Journal of Machine Learning Research (2013) Submitted 11/13; Published x/xx

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

Chinmay Hegde CHINMAY@CSAIL.MIT.EDU
CSAIL, MIT
Cambridge, MA 02139

Aswin C. Sankaranarayanan SASWIN@ECE.CMU.EDU
ECE Department, CMU
Pittsburgh, PA 15213

Wotao Yin WOTAOYIN@MATH.UCLA.EDU
Department of Mathematics, UCLA
Los Angeles, CA 90095

Richard G. Baraniuk RICHB@RICE.EDU
ECE Department, Rice University
Houston, TX 77005

Editor: TBD

Abstract

We propose a novel framework for the *deterministic* construction of *linear, near-isometric* embeddings of finite sets 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 Ψ that preserves the norms of all the vectors in $\mathcal{S}(\mathcal{X})$ up to a distortion parameter δ. While affine rank minimization is NP-hard, we show that this problem can be relaxed to a convex program that can be solved using a tractable semidefinite program (SDP). To enable scalability of the 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). 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 applications in machine learning and signal processing via a range of experiments on large-scale synthetic and real datasets.

Keywords: Linear Embeddings, Dimensionality Reduction, Compressive Sensing

1. Introduction

1.1 Motivation

We are in the throes of a "data crisis". The sheer size of raw data acquired and processed by data sources of diverse modalities poses a challenge to current state-of-the-art information processing systems. Fortunately, a significant body of work has emerged in machine learning research that can help counter this formidable challenge. This line of work is often termed *dimensionality reduction* and typically involves devising a very concise representation of the high-dimensional data, with as little loss of intrinsic information as possible. Such a concise representation is often called a low-dimensional *embedding*.

The canonical approach in statistics to constructing such an embedding is principal components analysis (PCA) (Moore, 1981). A closely-related technique, better suited for classification applications, is Linear Discriminant Analysis (LDA) (Fisher, 1936). PCA and LDA are both *linear* techniques. Subsequently, several *non-linear* generalizations (e.g., metric- and non-metric multi-dimensional scaling (Cox and Cox, 1994), kernel PCA (Mika et al., 1998)) have been developed; see Section 2.2 for a detailed discussion. Linear techniques enjoy two broad advantages:

1. *Computational efficiency*: The dimensionality reduction process for linear embeddings can be explicitly represented by a matrix mapping with fewer rows than columns. For a specific high-dimensional data vector, the dimensionality reduction is achieved via a simple matrix-vector multiplication.

2. *Generalizability*: Linear embeddings produce a smooth, globally defined mapping that can be easily applied to unseen, out-of-sample data vectors.

In this paper, we will exclusively focus on linear dimensionality reduction techniques. Consider PCA as a conceptual representative for such techniques. It is well-known that a PCA embedding has an important drawback: PCA can *arbitrarily distort pairwise distances* between sample data points (Achlioptas, 2001). Due to this behaviour, 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. Such a distortion of pairwise distances is symptomatic of other linear embedding techniques developed in the literature, including LDA, metric multi-dimensional scaling (Cox and Cox, 1994), metric learning (Yang and Jin, 2006), locality-preserving projections (LPP) (He and Niyogi, 2010), and many others.

The susceptibility of linear embedding methods to produce arbitrary distortions of pairwise distances is an important pitfall, both in theory and practice. Algorithms in machine learning often assume the availability of (reasonably) accurate estimates of pairwise distances between data points, and we will discuss some of these algorithms in Section 5. Any embedding technique that does not guarantee preservation of pairwise distances, therefore, violates this assumption and can potentially hamper the performance of such algorithms.

An example of a linear embedding technique that avoids the above pitfall is the method of *random projections*. Consider $\mathcal{X}$, a cloud of Q points in a high-dimensional Euclidean space $\mathbb{R}^N$. The Johnson-Lindenstrauss Lemma (Johnson and Lindenstrauss, 1982) states that $\mathcal{X}$ can be linearly mapped to a subspace of dimension $M = \mathcal{O}\left(\log Q\right)$ with *very small* 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 $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ with $M \ll N$ 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 (Johnson and Lindenstrauss, 1982; Achlioptas, 2001).

Random projections can be extended to more general signal classes beyond finite point clouds. For example, random linear projections provably preserve, up to a given distortion, all pairwise distances between points lying on compact, differentiable low-dimensional manifolds (Baraniuk and Wakin, 2009; Clarkson, 2008) as well as pairwise distances between all sparse signals (Baraniuk et al., 2008). Random projections have attracted significant attention in machine learning and theoretical computer science over the last two decades. The intuition of using random linear projections is also a fundamental component of compressive sensing (CS), an emergent framework for signal acquisition and reconstruction (Candès, 2006; Donoho, 2006).

Despite their conceptual simplicity, random projections also suffer from certain shortcomings. Their theoretical guarantees are *probabilistic* (i.e., there is a non-zero chance that the obtained embedding does *not* satisfy a (near) isometry), and *asymptotic* (i.e., the guarantees hold only when the problem dimensions are sufficiently high). Further, a random mapping is independent of the data under consideration and hence *cannot leverage* any special geometric structure of the data if present.

1.2 Our Contributions

In this paper, we propose a general *deterministic* approach for constructing linear, near-isometric embeddings of a finite high-dimensional point cloud. Our approach is based on a specific convex relaxation that lends itself both to efficient algorithms for constructing embeddings as well as easy specialization to specific applications.

Optimization framework. 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 $\mathbf{\Psi}$ that preserves the norms of all of the vectors in $\mathcal{S}(\mathcal{X})$ up to a desired distortion parameter δ. The affine rank minimization problem is known to be NP-hard, and so we perform a convex relaxation to obtain a trace-norm minimization (4), that is equivalent to a tractable semidefinite program (SDP). The SDP can be solved using any generic interior-point method for convex programming (for example, the solvers SDPT3 (Tütüncü et al., 2003) or SeDuMi (Polik, 2010)).

Efficient algorithms. While our proposed formulation can be solved using out-of-the-box convex solvers, the convergence of generic SDP solvers for our specific problem is typically very slow, even for small problem sizes. Further, the presence of the max-norm constraints in (4), though convex, negates the direct application of existing first-order methods for large-scale semidefinite programming (Wen, 2009).

We resolve this issue by developing two new algorithms. First, we develop an algorithm that we call *Nuclear norm minimization with Max-norm constraints* (NuMax) to solve (4). NuMax is based on the Alternating Direction Method of Multipliers (ADMM); it decouples the complex SDP formulation (4) into a sequence of easy-to-solve subproblems and enables much faster rates of convergence than standard approaches. Second, we propose a modified, greedy version of NuMax that mirrors the *column generation* (CG) approach commonly used to solve large-scale linear programs (Dantzig and Wolfe, 1960). 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., 10^9 or greater).

Applications. We demonstrate that the NuMax framework is useful for a number of applications in machine learning and signal processing. First, if the training set $\mathcal{X}$ comprises sufficiently many points that are uniformly drawn from a low-dimensional smooth manifold $\mathcal{M}$, then the matrix $\mathbf{\Psi}$ represents a near-isometric linear embedding over *all* pairwise secants of $\mathcal{M}$. In other words, $\mathbf{\Psi}$ satisfies the restricted isometry property (RIP) for signals belonging to $\mathcal{M}$ and therefore 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.*

Going further, by carefully pruning the secant set $\mathcal{S}(\mathcal{X})$, we can tailor $\mathbf{\Psi}$ for more general signal inference tasks, such as *supervised classification.* Specifically, in the context of classification, instead of preserving the lengths of the secants, we can seek to increase the inter-class distances while shrinking the intra-class distances. For a fixed distortion parameter, this has the dual benefit of decreasing the dimensionality of the embedded space while increasing the classification rate. Several numerical experiments in Section 5 demonstrate the advantages of this approach.

1.3 Paper outline

We organize this paper as follows. In Section 2 we provide a brief background on existing methods for linear dimensionality reduction and highlight some connections with compressive sensing. In Section 3 we introduce our main theoretical contributions and propose the SDP formulation for designing "good" linear embeddings. In Section 4 we develop efficient algorithms that can solve our proposed SDP for large-scale problems. In Section 5 we apply our linear embedding framework to a number of diverse problems and demonstrate its efficiency both on synthetic and real-world datasets. In Section 6 we provide concluding remarks and list potential directions for future work.

2. Background

2.1 Notation

In this paper, we will exclusively work with real-valued vectors and matrices; howecver our techniques can be extended to the complex case *mutatis mutandis.* We use lowercase boldface letters to denote vectors, uppercase boldface letters to denote matrices, and calligraphic letters to denote sets or set-valued operators. The ℓ_p-norm of a vector $\mathbf{x} = [x_1, \ldots, x_N]^T \in \mathbb{R}^N$ is defined as

$$\|\mathbf{x}\|_p = \begin{cases} \left(\sum_{i=1}^N |x_i|^p\right)^{\frac{1}{p}}, & p \in [1, \infty) \\ \max\limits_{i=1,2,\ldots,N} |x_i|, & p = \infty. \end{cases}$$

Given two symmetric matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{N \times N}$, we write $\mathbf{X} \succeq \mathbf{Y}$ if the matrix $\mathbf{X} - \mathbf{Y}$ 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 of $\mathbf{X}$. The rank of $\mathbf{X}$ is equal to the number of nonzero entries in $\boldsymbol{\sigma}$. 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 ℓ_2-norm of $\boldsymbol{\sigma}$. The *nuclear norm* of $\mathbf{X}$, denoted by $\|\mathbf{X}\|_*$, is equal to the sum of its singular values, or equivalently, the ℓ_1-norm of

$\boldsymbol{\sigma}$. When $\mathbf{X}$ is a positive semidefinite (PSD) symmetric matrix, $\|\mathbf{X}\|_*$ is equal to the trace, or the sum of diagonal values, of $\mathbf{X}$.

2.2 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 was first proposed by Pearson (Pearson, 1901) and is also sometimes referred to as the Karhunen-Loéve transform or the Hotelling transform. The work of Eckart and Young showed that the principal components can be efficiently discovered via a singular value decomposition (SVD) of the data covariance matrix (Eckart and Young, 1936).

PCA is ubiquitous in machine learning and statistics (Dony and Haykin, 1995; Tipping and Bishop, 1999). The method proceeds 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). Denote the projected columns by the rank-r matrix $\mathbf{X}_r$ (of size $N \times Q$). Then, it is well-known that $\mathbf{X}_r$ is the optimal approximation to $\mathbf{X}$ in terms of the Frobenius norm, i.e., $\|\mathbf{X} - \mathbf{X}_r\|_F \leq \|\mathbf{X} - \mathbf{Y}\|_F$, where $\mathbf{Y}$ is any other rank-r matrix.

Since the Frobenius norm measures the error aggregated over all columns of a matrix, PCA can be viewed as an efficient *linear* embedding method that incurs minimal distortion of the data *on average.* Furthermore, PCA can be adapted to account for problem-specific requirements. For example, if the data vectors originate from one of two classes and the goal is to maintain class separability, then PCA can be modified to produce related techniques such as Fisher's Linear Discriminant Analysis (LDA) or Factor Analysis (Fisher, 1936; Harman, 1976).

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 the pairwise dissimilarity matrix in the lower-dimensional space, $D(f(\mathcal{X})) \approx D(\mathcal{X})$. If the pairwise dissimilarities correspond to Euclidean distances, then MDS is equivalent to PCA (Cox and Cox, 1994) and $f(\mathcal{X})$ is simply a *linear* projection of $\mathcal{X}$. If the pairwise dissimilarities are captured by some other distance metric, then the embedding is *nonlinear,* in general; see Section 2.3 for a discussion of nonlinear embedding techniques.

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 (Achlioptas, 2001). Therefore, any information contained in the geometric inter-relationships between data points is irrevocably lost. One can easily generate examples of datasets where the distance between the PCA embeddings of two distinct high-dimensional points is vanishingly small. In other words, PCA and MDS are not guaranteed to be *isometric* (i.e., distance-preserving) or even *invertible*. This can severely affect both algorithm design and analysis.

2.3 Nonlinear Embeddings

While the focus of this paper is primarily on *linear* embeddings, we point out that several sophisticated *nonlinear* data embedding methods have emerged over the last decade; see, for example, Tenenbaum et al. (2000); Roweis and Saul (2000); Belkin and Niyogi (2004); Donoho and Grimes (2003); Weinberger and Saul (2006). These methods are sometimes referred to as *manifold learning* algorithms. The list of manifold learning methods 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 (Broomhead and Kirby, 2001, 2005). 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, therefore cannot ensure that the final obtained mapping is (near) isometric.

Our approach also has connections to Locally Linear Embedding (LLE), proposed in Roweis and Saul (2000). 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 (as specified by the orthogonal projection onto the corresponding subspace), but is *globally* nonlinear. It is unknown whether or not the LLE ensures a (near)-isometry. The more recent Sparse Manifold Learning and Clustering (SMCE) approach, proposed in Elhamifar (2011) aims to address this issue by constructing an embedding by directly operating on the normalized secant set $\mathcal{S}(\mathcal{X})$; however, the algorithm relies on a spectral decomposition that does not seem to be accompanied by isometry guarantees.

Finally, we note that the idea of using semidefinite programming (SDP) to construct low-dimensional embeddings has been explored; see, for example, the algorithms of Weinberger and Saul (2006) and Shaw and Jebara (2007). 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 above, the mappings obtained are highly nonlinear and consequently 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 very large datasets.

2.4 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 quickly becoming classical (for an excellent introduction to this subject, see Linial et al. (1995)). Concretely, we seek an embedding that satisfies 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) *on $\mathcal{X}$ if there exists a positive constant $\delta > 0$ such that, 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 δ 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 $M = N$. It is less clear whether embedding operators that satisfy the RIP even exist for $M < N$. The celebrated *Johnson-Lindenstrauss* (JL) Lemma answers this question in the affirmative (Johnson and Lindenstrauss, 1982). A simplified version of the JL Lemma is as follows.

Lemma 1 *Consider a dataset $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_Q\} \subset \mathbb{R}^N$. Let $M \geq \mathcal{O}\left(\delta^{-2} \log Q\right)$. 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 $\mathcal{X}$.*

We highlight some important features of the JL Lemma. First, like PCA, the linear embedding $\mathbf{\Phi}$ is computationally efficient and can be applied to out-of-sample points. Second, unlike PCA, the constructed embedding $\mathbf{\Phi}$ is *universal*, i.e., fully independent of the dataset $\mathcal{X}$. Instead of projecting the data onto the basis vectors of the subspace formed by the singular vectors of $\mathcal{X}$, one simply picks a few basis vectors at random and projects the data onto these vectors. The JL Lemma guarantees that such an embedding preserves the local geometric structure of $\mathcal{X}$. Third, the dimension M of the lower-dimensional embedding is only logarithmic in the number of data points and is independent of the ambient dimension N; therefore, potentially $M \ll N$.

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 (Baraniuk and Wakin, 2009; Clarkson, 2008). A particularly striking 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 (Baraniuk et al., 2008). 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 $K \log N/K$. Further, this recovery can be achieved efficiently via a convex program or a greedy pursuit (Candès, 2006; Donoho, 2006; Tropp and Gilbert, 2007). From a practical signal acquisition perspective, it is even possible to build practical *signal acquisition* systems where the embedding $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$ is performed in real-time (Wakin et al., 2006; Laska et al., 2007).

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 δ, there exist datasets that cannot be embedded into any M-dimensional space where $M \leq \delta^{-2}\log(\delta^{-1})\log Q$ (Alon, 2003). However, this worst case only occurs for a specific configuration of data points that rarely occurs in practice. Further, the universality property of random projections negates its ability to construct embeddings that leverage the intrinsic geometry of a given set of data vectors.

2.5 Metric learning

Closely related to the ideas proposed in this paper is the body of work on *metric learning*; see the survey articles (Yang and Jin, 2006; Kulis, 2012) for a full description. Given a dataset and an intended task (for example, classification), the goal of metric learning is to *learn* a distance metric that is better than (or at least, as good as) the Euclidean distance. There has been significant recent work in this context for learning *Mahalanobis* distances, i.e, metrics of the form $d^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \Sigma (\mathbf{x} - \mathbf{y})$. Here, the metric is fully specified by the positive semi-definite matrix Σ.

Xing et al. (2002) use the concept of metric learning for improved clustering by tuning the metric to user labels. Blitzer et al. (2005) and Globerson and Roweis (2005) promote nearest neighbor classification by learning metrics that seek to preserve local neighborhoods containing points from the same class. Jain et al. (2012) show that metric learning approaches can be scaled to high-dimensional data by using a LogDet divergence-based regularization term. In all the cases discussed above, the problem is formulated as a semidefinite program whose solution is the matrix Σ that defines the Mahalanobis distance.

There are several apparent similarities between metric learning and NuMax. For example, both use convex optimization techniques to estimate a positive semi-definite matrix. The Mahalanobis distance can be viewed as a linear transformation of the data points, and if the solution to the metric learning problem happens to be low-rank, then this linear transformation is also dimensionality reducing. However, there are several important differences. First, NuMax is a *dimensionality reduction* technique and hence, expressly optimizes for *low rank* solutions. To the best of our knowledge, none of the metric learning techniques actively seek low rank solutions. In some metric learning approaches, a low rank *approximation* of the final solution can be obtained, but often only via a post-processing step after the main optimization (e.g. see Jain et al. (2012)). Second, NuMax is *explicitly* geared towards producing embeddings that are (near) isometric, and this is typically not the goal of metric learning.

3. Near-Isometric Linear Embeddings

3.1 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 (Baraniuk and Wakin, 2009), we will refer to δ as the *isometry constant.* We form the secant set $\mathcal{S}(\mathcal{X})$ using (1) to obtain a set of $S = \binom{Q}{2}$ unit vectors $\mathcal{S}(\mathcal{X}) = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_S\}$. Then, we seek a *projection matrix* $\mathbf{\Psi} \in \mathbb{R}^{M \times N}$ with as few rows as possible that satisfies the RIP on $\mathcal{S}(\mathcal{X})$.

We cast this problem in terms of an optimization over the space of PSD symmetric matrices. Let $\mathbb{S}^{N \times N}$ be the set of symmetric $N \times N$ matrices. Define $\mathbf{P} \doteq \mathbf{\Psi}^T \mathbf{\Psi} \in \mathbb{S}^{N \times N}$;

then, $\mathrm{rank}(\mathbf{P}) = M$. We also have the constraints that $|\,\|\mathbf{\Psi v}_i\|_2^2 - 1| = \left|\mathbf{v}_i^T \mathbf{P} \mathbf{v}_i - 1\right|$ is no greater than δ 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, the matrix $\mathbf{P}$ we seek is the solution to the optimization problem

$$\begin{aligned} \text{minimize} \quad & \mathrm{rank}(\mathbf{P}) \\ \text{subject to} \quad & \|\mathcal{A}(\mathbf{P}) - \mathbf{1}_S\|_\infty \leq \delta, \\ & \mathbf{P} \succeq 0. \end{aligned} \tag{3}$$

Rank minimization is a non-convex problem and is known to be NP-hard, in general. Therefore, following Fazel (2002), we propose to instead solve a nuclear-norm relaxation of (3):

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{P}\|_* \\ \text{subject to} \quad & \|\mathcal{A}(\mathbf{P}) - \mathbf{1}_S\|_\infty \leq \delta, \\ & \mathbf{P} \succeq 0. \end{aligned} \tag{4}$$

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 (Alizadeh, 1995). Once the solution $\mathbf{P}^* = \mathbf{U \Lambda U}^T$ to (4) is found, $\mathrm{rank}(\mathbf{P}^*)$ 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 $\mathbf{\Lambda}_M = \mathrm{diag}\{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_M\}$ denotes the M leading (non-zero) eigenvalues of $\mathbf{P}^*$, and $\mathbf{U}_M$ denotes the set of corresponding eigenvectors.[1] 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 δ. The convex optimization formulation (4) is conceptually very simple, the only inputs being the input dataset $\mathcal{X}$ and the desired isometry constant $\delta > 0$.

3.2 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 thoroughly examined in a number of different settings (Recht et al., 2010; Candès and Recht, 2012). However, we highlight two unique aspects of the optimization problem (4). First, the ℓ_∞-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,

[1] An interesting question is whether other types of matrix square-roots provide added benefits. We defer this to future work.

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, the following classical result from SDP provides an upper bound of the rank of the optimum $\mathbf{P}^*$ in (4).

Proposition 1 *(Barvinok, 1995; Moscato et al., 1998) Let r^* be the rank of the optimum to the SDP (4). Then,*

$$r^* \leq \left\lceil \frac{\sqrt{8|\mathcal{S}(\mathcal{X})|+1}-1}{2} \right\rceil. \tag{6}$$

In essence, the rank of $\mathbf{P}^*$ grows as the square root of the cardinality of the secant set $\mathcal{S}(\mathcal{X})$. The upper bound on the optimal rank r^* provided in (6) can be loose, since the cardinality of $\mathcal{S}(\mathcal{X})$ can potentially be very large. Additionally, one might intuitively expect the optimal rank r^* to depend on the geometric arrangement of the data vectors in $\mathcal{X}$, as well as the input isometry constant δ; however, the bound in Proposition 1 does not reflect this dependence.

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 $\mathcal{X}$. 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 Candès et al. (2013) has addressed this question in a somewhat different context. Specifically, they provide sharp theoretical guarantees under which a similar nuclear-norm relaxation with rank-1 constraints produces the desired low-rank solution. However, the underlying assumption in their work is that the rank-1 constraint vectors are independently and randomly generated from a Gaussian distribution. This assumption does not typically hold for an arbitrary dataset $\mathcal{X}$, and therefore that theory does not apply in our case.

We also note that the recent results by Bah et al. (2013) and Grant et al. (2013) address the theoretical properties of a convex program that resembles (4), albeit under more stringent assumptions on the target embedding matrix Φ.

3.3 Linear Embeddings of Manifolds

The optimization framework (4) provides a novel method for producing an efficient, low-dimensional, linear embedding of an arbitrary high-dimensional dataset $\mathcal{X}$. More specifically, suppose $\mathcal{X}$ comprises points that are sampled from a K-dimensional *smooth, compact* manifold $\mathcal{M} \subset \mathbb{R}^N$. In such a scenario, we can make a stronger claim than Proposition 1: under certain assumptions on $\mathcal{M}$ and $\mathcal{X}$, the near-isometry property of the proposed embedding Ψ holds not only on pairwise secants of $\mathcal{X}$, but more generally to all pairwise secants in $\mathcal{M}$. In other words, the near-isometry property of Ψ can be extended to new, unseen data from the underlying manifold $\mathcal{M}$.

More precisely, suppose that $\mathbf{\Psi}$ satisfies the RIP with constant δ over a training set $\mathcal{X} \in \mathbb{R}^N$. If the training set $\mathcal{X}$ comes from a particular high-resolution sampling of points on or close to $\mathcal{M}$, then $\mathbf{\Psi}$ provably satisfies the RIP with a slightly larger constant over the *entire manifold* $\mathcal{M}$; see Section 3.2.5 of Baraniuk and Wakin (2009) for the detailed

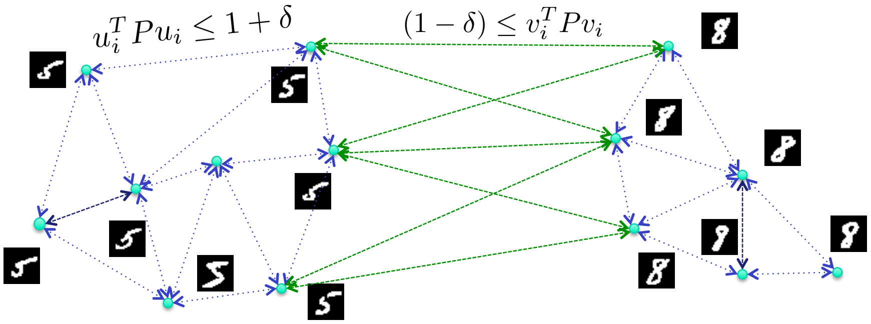


Figure 1: *A desirable objective for classification is to promote nearest neighbors of a point to come from its own class. We achieve this by altering the near-isometric constraints for secants. First, we relax the upper bound on the near-isometry for the inter-class secants; hence, they can expand in 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 a lower-rank solution and higher-classification rates.*

derivation. We numerically validate this phenomenon on synthetic manifold-modeled data below in Section 5.

Several SDP-based approaches for processing manifold-modeled data have been developed in the literature (Weinberger and Saul, 2006; Kulis et al., 2007), but their focus has been on producing general *nonlinear* mappings of high-dimensional datasets into a low-dimensional space. In contrast, our optimization formulation (4) produces an explicit *linear* embedding operator $\mathbf{\Psi} \in \mathbb{R}^{M \times N}$, thereby enabling easy application to out-of-sample, unseen data points. Furthermore, since $\mathbf{\Psi}$ satisfies the RIP for all signals belonging to $\mathcal{M}$, $\mathbf{\Psi}$ can be interpreted as a highly optimized *compressive sensing* matrix specifically tailored for signals belonging to $\mathcal{M}$. We explore this further in our numerical experiments below in Section 5.

An important challenge in this context is the size of the sampled dataset $\mathcal{X}$. The proof techniques of Baraniuk and Wakin (2009) assume that the training set $\mathcal{X}$ is an ϵ-cover of $\mathcal{M}$, i.e., for every $\mathbf{m} \in \mathcal{M}$, there exists an $\mathbf{x} \in \mathcal{X}$ such that $\min_{\mathbf{x} \in \mathcal{X}} d_{\mathcal{M}}(\mathbf{m}, \mathbf{x}) \leq \epsilon$ for a small constant $\epsilon > 0$. However, covering results from high-dimensional geometry state that the cardinality of such a set $\mathcal{X}$, in the worst case, can be exponential in the manifold dimension K, i.e.,

$$|\mathcal{X}| = \mathcal{O}\left(\left(\frac{1}{\delta}\right)^K\right).$$

For practical real-world problems, computations involving such large training sets $\mathcal{X}$ may be intractable, and thus one may have to resort to heuristic sub-optimal methods. In Section 4 below, we develop one such sub-optimal but efficient method. Nevertheless, our approach can be viewed as an initial step towards obtaining provably efficient linear embeddings that preserve the isometric structure of arbitrary nonlinear submanifolds of the signal space.

3.4 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 the dataset $\mathcal{X}$ 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

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{P}\|_* && (7) \\ \text{subject to} \quad & \mathbf{v}_i^T \mathbf{P} \mathbf{v}_i \geq 1 - \delta, \quad && \forall \ \mathbf{v}_i \in \text{inter-class secants} \\ & \mathbf{u}_i^T \mathbf{P} \mathbf{u}_i \leq 1 + \delta, \quad && \forall \ \mathbf{u}_i \in \text{intra-class secants} \\ & \mathbf{P} \succeq 0. \end{aligned}$$

This convex program 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 in Section 5.

4. Efficient 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 (Tütüncü et al., 2003; Polik, 2010) 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.

4.1 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

$$\begin{array}{ll} \min\limits_{\mathbf{P},\mathbf{L},\mathbf{q}} & \|\mathbf{P}\|_* \\ \text{subject to} & \mathbf{P} = \mathbf{L}, \;\; \mathcal{A}(\mathbf{L}) = \mathbf{q}, \;\; \|\mathbf{q} - \mathbf{1}_S\|_\infty \le \delta, \;\; \mathbf{P} \succeq 0. \end{array} \tag{8}$$

This approach can be viewed as an instance of the Douglas-Rachford variable splitting method in convex programming (Douglas and Rachford, 1956). Next, we relax the linear constraints and form an *augmented Lagrangian* of (8) as follows:

$$\begin{array}{ll} \min\limits_{\mathbf{P},\mathbf{L},\mathbf{q}} & \|\mathbf{P}\|_* + \frac{\beta_1}{2}\|\mathbf{P} - \mathbf{L} - \mathbf{\Lambda}\|_F^2 + \frac{\beta_2}{2}\|\mathcal{A}(\mathbf{L}) - \mathbf{q} - \boldsymbol{\omega}\|_2^2 \\ \text{subject to} & \|\mathbf{q} - \mathbf{1}_S\|_\infty \le \delta, \;\; \mathbf{P} \succeq 0. \end{array} \tag{9}$$

Here, the symmetric matrix $\mathbf{\Lambda} \in \mathbb{S}^{N\times N}$ and vector $\boldsymbol{\omega} \in \mathbb{R}^S$ represent the scaled Lagrange multipliers. The optimization in (9) is carried out over the variables $\mathbf{P}, \mathbf{L} \in \mathbb{S}^{N\times N}$ and $\mathbf{q} \in \mathbb{R}^S$, while $\mathbf{\Lambda}$ 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 (9) 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^{th} iteration of the algorithm. The following steps are performed until convergence.

1. **Update q**: Isolating the terms that involve $\mathbf{q}$, we obtain a new estimate $\mathbf{q}_{k+1}$ as the solution of the constrained optimization problem

$$\mathbf{q}_{k+1} \leftarrow \arg\min_{\mathbf{q}} \frac{\beta_2}{2}\|\mathcal{A}(\mathbf{L}_k) - \boldsymbol{\omega}_k - \mathbf{q}\|_2^2, \;\; \text{s.t. } \|\mathbf{q} - \mathbf{1}_S\|_\infty \le \delta.$$

   This problem has a closed-form solution using a component-wise truncation procedure for the entries in $\mathbf{q}$. Denote $\mathbf{z} = \mathcal{A}(\mathbf{L}^k) - \boldsymbol{\omega}^k - \mathbf{1}_S$. Then, it is easily seen that

$$\mathbf{q}_{k+1} = \mathbf{1}_S + \text{sign}(\mathbf{z}) \cdot \min(|\mathbf{z}|, \delta), \tag{10}$$

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

2. **Update P**: Isolating the terms that involve $\mathbf{P}$, we obtain a new estimate $\mathbf{P}_{k+1}$ as the solution of the constrained optimization problem

$$\mathbf{P}_{k+1} \leftarrow \arg\min_{\mathbf{P}} \|\mathbf{P}\|_* + \frac{\beta_1}{2}\|\mathbf{P} - \mathbf{L}_k - \mathbf{\Lambda}_k\|_F^2, \;\; \text{s.t. } \mathbf{P} \succeq 0.$$

   This problem also admits an efficient closed form solution via the *eigenvalue shrinkage operator* (similar to the approach described in (Ma et al., 2011)). Denote $\mathbf{P}' = \mathbf{L}_k + \mathbf{\Lambda}_k$

and perform the eigen decomposition $\mathbf{P}' = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{\Sigma} = \mathrm{diag}(\boldsymbol{\sigma})$. Then, the optimum $\mathbf{P}_{k+1}$ can be expressed as

$$\mathbf{P}_{k+1} = \mathbf{V}\mathcal{D}_\alpha(\mathbf{\Sigma})\mathbf{V}^T, \quad \mathcal{D}_\alpha(\mathbf{\Sigma}) = \mathrm{diag}(\{(\sigma_i - \alpha)_+\}), \tag{11}$$

where $\alpha = \frac{1}{\beta}$ 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 numerical linear algebra (RandNLA) techniques (Halko et al., 2011).

3. **Update L**: Isolating the terms that involve $\mathbf{L}$, we obtain a new estimate $\mathbf{L}_{k+1}$ as the solution of the unconstrained optimization problem

$$\mathbf{L}_{k+1} \leftarrow \arg\min_{\mathbf{L}} \frac{\beta_1}{2}\|\mathbf{P}_k - \mathbf{L} - \mathbf{\Lambda}_j\|_F^2 + \frac{\beta_2}{2}\|\mathcal{A}(\mathbf{L}) - \mathbf{q}_{k+1} - \boldsymbol{\omega}_k\|_2^2. \tag{12}$$

This is a least-squares problem, and the minimum is achieved by solving the linear system of equations

$$\beta_1(\mathbf{P}_k - \mathbf{L} - \mathbf{\Lambda}_j) = \beta_2\mathcal{A}^*(\mathcal{A}(\mathbf{L}) - \mathbf{q}_{k+1} - \boldsymbol{\omega}_k), \tag{13}$$

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 $\mathcal{O}\left(N^2 S^2\right)$. The least-squares solution to (13) can be calculated using a number of existing methods for solving large-scale linear equations, such as conjugate gradients (Meijerink and van der Vorst, 1977; Liu and Nocedal, 1989).

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

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

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 η, β_1, and β_2. In all of the experiments below in Section 5, we set $\eta = 1.618$ and $\beta_1 = \beta_2 = 1$.

4.2 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 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

Algorithm 1 NuMax

```
Inputs: Secant set $\mathcal{S}(\mathcal{X}) = \{\mathbf{v}_i\}_{i=1}^{S}$, isometry constant $\delta$
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
    $k \leftarrow k + 1$

    $\mathbf{z} \leftarrow \mathcal{A}(\mathbf{L}^k) - \boldsymbol{\omega}^k - \mathbf{b}$
    $\mathbf{q}_{k+1} \leftarrow \mathbf{b} + \mathrm{sign}(\mathbf{z}) \cdot \min(|\mathbf{z}|, \delta)$                {Truncation}

    $\mathbf{P}' \leftarrow \mathbf{L}_k + \boldsymbol{\Lambda}_k, \quad \mathbf{P}' = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^T$
    $\mathbf{P}_{k+1} \leftarrow \mathbf{V}\mathcal{D}_\alpha(\boldsymbol{\Sigma})\mathbf{V}^T$                              {Eigenvalue shrinkage}

    $\mathbf{Z} \leftarrow \beta_2 \mathcal{A}^*(\mathbf{q}_{k+1} + \boldsymbol{\omega}_k), \ \mathbf{Z}' \leftarrow \beta_1(\mathbf{P}_k - \boldsymbol{\Lambda}_k)$
    $\mathbf{L}_{k+1} \leftarrow \beta_2(\mathcal{A}^*\mathcal{A} + I)^\dagger(\mathbf{Z} + \mathbf{Z}')$                   {Least squares}

    $\boldsymbol{\Lambda}_{k+1} \leftarrow \boldsymbol{\Lambda}_k - \eta(\mathbf{P}_k - \mathbf{L}_k)$
    $\boldsymbol{\omega}_{k+1} \leftarrow \boldsymbol{\omega}_k - \eta(\mathcal{A}(\mathbf{L}_k) - \mathbf{q}_k)$                   {Update Lagrange multipliers}

end while
return $\widehat{\mathbf{P}} \leftarrow \mathbf{P}_k$
```

primal and dual optimal values of (4) are equal. Then, classical results in optimization theory (Boyd and Vanderberghe, 2004) state that *complementary slackness* holds and that 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). We prescribe the following steps:

1. Solve (9) 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 $\widehat{\mathbf{P}}$.

Algorithm 2 NuMax-CG

```
Inputs: Secant set $\mathcal{S} = \{\mathbf{v}_i\}_{i=1}^{S}$, isometry constant $\delta$, the NuMax algorithm
Parameters: Size of selected secant sets $S', S''$
Output: Symmetric PSD matrix $\widehat{\mathbf{P}}$

Initialize: Select a subset of $S'$ secants $\mathcal{S}_0$, set $\mathcal{A} : \mathbf{X} \mapsto \{\mathbf{v}_i^T \mathbf{X} \mathbf{v}_i\}_{i=1}^{S'}$
            Obtain initial estimate $\mathbf{P} \leftarrow \mathrm{NuMax}(\mathcal{S}_0, \delta)$
while not converged do
    $\widehat{\mathcal{S}} \leftarrow \{\mathbf{v}_i \in \mathcal{S}_0 : |\mathbf{v}_i^T \mathbf{P} \mathbf{v}_i - 1| = \delta\}$            {Retain active constraints}
    $\mathcal{S}_1 \leftarrow \{\mathbf{v}_i \in \mathcal{S} : \mathbf{v}_i \notin \mathcal{S}_0\}_{i=1}^{S''}$                  {Select additional test secants}
    $\widehat{\mathcal{S}} \leftarrow \widehat{\mathcal{S}} \bigcup \{\mathbf{v}_i \in \mathcal{S}_1 : |\mathbf{v}_i^T \mathbf{P} \mathbf{v}_i - 1| \geq \delta\}$   {Add secants that violate constraints}

    $\mathbf{P} \leftarrow \mathrm{NuMax}(\widehat{\mathcal{S}}, \delta)$                              {Update estimate}
    $\mathcal{S}_0 \leftarrow \widehat{\mathcal{S}}$
end while
return $\widehat{\mathbf{P}} \leftarrow \mathbf{P}$
```

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 (Dantzig and Wolfe, 1960). 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 5, 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 $\mathbf{q} = \mathcal{A}(\mathbf{L})$

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

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

4.3 Convergence

The convergence of NuMax can be understood in terms of the convergence properties of a more general ADMM. An important distinction is that although there are three variables $\mathbf{L}, \mathbf{P}, \mathbf{q}$ in (8), 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 (8), 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. This observation allows us to invoke the existing convergence results of standard ADMM.

For certain types of convex problems, ADMM converges at a rate of $O(1/k)$ (He and Yuan, 2012) (more recently, the rate has been slightly improved to $o(1/k)$ (Deng et al., 2013)). Although we have observed NuMax to have a rate of convergence that appears to be linear, we are not able to establish its linear convergence for arbitrary data. In particular, recent results in Deng and Yin (2012), Hong and Luo (2012), and Boley (2012) prove the linear convergence of ADMM under assumptions such as a strongly convex objective function or the underlying problem being a quadratic program. Unfortunately, these results do not appear to apply to the problem formulation (8).

NuMax-CG calls NuMax to solve a sequence of instances of (8) 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, it is difficult to estimate the actual number of iterations, since it will vary significantly depending on data, parameter choices, and the specific order in which the column generation procedure adds constraints to (8).

4.4 Class-specific NuMax

We now discuss how to solve the classification optimization problem (7) using minor modifications to NuMax and NuMax-CG. 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 δ, 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:

$$\mathcal{A}_c(\mathbf{P}) = \begin{pmatrix} \vdots \\ -\mathbf{v}_i^T\mathbf{P}\mathbf{v}_i \\ \vdots \\ \mathbf{u}_i^T\mathbf{P}\mathbf{u}_i \\ \vdots \end{pmatrix}, \quad \mathbf{b}_c = \begin{pmatrix} \vdots \\ -(1-\delta) \\ \vdots \\ 1+\delta \\ \vdots \end{pmatrix} \tag{14}$$

The convex program (7) can now be succinctly represented as

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{P}\|_* \\ \text{subject to} \quad & \mathcal{A}_c(\mathbf{P}) \leq \mathbf{b}_c \\ & \mathbf{P} \succeq 0. \end{aligned} \tag{15}$$

Algorithm 3 NuMax-Class

```
Inputs: $\mathcal{A}_c, \mathcal{A}_c^*, b$
Parameters: Weights $\beta_1, \beta_2$, step size $\eta$
Output: Symmetric PSD matrix $\widehat{\mathbf{P}}$ that solves (15)

Initialize: $\mathbf{P}_0, \mathbf{L}_0, \boldsymbol{\omega}_0, \mathbf{q}_0, k \leftarrow 0$
while not converged do
    $k \leftarrow k + 1$

    $\mathbf{z} \leftarrow \mathcal{A}_c(\mathbf{L}^k) - \boldsymbol{\omega}^k - \mathbf{b}$
    $\mathbf{q}_{k+1} \leftarrow \min(\mathbf{b}, \mathbf{z})$                                {Truncation}

    $\mathbf{P}' \leftarrow \mathbf{L}_k + \mathbf{\Lambda}_k, \quad \mathbf{P}' = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T$
    $\mathbf{P}_{k+1} \leftarrow \mathbf{V}\mathcal{D}_\alpha(\mathbf{\Sigma})\mathbf{V}^T$                    {Eigenvalue shrinkage}

    $\mathbf{Z} \leftarrow \beta_2\mathcal{A}_c^*(\mathbf{q}_{k+1} + \boldsymbol{\omega}_k), \ \mathbf{Z}' \leftarrow \beta_1(\mathbf{P}_k - \mathbf{\Lambda}_k)$
    $\mathbf{L}_{k+1} \leftarrow \beta_2(\mathcal{A}_c^*\mathcal{A}_c + I)^\dagger(\mathbf{Z} + \mathbf{Z}')$          {Least squares}

    $\mathbf{\Lambda}_{k+1} \leftarrow \mathbf{\Lambda}_k - \eta(\mathbf{P}_k - \mathbf{L}_k)$
    $\boldsymbol{\omega}_{k+1} \leftarrow \boldsymbol{\omega}_k - \eta(\mathcal{A}_c(\mathbf{L}_k) - \mathbf{q}_k)$            {Update Lagrange multipliers}

end while
return $\widehat{\mathbf{P}} \leftarrow \mathbf{P}_k$
```

Here, the $\mathcal{A}_c$ operator captures the specifics of the modified/relaxed isometry constraints on the intra- and inter-class secants. Note that (15) is a more general form of the convex program in (4). Specifically, solvers for either of the problems can be easily modified for the other. Algorithm 3 summarizes *NuMax-Class*, a modification of NuMax for solving (15). The sole difference between the two algorithms is in the truncation step, where a (slightly) different truncation operator is used. Similarly, a CG version of NuMax-Class can be easily derived with minor modifications.

5. 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. Further, 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 will interchangeably use the terms “projections” and “measurements” whenever the context is clear.

5.1 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 (see Figure 2 for example images). 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 via PCA. We achieve this by projecting the secants onto M PCA basis functions learned on the secant set $\mathcal{S}(\mathcal{X})$ and calculating the norms of the projected secants. The worst-case deviation from unity gives the estimate of the isometry constant δ. 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 δ, we solve (4) using NuMax (Algorithm 1) to obtain a positive semidefinite symmetric matrix $\mathbf{P}^*$. We measure the rank of $\mathbf{P}^*$ and denote it by M. We are interested in characterizing the variation of the isometry constant δ with the number of measurements M.

Figure 3(a) plots the variation of the number of measurements M as a function of the isometry constant δ. 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 *4 times fewer* measurements than the next best algorithm (PCA). In Fig. 3(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.

We have defined the isometry constant in terms of the worst-case distortion of the norms of the secants. However, for practical applications, it might be more instructive to consider how the curves for the competing algorithms look like, when all but a fraction of the S secant constraints are satisfied. Therefore, in Fig. 3(a), we have also included curves for PCA and Random Projections which indicate the number of measurements at which all but 1% of the secants achieve a distortion δ. It is clear that NuMax outperforms the other algorithms even in this less restrictive setting.

This phenomenon can be better understood by considering Figure 4. For an embedding dimension of $r = 30$, we record the norms of the projected secants using NuMax, PCA, and random projections, and plot histograms of the secant distortions. We observe that for NuMax, the norms of the (embedded) secants are sharply concentrated at $1 \pm \delta,\ \delta = 0.03$. On the other hand, the norms of the embedded secants using PCA are more spread-out (in fact, they are all smaller than 1, since PCA is a contractive mapping). Finally, the norms of the secants under random projections are much more widely distributed.

Figures 3(a) and (b) can be viewed as analogous to the *rate-distortion curve* commonly studied in information theory; here, δ represents the distortion and the undersampling factor

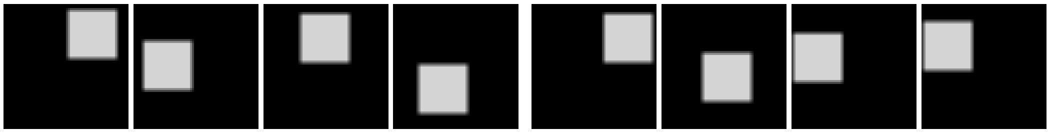


Figure 2: *Example images from a dataset of translating squares. Each image is a point in* $N = 16 \times 16 = 256$*-dimensional space.*

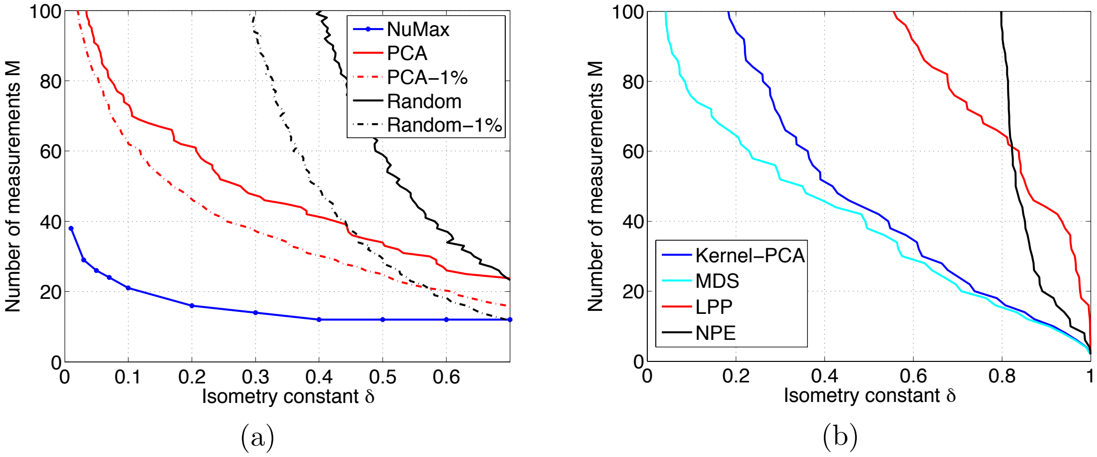


Figure 3: *(a) Empirical isometry constant* δ *vs. number of measurements* M *using NuMax, PCA, and random embeddings. (b) Empirical isometry constant vs. number of measurements using various other embeddings. NuMax ensures global approximate isometry using by far the fewest measurements.*

M/N represents the compression rate. For illustration purposes, we display (in montage form) the measurement basis functions (i.e., rows of Ψ) obtained by NuMax in Figure 5.

Next, we consider a more challenging real-world dataset. The MNIST dataset (LeCun and Cortes, 1998) contains a large number of digital images of handwritten digits and is commonly used as a benchmark for machine learning algorithms. The images exhibit dramatic variations (see Figure 6(a)) and presumably lie on a highly nonlinear, non-differentiable submanifold of the image space. We construct a training dataset $\mathcal{S}(\mathcal{X})$ comprising $S = 3000$ secants and estimate the variation of the isometry constant δ with the number of measurements M. The results of this experiment are plotted in Figure 6(b). Once again, we observe that NuMax provides the best linear embedding for a given value of δ in terms of reduced dimensionality and that both NuMax and PCA outperform random projections. For instance, for a distortion parameter $\delta = 0.2$, NuMax produces an embedding with 8×

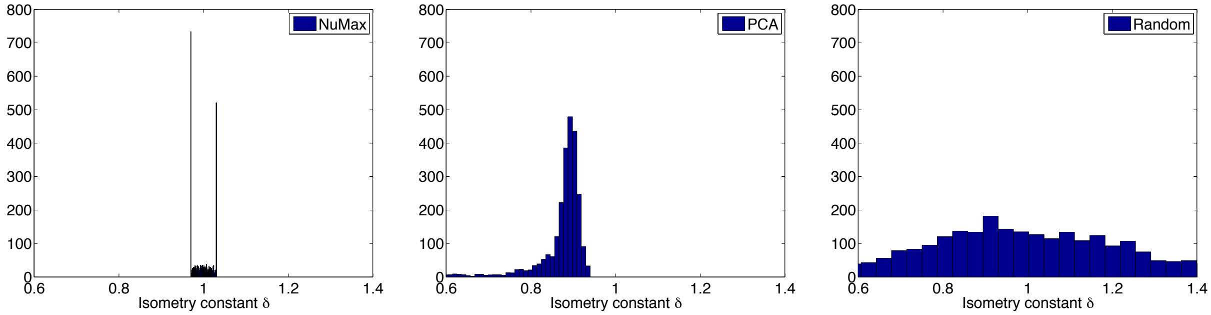


Figure 4: *Histograms of secant distortions using various embedding methods, for the translating squares dataset and an embedding dimension of $r = 30$. The input distortion to NuMax is $\delta = 0.03$.*

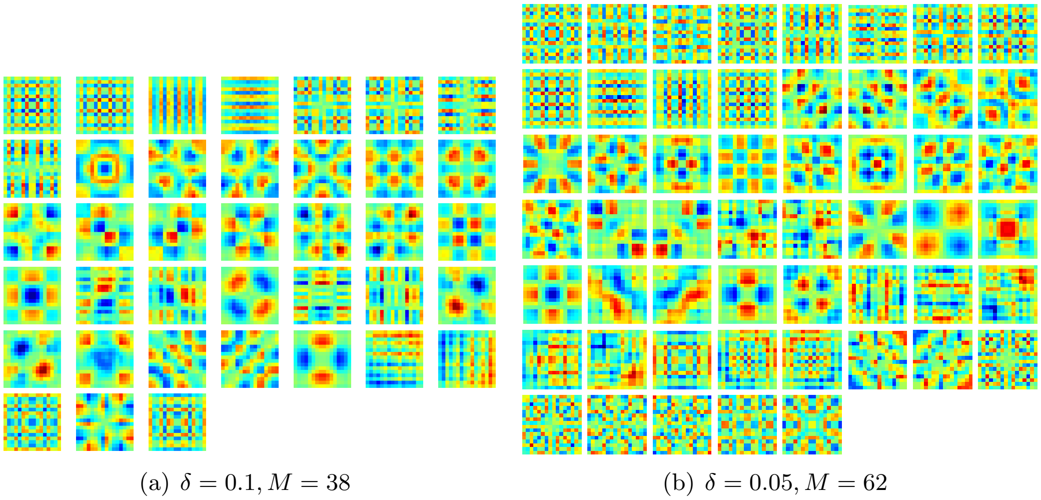
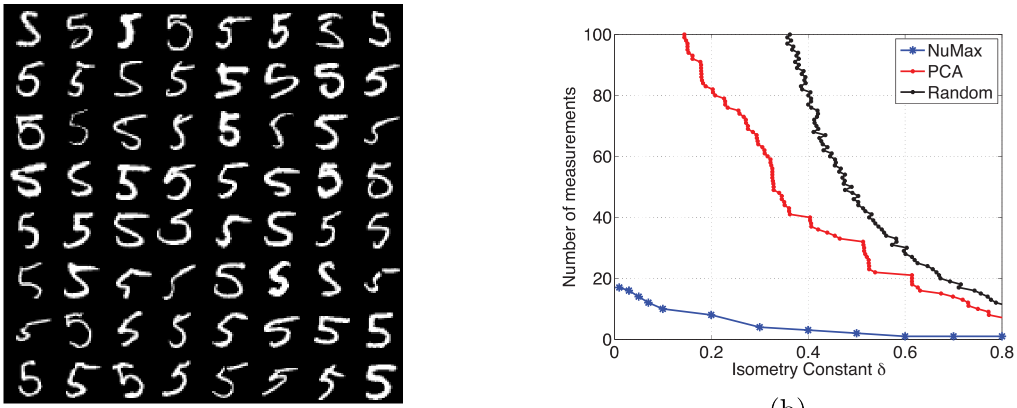


Figure 5: *Montage of basis functions (rows of Ψ) obtained by NuMax for the dataset in Figure 2 with different values of δ.*

fewer measurements than PCA. In essence, NuMax provides the best possible rate-distortion curve in terms of compressing the given image database.

Next, we compare runtime performance 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 7(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 7(b) confirms this fact: for a problem size with $S = 5 \times 10^4$, NuMax-CG outperforms NuMax in terms of running time by a full order of magnitude. Moreoever,



(a)

(b)

Figure 6: *(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 δ 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 $\delta = 0.2$, it produces an embedding with $8\times$ fewer measurements than PCA.*

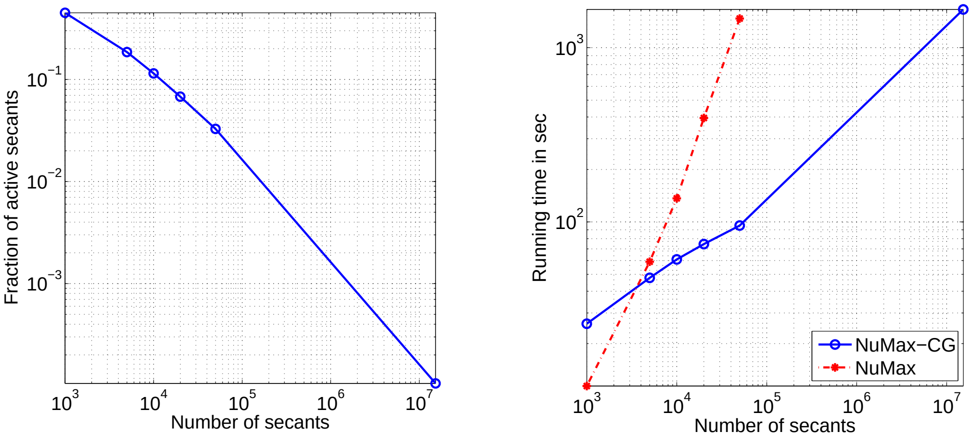
despite the heuristic nature of NuMax-CG, we observed in practice that the solutions obtained NuMax and NuMax-CG are virtually identical. Table 2 provides runtime values on the entire MNIST dataset for different values of δ. MNIST dataset has 60,000 datapoints; thereby, producing a total of 1.8 billion 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.

5.2 Approximate Nearest Neighbors (ANN)

The notion of *nearest neighbors* is vital to numerous problems in estimation, classification, and regression (Cover and Hart, 1967); the ubiquity of NN-based machine learning in part stems from its conceptual simplicity and good performance. Suppose that a large dataset of training examples is available. Then, given a new (query) data point, nearest neighbor-based machine learning techniques identify the k points in the training dataset closest to the query point and use these points for further processing.

Suppose that the data points are modeled as elements of a vector space. As the dimension N of the data grows, the computational cost of finding the k nearest neighbors becomes challenging (Arya et al., 1998). 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. By carefully controlling the distortion in distance caused by the lower-dimensional embedding, efficient inference techniques can be performed with little loss in performance.

The ANN principle forms the core of *locality sensitive hashing* (LSH), a popular technique for high-dimensional pattern recognition and information retrieval (Indyk and Mot-



(a) (b)

Figure 7: *Performance of NuMax-CG on the MNIST handwritten digit database (LeCun and Cortes, 1998). (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.*

wani, 1998; Shakhnarovich et al., 2005). Given a fixed dataset, the time complexity of a particular ANN method directly depends upon the dimension M of the embedded space; the smaller the embedding dimension, the faster the ANN method. Most existing ANN methods (including LSH) either compute a randomized linear dimensionality reduction or a PCA decomposition of the data. In contrast, we immediately observe that NuMax provides a linear near-isometric embedding that achieves a given distortion δ while *minimizing* M. In other words, NuMax can potentially enable far 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 (Russell et al., 2008). This database consists of high-resolution photographs of both indoor and outdoor scenes (see Figure 8) for several examples). We compute GIST feature descriptors (Oliva and Torralba, 2001) 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 δ. 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 9 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

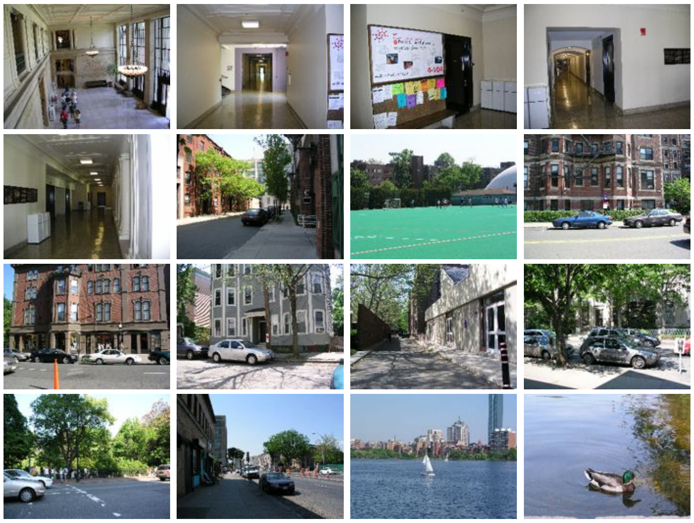


Figure 8: *Example images from the LabelMe dataset (Russell et al., 2008).*

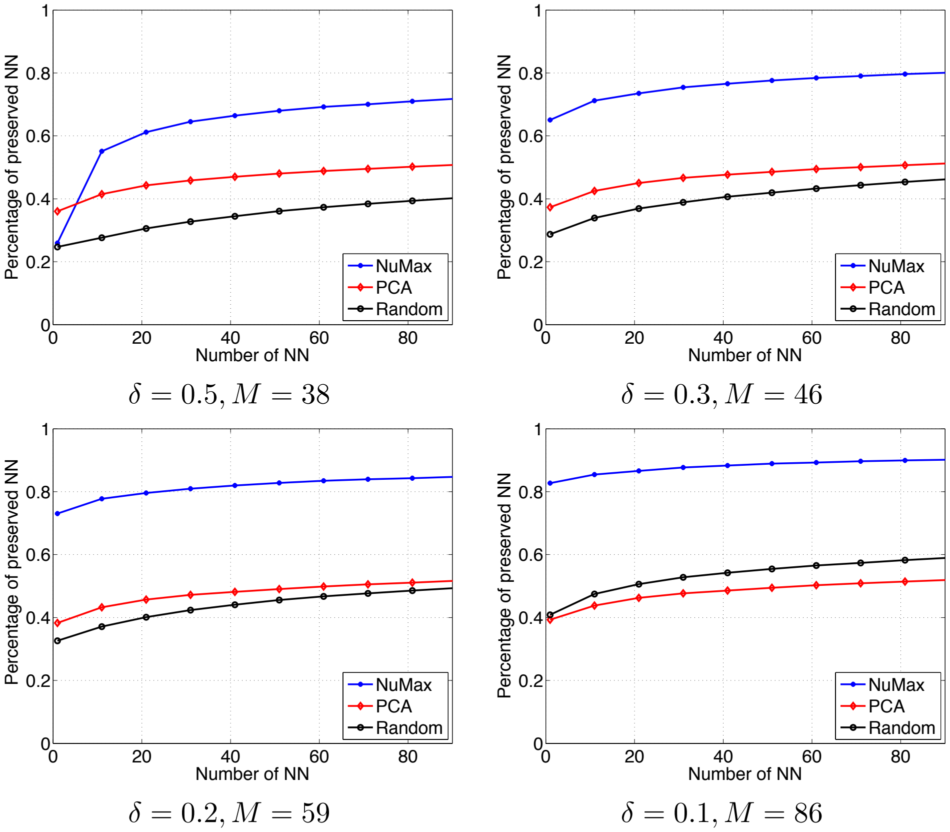
neighbors computed using the full (ground truth) N-dimensional data that are also k-nearest neighbors in the corresponding M-dimensional embedding. We observe from Figure 9 that the linear embedding obtained by NuMax-CG provides the best embedding results for a wide range of measurements M and neighborhood 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.

5.3 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 2, 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 δ) on the set of signals of interest. Typically, such matrices are constructed simply by drawing elements from, say, a standard normal probability distribution. Such matrices are universal in the sense that they can be constructed independently from the signal set of interest. 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 δ. 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 10 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 $N = 32 \times 32 = 256$ pixels. The standard deviation



$\delta = 0.5, M = 38$

$\delta = 0.3, M = 46$

$\delta = 0.2, M = 59$

$\delta = 0.1, M = 86$

Figure 9: *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.*

of the blob is chosen as 6 pixels. As the 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 10(a) compares the number of measurements required to reach a specified isometry constant δ_{learn} for both NuMax and random measurement matrices. As in earlier experiments, NuMax requires significantly fewer measurements, as compared to more conventional (random) CS matrices, to achieve the same value of δ_{learn}.

Figure 10(b) demonstrates the variation of the empirical isometry constant of both on new, unseen secants from the test dataset. In Figure 10(b), δ_{learn} is the parameter used for applying NuMax to the training dataset, while δ_{test} is the worst-case distortion among all pairwise secants from the test dataset. Thanks to their universality, we observe that traditional (random) CS matrices enjoy the same isometry constant on both training and test datasets. However, we observe that for the matrix $\mathbf{\Psi}$ generated by NuMax, δ_{test} is

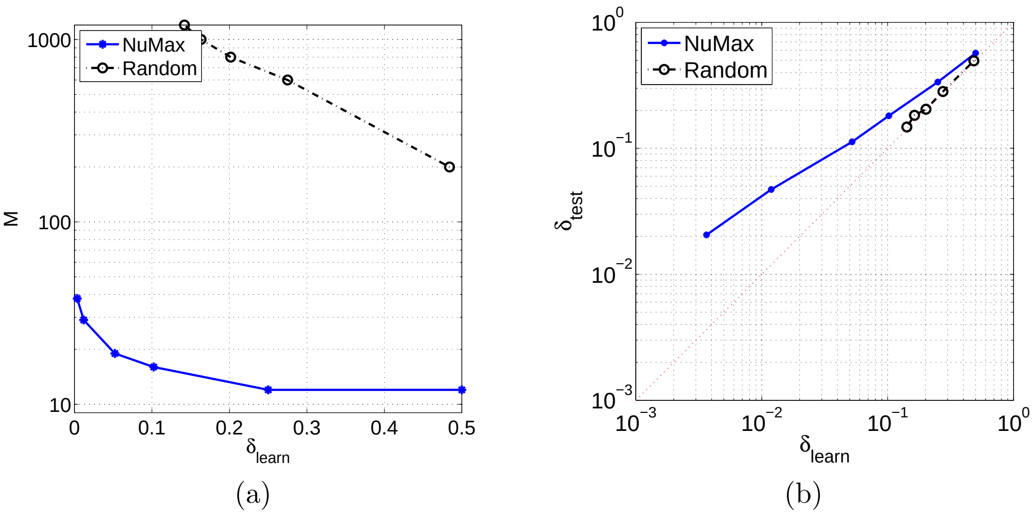


Figure 10: *(a) Number of measurements M vs. input isometry constant δ_{learn}. (b) Empirical (observed) isometry constant δ_{test} vs. input isometry constant δ_{learn} for NuMax and random projections.*

marginally greater than δ_{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 (Shah and Chandrasekharan, 2011). Figure 11 compares the recovery performance for different SNRs and different number of measurements for both random Gaussian and NuMax measurement matrices. 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.

5.4 Supervised Classification

5.4.1 Black and white images

First, we consider a toy supervised classification problem, where our training classes consist of binary images of shifts of a translating disk and a translating square; several example images are shown in Figure 12(a). We construct a training dataset of $S = 2000$ *inter-class* secants and obtain a measurement matrix $\mathbf{\Psi}_{\text{inter}}$ via NuMax. Using a small number of measurements of a test signal, we estimate the class label using a Generalized Maximum Likelihood Classification (GMLC) approach following the framework in (Davenport et al., 2007); assuming the availability of sufficiently many training examples, this formulation is essentially equivalent to ANN-based classification. We repeat this classification experiment using measurement basis functions learned via PCA on the inter-class secants as well as random projections. Figure 12(c) plots the variation of the number of measurements M vs.

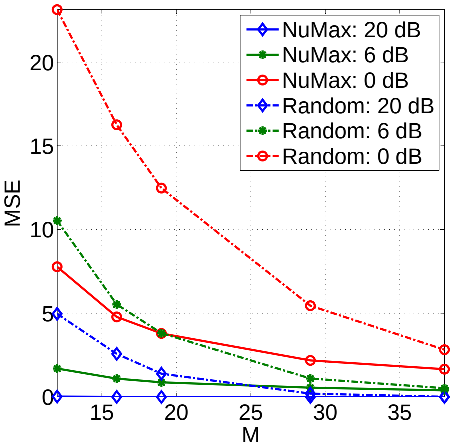


Figure 11: *CS recovery performance for NuMax and random projections. NuMax far outperforms random Gaussian projections in terms of recovered signal MSE for all ranges of measurements M as well as SNR.*

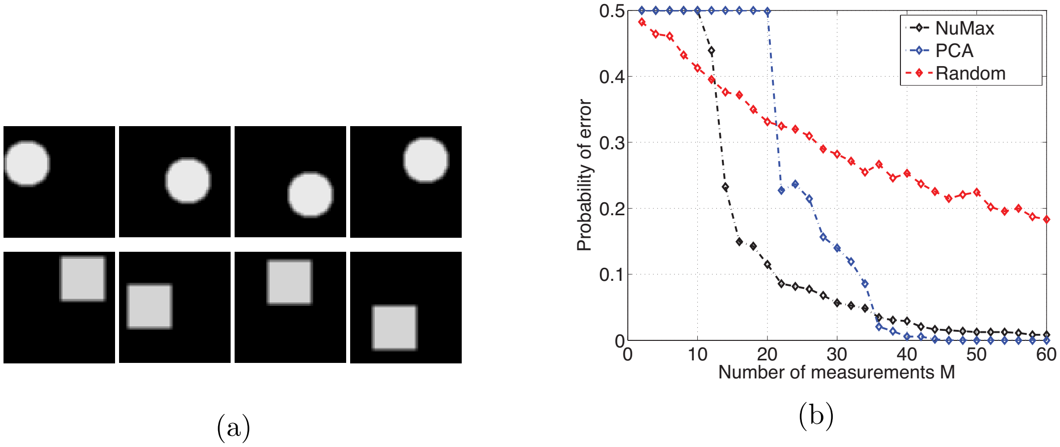


Figure 12: *Binary classification from low-dimensional linear embeddings. (a) The signals of interest comprise shifted images of a white disk/square on a black background. We observe M linear measurements of a test image using different matrices, and classify the observed samples using a GMLC approach. (b) Observed probability of classification error as a function of M. NuMax approach yields high classification rates using very few measurements.*

the probability of error. Again, we observe that NuMax significantly outperforms PCA and random projections.

Table 1: *Misclassification 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		$M = 72$	$M = 97$	$M = 167$
Distortion		$\delta = 0.40$	$\delta = 0.25$	$\delta = 0.1$
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

5.4.2 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. The number of secants (or equivalently, constraints for the SDP) is extraordinarily large, up to $\binom{60000}{2} = 1.6 \times 10^9 = 1.6$ billion secants.

Table 1 shows NN classification performance of NuMax, PCA and Gaussian projections for various lower-dimensional embedding dimensions, corresponding to several values of δ 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 1, 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 now illustrate the improvements in classification performance provided by NuMax-Class (Alg. 3). In particular, we allow for inter-class secants to expand and intra-class secants to shrink without qualifications. As a consequence, in comparison to (8), NuMax-Class optimizes over a (somewhat) larger feasible set. First, we wish to verify if this larger feasibility set indeed translates into a solution of lower rank. Second, we wish to verify if the asymmetric isometry conditions lead to improved classification performance.

We compare the classification performance of NuMax and NuMax-Class in Table 2. For the same value of δ, 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 experiment demonstrates the considerable potential gains using class-specific dimensionality reduction.

Table 2 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 δ, it takes approximately 2 hours to obtain the solution. The runtime increases by a factor of 5× when we decrease

Table 2: *Comparison of the classification rates of NuMax and NuMax-Class over the MNIST dataset (see Table 1 for comparisons with other linear DR techniques). Note that for the value of distortion δ, NuMax-Class provides both lower-rank solution as well as lower misclassification rates. The last two rows provides run-time in hours and the total number of active secants/constraints at the final solution for various values of δ. The number of active secants is a tiny fraction of the total* 1.8 *billion secant set; this demonstrates the scalability of the CG version of the algorithms.*

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

the distortion parameter δ to 0.1. This reflects the general intuition that smaller values of δ result in a larger number of active constraints, which leads to more computationaly intensive sub-problems.

5.4.3 Spoken letter recognition

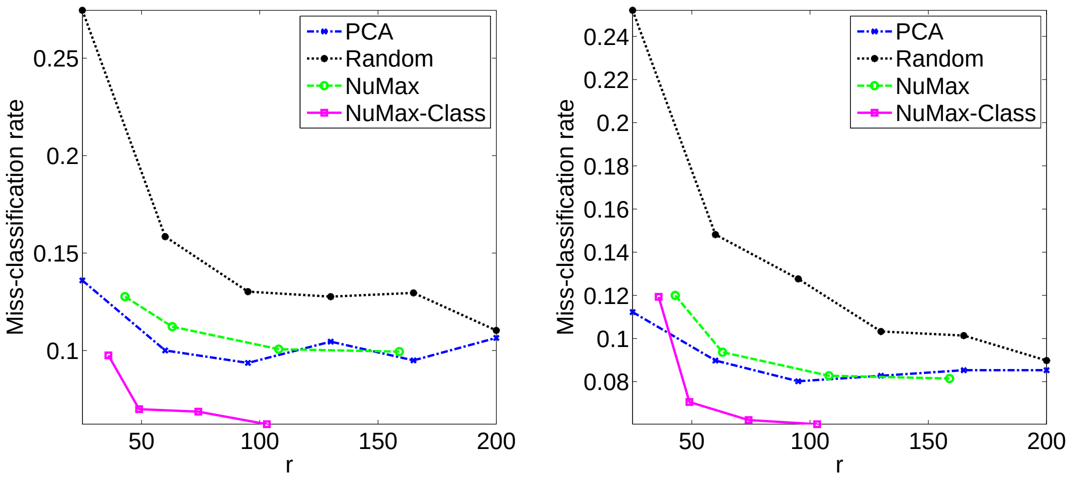
We tested NuMax and its classification variant, NuMax-Class, on the Isolet dataset obtained from the UCI Machine learning repository.[2] This dataset comprises of 26 classes, one for each alphabet in English language. The dataset set consists of 617-dimensional datapoints, with 6238 training points and 1559 test points. In Fig. 13, 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 13(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%.

6. Discussion

In this paper, we have taken some steps towards constructing a comprehensive algorithmic framework that creates a *linear*, *isometry-preserving* embedding of high-dimensional datasets. 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 two algorithms, NuMax and NuMax-CG, that efficiently construct the desired embedding with considerably smaller computational complexity than existing ap-

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



(a) Cross-validation

(b) Testing

Figure 13: *Performance of NuMax, its classification variant, PCA and Random projections on the ISOLET dataset.*

proaches. Our NuMax methods can be easily adapted to perform more complicated machine learning tasks, such as approximate nearest neighbors (ANN) as well as supervised classification. In addition, the NuMax embeddings can be successfully used in compressive sensing applications when the signals of interest can be modeled as elements lying on a smooth, low-dimensional manifold.

The problem of constructing linear, information-preserving embeddings of high-dimensional signals into a low-dimensional space is of central importance in a wide range of machine learning and signal processing applications. Despite their practical significance, surprisingly little is known about near-isometric linear embeddings beyond the Johnson-Lindenstrauss Lemma (Matousek, 2011). For example, it is unclear, even from a theoretical perspective, how to establish the dimension M of the smallest possible subspace into which a specific dataset $\mathcal{X}$ can be embedded using a linear mapping. The framework proposed in this paper adopts a *deterministic,* algorithmic approach to answering this important question. While we do not provide a full analytical characterization for our framework, we hope to initiate a line of work that might lead to some interesting theoretical conclusions.

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 we gave shown empirically that the speed of convergence of our proposed algorithms (NuMax and NuMax-CG) is far better than conventional methods, the analysis of our algorithms from a theoretical perspective remains a challenging task. Finally, from a practical perspective, it is common today in machine learning to encounter datasets that involve millions (or even billions) of training signals, and optimization on such datasets is only feasible when performed in a highly parallel, decentralized, and distributed fashion. How, then, should we extend our proposed algorithms to such scenarios? We defer such important challenges to future research.

Acknowledgments

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. A preliminary conference version of this paper appeared in the Proceedings of the Statistical Signal Processing Workshop (SSP) in August 2012 (Hegde et al., 2012).

References

D. Achlioptas. Database-friendly random projections. In *Proc. Symp. Principles of Database Systems (PODS)*, Santa Barbara, CA, May 2001.

F. Alizadeh. Interior-point methods in semidefinite programming with applications to combinatorial optimization. *SIAM J. Optimization*, 5(01), 1995.

N. Alon. Problems and results in extremal combinatorics. *Discrete Math.*, 273(1):31–53, 2003.

S. Arya, D. Mount, N. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, 1998.

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.

R. Baraniuk and M. Wakin. Random projections of smooth manifolds. *Found. Comput. Math.*, 9(1):51–77, 2009.

R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. A simple proof of the restricted isometry property for random matrices. *Const. Approx.*, 28(3):253–263, 2008.

A. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete and Comput. Geometry*, 13(1):189–202, 1995.

M. Belkin and P. Niyogi. Semi-supervised learning on Riemannian manifolds. *Machine Learning*, 56:209–239, 2004.

J. Blitzer, K. Q. Weinberger, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Proc. Adv. in Neural Processing Systems (NIPS)*, pages 1473–1480, 2005.

D. Boley. Linear convergence of ADMM on a model problem. Technical report, University of Minnesota, Department of Computer Science and Engineering, TR 12-009, 2012.

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

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

D. Broomhead and M. Kirby. Dimensionality reduction using secant-based projection methods: The induced dynamics in projected systems. *Nonlinear Dynamics*, 41(1):47–67, 2005.

E. Candès. Compressive sampling. In *Proc. Int. Congress of Math.*, Madrid, Spain, Aug. 2006.

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

E. Candès, T. Strohmer, and V. Voroninski. PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming. *Comm. Pure and Appl. Math.*, 66(8):1241–1274, Aug. 2013.

K. Clarkson. Tighter bounds for random projections of manifolds. In *Proc. Symp. Comp. Geom.*, pages 39–48. ACM, 2008.

T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Trans. Inform. Theory*, 13(1):21–27, 1967.

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

G. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Math. Operations Research*, 8(1):101–111, 1960.

M. Davenport, M. Duarte, M. Wakin, J. Laska, D. Takhar, K. Kelly, and R. Baraniuk. The smashed filter for compressive classification and target recognition. In *Proc. IS&T/SPIE Symp. Elec. Imag.: Comp. Imag.*, San Jose, CA, Jan. 2007.

W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. Technical report, Rice CAAM 12-14, 2012.

W. Deng, M.-J. Lai, and W. Yin. On the $o(1/k)$ convergence and parallelization of the alternating direction method of multipliers. Technical report, UCLA CAM 13-64, 2013.

D. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, 2006.

D. Donoho and C. Grimes. Hessian Eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proc. Natl. Acad. Sci.*, 100(10):5591–5596, 2003.

R. Dony and S. Haykin. Optimally adaptive transform coding. *IEEE Trans. Image Processing*, 4(10):1358–1370, 1995.

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

C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.

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

M. Fazel. *Matrix Rank Minimization With Applications.* PhD thesis, Stanford Univ., 2002.

R. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Human Genetics*, 7(2):179–188, 1936.

A. Globerson and S. T. Roweis. Metric learning by collapsing classes. In *Proc. Adv. in Neural Processing Systems (NIPS)*, pages 451–458, 2005.

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.

N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011.

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

B. He and X. Yuan. On the $o(1/n)$ convergence rate of the douglas-rachford alternating direction method. *SIAM J. Num. Anal.*, 50(2):700–709, 2012.

X. He and P. Niyogi. Locality preserving projections. In *Proc. Adv. in Neural Processing Systems (NIPS)*, Vancouver, BC, Dec. 2010.

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.

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

P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. ACM Symp. Theory of Comput.*, pages 604–613, New York, NY, 1998.

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

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.

B. Kulis. Metric learning: a survey. *Found. and Trends in Machine Learning*, 5(4):287–364, 2012.

B. Kulis, A. Surendran, and J. Platt. Fast low-rank semidefinite programming for embedding and clustering. In *Proc. Int. AISTATS Conf.*, 2007.

J. Laska, S. Kirolos, M. Duarte, T. Ragheb, R. Baraniuk, and Y. Massoud. Theory and implementation of an analog-to-information conversion using random demodulation. In *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS)*, New Orleans, LA, May 2007.

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

N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

D. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Prog.*, 45(1):503–528, 1989.

S. Ma, D. Goldfarb, and L. Chen. Fixed point and bregman iterative methods for matrix rank minimization. *Math. Prog.*, 128(1):321–353, 2011.

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

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.*, 31(137):148–162, 1977.

S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel pca and de-noising in feature spaces. In *Proc. Adv. in Neural Processing Systems (NIPS)*, 1998.

B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, 26(1):17–32, 1981.

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*, 23(2): 339–358, 1998.

A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, 2001.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Mag.*, 2(11):559–572, 1901.

I. Polik. Sedumi 1.3, 2010. Available online at `http://sedumi.ie.lehigh.edu`.

B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–500, 2010.

S. Roweis and L. Saul. Nonlinear dimensionality reduction by local linear embedding. *Science*, 290:2323–2326, 2000.

B. Russell, A. Torralba, K. Murphy, and W. Freeman. LabelMe: A database and web-based tool for image annotation. *Int. J. Comput. Vision*, 77(1):157–173, 2008.

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.

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

B. Shaw and T. Jebara. Minimum volume embedding. In *Intl. Conf. Artificial Intell. Stat.*, pages 460–467, 2007.

J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

M. Tipping and C. Bishop. Probabilistic principal component analysis. *J. Royal Statist. Soc B*, 61(3):611–622, 1999.

J. Tropp and A. Gilbert. Signal recovery from partial information via orthogonal matching pursuit. *IEEE Trans. Inform. Theory*, 53(12):4655–4666, 2007.

R. Tütüncü, K. Toh, and M. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Math. Prog.*, 95(2):189–217, 2003.

M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk. An architecture for compressive imaging. In *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Atlanta, GA, Oct. 2006.

K. Weinberger and L. Saul. Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comput. Vision*, 70(1):77–90, 2006.

Z. Wen. *First-order Methods for Semidefinite Programming.* PhD thesis, Columbia University, 2009.

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)*, pages 505–512, 2002.

L. Yang and R. Jin. Distance metric learning: A comprehensive survey. *Michigan State Universiy*, pages 1–51, 2006.