Isometry pursuit

Samson Koelle, Marina Meila

July 26, 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 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 unsupervised learning. Given a rank D matrix $\mathcal{X} \in \mathbb{R}^{D \times P}$ with $P > D$, select a square submatrix $\mathcal{X}_{\cdot\mathcal{S}}$ where subset $\mathcal{S} \subset P$ satisfies $|\mathcal{S}| = D$ that is as unitary as possible.

This problem is motivated by applications in diversification and non-linear dimension reduction. In particular, the name of the method comes from the fact that isometric embeddings have unitary differentials. 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 **Dey2017-mx**, in which a subset of variables are used to generate low-dimensional projections. Within non-linear dimension reduction dictionaries can be either given **Koelle2022-ju**; **Koelle2024-no** or learned **Kohli2021-lr**. In order of specificity, these methods may seek to optimize independent coordinates **Chen2019-km**; **He2023-ch**, low distortion embeddings, or isometric embeddings. Optimization can be global or local. These coordinate selection algorithms can be greedy **NEURIPS2019˙6a10bbd4**; **Kohli2021-lr**; **Jones2007-uc** or convex **Koelle2022-ju**; **Koelle2024-no**.

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. This normalization is log-symmetric length in the column-space of $\mathcal{X}$ and favors vectors closer to unit length. This property is the focus of Section **??**

The basis pursuit formulation is desirable for several reasons. First, it is convex and therefore computationally expedient. Second, while not D-sparse, it is relatively sparse, and so can be used for pruning. Third, it admits a lasso dual problem which is particularly useful in high dimensions.

2 Background

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 $|\mathcal{S}| = D$ such that $X_{\cdot \mathcal{S}}$ 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.

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 4 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 equivalent 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 $D_F^E \phi = UD\phi V^T$ 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. **Kohli2021-lr** showed how pointwise isometries selected from a dictionary may be stitched together to form global embeddings. Another line of work **Koelle2022-ju**; **Koelle2024-no** 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 **Koelle2022-ju**. 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 Applications

2.3.1 Diversification

Say you are hosting an elegant dinner party.

2.3.2 Embedding

2.3.3 Interpretability

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 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 6.1. 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 5

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 1. 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

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

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

While this normalization satisfies 3, 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.

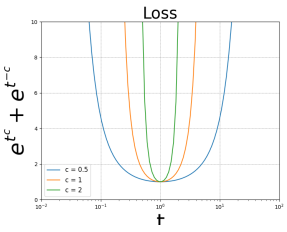

Figure 1: Ground truth loss scaling function g as a function of t

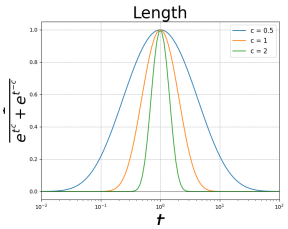

Figure 2: Length as a function of t

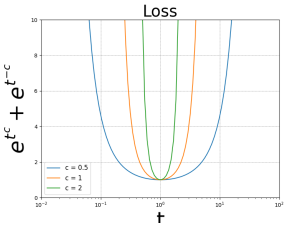

Figure 3: Basis pursuit losses as a function of t

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.

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

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

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

The isometry pursuit program is then

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

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

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

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

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

IsometryPursuit(Matrix $\mathcal{X} \in \mathbb{R}^{D\times P}$, scaling constant c)

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 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 6.1.1 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. From an intuitive perspective, this is akin to saying selected columns are not only complementary in that they are orthogonal, but also well-balanced in that their columns are of a length 1 rather than length 0.

Proposition 5 (Unitary selection) *Given a matrix $\mathcal{X} \in \mathbb{R}^{D\times P}$ with a rank D submatrix $\mathcal{X}_{\cdot\mathcal{S}} \in \mathbb{R}^{D\times D}$ 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 5

Proposition 6 (Loss equivalence) *Given a diagonalizable matrix $\mathcal{X} \in \mathbb{R}^{D\times D}$, $\|\widehat{\beta}_c^P(\mathcal{X})\|_{1,2} = l_c(\mathcal{X})$.*

This proposition should be thought of as applying to D column matrices after selection. We know that unitary submatrices minimize loss, and should they be present, they will be selected by either method. This proposition shows that should diagonalizable matrices be selected, they will also have equivalent loss, but not necessarily that they will be selected.

3.4 Isometric lasso

The convex loss function 17 and linear constraint in 19 admit a Lagrangian dual which we shall call Isometric Lasso. While we defer full discussion of computational complexity to Section **??**, the general principal which motivates the use of this formulation in our setting is the increased computational expediency in high-dimensions.

The Isometric Lasso loss is

$$l_\lambda(\mathcal{X}, \beta) = \|I_D - \tilde{\mathcal{X}}_c \beta\|_2^2 + \lambda \|\beta\|_{1,2} \tag{22}$$

which can be optimized as

$$\hat{\beta}_\lambda(\mathcal{X}) = \arg \min_{\beta \in \mathbb{R}^{P \times D}} l_\lambda(\mathcal{X}, \beta) \tag{23}$$

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

Proposition 7 (Lasso selection equivalence) *Let $U \in \mathbb{R}^{D \times D}$ be unitary. Then $S(\widehat{\beta}_\lambda(U\mathcal{X})) = S(\widehat{\beta}_\lambda(\mathcal{X}))$.*

This also covers changing the target variable.

Note that it also may be possible to argue the basis pursuit invariance from the lasso ones plus Lagranian duality, but to avoid taking the limit we prove both propositions independently.

3.5 Extension to non-linear spaces

Proposition 8 (Isometry at a point 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 6.1.3.

3.6 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 **Hesterberg2008-iy** 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 5 for the Lasso formulation, and similar lasso-specific conditions such as those found in **Koelle2024-no** are not satisfied by overcomplete dictionaries.

3.7 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.

3.8 Computational complexity

4 Discussion

This extension is a local version of Tangent Space Lasso.

Tangent space basis pursuit satisfies a similar property **Koelle2022-lp** but the normalization process differs.

It could be used in the stiching step of an algorithm like the kohli one We leave aside the question of patch alignment **https://arxiv.org/pdf/2303.11620.pdf**; **LDLE paper**. The full gradient approach. In this case normalization prior to projection is subsumbmed by the larger coefficients needed to get the tangent space. Good news is tangent space estimation need not be performed. Let's compare the coefficients involved in projecting versus not projecting. We can perform regression in the high dimensional space instead of projecting on span of target variable.

With respect to pseudoinverse estimation, sparse methods have been applied in **Sun2012-vp**

Even though by Lagrangian duality, the basis pursuit solution corresponds to λ approaching 0, the solution is sparse **Tropp04-ju**. about the lasso is that all coefficients enter the regularization path. As we see by the correspondence between λ approaching 0 and the basis pursuit problem, some coefficients in fact do not go to 0.

Proof of local isometry (simpler proof since no oscillation game)

Bertsimas2022-qo gives a method for solving the sparse-PCA method more efficiently than the original greedy approach. Compared with the FISTA method used in **Koelle2022-ju**; **Koelle2024-no**, coordinate descent **Friedman-2007-yb**; **Meier2008-ts**; **Qin2013-tx** is faster **Catalina2018-ek**; **Zhao2023-xn**. Compared with **Liu2009-yo**, the sklearn multitask lasso is $2,1$ rather than $\infty,1$ regularized.

Compared with Gram-Schmidt It is likely that the transformed singular value loss could be reframed as a semdefinite programming problem, since the composition of two convex functions is convex **Boyd2004-ql**.

Multitask lasso **Obozinski2006-kq**; **Yeung2011-fg** is a form of group lasso **Yuan2006-bt** where coefficients are group by response variable.

See **Obozinski2006-kq** for a comparison of forward and backward selection with lasso.

Our notion of isometric recovery is distinct from the restricted isometry property **Candes2005-dd**; **Hastie2015-qa**, which is used to show guaranteed recovery at fast convergence rates in supervised learning. In particular, our approach does not consider statistical error or the presence of a true underlying model. However, we note that disintigration of performance at high λ values in the lasso formulation may have some relation to these properties, as discussed in **Koelle2022-ju**; **Koelle2024-no**.

A major area of comparison is in diversification in recommendation systems. Greedy algorithms are used **Carbonell2017-gi**; **Wu2019-uk**

Compared with sparse pca **Bertsimas2022-qo**; **Bertsimas2022-dv**,

we are not concerned with variability in the dataset, and select. While the sparse PCA problem is non-convex, our approach can be taken as a simpler version in the sense that the loadings are constrained to be the identify matrix. **Tropp06-sg** and **Liu2009-yo** use a $1, \infty$ norm to induce sparsity that misses the utility of our normalization for finding unitary matrices. since isometry embeddings preserve important properties like distances between points.

5 Experiments

Comparison with isometry loss.

6 Supplement

We give proofs in support of the propositions in the main text and supplemental experimental information to better contextualize the results.

6.1 Proofs

6.1.1 Proof of Proposition ??

Proposition 9 *Let $U \in \mathbb{R}^{D\times D}$ be unitary. Then $\hat{\beta}_\lambda(U\mathcal{X}) = \hat{\beta}_\lambda(U\mathcal{X})$.*

Proposition 10 *Loss equivalence Let $U \in \mathbb{R}^{D\times D}$ be unitary. Then $\|\beta\|_{1,2} = \|\beta U\|$.*

6.1.2 Proof of Propositions 7 and ??

These proofs rely on some elementary applications of linear algebra. Proposition 7 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 11 *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

$$l^*(X^i) = l(\beta^i) = \sum_{j=1}^{p}(\sum_{i'=2}^{n} \|\beta_{i'j.}\|_2^2 + \|\beta_{1j.}^i\|_2^2)^{1/2} = \sum_{j=1}^{p}(\sum_{i'=1}^{n} \|\beta_{i'j.}U\|_2^2)^{1/2} = l^*(X) \quad (24)$$

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

Proposition 12 *Programmatic equivalence Let $U \in \mathbb{R}^{D\times D}$ be unitary. Then $\hat{\beta}_\lambda(U\mathcal{X}) = U\hat{\beta}_\lambda(\mathcal{X})$.*

6.1.3 Proof of Proposition 8

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

Proposition 13 *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

$$\beta^i_{ijk} = \begin{cases} \|\tilde{X}'_{.S_j}\|_2^{-1} & j = k \in \{1 \dots d\} \\ 0 & \text{otherwise} \end{cases}. \tag{25}$$

Therefore,

$$\tilde{l}^*(X') = \sum_{j=1}^{d} \sqrt{\sum_{i=1}^{n} \|\tilde{X}'_{i.S_j}\|_2^{-2}}. \tag{26}$$

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

$$R = \begin{bmatrix} \langle e^1, \tilde{X}_{i.S_1} \rangle & \langle e^1, \tilde{X}_{i.S_2} \rangle & \dots & \langle e^1, \tilde{X}_{i.S_d} \rangle \\ 0 & \langle e^2, \tilde{X}_{i.S_2} \rangle & \dots & \langle e^2, \tilde{X}_{i.S_d} \rangle \\ 0 & 0 & \dots & \dots \\ 0 & 0 & 0 & \langle e^d, \tilde{X}_{i.S_d} \rangle \end{bmatrix}. \tag{27}$$

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 14 *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

$$\tilde{X}_{.S_j} = \begin{bmatrix} c_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & c_d\rangle \end{bmatrix}. \tag{28}$$

Thus

$$\tilde{\exp}_1 X_{.S_j} = \begin{bmatrix} \exp(-|\log \|c_1\|_2)|) & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \exp(-|\log \|c_d\|_2)|)\rangle \end{bmatrix}. \tag{29}$$

and therefore

$$\tilde{\beta}_{.S_j} = \begin{bmatrix} \exp(-|\log \|c_1\|_2)|)^{-1} & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \exp(-|\log \|c_d\|_2)|)^{-1}\rangle \end{bmatrix}. \tag{30}$$

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.

6.2 Experiments