Isometry pursuit

Anonymous Author(s)
Affiliation
Address
email

Abstract

Isometry pursuit is an algorithm for identifying unitary column-submatrices of wide matrices. It achieves sparsity via use of the group lasso penalty. Applied to Jacobians of putative coordinate functions, it helps identity isometric embeddings from within dictionaries. It therefore has relevance to interpretability of learned representations.

1 Introduction

Many real-world problems may be abstracted as selecting a subset of the columns of a matrix representing stochastic observations or analytically exact data. This paper focuses on a simple such problem that appears in interpretable learning. Given a rank D matrix $\mathcal{X} \in \mathbb{R}^{D \times P}$ with $P > D$, select a square submatrix $\mathcal{X}_{.\mathcal{S}}$ where subset $\mathcal{S} \subset P$ satisfies $|\mathcal{S}| = D$ that is as isometric as possible.

This problem arises in interpretable learning because while the coordinate functions of a learned latent space may have no intrinsic meaning, it is sometimes possible to generate a dictionary of interpretable features which may be considered as potential parametrizing coordinates. When this is the case, selection of candidate interpretable features as coordinates data representation may take the above form. While implementations vary across data and algorithmic domains, identification of such coordinates generally aids mechanistic understanding, generative control, and statistical efficiency.

In this paper we show that an adapted version of the group lasso algorithm in **?** leads to a convex procedure competitive with greedy approaches such as those found in **???** for finding isometries. The insight that leads to isometry pursuit is that D function solutions multitask basis pursuit applied to an appropriately normalized $\mathcal{X}$ selects unitary submatrices. In particular, this normalization log-symmetrizes length in the column-space of $\mathcal{X}$ and favors vectors closer to unit length. The main advantage of this basis pursuit formulation is that it is convex and therefore computationally expedient. We exhibit the effectiveness of this approach experimentally in lieu of theoretical results.

2 Background

In this section we give background on both the techniques used in our method, and also the motivating applications on which it is applied.

2.1 Problem

Our goal is, given a matrix $\mathcal{X} \in \mathbb{R}^{D \times P}$, select a subset $\mathcal{S} \subset [P]$ with $|\mathcal{S}| = D_-$ and $D_- \leq D$ such that $X_{.\mathcal{S}}$ is as unitary as possible in a computationally efficient way. To that end, define a ground truth loss function that measures unitariness, and then introduce a surrogate loss function that convexifies the problem so that it may be efficiently solved.

Submitted to Interpretable AI: Past, Present and Future Workshop @ NeurIPS 2024

2.2 Techniques

2.2.1 Unitary matrices

Definition 1 (Unitary) *A rank D matrix $U \in \mathbb{R}^{D_\alpha \times D_\beta}$ is said to be **unitary** if $U^T U = I_D$.*

While this definition implies $D_\alpha, D_\beta \geq D$, this paper typically works with equality. Even so, this paper uses the term singular values rather than eigenvalues, and Section **??** contains speculation that a similar set of techniques hold for rank-d unitary rectangular matrices.

The property of unitary matrices that motivates our ground truth loss function comes from spectral analysis.

Proposition 1 (Singular values of a unitary matrix) *The singular values $\sigma_1 \dots \sigma_D$ are equal to 1 if and only if $U \in \mathbb{R}^{D_\alpha \times D_\beta}$ is unitary.*

As a ground truth loss function, we'd like the loss to be minimized uniquely by unitary matrices, invariant under rotation, and depend on all changes in the matrix. The first desired property precludes, for example, the log determinant, while the last precludes the operator norm, also known as the deformation. Finally, to simplify our overall exposition and exemplify the key features of the method, we'd like this ground truth loss to be as similar as possible to the convex loss we introduce. In fact, these losses will behave equivalently over diagonalizable matrices. That is, in a sense, the ground truth loss will be as convex as possible.

The property of unitary matrices that we use to define our alternative convex loss is slightly different from Proposition 3.

Proposition 2 (Basis vectors of a unitary matrix) *The component vectors $u^1 \dots u^D \in \mathbb{R}^B$ form a unitary matrix if and only if, for all $d_1, d_2 \in [D]$, $u_{d_1} u^{d_2} = \begin{cases} 1 & d_1 = d_2 \\ 0 & d_1 \neq d_2 \end{cases}$.*

We will show that the conditions of the antecedent of Proposition 2 are satisfied by the solutions to the multitask basis pursuit problem applied to matrices consisting of suitably normalized vectors. The vectors are in particular length normalized so that the post-normalization vectors with the longest length have length 1. As we show in Proposition , our main theoretical result, the tendency of the multitask basis pursuit problem to select orthogonal features then ensures that a unitary matrix is deterministically selected, given that one is present.

2.2.2 Basis pursuit and lasso

We use "multitask" instead of "group" to refer to this algorithm, as grouping is entirely determined by the dependent variable.

Basis pursuit programs are of the form

$$\min c(\beta) : d(y, x\beta) \leq o \tag{1}$$

Basis pursuit programs are convex, and solvable via standard interior point methods.

On the other hand, the basis pursuit program admits

2.2.3 Isometries

Another motivating example of this paper is the embedding of non-linear data manifolds by selecting functions from within a dictionary that preserve distances and angles. These sets of functions are called isometries if they do so globally and local isometries if they do so almost everywhere. If more pointwise specificity is required, they are known at isometric at a point. To avoid abstractness, we give the following definition in coordinates.

Definition 2 *Isometric at a point A map ϕ between D dimensional submanifolds with inherited Euclidean metric $\mathcal{M} \subseteq R^{B_\alpha}$ and $\mathcal{N} \subseteq R^{B_\beta}$ is an **isometry at a point** $p \in \mathcal{M}$ if*

$$\|v^T D\phi(p)\| = \|v\| \forall v \in \mathbb{R}^D \tag{2}$$

$D\phi$ is the differential implicitly given by $D_F^E \phi = U D\phi V^T$ where U and V are bases for $(T_p\mathcal{M})$ and $T_{\phi(p)}\mathcal{N}$, the tangent spaces of $\mathbb{R}$ and $\mathcal{N}$ at p and $\phi(p)$, respectively.

The reasons that pointwise isometry is interesting are themselves manifold. **?** showed how pointwise isometries selected from a dictionary may be stitched together to form global embeddings. Another line of work **??** has shown how independent functions may be selected globally using group lasso, and that this selection process has implications for the metric properties of the selected embedding **?**. Other approaches like Local Tangent Space Alignment seek to identify isometric coordinates non-parametrically through tangent space estimation prior to stitching, while multidimensional scaling and isomap do so directly through preservation of distances. A classic result in differential geometry equilibrates these two approaches.

Given the metric characterization of isometry in Definition 2, is it appropriate to recall another equivalent characterization of unitary matrices.

Proposition 3 (Metric properties of a unitary matrix) *$U \in \mathbb{R}^{D_\alpha \times D_\beta}$ is unitary if and only if $\|v^T U$ for all $v \in \mathbb{R}^{D_\beta}$.*

Thus, in coordinates, $D\phi(p)$ is unitary if ϕ is isometric at a point p.

X could be, for example, the Jacobian matrix dg of a set of candidate coordinate functions $g = [g^1, \dots g^P]$.

2.3 Interpretable manifold learning

While variable selection in unsupervised learning is comparably less studied than in supervised learning, substantial literature exists. One method that exemplifies this area is Sparse PCA **?**, in which a subset of variables are used to generate low-dimensional projections. Within non-linear dimension reduction dictionaries can be either given **??** or learned **?**. In order of specificity, these methods may seek to optimize independent coordinates **??**, low distortion embeddings, or isometric embeddings. Optimization can be global or local. These coordinate selection algorithms can be greedy or convex **??**.

The motivating example for this research is the computation of embeddings by selection from within sets of embedding coordinates. However, as the general approach has usefulness outside of non-linear dimension reduction, we first give examples in the relatively simpler setting of diversification.

3 Method

Recall that our objective is to, given a rank D matrix $\mathcal{X} \in \mathbb{R}^{D \times P}$ with $P > D$, select a square submatrix $\mathcal{X}_{.\mathcal{S}}$ where subset $\mathcal{S} \subset P$ satisfies $|\mathcal{S}| = D$ that is as unitary as possible. Thus, we first will define a function that is uniquely minimized by unitary matrices and some favorable properties for optimization that will be the ground truth we evaluate the success of our method against. We then define the combination of normalization and multitask basis pursuit that approximates this ground truth loss function. We finally define a post processing method for ensuring that the solution is D sparse. In lieu of theoretical results, we give experimental evidence of the efficacy of this approach in in Section **??**

3.1 Ground truth

The main goal of isometry pursuit is to expediate the selection of unitary submatrices. More traditional measures of unitariness which use the singular values of a matrix like the log operator norm (i.e. log deformation) and nuclear norm are poorly suited for optimization since they use a subset of the matrix's information and are not uniquely minimized at unitarity, respectively. Thus, we define the loss

$$l_c : \mathbb{R}^{D \times P} \to \mathbb{R}^+ \tag{3}$$

$$\mathcal{X} \mapsto \sum_{d=1}^{D} g(\sigma^d(\mathcal{X}), c) \tag{4}$$

where $\sigma^d((X))$ is the d-th singular value of $\mathcal{X}$ and

$$g : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+ \tag{5}$$

$$t, c \mapsto e^{t^c} + e^{t^{-c}}. \tag{6}$$

Plainly, g is uniquely maximized by unitary matrices, and $g(\mathcal{X}^\dagger) = g(\mathcal{X})^{-1}$. The former condition is necessary for success of the method, while the latter, as well as the convexity of g, are somewhat aesthetic choices. A graph of g is given in Figure **??**. Most importantly, this loss enables comparison with produced after normalization as in Section 3.2.

The overall algorithm we seek to improve upon is

$$\widehat{\mathcal{S}}_{GT} = \arg \min_{\mathcal{S} \subseteq [P] : |\mathcal{S}| = D} l_c(\mathcal{X}_{.\mathcal{S}}) \tag{7}$$

In practice, non-convexity occurs in two places, but only one is essential. The inessential non-convexity is in the computation of l_c. While this function is in fact convex, computation of the individual singular values prior to summation is not, and our experiments rely on such piecemeal computation rather than implementing an end-to-end method. However, the combinatorial search over $[P]$ is inherently non-convex, and requires combinatorial search over all combinations.

3.2 Normalization

We will propose a basis pursuit method which approximates the results of Program 7. Since basis pursuit methods tend to select longer vectors, selection of unitary submatrices requires normalization such that long and short candidate basis vectors are penalized in the subsequent regression. This calls for a "normalization" method that differs from other forms in its requirements, and we can't yet prove that these conditions relate it to any sort of norm, even on an appropriately chosen space. This normalization is Now establish some basic conditions for normalization of vectors $v \in \mathbb{R}^D$.

Definition 3 (Symmetric normalization) *A function $q : \mathbb{R}^D \to \mathbb{R}^+$ is a symmetric normalization if*

$$\arg \max_{v \in \mathbb{R}^D} q(v) = \{v : \|v\| = 1\} \tag{8}$$

$$q(v) = q(\frac{v}{\|v\|^2}) \tag{9}$$

$$q(v^1) = q(v^2) \; \forall \; v^1, v^2 : \|v^1\| = \|v^2\| \tag{10}$$

Note that requiring the full structure of a multiplicative norm here is unnecessary for basic success of the algorithm, but certain characteristics such as $q(v^{-1}) = q(v)$ seem desirable, provided one can give a reasonable way to compute v^{-1}, such as by considering each vector as a scaled rotation subgroup of the general linear group. Mindful of this opportunity, and also of the desire to compare with the ground truth and provide computational expediency, consider the normalization by

$$q : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+ \tag{11}$$

$$t, c \mapsto \frac{e^{t^c} + e^{t^{-c}}}{2e}, \tag{12}$$

and use this to define the vector normalization

$$n : \mathbb{R}^D \times \mathbb{R}^+ \to \mathbb{R}^D \tag{13}$$

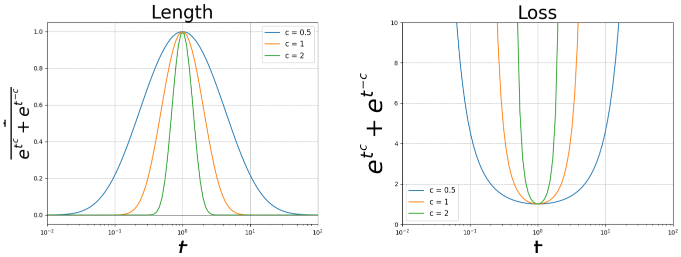
$$n, c \mapsto \frac{n}{q(\|n\|_2, c)} \tag{14}$$

and matrix normalization

$$w : \mathbb{R}^{D \times P} \times \mathbb{R}^+ \to \mathbb{R}^D \tag{15}$$

$$\mathcal{X}_{.p}, c \mapsto n(\mathcal{X}_{.p}, c) \; \forall \; p \in [P]. \tag{16}$$

While this normalization satisfies **??**, it also has some additional nice properties. First, q is convex. Second, it grows asymptotically log-linearly. Third, while $\exp(-|\log t|) = \exp(-\max(t, 1/t))$ is a seemingly natural choice for normalization, it is non smooth, and the LogSumExp replacement of $\max(t, 1/t)$ with $\log(\exp(t) + \exp(1/t))$ simplifies to 11 upon exponentiation. Finally, the parameter c grants control over the width of the basin, which is important in avoiding numerical issues arising close to 0 and ∞. This completes the deterministic data preprocessing.



(a) Ground truth loss scaling function g as a function of t.

(b) Length as a function of t

Figure 1: Plots of Length and Loss for different values of c. Since t is one dimensional and therefore diagonalizable, basis pursuit and ground truth give identical loss values.

3.3 Isometry pursuit

We will show how to use an appropriate normalized matrix $w(\mathcal{X})$ in multitask basis pursuit to identify submatrices of $\mathcal{X}$ that are as unitary as possible. Multitask basis pursuit is a method for identifying sparse signals from overcomplete dictionaries, and the intuition behind its application in our setting is that submatrices consisting of vectors which are closer to 1 in length and more orthogonal will have smaller loss. In contrast to the typical statistical setting, these features correspond to individual observations in our diversification example, and basis vectors of data manifold tangent spaces in our non-linear dimension reduction example.

Define the multitask basis pursuit penalty

$$\|\cdot\|_{1,2} : \mathbb{R}^{P \times D} \to \mathbb{R}^{+} \tag{17}$$

$$\beta \mapsto \sum_{p=1}^{P} \|\beta_{p.}\|_2. \tag{18}$$

The isometry pursuit program is then

$$\widehat{\beta}_c^D(\mathcal{X}) := \arg \min_{\beta \in \mathbb{R}^{P \times D}} \|\beta\|_{1,2} \;:\; I_D = w(\mathcal{X}, c)\beta. \tag{19}$$

The recovered functions are the indices of the dictionary elements with non-zero coefficients. That is, they are given by $S(\beta)$ where

$$S : \mathbb{R}^{p \times d} \to \binom{[P]}{d} \tag{20}$$

$$\beta \mapsto \{p \in [P] : \|\beta_{p.}\| > 0\} \tag{21}$$

and $\binom{[P]}{d} = \{A \subseteq [P] : |A| = d\}$.

```
IsometryPursuit(Matrix $\mathcal{X} \in \mathbb{R}^{D \times P}$, scaling constant $c$)
Output $\widehat{S} = S(\widehat{\beta}_P(w_c(\mathcal{X}))$
```

A key theoretical assertion for the feasibility of IsometryPursuit is that it is invariant to choice of basis for $\mathcal{X}$.

Proposition 4 (Basis pursuit selection invariance) *Let $U \in \mathbb{R}^{D \times D}$ be unitary. Then $S(\widehat{\beta}(U\mathcal{X})) = S(\widehat{\beta}(\mathcal{X}))$.*

A proof is given in Section **??** This fact has as an immediate corollary that we may replace I_D in the constraint by any unitary $D \times D$ matrix.

3.4 Two-stage isometry pursuit

A standard approach in the lasso literature is to first use the lasso to prune, prior to a final feature selection step **?** This avoids issues from shrinkage caused by the lasso estimator at high values of λ. For example, we cannot as of yet prove analogs of Proposition **??** for the Lasso formulation, and similar lasso-specific conditions such as those found in **?** are not satisfied by overcomplete dictionaries.

3.5 Implementation

We use the cvxpy package for basis pursuit. We use the SCS interior point solver from CVXPY, which is able to push sparse values arbitrarily close to 0 **?**. Data is IRIS and Wine, as well as flat torus from ldle.

3.6 Computational complexity