Isometry pursuit

Samson Koelle, Marina Meila

July 24, 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

First, we will define an function that is uniquely minimized by unitary matrices and some favorable properties for optimization. This will be the ground truth we evaluate the success of our method against. We then define the multitask lasso method that approximates the ground truth loss function in the sense that loss values are the same for all diagonalizable matrices. We show that this program recovers to optimum in a deterministic manner should it exist, but that the solution is not necessarily $D-sparse$. Therefore, we will demonstrate the ways in which sparse

1.1 Ground truth

The main goal of sparse isometry pursuit is to expediate the selection of unitary submatrices. Typical measures of unitariness which use the singular values of a matrix like the operator norm (i.e. 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.

Define the loss

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

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

where $\sigma^d((X))$ is the d-th singular value of $\mathcal{X}$. However, this would not result in a sparse solution.

This loss is an appropriate choice for comparison because it is equal to the basis pursuit loss for suitably normalized orthogonal matrices. This will be discussed

1.2 Normalization

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 on an appropriately chosen space. However, this transformation of individual features should be familiar to statisticians as a form of feature engineering, and whether the transformed vectors are related to a norm or not, the concept is similar.

We may 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*

$$\arg\max_{v \in \mathbb{R}^D} q(v) = \{v \; \|v\| = 1\} \tag{3}$$

$$q(v) = q(\frac{v}{\|v\|^2}) \tag{4}$$

$$q(v^1) = q(v^2) \; \forall v^1, v^2 : \|v^1\| = \|v^2\| \tag{5}$$

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

$$q : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+ \tag{6}$$

$$t, c \mapsto \frac{\exp(t^c) + \exp(t^{-c})}{2e}, \tag{7}$$

and use this to define the normalization

$$n : \mathbb{R}^D \times \mathbb{R}^+ \to \mathbb{R}^D \tag{8}$$

$$n^d, c \mapsto \frac{n^d}{q(\|n\|_2, c)} \forall d \in [D]. \tag{9}$$

While this normalization satisfies 1, it also has some additional nice properties. First, q is convex and smooth. 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 6 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 ∞.

Using this, define the matrix-wide normalization vector

$$\mathcal{D} : \mathbb{R}^{D \times P} \times \mathbb{R}^+ \to \mathbb{R}^P \tag{10}$$

$$\mathcal{X}_{.p}, c \mapsto n(\mathcal{X}_{.p}, c) \tag{11}$$

and the normalized matrix $\tilde{\mathcal{X}}_c = \mathcal{X}\mathcal{D}(\mathcal{X}, c)$. This completes the data preprocessing.

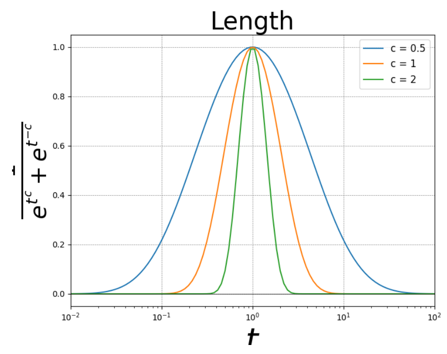

Figure 1: Length as a function of t

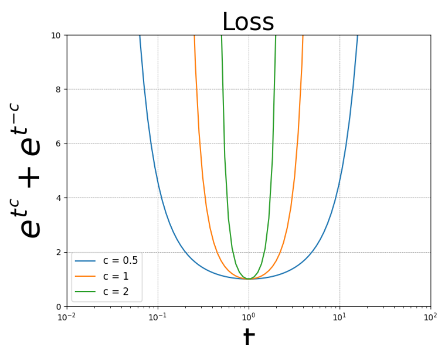

Figure 2: Loss as a function of t

Figure 3: 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

Define the multitask group basis pursuit penalty

$$\|\cdot\|_{1,2} : \mathbb{R}^{P\times D} \to \mathbb{R}^+ \tag{12}$$

$$\beta \mapsto \sum_{p=1}^{P} \|\beta_{p.}\|_2. \tag{13}$$

The isometry pursuit program is then

$$\hat{\beta_P}(\mathcal{X}) = \arg\min_{\beta\in\mathbb{R}^{P\times D}} \|\beta\|_{1,2} \; s.t. \; I_D = \tilde{\mathcal{X}}_c\beta. \tag{14}$$

The intuition is that vectors which are closer to 1 in length and more orthogonal will be smaller in loss.

1.4 Isometric lasso

The convex loss function 12 and linear constraint in 14 admit a Lagrangian dual which we shall call Isometric Lasso. The Isometric Lasso loss is

$$l_\lambda(\mathcal{X},\beta) = \|I_D - \tilde{\mathcal{X}}_c\beta\|_2^2 + \lambda\|\beta\|_{1,2} \tag{15}$$

which can be optimized as

$$\hat{\beta_\lambda}(\mathcal{X}) = \arg\min_{\beta\in\mathbb{R}^{P\times D}} l_\lambda(\mathcal{X},\beta) \tag{16}$$

The recovered supports are then given by $S(\hat{\beta_\lambda}(\mathcal{X}))$ where

$$S : \mathbb{R}^{p\times d} \to \binom{\{1,2,\ldots,P\}}{d} \tag{17}$$

$$\beta \mapsto \{p \in \{1,2,\ldots,P\} : \|\beta_{p.}\| > 0\} \tag{18}$$

and $\binom{\{1,2,\dots,P\}}{d} = \{A \subseteq \{1,2,\dots,P\} : |A| = d\}$ are the indices of the dictionary elements with non-zero coefficients.

1.5 Theory

A key theoretical assertion is that selection methods $S(\widehat{\beta}_\lambda(\mathcal{X}))$ and $S(\widehat{\beta}(\mathcal{X}))$ are invariant to choice of basis for $\mathcal{X}$.

Proposition 1 (Basis pursuit selection equivalence) *Let $U \in \mathbb{R}^{D \times D}$ be unitary. Then $S(\widehat{\beta}(U\mathcal{X})) = S(\widehat{\beta}(\mathcal{X}))$.*

Proposition 2 (Lasso selection equivalence) *Let $U \in \mathbb{R}^{D \times D}$ be unitary. Then $S(\widehat{\beta}_\lambda(U\mathcal{X})) = S(\widehat{\beta}_\lambda(\mathcal{X}))$.*

With these preliminaries, we may state our main result.

Proposition 3 (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, $\mathcal{S} = S(\widehat{\beta}(\mathcal{X})))$*

A proof is given in Section

This proof admits two immediate generalizations. First, any normalization function that satisfies the normalization conditions will do. Second, the ground truth and convex losses are equivalent for diagonalizable matrices.

Proposition 4

Proof:

$$\tag{19}$$

Then, singular values and regressands are analytically determined. cont. $\square$

Proposition 5 (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 **??**.

Algorithm (Local tangent Space basis pursuit)

Algorithm (Local two stage tangent space basis pursuit)

This provides an approach for the problem put forward in (cite) LDLE paper.

Experiments (Loss)

Compare with isometry loss (2 norm of singular values).

1.6 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.

1.7 Computational complexity

2 Experiments

Comparison with isometry loss.