# NuMax: A Convex Approach for Learning Near-Isometric Linear Embeddings

*Chinmay Hegde, Aswin C. Sankaranarayanan, Wotao Yin, Richard G. Baraniuk*

**Abstract**

We propose a novel framework for the *deterministic* construction of *linear, near-isometric* embeddings of a finite set of data points. Given a set of training points $\mathcal{X} \subset \mathbb{R}^N$, we consider the *secant set* $\mathcal{S}(\mathcal{X})$ that consists of all pairwise difference vectors of $\mathcal{X}$, normalized to lie on the unit sphere. We formulate an affine rank minimization problem to construct a matrix $\Psi$ that preserves the norms of all the vectors in $\mathcal{S}(\mathcal{X})$ up to a distortion parameter $\delta$. While affine rank minimization is NP-hard, we show that this problem can be relaxed to a convex formulation that can be solved using a tractable semidefinite program (SDP). In order to enable scalability of our proposed SDP to very large-scale problems, we adopt a two-stage approach. First, in order to reduce compute time, we develop a novel algorithm based on the Alternating Direction Method of Multipliers (ADMM) that we call *Nuclear norm minimization with Max-norm constraints* (NuMax) to solve the SDP. Second, we develop a greedy, approximate version of NuMax based on the *column generation* method commonly used to solve large-scale linear programs. We demonstrate that our framework is useful for a number of signal processing applications via a range of experiments on large-scale synthetic and real datasets.

**Index Terms**

Dimensionality reduction, compressive sensing, approximate nearest neighbors, classification

## I. Introduction

In many applications, we seek a low-dimensional representation (or *embedding*) of data that are elements of a high-dimensional ambient space. The classical approach to constructing such an embedding



CH is with the Massachusetts Institute of Technology. ACS is with Carnegie Mellon University. WY is with the University of California at Los Angeles. RGB is with Rice University. A preliminary version of this paper [1] appeared in the Proceedings of the Statistical Signal Processing Workshop. This work was supported in part by the grants NSF CCF-0431150, CCF-0926127, CCF-1117939, DMS-0748839 and ECCS-1028790; DARPA/ONR N66001-11-C-4092 and N66001-11-1-4090; ONR N00014-08-1-1101, N00014-08-1-1112, N00014-10-1-0989, and N00014-11-1-0714; AFOSR FA9550-09-1-0432; ARO MURI W911NF-07-1-0185 and W911NF-09-1-0383; and the Texas Instruments Leadership University Program.

is principal components analysis (PCA) [2], which involves *linearly* mapping the $N$-dimensional data into the $K$-dimensional subspace spanned by the dominant eigenvectors of the data covariance matrix, typically with $K \ll N$. A key appeal of PCA is its *computational efficiency*; it can be very efficiently performed using a singular value decomposition (SVD) on the data. Another key appeal is its *generalizability*; PCA produces a smooth, globally defined mapping that can be easily applied to unseen, out-of-sample test data points. Nevertheless, PCA has an important drawback: the produced embedding can *arbitrarily distort pairwise distances* between sample data points. This phenomenon is exacerbated when the data arises from a nonlinear submanifold of the signal space [3]. Due to this, PCA can potentially map two distinct points in the ambient signal space to a single point in the low-dimensional embedding space, rendering them indistinguishable. This hampers the application of PCA-like techniques to important signal processing problems such as reconstruction and parameter estimation.

An alternative to PCA is the approach of *random projections*. Consider $\mathcal{X}$, a cloud of $Q$ points in a high-dimensional Euclidean space $\mathbb{R}^N$. The Johnson-Lindenstrauss Lemma [4] states that $\mathcal{X}$ can be linearly mapped to a subspace of dimension $M = \mathcal{O}(\log Q)$ with minimal distortion of the $\binom{Q}{2}$ pairwise distances between the $Q$ points (in other words, the mapping is *near-isometric*). Further, this linear mapping can be easily implemented in practice; one simply constructs a matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ whose elements are drawn randomly from a certain probability distribution. Then, with high probability, $\mathbf{\Phi}$ is near-isometric under a certain lower-bound on $M$ [3, 4]. This approach can be extended to signal classes beyond finite point clouds including points that lie on compact, differentiable low-dimensional manifolds [5, 6] as well as pairwise distances between all sparse signals [7]. This intuition is a fundamental component of compressive sensing (CS), an emergent framework for signal acquisition and reconstruction [8]. Despite their simplicity, random projections are oblivious of the data under consideration and hence cannot leverage any special geometric structure of the data if present.

In this paper, we propose a novel *deterministic* framework for constructing *linear, near-isometric* embeddings of a finite high-dimensional dataset. Given a set of training points $\mathcal{X} \subset \mathbb{R}^N$, we consider the *secant set* $\mathcal{S}(\mathcal{X})$ consisting of all pairwise difference vectors of $\mathcal{X}$ normalized to lie on the unit sphere. We formulate an affine rank minimization problem (3) to construct a matrix $\Psi$ that preserves the norms of all of the vectors in $\mathcal{S}(\mathcal{X})$ up to a desired distortion parameter $\delta$. We perform a *convex* relaxation to obtain a trace-norm minimization (4), which is equivalent to a tractable semidefinite program (SDP). The SDP (4) can be solved using ellipsoid or interior-point methods for convex programming (for example, [9] and [10]). However, the convergence of such generic methods is typically very slow. Further, the presence of the max-norm constraints in (4), though convex, negates the direct application of existing

first-order methods [11].

To remedy this situation, we develop a novel algorithm that we call *Nuclear norm minimization with Max-norm constraints* (NuMax). NuMax is based on the Alternating Direction Method of Multipliers (ADMM); it decouples the complex SDP formulation into a sequence of easy-to-solve subproblems. In order to achieve scalability to large-scale problems, we propose a modified, greedy version of NuMax that mirrors the *column generation* approach commonly used to solve large-scale linear programs [12]. With this modification, NuMax can efficiently solve problems where the number of elements in the secant set $\mathcal{S}(\mathcal{X})$, i.e., the number of constraints in (4), is extremely large (e.g., $10^7$ or greater).

Via a series of numerical experiments, we demonstrate that NuMax is useful for a number of signal processing applications. First, if the training set $\mathcal{X}$ comprises sufficiently many points that are uniformly drawn from a low-dimensional smooth manifold $\mathcal{M}$, then we show that the matrix $\mathbf{\Psi}$ satisfies the restricted isometry property (RIP) for signals belonging to $\mathcal{M}$ and hence enables the design of *efficient measurement matrices* for the compressive sensing of manifold-modeled datasets. Second, since the embedding $\mathbf{\Psi}$ (approximately) preserves all pairwise secants in the training set $\mathcal{X}$, it is also guaranteed to (approximately) preserve nearest-neighbors of all points of $\mathcal{X}$. Therefore, NuMax produces an efficient method to design *linear hash functions* for high-dimensional data retrieval. Third, by carefully pruning the secant set $\mathcal{S}(\mathcal{X})$, we can tailor $\mathbf{\Psi}$ for more general signal inference tasks, such as *supervised binary classification.*

## II. Background

### A. Notation

In this paper, we will exclusively work with real-valued vectors and matrices. We use lowercase boldface to denote vectors, uppercase boldface to denote matrices, and calligraphic letters to denote sets or set-valued operators. Given a symmetric matrix $\mathbf{X} \in \mathbb{R}^{N \times N}$, we write $\mathbf{X} \succeq 0$ if $\mathbf{X}$ is positive semidefinite (PSD). Denote the singular value decomposition (SVD) of a matrix $\mathbf{X} \in \mathbb{R}^{N \times N}$ as $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{\Sigma} = \mathrm{diag}(\boldsymbol{\sigma})$ is a diagonal, non-negative matrix where $\boldsymbol{\sigma}$ is the vector of (sorted) singular values. The *Frobenius norm* of $\mathbf{X}$, denoted by $\|\mathbf{X}\|_F$, is the square root of the sum of squared entries of $\mathbf{X}$, or equivalently, the $\ell_2$-norm of $\boldsymbol{\sigma}$. The rank of $\mathbf{X}$ is equal to the number of nonzero entries in $\boldsymbol{\sigma}$. The *nuclear norm* of $\mathbf{X}$, denoted by $\|\mathbf{X}\|_*$, is equal to the sum of its singular values, or equivalently, the $\ell_1$-norm of $\boldsymbol{\sigma}$.

### B. PCA and MDS

Consider a set of $Q$ data vectors $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_Q\} \subset \mathbb{R}^N$, where $N, Q$ are potentially very large. We group the elements of $\mathcal{X}$ as columns in the matrix $\mathbf{X} \in \mathbb{R}^{N \times Q}$, which we term the *data matrix*. Given a data matrix, a natural question is whether the $Q$ points can be embedded into a lower-dimensional space $\mathbb{R}^M,\ M < N$ with minimal distortion.

One such embedding can be obtained via a popular statistical technique known as *principal components analysis* (PCA). PCA obtains an embedding as follows; given $\mathbf{X}$, we perform an SVD of $\mathbf{X}$, i.e., compute $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, and then linearly project the columns of $\mathbf{X}$ onto the subspace spanned by the $r$ leftmost columns of $\mathbf{U}$ (termed the PCA basis vectors). The projected data points provide the optimal $r-$ dimensional approximation to $\mathbf{X}$ in terms of the Frobenius norm. Furthermore, PCA can be adapted to account for problem-specific requirements. For example, if the data vectors originate from one of two classes, then PCA can be modified to maintain class separability using related techniques such as Fisher's Linear Discriminant Analysis (LDA) or Factor Analysis [13, 14].

PCA can be viewed as a special case of the more general technique of *multi-dimensional scaling* (MDS). Given a high-dimensional dataset $\mathcal{X} \in \mathbb{R}^{Q \times N}$, MDS constructs a $Q \times Q$ matrix $D(\mathcal{X})$ of pairwise dissimilarities and tries to construct a lower-dimensional dataset $f(\mathcal{X}) \in \mathbb{R}^{M \times N},\ M < N$ such that $D(f(\mathcal{X})) \approx D(\mathcal{X})$. If the pairwise dissimilarities correspond to Euclidean distances, then MDS is equivalent to PCA [15] and $f(\mathcal{X})$ is simply a *linear* embedding of $\mathcal{X}$. If the pairwise dissimilarities are captured by some other distance metric, then the embedding is *nonlinear* in general.

PCA and MDS are conceptually simple. However, the convenience of PCA-like techniques are balanced by certain drawbacks. Crucially, their optimality is not accompanied by any guarantees regarding the local geometric properties of the resulting embedding [3]. In other words, PCA and MDS are not guaranteed to be *isometric* or even *invertible*.

### C. Nonlinear Embeddings

The focus of this paper is primarily on designing *linear* embeddings of data into low dimensions. Linear embedings can be *explicitly* stored in terms of a matrix operator; they are *generalizable*, i.e., they can be applied to any out-of-sample data point; and they naturally lend themselves to applications such as compressive sensing. However, several sophisticated *nonlinear* data embedding methods have also emerged over the last decade; see, for example, [16–20]. These methods are sometimes referred to as *manifold learning* algorithms. The list of manifold learning methods in the literature is far too long to enumerate in full, so we will simply discuss a few representative approaches.

Our approach bears some resemblance to the *Whitney Reduction Network* (WRN) approach for computing auto-associative graphs [21, 22]. The WRN is a heuristic that is algorithmically similar to PCA. An important notion in the WRN approach is the normalized *secant set* of $\mathcal{X}$:

$$\mathcal{S}(\mathcal{X}) = \left\{ \frac{\mathbf{x} - \mathbf{x}'}{\|\mathbf{x} - \mathbf{x}'\|_2}, \quad \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \mathbf{x} \neq \mathbf{x}' \right\}. \tag{1}$$

The approach initializes an estimate of the desired embedding and iteratively refines the embedding so as to ensure that the norms of the secants in $\mathcal{S}(\mathcal{X})$ deviate from unity as little as possible. Unfortunately, the WRN algorithm only makes locally optimal decisions and cannot ensure that the final mapping is (near) isometric. The more recent work by [23] proposes algorithms to construct near-isometric low-dimensional of manifold data based on Nash's embedding technique. Again, here the resulting mapping is nonlinear, and the discovery of the mapping is computationally challenging.

Our approach has connections to Locally Linear Embedding (LLE), proposed in [17]. LLE takes as input an arbitrary dataset $\mathcal{X}$ and outputs a set of (possibly overlapping) $M$-dimensional subspaces, each of which approximates a small subset of $\mathcal{X}$ according to a Euclidean error criterion. Therefore, the embedding is locally linear, but is *globally* nonlinear. It is unclear whether LLE ensures a (near)-isometry. The Sparse Manifold Learning and Clustering (SMCE) approach, proposed in [24] aims to address this issue by constructing an embedding by directly operating on the normalized secant set $\mathcal{S}(\mathcal{X})$; however, SMCE relies on a spectral decomposition that does not seem to lead to isometry guarantees.

We also note that using semidefinite programming (SDP) to construct low-dimensional embeddings of data have been explored before; see, for example, [20] and [25]. Such approaches construct a low-dimensional representation of an input data set $\mathcal{X}$ by performing a trace-norm optimization, subject to a set of distance constraints. It is likely that these approaches can be modified to produce near-isometric (nonlinear) embeddings of datasets. However, as mentioned above, the mappings obtained are not easily generalizable to out-of-sample data points. Further, it is unclear if the corresponding SDP formulations can be modified to scale to large datasets.

### *D. Random Projections*

The problem of constructing a low-dimensional isometric embedding of a dataset, i.e., embeddings that preserves all pairwise distances between the data points, has been studied in depth and is now classical (for an excellent introduction to this subject, see [26]). Concretely, we seek an embedding that obeys the following relaxed notion of isometry:

*Definition 1:* Suppose $M \leq N$ and consider $\mathcal{X} \subset \mathbb{R}^N$. An embedding operator $\mathcal{P} : \mathcal{X} \to \mathbb{R}^M$ satisfies

the *restricted isometry property (RIP)* with constant $\delta > 0$ on $\mathcal{X}$ if, for every $\mathbf{x}, \mathbf{x}'$ in $\mathcal{X}$, the following relations hold:

$$(1-\delta)\left\|\mathbf{x}-\mathbf{x}'\right\|_2^2 \leq \left\|\mathcal{P}\mathbf{x}-\mathcal{P}\mathbf{x}'\right\|_2^2 \leq (1+\delta)\left\|\mathbf{x}-\mathbf{x}'\right\|_2^2. \tag{2}$$

The quantity $\delta$ encapsulates the deviation from perfect isometry and is called the *isometry constant*. We (trivially) observe that the identity operator on $\mathcal{X}$ always satisfies the RIP with $\delta = 0$; however, in this case $M = N$. For the range $M < N$, the celebrated *Johnson-Lindenstrauss* (JL) Lemma confirms the existence of such operators [4]:

*Lemma 1:* [4] Consider a dataset $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_Q\} \subset \mathbb{R}^N$. Let $M = \Omega(\delta^{-2} \log Q)$. Construct a matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ by drawing each element of $\mathbf{\Phi}$ independently from a Gaussian distribution with zero mean and variance $1/M$. Then, with high probability, the linear operator $\mathbf{\Phi} : \mathbb{R}^N \to \mathbb{R}^M$ satisfies the RIP on $\mathcal{X}$.

The method of random projections can be extended to more general signal classes beyond finite point clouds. For example, random linear projections provably satisfy the RIP for data modeled as compact, differentiable low-dimensional submanifolds [5, 6]. A particularly interesting connection has been made with compressive sensing (CS), an emergent paradigm for efficient acquisition and processing of $K$-*sparse signals*, i.e., signals that can be expressed as the sum of only $K$ elements from a basis [7]. The central result of CS asserts that if a matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ satisfies the RIP on the set of all $K$-sparse signals, then it is possible to *stably* recover a sparse signal $\mathbf{x}$ from the linear embedding (or "measurements") $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$, even when $M$ is only proportional to $K \log(N/K)$.

Random projections provide a simple method to construct embeddings that satisfy the RIP for arbitrary datasets. It can be shown that, in the worst case for a given isometry constant $\delta$, there exist datasets that cannot be embedded into any $M$-dimensional space where $M \leq \delta^{-2} \log^{-1}(\delta^{-1}) \log Q$ [27]. However, this worst case only occurs for a specific configuration of points that seldom occurs in practice. Further, the universality property of random projections negates its ability to leverage the intrinsic geometry of a given data set.

### *E. Metric learning*

Related to the ideas proposed in this paper is a framework known as *metric learning*. Given a dataset and an intended task (for example, classification), the goal of metric learning is to learn a distance metric that improves over the Euclidean distance. There has been significant recent work in this context for learning *Mahalanobis* distances, i.e, metrics of the form $d^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x}-\mathbf{y})^T \Sigma (\mathbf{x}-\mathbf{y})$, where $\Sigma$ is

positive semidefinite. See [28–30] and references therein.

There are several differences between NuMax and metric learning. First, we consider NuMax to be a *dimensionality reduction* technique. In contrast, metric learning is a fairly general framework that attempts to learn a suitable depending on the particular application. Second, NuMax is geared towards producing linear embeddings that are nearly *isometric* with respect to a set of vectors, and this motivates specific computational challenges and solutions. In contrast, metric learning does not explicitly emphasize isometry, and as a consequence, the optimization problems that arise within that framework are very different from ours. Lastly, our focus is on designing *efficient, scalable* algorithms for embedding design. To the best of our knowledge, this issue is not addressed in detail in the metric learning literature.[1]

### *F. Other related work*

A conference version of this manuscript [1] introduced the basic convex formulation (Eq. 4), and explored its use in two applications. The current manuscript contains several new extensions, including: (i) two efficient algorithms to solve the convex optimization problem proposed in Eq. 4; (ii) a novel formulation for classification applications (Eq. 6) that improves upon the conference version. (iii) A large number of experiments on both synthetic and real-world datasets that validate our approach for applications like nearest-neighbor based retrieval, binary classification, and compressive sensing.

Since the appearance of a preliminary version of this manuscript, several works have pursued similar goals of learning norm-preserving linear embeddings using optimization methods. The authors of [32] discuss the specialized problem of learning *orthonormal* linear embeddings, and develop polynomial-time algorithms with provable approximation guarantees. The authors of [33] propose learning embeddings under a Frobenius norm constraint, and propose a different optimization approach with provable guarantees.

We point out that surprisingly little is known about (near) isometric linear embeddings in Euclidean space beyond the Johnson-Lindenstrauss Lemma [34]. While we fall short of a rigorous analytical characterization for our framework, our algorithmic techniques might lead to some interesting progress in this regard.

[1]While preparing the final version of this manuscript, we became aware of the recent paper [31] that performs computationally efficient metric learning using somewhat different algorithmic techniques. We defer an in-depth comparison to future work.

## III. Near-Isometric Linear Embeddings

### A. Optimization Framework

Given a dataset $\mathcal{X} \subset \mathbb{R}^N$, our goal is to find a linear embedding $\mathcal{P} : \mathbb{R}^N \to \mathbb{R}^M,\ M \ll N$, that satisfies the RIP (2) on $\mathcal{X}$ with parameter $\delta > 0$. Following [5], we will refer to $\delta$ as the *isometry constant*. We form the secant set, a set of $S = \binom{Q}{2}$ unit vectors $\mathcal{S}(\mathcal{X}) = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_S\}$ as defined in (1). Then, we seek a *embedding matrix* $\mathbf{\Psi} \in \mathbb{R}^{M \times N}$ with as few rows as possible that satisfies the RIP on $\mathcal{S}(\mathcal{X})$. [2]

We cast this problem in terms of an optimization over the space of symmetric PSD matrices. Define $\mathbf{P} \doteq \mathbf{\Psi}^T\mathbf{\Psi} \in \mathbb{R}^{N \times N}$; then, $\text{rank}(\mathbf{P}) = M$. We also have the constraints that $|\,\|\mathbf{\Psi}\mathbf{v}_i\|_2^2 - 1| = \left|\mathbf{v}_i^T\mathbf{P}\mathbf{v}_i - 1\right|$ is no greater than $\delta$ for every secant $\mathbf{v}_i$ in $\mathcal{S}(\mathcal{X})$. Let $\mathbf{1}_S$ denote the $S$-dimensional all-ones vector, and let $\mathcal{A}$ denote the linear operator that maps a symmetric matrix $\mathbf{X}$ to the $S$-dimensional vector $\mathcal{A} : \mathbf{X} \to (\mathbf{v}_i^T\mathbf{X}\mathbf{v}_i)_{i=1}^S$. Then, we seek the solution to the optimization problem

$$\min_{\mathbf{P}^T = \mathbf{P} \succeq 0} \text{rank}(\mathbf{P}) \quad \text{subject to} \quad \|\mathcal{A}(\mathbf{P}) - \mathbf{1}_S\|_\infty \leq \delta. \tag{3}$$

Rank minimization is both non-convex and NP-hard, in general. Therefore, following [35], we propose to instead solve a nuclear-norm relaxation of (3):

$$\min_{\mathbf{P}^T = \mathbf{P} \succeq 0} \|\mathbf{P}\|_* \quad \text{subject to} \quad \|\mathcal{A}(\mathbf{P}) - \mathbf{1}_S\|_\infty \leq \delta. \tag{4}$$

Since $\mathbf{P}$ is a PSD symmetric matrix, the nuclear norm of $\mathbf{P}$ is equal to its trace. Thus, the problem (4) consists of minimizing a linear objective function subject to linear inequality constraints over the cone of PSD symmetric matrices. Hence, it is equivalent to a semidefinite program (SDP) and can be solved in polynomial time [36]. Once the solution $\mathbf{P}^* = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ to (4) is found, $\text{rank}(\mathbf{P}^*)$ determines the value of $M$, the dimensionality of the linear embedding. The desired linear embedding $\mathbf{\Psi}$ can then be calculated using a simple matrix square root

$$\mathbf{\Psi} = \mathbf{\Lambda}_M^{1/2}\mathbf{U}_M^T, \tag{5}$$

where $\mathbf{\Lambda}_M = \text{diag}\{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_M\}$ denotes the $M$ leading (non-zero) eigenvalues of $\mathbf{P}^*$, and $\mathbf{U}_M$ denotes the set of corresponding eigenvectors. In this manner, we obtain a low-rank matrix $\mathbf{\Psi} \in \mathbb{R}^{M \times N}$ that satisfies the RIP on the secant set $\mathcal{S}(\mathcal{X})$ with isometry constant $\delta$. The convex optimization formulation

[2]Equivalently, we can state the constraints in terms of the elements of all pairwise difference vectors $\mathbf{x}_i - \mathbf{x}_j$ and their Euclidean norms; we prefer the secant notation due to its conciseness.

(4) is conceptually simple, the only inputs being the dataset $\mathcal{X}$ and the desired isometry constant $\delta > 0$.

### B. Analysis

Since we seek an embedding matrix $\mathbf{\Psi}$ with a minimal number of rows, a natural question to ask is whether the nuclear-norm relaxation (4) is guaranteed to produce solutions $\mathbf{P}^*$ of minimum rank. The efficiency of nuclear-norm minimization for low-rank matrix recovery has been studied in a number of settings [37, 38]. However, we highlight two unique aspects of the optimization problem (4). First, the $\ell_\infty$-norm constraints in (4) are non-standard. Second, the best known theoretical results make certain restrictive assumptions on the linear operator $\mathcal{A}$ in (4); for example, one common assumption is that the entries of the matrix representation of $\mathcal{A}$ are independently drawn from a standard normal distribution. This assumption is clearly violated in our case, since $\mathcal{A}$ is a function of the secant set $\mathcal{S}(\mathcal{X})$, which depends heavily on the geometry of the data at hand. Nevertheless, a classical result from SDP provides an upper bound on the rank of the optimum $\mathbf{P}^*$ in (4).

*Proposition 1:* [39, 40] Let $r^*$ be the rank of the optimum to the SDP (4). Then, $r^* \leq \left\lceil \frac{\sqrt{8|\mathcal{S}(\mathcal{X})|+1}-1}{2} \right\rceil$. In essence, the rank of $\mathbf{P}^*$ grows as the square root of the cardinality of the secant set $\mathcal{S}(\mathcal{X})$. Note that the upper bound on the optimal rank $r^*$ provided in Prop. 1 can be very loose, since the cardinality of $\mathcal{S}(\mathcal{X})$ is potentially large.

A full analytical characterization of the optimal rank obtained by the program (4) is of considerable interest both in theory and practice. However, this seems to an extremely challenging analytical problem for a generic point set $\mathcal{X}$. For some initial progress in this direction (albeit under somewhat more restrictive settings), see the recent results by [32] and [33]. The main question is to verify the efficiency of the convex relaxation (4), which is essentially an SDP with rank-1 constraints (specified by the secant set $\mathcal{S}(\mathcal{X})$). The *PhaseLift* approach proposed by [41] has addressed this question in a somewhat different context. However, the underlying assumption in their work is that the rank-1 constraint vectors are independently and randomly for an arbitrary dataset $\mathcal{X}$, and therefore that theory does not apply in our case.

To the best of our knowledge, no strengthening of Prop. 1 is known even for special instances of the dataset $\mathcal{X}$. Indeed, surprisingly little is known about (near) isometric linear embeddings in Euclidean space beyond the Johnson-Lindenstrauss Lemma [34]. While we fall short of a rigorous analytical characterization for our framework, our algorithmic techniques might lead to some progress in this respect.

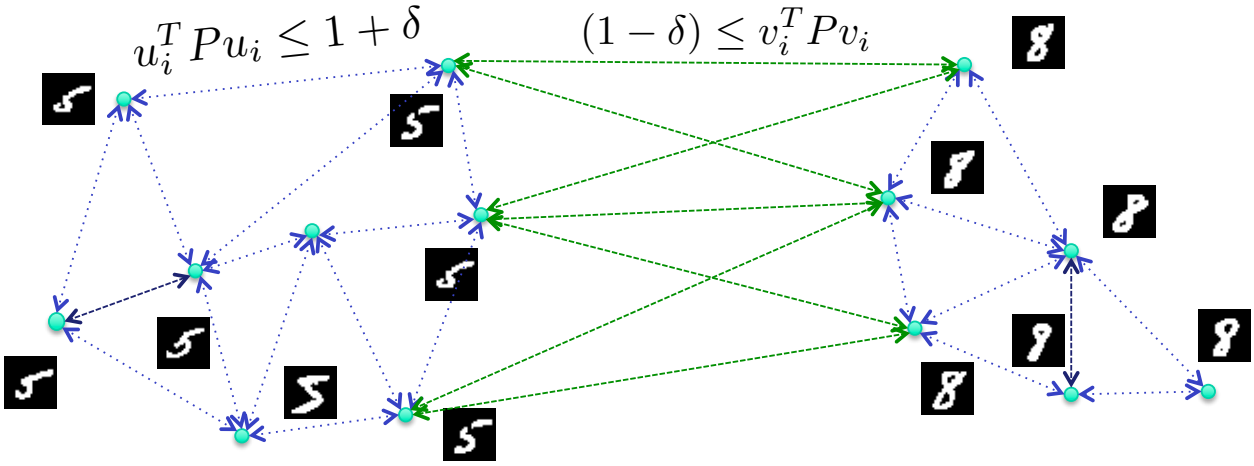



Fig. 1. *A desirable objective for classification is to promote nearest neighbors of a point to come from its own class. We achieve this by altering the near-isometric constraints. First, we relax the upper bound on the near-isometry for the inter-class secants; hence, they can expand in length unconstrained. Second, we relax the lower-bound on the intra-class secants; hence, they can shrink in length unconstrained. For the same distortion parameters, we observe lower-rank solutions with higher-classification rates.*

### *C. Class-specific Linear Dimensionality Reduction*

We observe that the inequality constraints in (4) are derived by enforcing an approximate isometry condition on *all* pairwise secants $\{\mathbf{v}_i\}_{i=1}^S$. While the need to enforce the (approximate) isometry of all pairwise secants might be important in applications such as signal *reconstruction*, such a criterion could prove to be too restrictive for other tasks.

For example, consider a *supervised classification* scenario, where the points in $\mathcal{X}$ arise from two classes of interest. Suppose that we wish to use the classical nearest neighbor (NN) classifier to classify data points based on the labeled training data. In this scenario, preserving the lengths of the secants is no longer the goal; instead we really need an embedding matrix $\mathbf{\Psi}$ that tries to *separate* the two classes. It would not really affect classification performance if two data points from the same class somehow were mapped to the same lower-dimensional point, as long as pairs of points from different classes were mapped to points sufficiently far apart.

There are many ways for translating this idea into a precise criterion for optimization. Here is one intuitive approach. Suppose that we have labeled training data from multiple classes. We can identify two flavors of secants — *inter-class* secants $\mathbf{v}_i$ which connect points from different classes, and *intra-class* secants $\mathbf{u}_i$ which connect points from the same class. A simple extension to (4) applies different constraints to the inter and intra-class secants (see Fig. 1). Specifically, we let the length of inter-class secants to expand by an arbitrary factor while not allowing their length to shrink; this enables points

from different classes to move apart from one another. Similarly, we let the length of intra-class secants to shrink by an arbitrary factor while not allowing their lengths to expand; this is formulated as

$$\begin{aligned} \min_{\mathbf{P}^T=\mathbf{P}\succeq 0} \quad & \|\mathbf{P}\|_* \\ \text{subject to} \quad & \mathbf{v}_i^T\mathbf{P}\mathbf{v}_i \geq 1-\delta, \quad \forall \ \ \mathbf{v}_i \in \text{inter-class} \\ & \mathbf{u}_i^T\mathbf{P}\mathbf{u}_i \leq 1+\delta \quad \forall \ \ \mathbf{u}_i \in \text{intra-class} \end{aligned} \tag{6}$$

This convex program in (6) has the same objective as the one in (4); however, the feasible set is vastly expanded since the near-isometric constraints are significantly weakened. Hence, we can hope not just to obtain a low-rank solution (since our feasibility set has been expanded) but also to promote improved classification (since we can expect points from different classes to be embedded differently). We examine this type of "class-specific" linear embeddings further in our numerical experiments.

## IV. Algorithms for Designing Embeddings

The SDP (4) admits a tractable solution in polynomial time using interior-point methods. However, for a generic SDP with $S$ constraints and a matrix variable of size $N \times N$, interior-point methods incur memory costs that scale as $\mathcal{O}\left(S^2\right)$ and time-complexity costs that scale as $\mathcal{O}\left(N^6\right)$. Therefore, solving (4) using traditional SDP solvers [9, 10] quickly becomes infeasible. Here, we develop two algorithms that exploit the special structure of the optimization problem (4) to produce very efficient solutions at vastly reduced costs.

### A. ADMM

We develop an efficient algorithm to solve (4) based on the Alternating Direction Method of Multipliers (ADMM). We dub our algorithm *NuMax*, an abbreviation for *Nuclear norm minimization with Max-norm constraints*. We rewrite (4) by introducing the auxiliary variables $\mathbf{L} \in \mathbb{S}^{N\times N}$ and $\mathbf{q} \in \mathbb{R}^S$ to obtain the optimization problem

$$\begin{aligned} \min_{\mathbf{P}\succeq 0,\mathbf{L},\mathbf{q}} \quad & \|\mathbf{P}\|_* \\ \text{subject to} \quad & \mathbf{P}=\mathbf{L}, \quad \mathcal{A}(\mathbf{L})=\mathbf{q}, \quad \|\mathbf{q}-\mathbf{1}_S\|_\infty \leq \delta. \end{aligned} \tag{7}$$

This approach can be viewed as an instance of the Douglas-Rachford variable splitting method in convex programming [42]. Next, we relax the linear constraints and form an *augmented Lagrangian* of (7) as

follows:

$$\begin{aligned} \min_{\mathbf{P}\succeq 0,\mathbf{L},\mathbf{q},\boldsymbol{\Lambda},\boldsymbol{\omega}} \|\mathbf{P}\|_* + \frac{\beta_1}{2}\|\mathbf{P}-\mathbf{L}-\boldsymbol{\Lambda}\|_F^2 \\ +\frac{\beta_2}{2}\|\mathcal{A}(\mathbf{L})-\mathbf{q}-\boldsymbol{\omega}\|_2^2 \\ \text{subject to} \quad \|\mathbf{q}-\mathbf{1}_S\|_\infty \leq \delta. \end{aligned} \tag{8}$$

Here, the symmetric matrix $\boldsymbol{\Lambda} \in \mathbb{S}^{N\times N}$ and vector $\boldsymbol{\omega} \in \mathbb{R}^S$ represent the scaled Lagrange multipliers. The optimization in (8) is carried out over the variables $\mathbf{P}, \mathbf{L} \in \mathbb{S}^{N\times N}$ and $\mathbf{q} \in \mathbb{R}^S$, while $\boldsymbol{\Lambda}$ and $\boldsymbol{\omega}$ are iteratively updated as well. Instead of jointly optimizing over all three variables, we optimize the variables one at a time while keeping the others fixed. That is, we can solve the optimization (8) via a sequence of three sub-problems, each of which admits a computationally efficient solution. Let the subscript $k$ denote the estimate of a variable at the $k^{\text{th}}$ iteration of the algorithm. The following steps are performed until convergence.

**Update $\mathbf{q}$**: Isolating the terms that involve $\mathbf{q}$, we obtain a new estimate $\mathbf{q}_{k+1}$ in closed form. Denote $\mathbf{z} = \mathcal{A}(\mathbf{L}^k) - \boldsymbol{\omega}^k - \mathbf{1}_S$. Then, it is seen that

$$\mathbf{q}_{k+1} = \mathbf{1}_S + \text{sign}(\mathbf{z}) \cdot \min(|\mathbf{z}|, \delta), \tag{9}$$

where the sign and min operators are applied component-wise. This step can be performed in $\mathcal{O}(S)$ operations.

**Update $\mathbf{P}$**: Isolating the terms that involve $\mathbf{P}$, we obtain a new estimate $\mathbf{P}_{k+1}$ via the *eigenvalue shrinkage operator* (similar to the approach described in [43]). Denote $\mathbf{P'} = \mathbf{L}_k + \boldsymbol{\Lambda}_k$ and perform the eigen decomposition $\mathbf{P'} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T$, where $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\sigma})$. Then, the optimum $\mathbf{P}_{k+1}$ can be expressed as

$$\mathbf{P}_{k+1} = \mathbf{V}\mathcal{D}_\alpha(\boldsymbol{\Sigma})\mathbf{V}^T, \quad \mathcal{D}_\alpha(\boldsymbol{\Sigma}) = \text{diag}(\{(\sigma_i - \alpha)_+\}), \tag{10}$$

where $\alpha = \frac{1}{\beta_1}$ and $t_+$ represents the positive part of $t$, i.e., $t_+ = \max(t, 0)$. The dominant computational cost for this update is incurred by performing the eigendecomposition of $\mathbf{P'} \in \mathbb{S}^{N\times N}$; in general this step can be carried out in $\mathcal{O}\left(N^3\right)$ operations. This step can potentially be made even faster by using randomized linear algebra techniques [44].

**Update $\mathbf{L}$**: Isolating the terms that involve $\mathbf{L}$, we obtain a new estimate $\mathbf{L}_{k+1}$ as the solution of the unconstrained least-squares problem whose minimum is achived by solving the following linear system

**Algorithm 1** NuMax

**Inputs**: Secant set $\mathcal{S}(\mathcal{X}) = \{\mathbf{v}_i\}_{i=1}^S$, parameter $\delta$
**Parameters**: Weights $\beta_1, \beta_2$, step size $\eta$
**Output**: Symmetric PSD matrix $\widehat{\mathbf{P}}$
**Initialize**: $\mathbf{P}_0, \mathbf{L}_0, \boldsymbol{\omega}_0, \mathbf{q}_0, k \leftarrow 0$, $\mathbf{b} \leftarrow \mathbf{1}_S$,
set $\mathcal{A} : \mathbf{X} \mapsto (\mathbf{v}_i^T \mathbf{X} \mathbf{v}_i)_{i=1}^S$
**while** not converged **do**
$\mathbf{z} \leftarrow \mathcal{A}(\mathbf{L}_k) - \boldsymbol{\omega}_k - \mathbf{b}$
$\mathbf{q}_{k+1} \leftarrow \mathbf{b} + \text{sign}(\mathbf{z}) \cdot \min(|\mathbf{z}|, \delta)$
$\mathbf{P}' \leftarrow \mathbf{L}_k + \boldsymbol{\Lambda}_k, \quad \mathbf{P}' = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T$
$\mathbf{P}_{k+1} \leftarrow \mathbf{V}\mathcal{D}_\alpha(\boldsymbol{\Sigma})\mathbf{V}^T$
$\mathbf{Z} \leftarrow \beta_2 \mathcal{A}^*(\mathbf{q}_{k+1} + \boldsymbol{\omega}_k), \ \mathbf{Z}' \leftarrow \beta_1(\mathbf{P}_k - \boldsymbol{\Lambda}_k)$
$\mathbf{L}_{k+1} \leftarrow \beta_2(\mathcal{A}^*\mathcal{A} + I)^\dagger(\mathbf{Z} + \mathbf{Z}')$
$\boldsymbol{\Lambda}_{k+1} \leftarrow \boldsymbol{\Lambda}_k - \eta(\mathbf{P}_k - \mathbf{L}_k)$
$\boldsymbol{\omega}_{k+1} \leftarrow \boldsymbol{\omega}_k - \eta(\mathcal{A}(\mathbf{L}_k) - \mathbf{q}_k)$
$k \leftarrow k + 1$
**end while**
return $\widehat{\mathbf{P}} \leftarrow \mathbf{P}_k$

of equations.

$$\beta_1(\mathbf{P}_k - \mathbf{L} - \boldsymbol{\Lambda}_j) = \beta_2 \mathcal{A}^*(\mathcal{A}(\mathbf{L}) - \mathbf{q}_{k+1} - \boldsymbol{\omega}_k), \tag{11}$$

where $\mathcal{A}^*$ represents the adjoint of $\mathcal{A}$. The dominant cost in this step arises due to the linear operator $\mathcal{A}^*\mathcal{A}$. A single application of this operator incurs a complexity of $\mathcal{O}\left(N^2S^2\right)$. The least-squares solution to (11) can be calculated using a number of existing methods for solving large-scale linear equations, such as conjugate gradients [45, 46].

**Update $\boldsymbol{\Lambda}, \boldsymbol{\omega}$**: Finally, as is standard in augmented Lagrange methods, we update the parameters $\boldsymbol{\Lambda}, \boldsymbol{\omega}$ according to the equations

$$\boldsymbol{\Lambda}_{k+1} \leftarrow \boldsymbol{\Lambda}_k - \eta(\mathbf{P}_k - \mathbf{L}_k), \quad \boldsymbol{\omega}_{k+1} \leftarrow \boldsymbol{\omega}_k - \eta(\mathcal{A}(\mathbf{L}_k) - \mathbf{q}_k).$$

The overall NuMax method is summarized in pseudocode form in Algorithm 1. The convergence properties of NuMax, both in terms of precision as well as speed, are affected by the user-defined parameters $\eta, \ \beta_1$, and $\beta_2$. In all of the experiments below in Section V, we set $\eta = 1.618$ and $\beta_1 = \beta_2 = 1$.

### *B. Column Generation*

NuMax (Algorithm 1) dramatically decreases the time-complexity of solving the SDP (4). However, for a problem with $S$ input secants, the memory complexity of NuMax still remains $\mathcal{O}\left(S^2\right)$, and this

**Algorithm 2** NuMax-CG

**Inputs**: Secant set $\mathcal{S} = \{\mathbf{v}_i\}_{i=1}^{S}$, parameter $\delta$
**Parameters**: Size of selected secant sets $S', S''$
**Output**: Symmetric PSD matrix $\widehat{\mathbf{P}}$
**Initialize**: Select a subset of $S'$ secants and call it $\mathcal{S}_0$,
set $\mathcal{A} : \mathbf{X} \mapsto (\mathbf{v}_i^T \mathbf{X} \mathbf{v}_i)_{i=1}^{S'}$
Obtain initial estimate $\mathbf{P} \leftarrow \text{NuMax}(\mathcal{S}_0, \delta)$
**while** not converged **do**
    $\widehat{\mathcal{S}} \leftarrow \{\mathbf{v}_i \in \mathcal{S}_0 : |\mathbf{v}_i^T \mathbf{P} \mathbf{v}_i - 1| = \delta\}$
    $\mathcal{S}_1 \leftarrow \{\mathbf{v}_i \in \mathcal{S} : \mathbf{v}_i \notin \mathcal{S}_0\}_{i=1}^{S''}$
    $\widehat{\mathcal{S}} \leftarrow \widehat{\mathcal{S}} \bigcup \{\mathbf{v}_i \in \mathcal{S}_1 : |\mathbf{v}_i^T \mathbf{P} \mathbf{v}_i - 1| \geq \delta\}$
    $\mathbf{P} \leftarrow \text{NuMax}(\widehat{\mathcal{S}}, \delta)$
estimate}
    $\mathcal{S}_0 \leftarrow \widehat{\mathcal{S}}$
**end while**
return $\widehat{\mathbf{P}} \leftarrow \mathbf{P}$

could be prohibitive in applications involving millions (or billions) of secants. We now develop a heuristic optimization method that only approximately solves (4) but that scales very well to such problem sizes.

Our key idea is based on the Karush-Kuhn-Tucker (KKT) conditions describing the optimum of (4). Recall that (4) consists of optimizing a linear objective subject to inequality constraints over the cone of PSD matrices. Suppose that strong duality holds, i.e., the primal and dual optimal values of (4) are equal. Then, by *complementary slackness* [47], the optimal solution is entirely specified by the set of those constraints that hold with equality. Such constraints are also known as *active* constraints. We propose a simple, greedy method to rapidly find the active constraints of (4).

1) Solve (8) with only a small subset $\mathcal{S}_0$ of the input secants $\mathcal{S}(\mathcal{X})$ using NuMax (Algorithm 1) to obtain an initial estimate $\widehat{\mathbf{P}}$. Identify the set $\widehat{\mathcal{S}}$ of secants that correspond to active constraints, i.e.,

$$\widehat{\mathcal{S}} \leftarrow \{\mathbf{v}_i \in \mathcal{S}_0 : |\mathbf{v}_i^T \widehat{\mathbf{P}} \mathbf{v}_i - 1| = \delta\}.$$

2) Select additional secants $\mathcal{S}_1 \subset \mathcal{S}$ that were not selected previously and identify all the secants among $S_1$ that *violate* the infinity norm constraints at the current estimate $\widehat{\mathbf{P}}$. Append these secants to the set of active constraints $\widehat{\mathcal{S}}$ to obtain an augmented set $\widehat{\mathcal{S}}$

$$\widehat{\mathcal{S}} \leftarrow \widehat{\mathcal{S}} \bigcup \{\mathbf{v}_i \in \mathcal{S}_1 : |\mathbf{v}_i^T \mathbf{P} \mathbf{v}_i - 1| \geq \delta\}.$$

3) Solve (4) with the augmented set $\widehat{\mathcal{S}}$ using NuMax (Alg. 1) to obtain an new estimate $\widehat{\mathbf{P}}$.

4) Identify the secants that correspond to active constraints. Repeat Steps 2 and 3 until convergence is reached in the estimated optimal matrix $\widehat{\mathbf{P}}$.

Instead of performing a large numerical optimization procedure on the entire set of secants $\mathcal{S}(\mathcal{X})$, we perform a sequence of optimization procedures on small subsets of $\mathcal{S}(\mathcal{X})$. When the number of active constraints is a small fraction of the overall secants, the computational gains are significant. This approach is analogous to the *column generation* (CG) method used to solve very large-scale linear programs [12]. Therefore, we dub our overall algorithm *NuMax-CG*; this algorithm is listed in pseudocode form in Algorithm 2.

A key benefit of NuMax-CG is that the set of secants upon which NuMax acts upon within each iteration never needs to be explicitly stored in memory and can in fact be generated *on the fly*. This can potentially lead to significant improvements in terms of memory complexity of the overall procedure. An important caveat is that we are no longer guaranteed to converge to the optimal solution of (4); nevertheless, as we see below in Section V, NuMax-CG yields excellent results on massively-sized, real-world datasets.

In practice, evaluating the KKT conditions for NuMax (and NuMax-CG) is computationally expensive. As a consequence, we use a notion of infeasibility as our main halting criterion. Specifically, we measure the errors in the strict enforcement of the equality constraints $\mathbf{P} = \mathbf{L}$ and $\mathbf{q} = \mathcal{A}(\mathbf{L})$

$$e_1 = \frac{2\|\mathbf{P} - \mathbf{L}\|_F}{\|\mathbf{P}\|_F + \|\mathbf{L}\|_F}, \qquad e_2 = \frac{2\|\mathbf{q} - \mathcal{A}(\mathbf{L})\|_2}{\|\mathbf{q}\|_2 + \|\mathcal{A}(\mathbf{L})\|_2} \ .$$

When $\max(e_1, e_2)$ is smaller than a user-specified parameter $\eta$, we proclaim convergence. For the numerical experiments below in Section V, we use $\eta = 5 \times 10^{-5}$.

### C. Convergence of NuMax and NuMax-CG

The convergence of NuMax can be understood in terms of the convergence properties of a more general ADMM. Although there are three variables $\mathbf{L}, \mathbf{P}, \mathbf{q}$ in (7), NuMax is indeed a standard ADMM (that is, with two blocks of variables) rather than a three-block ADMM, whose convergence is not guaranteed without extra assumptions or additional computation. In (7), one block of variables is $(\mathbf{P}, \mathbf{q})$, and the other is $\mathbf{L}$. In the standard ADMM, when one of the two blocks is fixed, the subproblem is minimized over the entire other block. In NuMax, when $\mathbf{L}$ is fixed, the subproblem is minimized over $\mathbf{P}, \mathbf{q}$ jointly. But since $\mathbf{P}, \mathbf{q}$ do not together appear any single objective term or constraint, the subproblem can be decoupled into minimizing over $\mathbf{P}$ and $\mathbf{q}$ separately.

For certain types of convex problems, ADMM converges at a rate of $O(1/k)$ [48] (more recently, the rate has been slightly improved to $o(1/k)$ [49]). Although we have observed NuMax to have a rate of convergence that appears to be linear, we have not been able to establish its linear convergence for arbitrary data. In particular, recent results in [50–52] prove the linear convergence of ADMM under extra assumptions. Unfortunately, these results do not appear to apply to (7) and establishing linear convergence remains open.

NuMax-CG calls NuMax to solve a sequence of instances of (7) with increasingly many constraints. Since there are finitely many secants and thus finitely many constraints in total, NuMax-CG is guaranteed to terminate after a finite number of iterations. However, the actual number of iterations will vary significantly depending on data, parameter choices, and the specific order in which the column generation procedure adds constraints to (7).

### *D. Class-specific NuMax*

We now discuss how to solve the classification optimization problem (6). Given the inter-class secants $\{\mathbf{v}_i, i = 1, \ldots, S_v\}$, the intra-class secants $\{\mathbf{u}_i, i = 1, \ldots, S_u\}$, and the distortion $\delta$, we can define a linear operator $\mathcal{A}_c : \mathbb{R}^{N\times N} \mapsto \mathbb{R}^{S_v+S_u}$, and the vector $\mathbf{b}_c \in \mathbb{R}^{S_v+S_u}$ as follows:

$$\mathcal{A}_c(\mathbf{P}) = \begin{pmatrix} \vdots \\ -\mathbf{v}_i^T\mathbf{P}\mathbf{v}_i \\ \vdots \\ \mathbf{u}_i^T\mathbf{P}\mathbf{u}_i \\ \vdots \end{pmatrix}, \quad \mathbf{b}_c = \begin{pmatrix} \vdots \\ -(1-\delta) \\ \vdots \\ 1+\delta \\ \vdots \end{pmatrix} \tag{12}$$

The convex program (6) can now be succinctly represented as

$$\underset{\mathbf{P}\succeq 0}{\text{minimize}} \ \ \|\mathbf{P}\|_* \quad \text{subject to} \quad \mathcal{A}_c(\mathbf{P}) \leq \mathbf{b}_c. \tag{13}$$

Here, the $\mathcal{A}_c$ operator captures the specifics of the modified/relaxed isometry constraints on the intra- and inter-class secants. Note that (13) is a more general form of the convex program in (4) and hence, solvers for (4) can be easily modified to solve (13). For example, Algorithm 1 can be modified to solve (13), simply, by modifying the truncation step to

$$\mathbf{q}_{k+1} \leftarrow \min(\mathbf{b}, \mathbf{z}).$$

We refer to this class-sensitive version of NuMax as *NuMax-Class*. Similarly, a CG version of NuMax-Class can be easily derived with minor modifications.

## V. Numerical Experiments

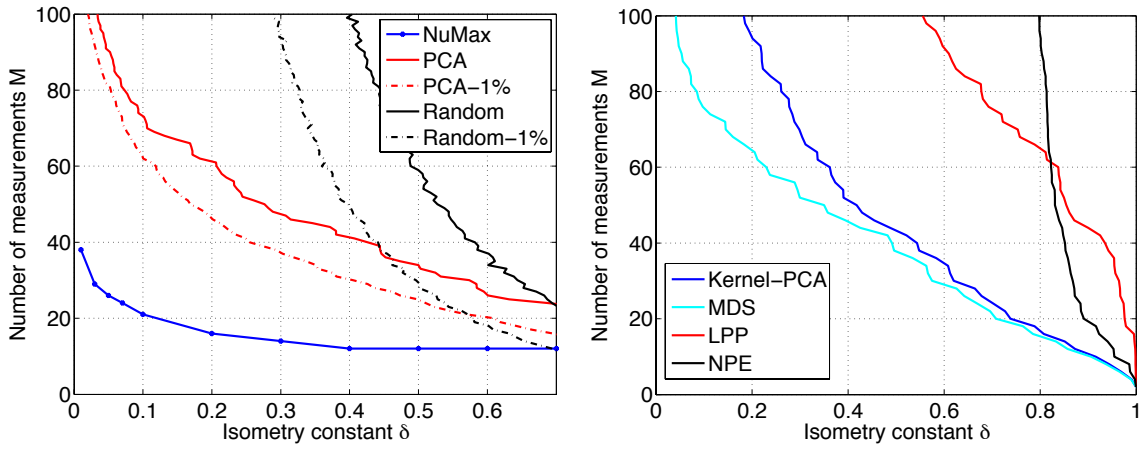
We illustrate the performance of the NuMax framework and algorithms via a number of numerical experiments and show that our approach enables improved performance in machine learning applications such as approximate nearest neighbor (ANN)-based data retrieval and supervised binary classification. We use $\eta = 1.6$ and $\beta_1 = \beta_2 = 1$ for all our numerical simulations. We use Algorithm 1 (NuMax) when $S$, the number of secants, is smaller than $5000$, and Algorithm 2 (NuMax-CG) for larger sized problems. For the rest of this section, we interchangeably use the terms "embeddings" and "measurements" whenever the context is clear.

### A. Linear Low-Dimensional Embeddings

We first demonstrate that NuMax can be used to design linear, low-dimensional embeddings of possibly complicated image datasets. We first consider a synthetic dataset $\mathcal{X}$ comprised of $N = 16 \times 16 = 256$-dimensional images of translations of a white square on a black background. We construct a training set $\mathcal{S}(\mathcal{X})$ of $S = 1000$ secants by randomly sampling pairs of images from $\mathcal{X}$, and normalizing the secants using (1). We are interested in quantitatively studying the performance of different types of linear as well as low-dimensional embeddings.

We begin with an empirical estimation of isometry constants using PCA, random Gaussian projections and NuMax. For each technique, we are interested in characterizing the variation of the isometry constant $\delta$ with the number of measurements $M$. For PCA, for a given dimensionality $M$, we project the secant set $S(\mathcal{X})$ onto the $M$ PCA basis functions learned from the $S(\mathcal{X})$ itself. We observe the worst-case deviation from unity in the norm of the projected secants; this gives the estimate of the isometry constant $\delta$. We also perform a similar isometry constant calculation using $M$ random Gaussian projections. Each entry of the $M \times N$ linear embedding matrix is sampled independently from a Gaussian distribution with zero mean and variance $1/M$. Third, for a desired value of isometry constant $\delta$, we solve (4) using NuMax (Algorithm 1) to obtain a positive semidefinite symmetric matrix $\mathbf{P}^*$. We denote the rank of $\mathbf{P}^*$ by $M$.

Figure 2(a) plots the variation of the number of measurements $M$ as a function of the isometry constant $\delta$. We observe that the NuMax embedding $\mathbf{\Psi}$ achieves the desired isometry constant on the secants using by far the fewest number of measurements. For example, NuMax attains a distortion of $\delta = 0.1$ with



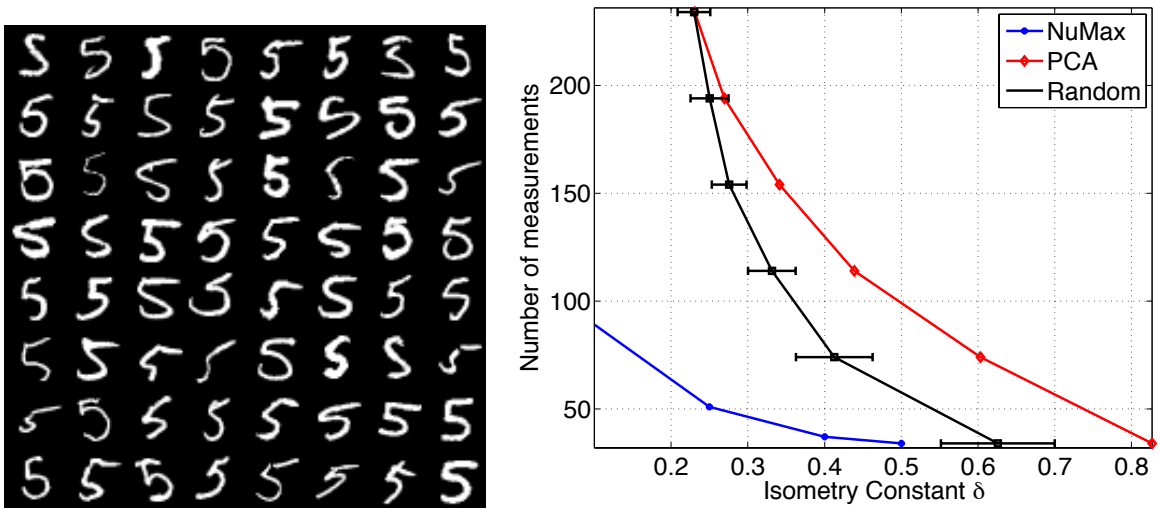

(a) (b)

Fig. 2. *(a) Empirical isometry constant $\delta$ vs. number of measurements $M$ using NuMax, PCA, and random embeddings. (b) Empirical isometry constant vs. number of measurements using various other embeddings. NuMax ensures global approximate isometry using by far the fewest measurements.*

*4× fewer* measurements than the next best algorithm (PCA). In Fig. 2(b), we include the numerical performance by several other techniques, such as Kernel-PCA (with an radial basis function kernel), metric MDS, locality preserving projections (LPP), and neighborhood preserving embedding (NPE). As in the comparison with linear techniques, NuMax outperforms the nonlinear techniques by achieving the desired isometric embedding using the fewest number of measurements.

For practical applications, it is often instructive to consider how the secant distortions look like when all but a fraction of the $S$ secant constraints are satisfied. In Fig. 2(a), we include curves for PCA and Random Projections which indicate the number of measurements at which all but $1\%$ of the secants achieve a distortion $\delta$. It is clear that NuMax outperforms the other algorithms even in this less restrictive setting.

Next, we consider a more challenging real-world dataset. The MNIST dataset [53] contains a large number of digital images of handwritten digits and is commonly used as a benchmark for machine learning algorithms. The images exhibit rich variations and presumably lie on a highly nonlinear submanifold of the image space. We construct a training dataset comprising of all images corresponding to the letter '5' (see Figure 3(a)); this forms a dataset with 4406 datapoints and hence, a secant set of cardinality 4.85 million. We estimate the variation of the isometry constant $\delta$ with the number of measurements $M$. For both PCA and random Gaussian embeddings, we obtain a $(\delta, M)$ pair by first selecting a value for $M$ and computing the value of $\delta$ associated with it; as with the Fig. 2(a), we discard $1\%$ of the secants




(a) (b)

Fig. 3. *(a) Example "5" images from the MNIST dataset. Each image is a point in $N = 28 \times 28 = 784$–dimensional space. (b) Empirical isometry constant $\delta$ vs. number of measurements $M$ using NuMax, PCA, and random embeddings. NuMax ensures global approximate isometry using the fewest number of measurements; for example, for a distortion parameter $\delta = 0.25$, it produces an embedding with nearly $5\times$ fewer measurements than PCA.*

that violate perfect isometry by the maximum amount. Further, to account for the inherent randomness in random Gaussian embeddings, we perform 100 independent trials by generating a Gaussian random matrix of size $M \times N$, and compute the near-isometry constant associated with each generated matrix. The results of this experiment are plotted in Figure 3(b). (The horizontal error bars in the figure represent the standard deviation of the observed isometry constants over the different trials.) Again, we observe that NuMax provides the best linear embedding uniformly over the entire range of the parameter $\delta$ (here, we refer to "best" in terms of reduced dimensionality.) For instance, for a distortion parameter $\delta = 0.25$, NuMax produces a linear embedding with $5\times$ fewer measurements than PCA.

Next, we compare the running times of NuMax and NuMax-CG by testing them on subsets of the MNIST dataset. We use the training dataset associated with the letter "5". We generate problems of different sizes by varying the number of secants. For each ensuing collection of secants, we solve both NuMax and NuMax-CG and observe the individual running times as well as the fraction of constraints that are active at the solution of NuMax-CG. For each problem size, we perform 10 trials and compile average statistics.

Figure 5(a) demonstrates that the fraction of *active* secants can be significantly smaller than the total number of secants, suggesting that NuMax-CG can be considerably faster than NuMax. Figure 5(b) confirms this fact: for a problem size with $S = 5 \times 10^4$, NuMax-CG outperforms NuMax in terms

| $\lvert\mathcal{S}(\mathcal{X})\rvert$ | 1000 | 2000 | 4000 | NuMax-CG |
|---|---|---|---|---|
| $\min\left(\frac{\lVert\Phi\mathbf{v}\rVert_2}{\lVert\mathbf{v}\rVert_2}\right)$ | 0.0204 | 0.0356 | 0.0822 | 0.60 |
| $\max\left(\frac{\lVert\Phi\mathbf{v}\rVert_2}{\lVert\mathbf{v}\rVert_2}\right)$ | 1.6009 | 1.5390 | 1.6376 | 1.37 |
| Runtime | 66.5s | 119.9s | 199.6s | 182.8s |
| Violations | 17.17% | 12.34% | 7.48% | 0.00 |

Fig. 4. *Performance of NuMax on sub-samplings of MNIST secant dataset.*

of running time by a full order of magnitude. Moreoever, despite the heuristic nature of NuMax-CG, we observed in practice that the solutions obtained NuMax and NuMax-CG are identical. Table II provides runtime values on the entire MNIST dataset for different values of $\delta$. MNIST dataset has 60,000 datapoints; thereby, producing a total of 1.8 billion secants (constraints). On this dataset, for values of $\delta \in [0.1, 0.4]$, NuMax-CG and NuMax-Class-CG converge within a few hours.

We compare the greedy secant selection strategy of NuMax-CG against a simpler subsampling of the data points and subsequently, applying the traditional NuMax (Algorithm 1). Using the following experiment, we empirically demonstrate why subsampling is not a good strategy. We considered the set of all the images corresponding to the letter "5" in the MNIST database. We set the distortion parameter $\delta = 0.4$, i.e., the lengths of secants were to be preserved between 0.6 and 1.4 times their original lengths. We applied NuMax-CG (Algorithm 2) on the entire dataset until convergence.

Next, we subsampled the secant set, and ran NuMax to find an embedding that guaranteed preservation of all distances for $\delta = 0.4$ *over the subsampled* secants. We repeated this for subsampled sets of sizes $1000, 2000,$ and $4000$. From Fig. V-A, we observed that the worst-case isometry constant $\delta$ in each of these cases is poor (ranging from 0.98 to .92), and that a significant fraction of the original secants violate the desired constant $\delta = 0.4$. The intuition is as follows. The total number of secants for this (moderately sized) dataset is nearly 4.85 million, and subsampling even up to 80,000 secants in this situation would constitute less than 2% of the total secants. Therefore, there is an overwhelming likelihood that most of the "important" secants that constitute the optimal solution would be absent in any reasonable subsampling. Further, we also point out that the runtime of NuMax-CG is not significantly greater than required for subsampling.

### *B. Approximate Nearest Neighbors (ANN)*

The notion of *nearest neighbors* is vital to numerous problems in estimation, classification, and regression [54]; Suppose that a large dataset of training examples is modeled as a subset of a Euclidean

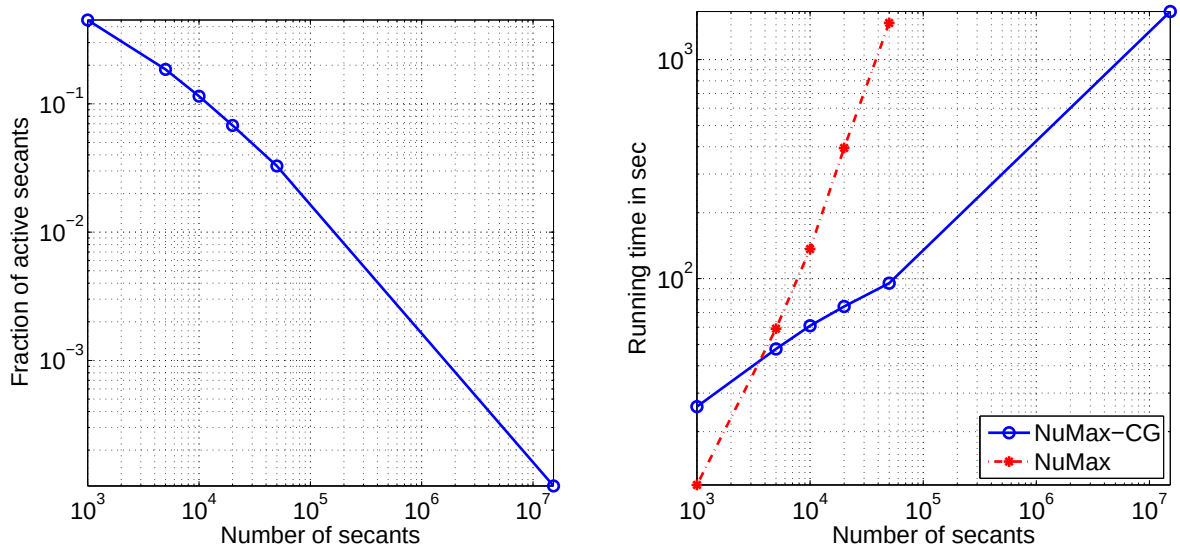



Fig. 5. *Performance of NuMax-CG on the MNIST handwritten digit database [53]. (a) Ratio of active secants to total number of secants for problems of different sizes. As the problem size (number of secants) increases, the ratio of* active *secants decreases exponentially; this implies dramatic improvements in computational cost for NuMax-CG over NuMax. (b) Timing plots comparing NuMax-CG and NuMax for problems of different sizes.*

space. Then, given a query data point, nearest neighbor-based techniques identify the $k$ points in the training dataset closest to the query point and use these points for further processing.

As the dimension $N$ of the data grows, the computational cost of finding the $k$ nearest neighbors becomes challenging [55]. To counter this challenge, as opposed to computing nearest neighbors of the query data point, one can instead construct a near-isometric embedding of the data into an $M$-dimensional space and estimate *approximate nearest neighbors* (ANN) in the embedded space. The ANN principle forms the core of *locality sensitive hashing* (LSH), a popular technique for high-dimensional pattern recognition and information retrieval [56, 57]. Many existing ANN methods (including LSH) either compute a randomized linear dimensionality reduction or a PCA decomposition of the data. In contrast, we observe that NuMax provides a linear near-isometric embedding that achieves a given distortion $\delta$ while *minimizing* $M$. In other words, NuMax can potentially enable more efficient ANN computations over conventional approaches.

We test the efficiency of our approach on a set of $Q = 4000$ images taken from the LabelMe database [58]. This database consists of high-resolution photographs of both indoor and outdoor scenes. We compute GIST feature descriptors [59] for every image. In our case, the GIST descriptors are vectors of size $N = 512$ that coarsely express the dominant spatial statistics of the scene; such descriptors

have been shown to be very useful for image retrieval purposes. Therefore our "ground truth" data consists of a matrix of size $N \times Q$. Since the number of pairwise secants in this case is extremely high ($S = \binom{Q}{2} \approx 8 \times 10^6$), we use NuMax-CG to estimate the linear embedding of lowest rank for a given distortion parameter $\delta$. We record $M$, the rank of the optimal linear embedding, and for comparison purposes we also compute $M$-dimensional random linear projections of the data as well as the best $M$-term PCA approximation of the data. We perform subsequent ANN computations for a set of 1000 test query points in the corresponding $M$-dimensional space.

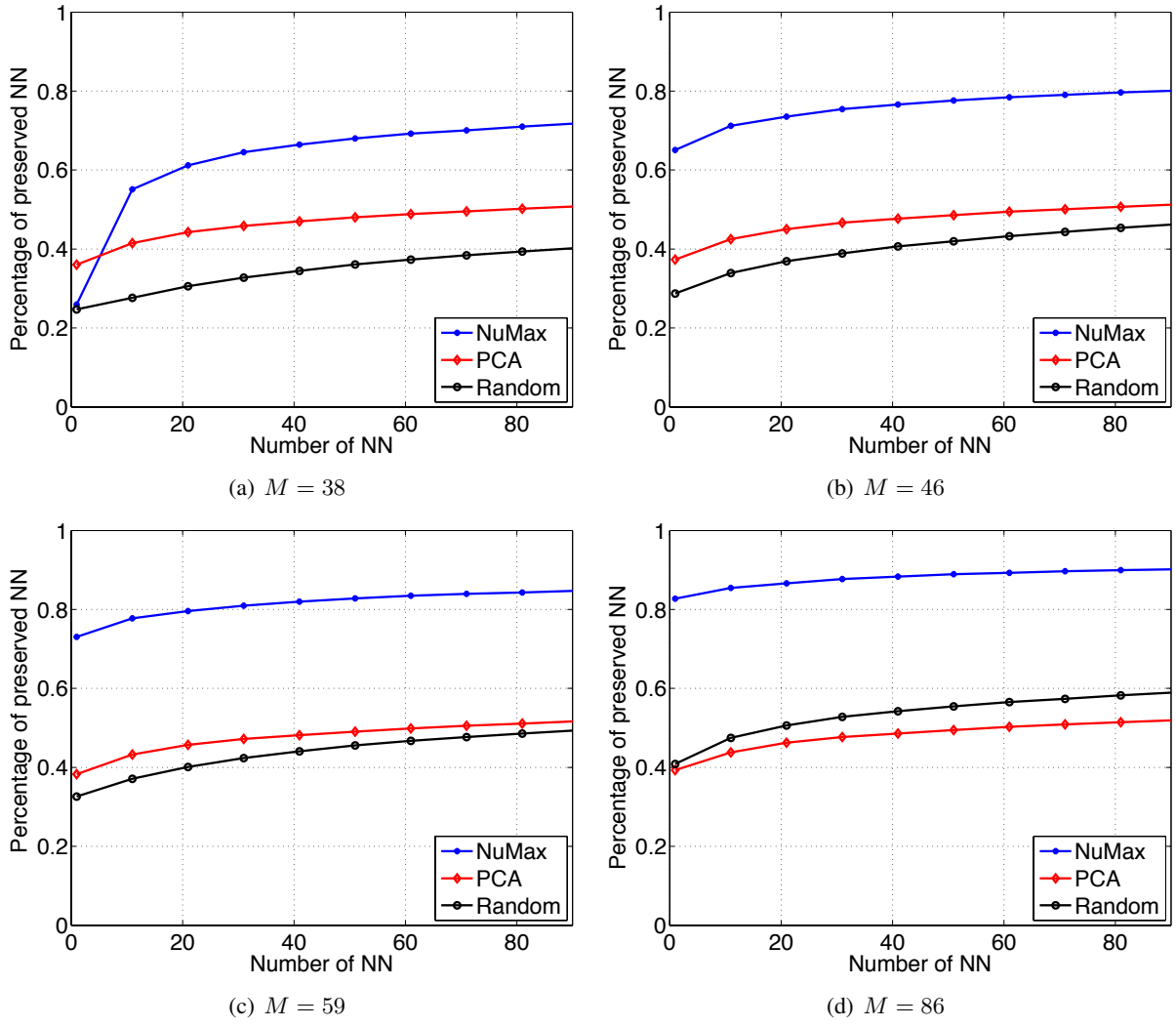
Figure 6 displays the benefits of using the linear embedding generated by NuMax-CG in ANN computations. For a given neighborhood size $k$, we plot the fraction of $k$-nearest neighbors computed using the full (ground truth) $N$-dimensional data that are also $k$-nearest neighbors in the corresponding $M$-dimensional embedding. We observe from Figure 6 that the linear embedding obtained by NuMax-CG provides the best results for a wide range of measurements $M$ and neighborhood sizes $k$. In particular, for embedding dimensions of $M > 45$, NuMax-CG outperforms both PCA and random projections for all values of $k$ by a significant margin.

### *C. Compressive Sensing of Manifold-Modeled Signals*

We demonstrate the utility of our framework for designing efficient compressive sensing (CS) measurement matrices. As discussed in Section II, the canonical approach in CS theory and practice is to construct matrices $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$, with as small $M$ as possible, that satisfy the RIP (with distortion parameter $\delta$) on the set of signals of interest. Typically, such matrices are constructed simply by drawing elements from, say, a standard normal probability distribution. Our proposed framework and NuMax algorithm suggests an alternate approach for constructing CS measurement matrices that are tailored to specific signal models.

We perform the following numerical experiment. Given a set of example signals originating from a low-dimensional manifold, we divide it into training and test datasets. Using the training dataset, we learn a measurement matrix $\mathbf{\Psi}$ that satisfies the RIP for all secants generated from the training dataset using NuMax-CG for a pre-chosen value of $\delta$. Given such a measurement matrix, we are interested in (a) characterizing the RIP of the matrix $\mathbf{\Psi}$ when applied to secants from the *test* dataset, and (b) characterizing the efficiency of CS recovery using $\mathbf{\Psi}$ on signals belonging to the test dataset.

Figure 7 displays the results of this experiment on an image dataset corresponding to a two-dimensional (2D) manifold of a translating Gaussian blob. Each element on this 2D-manifold corresponds to an image of size $N = 32 \times 32 = 1024$ pixels. The standard deviation of the blob is chosen as 6 pixels. As the




(a) $M = 38$ (b) $M = 46$

(c) $M = 59$ (d) $M = 86$

Fig. 6. *Approximate Nearest Neighbors (ANN) for the LabelMe dataset using various linear embedding methods. We choose a set of 4000 images and compute GIST features of size $N = 512$ for every image. For a given number of nearest neighbors $k$, we plot the average fraction of $k$-nearest neighbors that are retained in an $M$-dimensional embedding relative to the full $N$-dimensional data. NuMax-CG provides the best embedding results for a wide range of measurements $M$ and neighborhood sizes $k$.*

training dataset, we select images where the center pixel of the Gaussian blob is on an even row and column. All other images are considered to comprise the test dataset. Figure 7(a) compares the number of measurements required to reach a specified isometry constant $\delta_{\text{learn}}$. As in earlier experiments, NuMax requires significantly fewer measurements, as compared to more conventional (random) CS matrices, to achieve the same value of $\delta_{\text{learn}}$.

Figure 7(b) demonstrates the variation of the empirical isometry constant of both on new, unseen secants from the test dataset. In Figure 7(b), $\delta_{\text{learn}}$ is the parameter used for applying NuMax to the

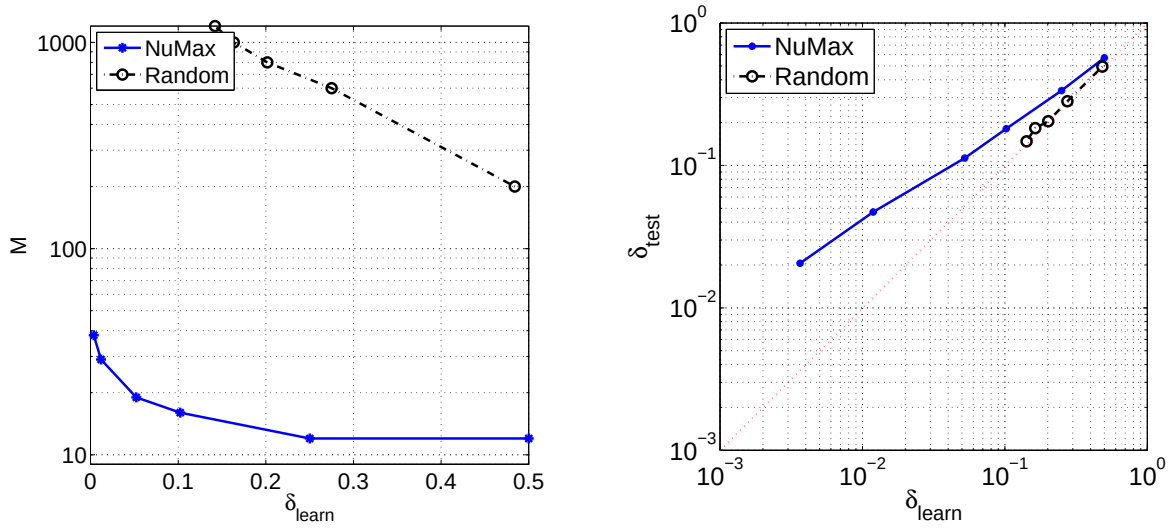



Fig. 7. *(a) Number of measurements $M$ vs. input isometry constant $\delta_{learn}$. (b) Empirical (observed) isometry constant $\delta_{test}$ vs. input isometry constant $\delta_{learn}$ for NuMax and random projections.*

training dataset, while $\delta_{\text{test}}$ is the worst-case distortion among all pairwise secants from the test dataset. Thanks to their universality, we observe that random matrices enjoy the same isometry constant on both training and test datasets. However, we observe that for the matrix $\mathbf{\Psi}$ generated by NuMax, $\delta_{\text{test}}$ is marginally greater than $\delta_{\text{learn}}$. This suggests a moderate loss of universality using $\mathbf{\Psi}$, but a significant gain in terms of lowering the number of measurements.

Finally, we demonstrate the improved performance of CS recovery using NuMax embeddings. We obtain (noisy) compressive measurements using both random Gaussian matrices and the matrices obtained by NuMax for different values of measurement SNR and $M$. Using the noisy measurements, we perform CS recovery via Manifold Iterative Pursuit (MIP), a projected-gradient type method for the recovery of manifold-modeled signals [60]. Figure 8 compares the recovery performance for different SNRs and different number of measurements. We observe that in terms of recovered signal MSE, NuMax outperforms random Gaussian measurements for all values of SNR and for all values of $M$.

### *D. Supervised Classification*

*1) MNIST digit classification:* The MNIST handwritten digits dataset consists of 10 classes, one for each digit from $0-9$, with 60,000 training data points and 10,000 test data points. We used the $N = 400$-dimensional version of the dataset that does not include extra space at the boundaries. Here, the number of secants is extraordinarily large, up to $\binom{60000}{2} = 1.8 \times 10^9 = 1.8$ billion secants.

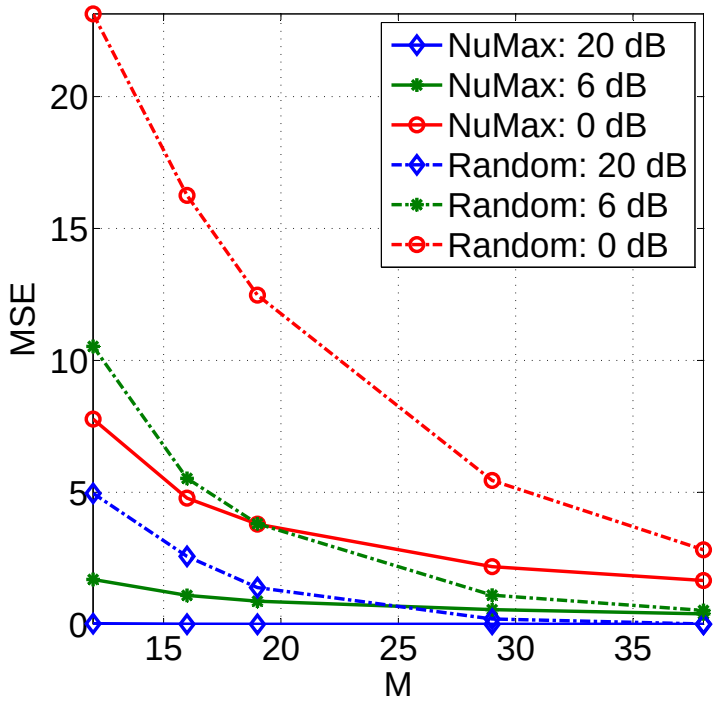



Fig. 8. *Compressive sensing recovery performance for NuMax and random projections. NuMax outperforms random Gaussian projections in terms of recovered signal MSE for all ranges of measurements $M$ as well as signal-to-noise ratios.*

Table I shows NN classification performance of NuMax, PCA and Gaussian projections for various lower-dimensional embedding dimensions, corresponding to several values of $\delta$ in NuMax. We used the rank of the NuMax solution to set the value of $M$ for PCA and Gaussian embeddings. As we see from Table I, NuMax outperforms both methods by a significant margin achieving a mis-classification rate of $2.99\%$ at a dimensionality of $M = 72$; in contrast, for the same dimensionality, Gaussian and PCA produce a mis-classification rate of $5.79\%$ and $4.40\%$.

We compare the classification performance of NuMax and NuMax-Class in Table II. For the same value of $\delta$, not only does NuMax-Class produce a lower-rank solution, but it also provides a lower mis-classification rate as compared to NuMax thereby outperforming all the linear DR techniques. Specifically, for $M = 52$, NuMax-Class achieves a mis-classification rate of $2.68\%$, while that NuMax achieves a mis-classification rate of $2.99\%$ at $M = 72$. This demonstrates the considerable potential gains using class-specific dimensionality reduction.

Table II also reports MATLAB processing times required to obtain NuMax and NuMax-Class solutions. We used the CG version of the algorithms for this dataset; both algorithms scale gracefully to large-scale problems. For larger values of $\delta$, it takes approximately $2$ hours to obtain the solution. The runtime

TABLE I

*Mis-classification rates on the MNIST dataset for all 10 classes. We compare the performance of NuMax, Gaussian matrices, and PCA for the same dimensionality of the lower-dimensional space. We used a nearest neighbor classifier for all dimensionality reduction techniques.*

| Rank of NuMax solution $M$ | | 72 | 97 | 167 |
|---|---|---|---|---|
| Distortion $\delta$ | | 0.40 | 0.25 | 0.1 |
| Mis-classification rate in % | NuMax | **2.99** | **3.11** | **3.31** |
| | Gaussian | 5.79 | 4.51 | 3.88 |
| | PCA | 4.40 | 4.38 | 4.41 |

TABLE II

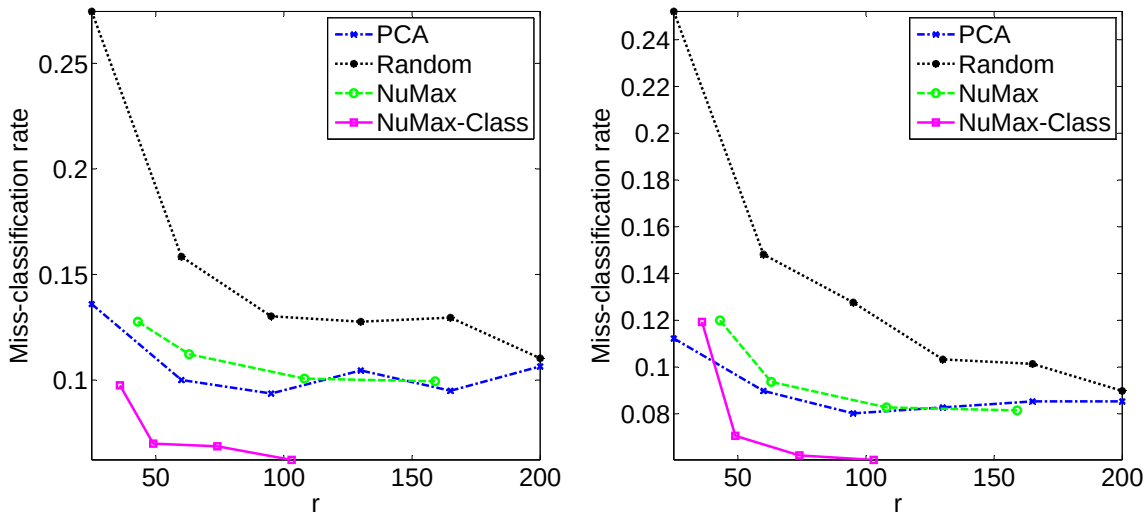
*Comparison of the classification rates of NuMax and NuMax-Class over the MNIST dataset (see Table I for comparisons with other linear DR techniques).*

| Distortion | $\delta = 0.4$ | | $\delta = 0.25$ | | $\delta = 0.1$ | |
|---|---|---|---|---|---|---|
| Algorithm | NuMax | NuMax-Class | NuMax | NuMax-Class | NuMax | NuMax-Class |
| Rank | 72 | 52 | 97 | 69 | 167 | 116 |
| Prob. error | 2.99 | **2.68** | 3.11 | **2.72** | 3.31 | **3.09** |
| Time (hrs) | 2.35 | 1.90 | 4.85 | 5.57 | 10.64 | 9.73 |
| Active secants | 6950 | 4068 | 12121 | 6746 | 29702 | 17323 |

increases by a factor of $5\times$ when we decrease the distortion parameter $\delta$ to $0.1$. This reflects the general intuition that smaller values of $\delta$ result in a larger number of active constraints, and more computationally intensive sub-problems.

*2) Spoken letter recognition:* We tested NuMax and its classification variant, NuMax-Class, on the Isolet dataset obtained from the UCI Machine learning repository.[3] This dataset comprises of 26 classes, one for each alphabet in English language. The dataset set consists of 617-dimensional datapoints, with 6238 training points and 1559 test points. In Fig. 9, we compare the performance of NuMax, NuMax-Class, PCA, and random Gaussian embeddings in $k$ nearest neighbor classification. To determine the optimal number of neighbors ($k$) to be used in the classifier, we used a cross-validation approach. Specifically, 10% of the training dataset was used as a cross-validation dataset, and was used to select the optimal parameter $k$. Figures 9(a) and (b) show cross-validation and test performance, respectively, for varying dimension of the embedded space. On the whole, NuMax-Class significantly outperforms other linear dimensionality reduction techniques; specifically, when projected to a $105-$dimensional space, the mis-classification rate offered by NuMax is merely 6%.

[3]http://archive.ics.uci.edu/ml/datasets/ISOLET




(a) Cross-validation (b) Testing

Fig. 9. *Performance of NuMax, its classification variant, PCA and Random projections on the ISOLET dataset.*

## VI. Discussion

In this paper, we have taken some initial steps towards constructing a comprehensive algorithmic framework that creates a *linear*, *isometry-preserving* embedding of a high-dimensional dataset. Our framework is based on a convex optimization formulation (in particular, the SDP (4)) that approximately preserves the norms of all pairwise secants of the given dataset. We have developed efficient algorithms, NuMax and NuMax-CG, that efficiently construct the desired embedding with considerably smaller computational complexity than existing approaches. Our NuMax methods can be easily adapted to perform more complicated machine learning tasks, such as approximate nearest neighbors (ANN) as well as supervised binary classification. In addition, the NuMax embeddings can be successfully used in CS applications where the signals of interest can be modeled as elements lying on a smooth, low-dimensional manifold.

Several challenges remain. First, our approach relies on the efficiency of the nuclear norm as a proxy for the matrix rank in the objective function in (4). A natural question is under what conditions the optimum of the convex relaxation (4) equals the optimum of the nonconvex problem (3). Moreover, while the speed of convergence of our proposed algorithms (NuMax and NuMax-CG) have been shown to empirically shown to be far better than traditional methods, a rigorous convergence analysis of our algorithms remains open. Finally, from a practical perspective, it is important nowadays to build algorithms for datasets that involve millions (or even billions) of training signals, and optimization on such datasets is only feasible

when performed in a highly distributed fashion. We defer these to future work.

## REFERENCES


[1] C. Hegde, A. Sankaranarayanan, and R. Baraniuk, "Near-isometric linear embeddings of manifolds," in *Proc. IEEE Work. Stat. Signal Processing (SSP)*, Ann Arbor, MI, Aug. 2012.

[2] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Trans. Automat. Control*, vol. 26, no. 1, pp. 17–32, 1981.

[3] D. Achlioptas, "Database-friendly random projections," in *Proc. Symp. Principles of Database Systems (PODS)*, Santa Barbara, CA, May 2001.

[4] W. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," in *Proc. Conf. Modern Anal. and Prob.*, New Haven, CT, Jun. 1982.

[5] R. Baraniuk and M. Wakin, "Random projections of smooth manifolds," *Found. Comput. Math.*, vol. 9, no. 1, pp. 51–77, 2009.

[6] K. Clarkson, "Tighter bounds for random projections of manifolds," in *Proc. Symp. Comp. Geom.* ACM, 2008, pp. 39–48.

[7] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Const. Approx.*, vol. 28, no. 3, pp. 253–263, 2008.

[8] E. Candès, "Compressive sampling," in *Proc. Int. Congress of Math.*, Madrid, Spain, Aug. 2006.

[9] R. Tütüncü, K. Toh, and M. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Math. Prog.*, vol. 95, no. 2, pp. 189–217, 2003.

[10] I. Polik, "Sedumi 1.3," 2010, Available online at http://sedumi.ie.lehigh.edu.

[11] Z. Wen, *First-order Methods for Semidefinite Programming*, Ph.D. thesis, Columbia University, 2009.

[12] G. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Math. Operations Research*, vol. 8, no. 1, pp. 101–111, 1960.

[13] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Human Genetics*, vol. 7, no. 2, pp. 179–188, 1936.

[14] H. Harman, *Modern Factor Analysis*, U. Chicago Press, 1976.

[15] T. Cox and M. Cox, *Multidimensional Scaling*, Chapman & Hall / CRC, Boca Raton, FL, 1994.

[16] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[17] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by local linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[18] M. Belkin and P. Niyogi, "Semi-supervised learning on Riemannian manifolds," *Machine Learning*, vol. 56, pp. 209–239, 2004.

[19] D. Donoho and C. Grimes, "Hessian Eigenmaps: Locally linear embedding techniques for high-dimensional data," *Proc. Natl. Acad. Sci.*, vol. 100, no. 10, pp. 5591–5596, 2003.

[20] K. Weinberger and L. Saul, "Unsupervised learning of image manifolds by semidefinite programming," *Int. J. Comput. Vision*, vol. 70, no. 1, pp. 77–90, 2006.

[21] D. Broomhead and M. Kirby, "The Whitney Reduction Network: A method for computing autoassociative graphs," *Neural Comput.*, vol. 13, pp. 2595–2616, 2001.

[22] D. Broomhead and M. Kirby, "Dimensionality reduction using secant-based projection methods: The induced dynamics in projected systems," *Nonlinear Dynamics*, vol. 41, no. 1, pp. 47–67, 2005.

[23] N. Verma, "Distance preserving embeddings for general n-dimensional manifolds," *J. Machine Learning Research*, vol. 14, pp. 2415–2448, 2013.

[24] E. Elhamifar, "Sparse manifold clustering and embedding," in *Proc. Adv. in Neural Processing Systems (NIPS)*, 2011.

[25] B. Shaw and T. Jebara, "Minimum volume embedding," in *Intl. Conf. Artificial Intell. Stat.*, 2007, pp. 460–467.

[26] N. Linial, E. London, and Y. Rabinovich, "The geometry of graphs and some of its algorithmic applications," *Combinatorica*, vol. 15, no. 2, pp. 215–245, 1995.

[27] N. Alon, "Problems and results in extremal combinatorics," *Discrete Math.*, vol. 273, no. 1, pp. 31–53, 2003.

[28] E. Xing, M. Jordan, S. Russell, and A. Ng, "Distance metric learning with application to clustering with side-information," in *Proc. Adv. in Neural Processing Systems (NIPS)*, 2002, pp. 505–512.

[29] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon, "Metric and kernel learning using a linear transformation," *J. Machine Learning Research*, vol. 13, pp. 519–547, 2012.

[30] B. Kulis, "Metric learning: a survey," *Found. and Trends in Machine Learning*, vol. 5, no. 4, pp. 287–364, 2012.

[31] B. Bah, S. Becker, V. Cevher, and B. Gozcu, "Metric learning with rank and sparsity constraints," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, May 2014.

[32] E. Grant, C. Hegde, and P. Indyk, "Nearly optimal linear embeddings into very low dimensions," in *IEEE GlobalSIP Symposium on Sensing and Statistical Inference*, Austin, TX, Dec. 2013.

[33] B. Bah, A. Sadeghian, and V. Cevher, "Energy aware adaptive bi-Lipschitz embeddings," in *Int. Conf. on Sampling Theory and Appl. (SAMPTA)*, Bremen, Germany, July 2013.

[34] J. Matousek, "Open problems on embeddings of finite metric spaces," 2011, Available online at http://kam.mff.cuni.cz/~matousek/metrop.ps.

[35] M. Fazel, *Matrix Rank Minimization With Applications*, Ph.D. thesis, Stanford Univ., 2002.

[36] F. Alizadeh, "Interior-point methods in semidefinite programming with applications to combinatorial optimization," *SIAM J. Optimization*, vol. 5, no. 01, 1995.

[37] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–500, 2010.

[38] E. Candès and B. Recht, "Simple bounds for low-complexity model reconstruction," to appear in *Math. Prog.*, 2012.

[39] A. Barvinok, "Problems of distance geometry and convex properties of quadratic maps," *Discrete and Comput. Geometry*, vol. 13, no. 1, pp. 189–202, 1995.

[40] P. Moscato, M. Norman, and G. Pataki, "On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues," *Math. Operations Research*, vol. 23, no. 2, pp. 339–358, 1998.

[41] E. Candès, T. Strohmer, and V. Voroninski, "PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming," *Comm. Pure and Appl. Math.*, vol. 66, no. 8, pp. 1241–1274, Aug. 2013.

[42] J. Douglas and H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Trans. Amer. Math. Soc*, vol. 82, pp. 421–439, 1956.

[43] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and bregman iterative methods for matrix rank minimization," *Math. Prog.*, vol. 128, no. 1, pp. 321–353, 2011.

[44] N. Halko, P. Martinsson, and J. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.

[45] J. Meijerink and H. van der Vorst, "An iterative solution method for linear systems of which the coefficient matrix is a symmetric $M$-matrix," *Math. Comp.*, vol. 31, no. 137, pp. 148–162, 1977.

[46] D. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Prog.*, vol. 45, no. 1, pp. 503–528, 1989.

[47] S. Boyd and L. Vanderberghe, *Convex Optimization*, Cambridge Univ. Press, Cambridge, England, 2004.

[48] B. He and X. Yuan, "On the $o(1/n)$ convergence rate of the douglas-rachford alternating direction method," *SIAM J. Num. Anal.*, vol. 50, no. 2, pp. 700–709, 2012.

[49] W. Deng, M.-J. Lai, and W. Yin, "On the $o(1/k)$ convergence and parallelization of the alternating direction method of multipliers," Tech. Rep., UCLA CAM 13-64, 2013.

[50] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," Tech. Rep., Rice CAAM 12-14, 2012.

[51] M. Hong and Z.-Q. Luo, "On the Linear Convergence of the Alternating Direction Method of Multipliers," *ArXiv e-prints*, Aug. 2012.

[52] D. Boley, "Linear convergence of ADMM on a model problem," Tech. Rep., University of Minnesota, Department of Computer Science and Engineering, TR 12-009, 2012.

[53] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 1998, Available online at http://yann.lecun.com/exdb/mnist.

[54] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inform. Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[55] S. Arya, D. Mount, N. Netanyahu, R. Silverman, and A. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *J. ACM*, vol. 45, no. 6, pp. 891–923, 1998.

[56] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. ACM Symp. Theory of Comput.*, New York, NY, 1998, pp. 604–613.

[57] G. Shakhnarovich, T. Darrell, and P. Indyk, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*, MIT Press, Cambridge, MA, 2005.

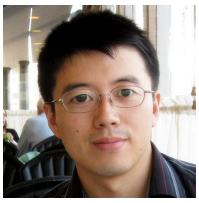
[58] B. Russell, A. Torralba, K. Murphy, and W. Freeman, "LabelMe: A database and web-based tool for image annotation," *Int. J. Comput. Vision*, vol. 77, no. 1, pp. 157–173, 2008.

[59] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[60] P. Shah and V. Chandrasekharan, "Iterative projections for signal identification on manifolds," in *Proc. Allerton Conf. on Comm., Contr., and Comp.*, Monticello, IL, Sept. 2011.

**Chinmay Hegde** joined the Theory of Computation (ToC) group at MIT in October 2012, where he is currently a Shell-MIT postdoctoral research associate. In Fall 2016, he will join Iowa State University as an assistant professor in the the ECpE Department. He received his B.Tech. degree in Electrical Engineering from IIT Madras (India) in 2006, and M.S. and Ph.D. degrees in Electrical and Computer Engineering from Rice University in 2010 and 2012, respectively. His research interests lie broadly in the areas of signal and image processing, theory of algorithms, and machine learning.

**Aswin Sankaranarayanan** is an assistant professor in the ECE Department at Carnegie Mellon University. Aswin received his B.Tech in Electrical Engineering from the Indian Institute of Technology, Madras in 2003, and M.Sc. and Ph.D. degrees from the ECE Department at the University of Maryland, College Park in 2007 and 2009, respectively. He was a post-doctoral researcher at the DSP group in Rice University, prior to joining Carnegie Mellon University. His research interests lie broadly in the areas of computer vision, signal processing, and image and video acquisition.



**Wotao Yin** is a professor in the Department of Mathematics in the University of California at Los Angeles. His research interests lie in computational optimization and its applications in image processing, machine learning, and other inverse problems. He received his B.S. in mathematics from Nanjing University in 2001, and his M.S. and Ph.D. in operations research from Columbia University in 2003 and 2006, respectively. During 2006 - 2013, he was with Rice University. He won the NSF CAREER award in 2008 and the Alfred P. Sloan Research Fellowship in 2009. His recent work has been in optimization algorithms for large-scale and distributed signal processing and machine learning problems.

**Richard G. Baraniuk** is currently the Victor E. Cameron Professor of Electrical and Computer Engineering at Rice University, a member of the Digital Signal Processing (DSP) group, Director of the Rice center for Digital Learning and Scholarship (RDLS), and Director/Founder of Connexions and OpenStax College. He received the B.Sc. degree in 1987 from the University of Manitoba, the M.Sc. degree in 1988 from the University of Wisconsin-Madison, and the Ph.D. degree in 1992 from the University of Illinois at Urbana-Champaign, all in Electrical Engineering. His research interests lie in the areas of signal, image, and information processing and include machine learning and compressive sensing.