Isometry pursuit

Samson Koelle, Marina Meila

July 25, 2024

Abstract

Isometry pursuit is an algorithm for identifying unitary column-submatrices of wide matrices in polynomial time. It achieves sparsity via use of the group lasso norm, and therefore has constrained and penalized formulations. Applied to tabular data, it selects a subset of columns that maximize diversity. Applied to Jacobians of putative coordinate functions, it identifies isometric embeddings from within dictionaries. It therefore has relevance to interpretability of learned representations.

1 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 include claims that ground truth and convex loss values are the same for all diagonalizable matrices, and that the convex basis pursuit program recovers to optimum in a deterministic manner should it exist; proofs are given in Section 2.1. We finally define the lasso dual to the basis pursuit program and a post processing method for ensuring that the solution is D sparse. Experimental results using these methods will then be given in Section **??**

1.1 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

$$l_c : \mathbb{R}^{D \times P} \to \mathbb{R}^+ \tag{1}$$

$$\mathcal{X} \mapsto \sum_{d=1}^{D} g(\sigma^d(\mathcal{X}), c) \tag{2}$$

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

$$g : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+ \tag{3}$$

$$t, c \mapsto e^{t^c} + e^{t^{-c}}. \tag{4}$$

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 1. Most importantly, this loss enables comparison with produced after normalization as in Section 1.2.

The overall algorithm we seek to improve upon is

$$\widehat{\mathcal{S}}_{GT} = \arg \min_{\mathcal{S} \subseteq [P] : |\mathcal{S}| = D} l_c(\mathcal{X}_{.\mathcal{S}}) \tag{5}$$

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.

1.2 Normalization

We will propose a basis pursuit method which approximates the results of Program 5. 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 1 (Symmetric normalization) *A function $q : \mathbb{R}^D \to \mathbb{R}^+$ is a symmetric normalization if*

$$\arg\max_{v \in \mathbb{R}^D} q(v) = \{v : \|v\| = 1\} \tag{6}$$

$$q(v) = q(\frac{v}{\|v\|^2}) \tag{7}$$

$$q(v^1) = q(v^2) \; \forall \, v^1, v^2 : \|v^1\| = \|v^2\| \tag{8}$$

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

$$q : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+ \tag{9}$$

$$t, c \mapsto \frac{e^{t^c} + e^{t^{-c}}}{2e}, \tag{10}$$

and use this to define the vector normalization

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

$$n, c \mapsto \frac{n}{q(\|n\|_2, c)} \tag{12}$$

and matrix normalization

$$w : \mathbb{R}^{D \times P} \times \mathbb{R}^+ \to \mathbb{R}^D \tag{13}$$
$$\mathcal{X}_{.p}, c \mapsto n(\mathcal{X}_{.p}, c) \ \forall \ p \in [P]. \tag{14}$$

While this normalization satisfies 1, 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 9 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.

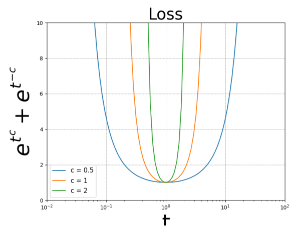

Figure 1: Ground truth loss scaling function g as a function of t

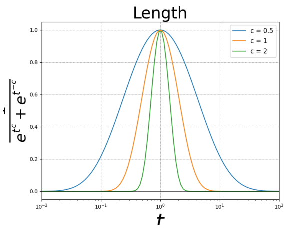

Figure 2: Length as a function of t

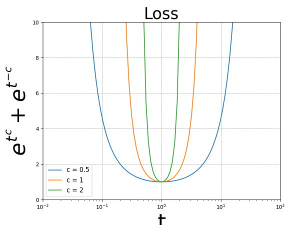

Figure 3: Basis pursuit losses as a function of t

Figure 4: 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.

1.3 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

$$\| \cdot \|_{1,2} : \mathbb{R}^{P \times D} \to \mathbb{R}^+ \tag{15}$$
$$\beta \mapsto \sum_{p=1}^{P} \|\beta_{p.}\|_2. \tag{16}$$

The isometry pursuit program is then

$$\widehat{\beta}_c^P(\mathcal{X}) := \arg \min_{\beta \in \mathbb{R}^{P \times D}} \|\beta\|_{1,2} \ : \ I_D = w(\mathcal{X}, c)\beta. \tag{17}$$

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

$$S : \mathbb{R}^{p \times d} \to \binom{[P]}{d} \tag{18}$$

$$\beta \mapsto \{p \in [P] : \|\beta_{p.}\| > 0\} \tag{19}$$

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

```
IsometryPursuit(Matrix $\mathcal{X} \in \mathbb{R}^{D \times P}$, scaling constant $c$)
1: 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 1 (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 2.1.1 This fact has as an immediate corollary that we may replace I_D in the constraint by any unitary $D \times D$ matrix. With these preliminaries, we may state our main result.

Proposition 2 (Unitary selection) *Given a matrix $\mathcal{X} \in \mathbb{R}^{D \times P}$ with a rank D submatrix $\mathcal{X}_{.\mathcal{S}} \in \mathbb{R}^{D \times D}$ that is unitary, $\mathcal{S} = S(\widehat{\beta}(\mathcal{X})))$*

This proof admits two immediate generalizations. First, any normalization function that satisfies the normalization conditions will do. Second, assuming that we do in fact use w for normalization, the ground truth and convex losses are equivalent for diagonalizable matrices. This is summarized in the following proposition, which is slightly stronger than Proposition

1.4 Isometric lasso

We defer discussion of computational complexity to Section **??**.

The convex loss function 15 and linear constraint in 17 admit a Lagrangian dual which we shall call Isometric Lasso. The Isometric Lasso loss is

$$l_\lambda(\mathcal{X}, \beta) = \|I_D - \tilde{\mathcal{X}}_c \beta\|_2^2 + \lambda \|\beta\|_{1,2} \tag{20}$$

which can be optimized as

$$\hat{\beta}_\lambda(\mathcal{X}) = \arg \min_{\beta \in \mathbb{R}^{P \times D}} l_\lambda(\mathcal{X}, \beta) \tag{21}$$

Similarly to Section **??**, we assert that $S(\widehat{\beta}_\lambda(\mathcal{X}))$.

Proposition 3 (Lasso selection equivalence) *Let $U \in \mathbb{R}^{D \times D}$ be unitary. Then $S(\widehat{\beta}_\lambda(U\mathcal{X})) = S(\widehat{\beta}_\lambda(\mathcal{X}))$.*

1.5 Extension to non-linear spaces

Proposition 4 (Local isometry selection) *Given a set of functions G that contains a subset that defines a locally isometric embedding at a point ξ, then these will be selected as $\arg\min_\beta$.*

A proof is given in Section 2.1.3.

1.6 Two-stage isometry pursuit

1.7 Implementation

We use the multitask lasso from sklearn and 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 **cvxpy˙sparse˙solution**. Data is IRIS and Wine, as well as flat torus from ldle.

1.8 Computational complexity

2 Supplement

2.1 Proofs

2.1.1 Proof of Proposition ??

Proposition 5 *Let $U \in \mathbb{R}^{D \times D}$ be unitary. Then $\hat{\beta}_\lambda(U\mathcal{X}) = \hat{\beta}_\lambda(U\mathcal{X})$.*

Proposition 6 *Loss equivalence Let $U \in \mathbb{R}^{D \times D}$ be unitary. Then $\|\beta\|_{1,2} = \|\beta U\|$.*

2.1.2 Proof of Propositions 3 and ??

These proofs rely on some elementary applications of linear algebra. Proposition 3 relies on the fact that its loss is invariant under any unitary transformation. As a corollary, this fact gives that the identify matrix which is the "dependent variable" in the regression equation may be replaced by any $d \times d$ unitary matrix. For Proposition **??**, the loss is also invariant under unitary, transformation, but we also check that this transformation. Once again, this also implies that any unitary matrix may replace the identity in the constraint.

Proposition 7 *Loss equivalence Let $U \in \mathbb{R}^{D \times D}$ be unitary. Then $l_\lambda(\mathcal{X}, \beta) = l_\lambda(U\mathcal{X}, \beta U)$.*

Proof: Without loss of generality, let $i = 1$. We can write

$$l^*(X^i) = l(\beta^i) = \sum_{j=1}^{p} (\sum_{i'=2}^{n} \|\beta_{i'j.}\|_2^2 + \|\beta_{1j.}^i\|_2^2)^{1/2} = \sum_{j=1}^{p} (\sum_{i'=1}^{n} \|\beta_{i'j.} U\|_2^2)^{1/2} = l^*(X) \quad (22)$$

where the second to last equality is because the norm $\|v\|_2^2$ is unitary invariant. $\square$

Proposition 8 *Programmatic equivalence Let $U \in \mathbb{R}^{D \times D}$ be unitary. Then $\hat{\beta}_\lambda(U\mathcal{X}) = U\hat{\beta}_\lambda(\mathcal{X})$.*

2.1.3 Proof of Proposition 4

The two main components of this proof are that vectors which are more orthogonal will be smaller in loss.

Proposition 9 *Let $X_{.S} \in \mathbb{R}^{d \times p}$ be defined as above and let $X'_{..S}$ be an array such that $\|X'_{.S_j}\|_2 = \|X_{.S_j}\|_2$ for all $j \in [d]$ and $X'_{.S}$ is column-orthogonal. Then $\tilde{l}^*(X_{..S}) > \tilde{l}^*(X'_{..S})$.*

Proof: By Lemma **??**, without loss of generality

$$\beta^i_{ijk} = \begin{cases} \|\tilde{X}'_{.S_j}\|_2^{-1} & j = k \in \{1 \dots d\} \\ 0 & \text{otherwise} \end{cases}. \tag{23}$$

Therefore,

$$\tilde{l}^*(X') = \sum_{j=1}^{d} \sqrt{\sum_{i=1}^{n} \|\tilde{X}'_{i.S_j}\|_2^{-2}}. \tag{24}$$

On the other hand, the invertible matrices $\tilde{X}_{.S}$ admit QR decompositions $\tilde{X}_{.S} = QR$ where Q and R are square unitary and upper-triangular matrices, respectively **Anderson1992-fb**. Since l^* is invariant to unitary transformations, we can without loss of generality, consider $Q = I_d$. Denoting I_d to be composed of basis vectors $[e^1 \dots e^d]$, the matrix R has form

$$R = \begin{bmatrix} \langle e^1, \tilde{X}_{i.S_1}\rangle & \langle e^1, \tilde{X}_{i.S_2}\rangle & \dots & \langle e^1, \tilde{X}_{i.S_d}\rangle \\ 0 & \langle e^2, \tilde{X}_{i.S_2}\rangle & \dots & \langle e^2, \tilde{X}_{i.S_d}\rangle \\ 0 & 0 & \dots & \dots \\ 0 & 0 & 0 & \langle e^d, \tilde{X}_{i.S_d}\rangle \end{bmatrix}. \tag{25}$$

The diagonal entries $R_{jj} = \langle q^j, \tilde{X}_{.S_j}\rangle$ of this matrix have form $\|\tilde{X}_{.S_j} - \sum_{j' \in \{1 \dots j-1\}} \langle \tilde{X}_{.S_j}, e^{j'}\rangle e^{j'}\|$. Thus, $R_j \in (0, \|\tilde{X}_{i.S_j}\|]$. On the other hand $\beta_{iS.} = R^{-1}$, which has diagonal elements $\beta_j = R_j^{-1}$, since R is upper triangular. Thus, $\beta_{jj} \geq \|\tilde{X}_{.S_j}\|^{-1}$, and therefore $\|\beta_{iS_j.}\| \geq \|\beta'_{S_j.}\|$. Since $\|\beta_{S_j.}\| \geq \|\beta'_{S_j.}\|$ for all i, then $\|\beta_{.S_j.}\| \geq \|\beta'_{.S_j.}\|$. □

The above proposition formalizes our intuition that orthogonality of X lowers $l^*(X)$ over non-orthogonality. We now show a similar result for the somewhat less intuitive heuristic that dictionary functions whose gradient fields are length 1 will be favored over those which are non-constant. Since the result on orthogonality holds regardless of length, we need only consider the case where the component vectors in our sets of vector fields are mutually orthogonal at each data point, but not necessarily of norm 1. Note that were they not orthogonal, making them so would also reduce l^*. We then show that vectors which are closer to length 1 are lower in loss. Since vectors which are closer to length 1 are shrunk in length less by $\exp_1$, their corresponding loadings are smaller. This is formalized in the following proposition

Proposition 10 *Let $X''_{.S}$ be a set of vector fields $X''_{.S_j}$ mutually orthogonal at every data point i, and $\|X''_{.S_j}\| = 1$. Then $\tilde{l}^*(X'_{.S}) \geq \tilde{l}^*(X''_{.S})$.*

Proof: Let $\|X''_{i.S_j}\| = c_j$. By Proposition **??**, we can assume without loss of generality (i.e without changing the loss) that

$$\tilde{X}_{.S_j} = \begin{bmatrix} c_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & c_d\rangle \end{bmatrix}. \tag{26}$$

Thus

$$\tilde{\exp}_1 X_{.S_j} = \begin{bmatrix} \exp(-|\log \|c_1\|_2)|) & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \exp(-|\log \|c_d\|_2)|)\rangle \end{bmatrix}. \tag{27}$$

and therefore

$$\tilde{\beta}_{.S_j} = \begin{bmatrix} \exp(-|\log \|c_1\|_2)|)^{-1} & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \exp(-|\log \|c_d\|_2)|)^{-1}\rangle \end{bmatrix}. \tag{28}$$

The question is therefore what values of c_j minimize $\exp(-|\log\ |c_1\|_2)|)^{-1}$. $|\log\ |c_1\|_2)|$ is minimized (evaluates to 0) when $c_j = 1$, so $-|\log\ |c_1\|_2)|$ is maximized (evaluates to 0, so $\exp(-|\log\ |c_1\|_2)|)$ is maximized (evaluates to 1), so $\exp(-|\log\ |c_1\|_2)|)^{-1}$ is minimized (evaluates to 1). □

For basis pursuit, the situation is similar.