$. Indeed the *Ky Fan Theorem*, states that the *Ky Fan Norm* defined to be the sum of the ℓ largest eigenvalues of a psd matrix is equal to

$$\|B\|_\ell := \sum_{i=1}^{\ell} \lambda_i(B) = \max_{Q^TQ=I_\ell} \langle B, QQ^T\rangle = \max_{M\in\mathcal{F}_\ell} \langle B, M\rangle \tag{3}$$

Here the first equality is an extremal property known as *Ky Fan's Maximum Principle*, and the second equality follows because the rank ℓ projection matrices are extremal points of $\mathcal{F}_\ell$. See [39]. We use this principle to generalize the min-max problem considered in the previous section. Let $Cost_{X,b,\ell}(\nu) : \mathbb{R}^d \to \mathbb{R}^+$ be defined

$$\begin{aligned} Cost_{X,b,\ell}(\nu) &= \min_{w\in\Phi_b} \max_{M\in\mathcal{F}_\ell} \langle \sum_{i=1}^{N} w_i(X_i-\nu)(X_i-\nu)^\top, M\rangle \\ &= \min_{w\in\Phi_b} \left\| \sum_{i=1}^{N} w_i(X_i-\nu)(X_i-\nu)^\top \right\|_\ell \end{aligned} \tag{4}$$

We call the above min-max formulation the dual and the associated minimizer w^* the dual minimizer or dual weights. By Von Neumann's min max theorem we have

$$Cost_{X,b,\ell}(\nu) = \max_{M\in\mathcal{F}_\ell} \min_{w\in\Phi_b} \langle \sum_{i=1}^{N} w_i(X_i-\nu)(X_i-\nu)^\top, M\rangle$$

Where we refer to the maximizer M^* as the primal maximizer. For the remainder of this section we will set $\ell = \frac{1}{\alpha}$ and $b = (\frac{1}{\alpha N}, ..., \frac{1}{\alpha N}) \in \mathbb{R}^N$ and drop the subscripts in $Cost(\cdot)$.

An Easier Problem: : To aid in exposition, we illustrate our algorithmic approach on the simpler and well understood problem of finding the $k = \frac{1}{\alpha}$ means of data drawn from a mixture of k bounded covariance distributions. That is $x_1, ..., x_N \sim \frac{1}{k}\sum_{i=1}^{k} \mathcal{D}(\mu_i)$ for a distribution $D(\mu) \preceq I$ with means $\{\mu_i\}_{i=1}^k$ and let T_i denote the set of points in each cluster $i \in [k]$. Consider a vector ν that is further than $O(\sqrt{k})$ away from all the means $\{\mu_i\}_{i=1}^k$. By standard duality arguments, we see that $Cost(\nu)$ is a good approximation to the distance to the closest cluster center denoted μ^* comprised of points T^*. Furthermore, $\mu^* - \nu$ is almost completely contained in the top-$O(k)$ singular subspace of M^*, denoted V. We may now project all the data points onto the affine subspace V offset to ν forming the set $X' := \{\Pi_V(x_i - \nu)\}_{i=1}^N$. The second observation is that a randomly chosen point $\bar{x} \in T^*$ satisfies $\|\Pi_V(\bar{x} - \mu^*)\| \leqslant O(\sqrt{k})$ with constant probability. Due to the fact that $\mu^* - \nu$ is almost completely contained in V, we get by picking a set of $p = \tilde{O}(k)$ random data points $R := \{x'_1, x'_2, ..., x'_p\} \in X'$, that there exists a point $\hat{x}' \in R$ substantially closer to μ^* than ν with high probability $1 - \frac{1}{\text{poly}(d)}$. We can efficiently certify this progress by computing the value of $Cost(\hat{x}')$. By iterating this procedure we converge to within $O(\sqrt{k})$ of the mean of a cluster center.

List Decoding Main Lemma: : In analogy to clustering, one should hope that for any $\nu \in R^d$ further than $O(\frac{\sigma}{\sqrt{\alpha}})$ from μ, that $Cost(\nu) \approx \|\nu - \mu\|^2$. Although this is impossible, it turns out that when it is false, there exists a corresponding "dual procedure" for outputting a sanitizing tuple. More precisely, we claim that either $0.4\|\nu-\mu\|^2 \leqslant Cost(\nu) \leqslant 1.1\|\nu-\mu\|^2$, or a simple procedure outputs a set of weights $\hat{w}$ identifying vastly more outliers than inliers i.e $\sum_{i\in\mathcal{O}} \hat{w}_i \geqslant \frac{\alpha}{2}\sum_{i\in\mathcal{I}} \hat{w}_i$, or both.

The dual procedure is as follows. Let $\widehat{\Sigma} := \sum_{i=1}^{n} w_i^*(x_i-\nu)(x_i-\nu)^T$ be the weighted second moment matrix centered at ν. Let V be the top $O(\frac{1}{\alpha})$ eigenspace of $\widehat{\Sigma}$. We project the dataset onto the affine subspace V with offset ν. We then sort the points $\{\Pi_V(x_i-\nu)\}_{i=1}^N$ by Euclidean lengths. This sorting determines an ordering of the weights $w_1^*, ..., w_N^*$. We pass through the sorted list, and find the smallest $m \in [N]$ such that $\sum_{i=1}^{m} w_i^* \geqslant 0.5$. We set $\hat{w}_i = w_i^*$ for $i = 1, ..., m$ and $\hat{w}_i = 0$ for $i > m$. The following lemma guarantees $\sum_{i\in\mathcal{O}} \hat{w}_i \geqslant \frac{\alpha}{2}\sum_{i\in\mathcal{I}} \hat{w}_i$.

Lemma II.1. *(Nonalgorithmic Filtering with Exact Cost Evaluation) Let $\nu \in \mathbb{R}^d$ be any vector satisfying $\|\nu - \mu\| \geqslant O(\frac{\sigma}{\sqrt{\alpha}})$. Let $Cost_{X,b,\ell}(\nu)$ be defined as in Eq. (4) for $b = (\frac{1}{\alpha N}, ..., \frac{1}{\alpha N}) \in \mathbb{R}^N$ and $\ell = \frac{1}{\alpha}$. Let $w^* \in \Phi_b$ be the corresponding dual minimizer. Let $\hat{w}$ be defined as follows*

$$\hat{w} := \operatorname*{arg\,min}_{p_i\in[0,w_i^*] \text{ and } \|p\|_1\geqslant 0.5} \sum_{i=1}^{N} p_i \|\Pi_V(x_i-\nu)\|$$

for V the subspace defined above. Then either the cost is a constant factor approximation to the distance to the true

mean, $0.4\|\nu-\mu\|^2 \leqslant Cost_{X,b,\ell}(\nu) \leqslant 1.1\|\nu-\mu\|^2$*, or* $\hat{w}$ *identifies a set of weights with vastly more outliers than inliers,* $\sum_{i\in\mathcal{O}} \hat{w}_i \geqslant \frac{\alpha}{2}\sum_{i\in\mathcal{I}} \hat{w}_i$ *(or both).*

In Lemma 4.5 of the full version we state the algorithmic version of the above lemma. There it is important to take into account technicalities involving the approximate evaluation of $Cost_{X,b,\ell}(\cdot)$, and provide a procedure for making progress when the cost is a constant approximation $\|\nu-\mu\|^2$. This will be done in a manner akin to the procedure for clustering described earlier. Nevertheless, Lemma II.1 captures the key guarantee that ensures our main algorithms in Section 4 succeed.

B. Generalized Packing/Covering Solvers and Improved Power Method Analysis

We start by considering the simpler problem of computing $Cost_{X,b,1}(\nu)$. The approach taken in [20] is to reduce the problem to a packing SDP via the introduction of an additional parameter λ; specifically, they solve the following packing SDP:

$$\begin{aligned} &\max_{w_i\geqslant 0}\sum_{i=1}^{N} w_i \\ \text{s.t } &\left\|\sum_{i=1}^{N} w_i(X_i-\nu)(X_i-\nu)^\top\right\| \leqslant \lambda \\ &w_i \leqslant b_i \end{aligned} \qquad \text{(Packing-SDP)}$$

for which there exist fast linear-time solvers [34], [10]. It can be shown that the value of the above program when viewed as a function of λ is monotonic, continuous and attains the value 1 precisely when $\lambda = Cost_{X,b,1}(\nu)$. Therefore, by performing a binary search over λ, one obtains accurate estimates of w^* and $Cost_{X,b,1}(\nu)$.

However, a similar approach for the problem of compute, $Cost_{X,b,\ell}(\nu)$ results in the following SDP:

$$\begin{aligned} &\max_{w_i\geqslant 0}\sum_{i=1}^{N} w_i \\ \text{s.t } &\|\sum_{i=1}^{N} w_i(X_i-\nu)(X_i-\nu)^\top\|_\ell \leqslant \lambda \\ &w_i \leqslant b_i \end{aligned}$$

which does not fall into the standard class of packing SDPs. We extend and generalize fast linear time solvers for packing/covering SDPs from [34] to this broader class of problems. However, this generalization is not straightforward.

To demonstrate the main difficulties, we will delve more deeply into the solver from [34] and state the packing/covering primal dual pairs they consider:

Covering (Primal)	Packing (Dual)
$\min_M \operatorname{Tr} M$	$\max_{w_i\geqslant 0}\sum_{i=1}^{N} w_i$
Subject to: $\langle A_i, M\rangle \geqslant 1$	Subject to: $\sum_{i=1}^{N} w_i A_i \preccurlyeq I$
$M \succcurlyeq 0$	

where $A_i \in \mathbb{S}_+^m$. The solver of [34] start by initializing a set of weights, w_i, feasible for Packing-SDP. Subsequently, in each iteration, t, they first compute the matrix

$$P_1 = \frac{\exp(\sum_{i=1}^{N} w_i A_i)}{\operatorname{Tr}\exp(\sum_{i=1}^{N} w_i A_i)}.$$

The algorithm then proceeds to increment the weights of all i such that $\langle P_1, A_i\rangle \leqslant (1+\varepsilon)$ for a user defined accuracy parameter, ε, by a multiplicative factor. Intuitively, these indices correspond to "directions", A_i, along which $\sum_{i=1}^{N} w_i A_i$ is small and therefore, their weights can be increased in the dual formulation. By incorporating a standard regret analysis from [40] for the matrices, P_1, they show that one either outputs a primal feasible, M, with $\operatorname{Tr} M \leqslant 1$ or a dual feasible w with $\sum_{i=1}^{N} w_i \geqslant (1-\varepsilon)$.

The construction of our solver follows the same broad outline as in [34]. While the regret guarantees we employ are a generalization of those used in [34], they still follow straightforwardly from standard regret bounds for mirror descent based algorithms [41]. Instead, the main challenge of our solver is computational. The matrix $P^{(t)}$ can be viewed as a maximizer to $f(X) = \langle X, F\rangle - \langle X, \log X\rangle$ where $F = \sum_{i=1}^{N} w_i A_i$ in $\mathcal{F}_1$. In our setting, we instead are required to compute the maximizer of f in $(\mathcal{F}_\ell/\ell)$ which we show is given by the following: Let H and τ^* be defined as:

$$H = \exp(F) = \sum_{i=1}^{m} \lambda_i u_i u_i^\top \text{ with } \lambda_1 \geqslant \lambda_2 \ldots \geqslant \lambda_m > 0$$

and

$$\tau^* = \max_{\tau>0}\left\{\frac{\tau^*}{\sum_{i=1}^{m}\min(\tau,\lambda_i)} = \frac{1}{\ell}\right\}.$$

Then, we have:

$$\begin{aligned} P_\ell &= \arg\max_{\ell X\in\mathcal{F}_\ell} f(X) \\ &= \frac{1}{\sum_{i=1}^{m}\min(\lambda_i,\tau^*)}\cdot\sum_{i=1}^{m}\min(\lambda_i,\tau^*)u_i u_i^\top. \end{aligned}$$

While the matrix, P_1, can be efficiently estimated by Taylor series expansion of the exponential function (see [40]), we need to estimate P_ℓ which is given by a careful truncation operation on $\exp(F)$. Note that given access to the exact

top-ℓ eigenvectors and eigenvalues of $\exp(F)$, one can efficiently obtain a good estimate of P_ℓ. However, the main technical challenge is establishing such good estimates given access only to *approximate* eigenvectors and eigenvalues of a truncated Taylor series approximation of H.

One of the main insights of our approach is that instead of analyzing the truncation operator directly, we instead view the matrix. P_ℓ as being the maximizer of $g(X, F) = \langle F, X\rangle - \langle X, \log X\rangle$ with respect to X. We then subsequently show that maximizer of $g(X, \widetilde{F})$ is close to P_ℓ for some $\widetilde{F}$ close to F which makes crucial use of the fact that $\log X$ is operator monotone. Our second main piece of insight is that if our approximate eigenvectors and eigenvalues, denoted by $(\widehat{\lambda}_i, v_i)_{i=1}^{\ell}$ satisfy:

$$(1-\varepsilon)\sum_{i=1}^{l}\widehat{\lambda}_i v_i v_i^\top + \mathcal{P}_V^\perp H \mathcal{P}_V^\perp \preccurlyeq H \preccurlyeq (1+\varepsilon)\sum_{i=1}^{l}\widehat{\lambda}_i v_i v_i^\top + \mathcal{P}_V^\perp H \mathcal{P}_V^\perp,$$

where V is the subspace spanned by the v_i and $\mathcal{P}_V^\perp$ is the projection onto the orthogonal subspace of V, then our approximate truncation operator can be viewed as the *exact* maximizer of $g(X, \widetilde{F})$ for some $\widetilde{F}$ close to F. From the previous discussion, this means that our truncation operator operating on the approximate eigenvectors v_i is a good estimate of P_ℓ. However, standard analysis of methods for the computation of eigenvalues and eigenvectors do not yield such strong guarantees [42], [43]. The final contribution of our work is a refined analysis of the power method that yields the required stronger guarantees which is formally stated in Theorem 6.1 of the full version.[2]

ACKNOWLEDGEMENTS

We would like to thank Sam Hopkins and Prasad Raghavendra for helpful conversations. We would also like to thank Bryan Chen for pointing out errors in an earlier version of this manuscript.

REFERENCES

[1] M. Charikar, J. Steinhardt, and G. Valiant, "Learning from untrusted data," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, 2017, pp. 47–60.

[2] Y. Cherapanamjeri, S. Mohanty, and M. Yau, "List decodable mean estimation in nearly linear time," 2020.

[3] J. Tukey, "A survey of sampling from contaminated distributions," 1960.

[4] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 03 1964.

[2]While our guarantees scale with $1/\varepsilon$ as opposed to $1/\sqrt{\varepsilon}$ as in [43], we suspect this dependence may be improved using techniques from [43].

[5] ——, *Robust Statistics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1248–1251.

[6] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust statistics. The approach based on influence functions*. Wiley New York, 1986.

[7] M.-F. Balcan, A. Blum, and S. Vempala, "A discriminative framework for clustering via similarity functions," in *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, ser. STOC '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 671–680.

[8] I. Diakonikolas, D. M. Kane, and A. Stewart, "List-decodable robust mean estimation and learning mixtures of spherical gaussians," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 1047–1060.

[9] S. Arora, E. Hazan, and S. Kale, "The multiplicative weights update method: a meta-algorithm and applications," *Theory of Computing*, vol. 8, no. 1, pp. 121–164, 2012.

[10] Z. Allen Zhu, Y. T. Lee, and L. Orecchia, "Using optimization to obtain a width-independent, parallel, simpler, and faster positive SDP solver," in *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, R. Krauthgamer, Ed. SIAM, 2016, pp. 1824–1831.

[11] A. Jambulapati, Y. T. Lee, J. Li, S. Padmanabhan, and K. Tian, "Positive semidefinite programming: Mixed, parallel, and width-independent," *CoRR*, vol. abs/2002.04830, 2020.

[12] J. W. Tukey, "Mathematics and the picturing of data," 1975.

[13] F. R. Hampel, "A general qualitative definition of robustness," *The Annals of Mathematical Statistics*, vol. 42, no. 6, pp. 1887–1896, 1971.

[14] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart, "Robust estimators in high dimensions without the computational intractability," in *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2016, pp. 655–664.

[15] K. A. Lai, A. B. Rao, and S. Vempala, "Agnostic estimation of mean and covariance," in *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2016, pp. 665–674.

[16] A. Klivans, P. K. Kothari, and R. Meka, "Efficient algorithms for outlier-robust regression," *arXiv preprint arXiv:1803.03241*, 2018.

[17] I. Diakonikolas, W. Kong, and A. Stewart, "Efficient algorithms and lower bounds for robust linear regression," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '19. USA: Society for Industrial and Applied Mathematics, 2019, p. 2745–2754.

[18] S. Balakrishnan, S. S. Du, J. Li, and A. Singh, "Computationally efficient robust sparse estimation in high dimensions," in *Proceedings of the 2017 Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, S. Kale and O. Shamir, Eds., vol. 65. Amsterdam, Netherlands: PMLR, 07–10 Jul 2017, pp. 169–212.

[19] I. Diakonikolas, D. Kane, S. Karmalkar, E. Price, and A. Stewart, "Outlier-robust high-dimensional sparse estimation via iterative filtering," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 10 689–10 700.

[20] Y. Cheng, I. Diakonikolas, and R. Ge, "High-dimensional robust mean estimation in nearly-linear time," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, T. M. Chan, Ed. SIAM, 2019, pp. 2755–2771.

[21] Y. Dong, S. Hopkins, and J. Li, "Quantum entropy scoring for fast robust mean estimation and improved outlier detection," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 6067–6077.

[22] Y. Cheng, I. Diakonikolas, R. Ge, and D. P. Woodruff, "Faster algorithms for high-dimensional robust covariance estimation," in *Proceedings of the Thirty-Second Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, A. Beygelzimer and D. Hsu, Eds., vol. 99. Phoenix, USA: PMLR, 25–28 Jun 2019, pp. 727–757.

[23] I. Diakonikolas, D. M. Kane, and A. Stewart, "Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures," in *In Proc. 58th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2017, pp. 73–84.

[24] S. B. Hopkins and J. Li, "How hard is robust mean estimation?" in *Proceedings of the Thirty-Second Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, A. Beygelzimer and D. Hsu, Eds., vol. 99. Phoenix, USA: PMLR, 25–28 Jun 2019, pp. 1649–1682.

[25] Y. Cheng, I. Diakonikolas, D. M. Kane, and A. Stewart, "Robust learning of fixed-structure bayesian networks," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 10304–10316.

[26] P. K. Kothari, J. Steinhardt, and D. Steurer, "Robust moment estimation and improved clustering via sum of squares," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 1035–1046.

[27] I. Diakonikolas and D. M. Kane, "Recent advances in algorithmic high-dimensional robust statistics," 2019.

[28] P. K. Kothari and J. Steinhardt, "Better agnostic clustering via relaxed tensor norms," *arXiv preprint arXiv:1711.07465*, 2017.

[29] S. B. Hopkins and J. Li, "Mixture models, robustness, and sum of squares proofs," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2018, pp. 1021–1034.

[30] P. Raghavendra and M. Yau, "List decodable learning via sum of squares," in *SODA*, 2020.

[31] S. Karmalkar, A. R. Klivans, and P. Kothari, "List-decodable linear regression," in *NeurIPS*, 2019.

[32] P. Raghavendra and M. Yau, "List decodable subspace recovery," 2020.

[33] A. Bakshi and P. Kothari, "List-decodable subspace recovery via sum-of-squares," 2020.

[34] R. Peng, K. Tangwongsan, and P. Zhang, "Faster and simpler width-independent parallel algorithms for positive semidefinite programming," *arXiv preprint arXiv:1201.5135*, 2012.

[35] A. Blum and J. Spencer, "Coloring random and semi-random k-colorable graphs," *Journal of Algorithms*, vol. 19, no. 2, pp. 204–234, 1995.

[36] U. Feige and J. Kilian, "Heuristics for semirandom graph problems," *Journal of Computer and System Sciences*, vol. 63, no. 4, pp. 639–671, 2001.

[37] T. McKenzie, H. Mehta, and L. Trevisan, "A new algorithm for the robust semi-random independent set problem," in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2020, pp. 738–746.

[38] J. Dattorro, *Convex optimization and Euclidean distance geometry*. USA: Meboo Publishing, 2005.

[39] M. L. Overton and R. S. Womersley, "On the sum of the largest eigenvalues of a symmetric matrix," *SIAM J. Matrix Anal. Appl.*, vol. 13, no. 1, p. 41–45, Jan. 1992.

[40] S. Arora and S. Kale, "A combinatorial, primal-dual approach to semidefinite programs," *J. ACM*, vol. 63, no. 2, pp. 12:1–12:35, 2016.

[41] E. Hazan, "Introduction to online convex optimization," *CoRR*, vol. abs/1909.05207, 2019.

[42] Z. Allen Zhu and Y. Li, "Even faster SVD decomposition yet without agonizing pain," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 974–982.

[43] C. Musco and C. Musco, "Randomized block krylov methods for stronger and faster approximate singular value decomposition," in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 1396–1404.