SPARSENESS BY ITERATIVE PROJECTIONS ONTO SPHERES

Fabian J. Theis[*]

Inst. of Biophysics, Univ. of Regensburg
93040 Regensburg, Germany

Toshihisa Tanaka[*]

Dept. EEE, Tokyo Univ. of Agri. and Tech.
Tokyo 184-8588, Japan

ABSTRACT

Many interesting signals share the property of being sparsely active. The search for such sparse components within a data set commonly involves a linear or nonlinear projection step in order to fulfill the sparseness constraints. In addition to the proximity measure used for the projection, the result of course is also intimately connected with the actual definition of the sparseness criterion. In this work, we introduce a novel sparseness measure and apply it to the problem of finding a sparse projection of a given signal. Here, sparseness is defined as the fixed ratio of p- over 2-norm, and existence and uniqueness of the projection holds. This framework extends previous work by Hoyer in the case of $p = 1$, where it is easy to give a deterministic, more or less closed-form solution. This is not possible for $p \neq 1$, so we introduce an algorithm based on alternating projections onto spheres (POSH), which is similar to the projection onto convex sets (POCS). Although the assumption of convexity does not hold in our setting, we observe not only convergence of the algorithm, but also convergence to the correct minimal distance solution. Indications for a proof of this surprising property are given. Simulations confirm these results.

1. INTRODUCTION

Sparseness is an important property of many natural signals, and various definitions exist. Intuitively, a signal $\mathbf{x} \in \mathbb{R}^n$ increases in sparseness with the increasing number of zeros; this is often measured by the 0-(pseudo)-norm $\|\mathbf{x}\|_0 := |\{i|x_i \neq 0\}|$, counting the number of non-zero entries of $\mathbf{x}$. It is a pseudo-norm because $\|\alpha\mathbf{x}\|_0 = |\alpha|\|\mathbf{x}\|_0$ does not necessarily hold; indeed $\|\alpha\mathbf{x}\|_0 = \|\mathbf{x}\|_0$ if $\alpha \neq 0$, so $\|.\|_0$ is scale-invariant.

A typical problem in the field is the search for sparse instances or representations of a data set. Using the above 0-pseudo-norm as sparseness measure quickly turns out to be both theoretically and algorithmically unfeasible. The former simply follows because $\|.\|_0$ is discrete, so the indeterminacies of the problem can be expected to be very high, and the latter because optimization on such a discrete function is a combinatorial problem and indeed turns out to be NP-complete. Hence, this sparseness measure is commonly approximated by some continuous measures e.g. by replacing it by the p-norm $\|\mathbf{x}\|_p := (\sum_{i=1}^n |x_i|^p)^{1/p}$ for $p \in \mathbb{R}^+$. As $\lim_{p\to 0+} \|\mathbf{x}\|_p^p = \|\mathbf{x}\|_0$, this can be interpreted as a possible approximation. This together with extensions to noisy situations can be used for measuring sparseness, and the connection with $\|.\|_0$, especially in the case of $p = 1$, has been intensively studied [1].

Often, we are not interested in the scale of the signals, so ideally the sparseness measure should be independent of the scale — which is the case for the 0-pseudo-norm, but not for the p-norms. In order to guarantee scaling invariance, some normalization has to be applied in the latter case, and a possible solution is the measure

$$\sigma_p(\mathbf{x}) := \|\mathbf{x}\|_p / \|\mathbf{x}\|_2 \tag{1}$$

for $\mathbf{x} \in \mathbb{R}^n \setminus \{0\}$ and $p > 0$. Then $\sigma_p(\alpha\mathbf{x}) = \sigma_p(\mathbf{x})$ for $\alpha \neq 0$; moreover the sparser $\mathbf{x}$, the smaller $\sigma_p(\mathbf{x})$. Indeed, it can still be interpreted as approximation of the 0-pseudo-norm in the sense that it is scale-invariant and that $\lim_{p\to 0+} \sigma_p(\mathbf{x})^p = \|\mathbf{x}\|_0$. Altogether we infer that by minimizing $\sigma_p(\mathbf{x})$ under some constraint, we can find a sparse representation of $\mathbf{x}$. Hoyer [2] noticed this in the important case of $p = 1$; he defined a normalized sparseness measure by $(\sqrt{n} - \sigma_1(\mathbf{x}))/(\sqrt{n} - 1)$, which lies in $[0, 1]$ and is maximal if $\mathbf{x}$ contains $n - 1$ zeros and minimal if the absolute value of all coefficients of $\mathbf{x}$ coincide.

Little attention has been paid to finding projections in the case of $p \neq 1$, which is particularly important for $p \to 0$ as better approximation of $\|.\|_0$. Hence, the goal of this manuscript is to explore the general notion of sparseness in the sense of equation (1) and to construct algorithms to project a vector to its closest vector of a given sparseness.

2. EUCLIDEAN PROJECTION

Let $M \subset \mathbb{R}^n$ be an arbitrary, non-empty set. A vector $\mathbf{y} \in M \subset \mathbb{R}^n$ is called *Euclidean projection of* $\mathbf{x} \in \mathbb{R}^n$ *in* M, in symbols $\mathbf{y} \lhd_M \mathbf{x}$, if $\|\mathbf{x} - \mathbf{y}\|_2 \leq \|\mathbf{x} - \mathbf{z}\|_2$ for all $\mathbf{z} \in M$.

2.1. Existence and uniqueness

We review conditions [3] for existence on uniqueness of the Euclidean projection. For this, we need the following notion: Let $\mathcal{X}(M) := \{\mathbf{x} \in \mathbb{R}^n \,|\, \text{there exists more than one point adjacent to } \mathbf{x} \text{ in } M\} = \{\mathbf{x} \in \mathbb{R}^n \,|\, \#\{\mathbf{y} \in M \,|\, \mathbf{y} \lhd_M \mathbf{x}\} > 1\}$ denote the *exception set* of M.

Theorem 2.1 (Euclidean projection)**.**

i. If M is closed and nonempty, then the Euclidean projection onto M is exists that is for every $\mathbf{x} \in \mathbb{R}^n$ there exists a $\mathbf{y} \in M$ with $\mathbf{y} \lhd_M \mathbf{x}$.

ii. The Euclidean projection onto M is unique from almost all points in $\mathbb{R}^n$ i.e. $\mathrm{vol}(\mathcal{X}(M)) = 0$.

Proof. See [3], theorems 2.2 and 2.6. □

So we can always project a vector $\mathbf{x} \in \mathbb{R}^n$ onto a closed set M, and this projection will be unique almost everywhere. In this case, we denote the projection by $\boldsymbol{\pi}_M(\mathbf{x})$ or $\boldsymbol{\pi}(\mathbf{x})$ for short. Indeed, in the case of the p-spheres S_p^{n-1}, the exception set consists of a single point $\mathcal{X}(S_p^{n-1}) = \{0\}$ if $p \geq 2$, hence $\boldsymbol{\pi}_{S_p^{n-1}}$ is well-defined on $\mathbb{R}^n \setminus \{0\}$. If $p < 2$, additional non-uniqueness points exists on the coordinate hyperplanes, which can be ignored thanks to theorem 2.1(ii).

[*]Partial financial support by the JSPS (PE 05543) and the DFG (GRK 638) is acknowledged.

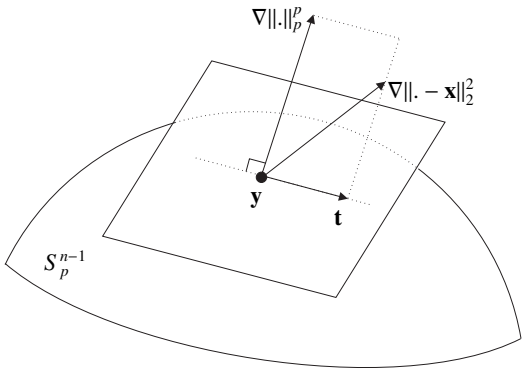


Fig. 1. Constrained gradient $\mathbf{t}$ on the p-sphere, given by the projection of the unconstrained gradient $\nabla\|. - \mathbf{x}\|_2^2$ onto the tangent space that is orthogonal to $\nabla\|.\|_p^p$, see equation (6).

2.2. Projection onto a p-sphere

Let $S_p^{n-1} := \{\mathbf{x} \in \mathbb{R}^n | \ \|\mathbf{x}\|_p = 1\}$ denote the $(n-1)$-dimensional sphere with respect to the p-norm ($p > 0$). A scaled version of this unit sphere is given by $cS_p^{n-1} := \{\mathbf{x} \in \mathbb{R}^n | \ \|\mathbf{x}\|_p = c\}$. The spheres are smooth C^1-submanifolds of $\mathbb{R}^n$ for $p \geq 2$. For $p < 2$, they have corners at the intersections with the coordinate axes.

Later we will need to explicitly find the Euclidean projection onto a p-sphere, so we introduce it algorithmically in the following. First note that without loss of generality we may assume that the p-sphere has been scaled to a unit sphere because the general case can be recovered by $\boldsymbol{\pi}_{cS_p^{n-1}}(\mathbf{x}) = c\boldsymbol{\pi}_{S_p^{n-1}}(\mathbf{x}/c)$ for $c > 0$.

Now, in the case $p = 2$, the projection is simply given by

$$\boldsymbol{\pi}_{S_2^{n-1}}(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2. \tag{2}$$

In the case $p = 1$, the sphere consists of a union of hyperplanes being orthogonal to $(\pm 1, \ldots, \pm 1)$. Considering only the first quadrant (i.e. $x_i > 0$), this means that $\boldsymbol{\pi}_{S_1^{n-1}}(\mathbf{x})$ is given by the projection onto the hyperplane $H := \{\mathbf{x} \in \mathbb{R}^n | \langle \mathbf{x}, \mathbf{e} \rangle = n^{-1/2}\}$ and setting resulting negative coordinates to 0; here $\mathbf{e} := n^{-1/2}(1, \ldots, 1)$. So with $x_+ := x$ if $x \geq 0$ and 0 otherwise, we get

$$\boldsymbol{\pi}_{S_1^{n-1}}(\mathbf{x}) = \left(\mathbf{x} + (n^{-1/2} - \langle \mathbf{x}, \mathbf{e} \rangle)\mathbf{e}\right)_+ . \tag{3}$$

In the case of arbitrary $p > 0$, the projection is given by the unique solution of

$$\boldsymbol{\pi}_{S_p^{n-1}}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{y} \in S_p^{n-1}} \|\mathbf{x} - \mathbf{y}\|_2^2. \tag{4}$$

Unfortunately, no closed-form solution exists in the general case, so we have to numerically determine the solution. We have experimented with a) explicit Lagrange multiplier calculation and minimization, b) constrained gradient descent and c) constrained fixed-point algorithm (best). Ignoring the singular points at the coordinate hyperplanes, let us first assume that all $x_i > 0$. Then at a regular solution $\mathbf{y}$ of equation (4), the gradient of the function to be minimized is parallel to the gradient of the constraint, i.e. $\mathbf{y} - \mathbf{x} = \lambda \ \nabla\|.\|_p^p\big|_{\mathbf{y}}$ for some Lagrange-multiplier $\lambda \in \mathbb{R}$, which can be calculated from the additional constraint equation $\|\mathbf{y}\|_p^p = 1$. Using the notation $\mathbf{y}^{\odot p} := (y_1^p, \ldots, y_n^p)^\top$ for the componentwise exponentiation, we therefore get

$$\mathbf{y} - \mathbf{x} = \lambda p \, \mathbf{y}^{\odot(p-1)} \qquad \text{and} \qquad \sum_i y_i^p = 1 \tag{5}$$

Algorithm 1: Projection onto S_p^{n-1} by constrained gradient descent. Commonly, the iteration is stopped after the update step-size lies below some given threshold.

Input: vector $\mathbf{x} \in \mathbb{R}^n$, learning rate $\eta(i)$
Output: Euclidean projection $\mathbf{y} = \boldsymbol{\pi}_{S_p^{n-1}}(\mathbf{x})$

1 Initialize $\mathbf{y} \in S_p^{n-1}$ randomly.
for $i \leftarrow 1, 2, \ldots$ **do**
2 $\quad \mathbf{df} \leftarrow \mathbf{y} - \mathbf{x}$, $\mathbf{dg} \leftarrow p \ \operatorname{sgn}(\mathbf{y}) \, |\mathbf{y}|^{\odot(p-1)}$
3 $\quad \mathbf{t} \leftarrow \mathbf{df} - \mathbf{df}^\top \mathbf{dg} \mathbf{dg} / (\mathbf{dg}^\top \mathbf{dg})$
4 $\quad \mathbf{y} \leftarrow \mathbf{y} - \eta(i)\mathbf{t}$
5 $\quad \mathbf{y} \leftarrow \mathbf{y} / \|\mathbf{y}\|_p$
end

For $p \notin \{1, 2\}$, these equations cannot be solved in closed form, hence we propose an alternative approach to solving the constrained minimization (4). The goal is to minimize $f(\mathbf{y}) := \|\mathbf{y} - \mathbf{x}\|_2^2$ under the constraint $g(\mathbf{y}) := \|\mathbf{y}\|_p^p = 1$. This can for example be achieved by gradient-descent methods, taking into account that the gradient has to be calculated on the submanifold given by the S_p^{n-1}-constraint, see figure 1 for an illustration. The projection of the gradient ∇f onto the tangent space of S_p^{n-1} at $\mathbf{y}$ can be easily calculated as

$$\mathbf{t} = \nabla f - \langle \nabla f, \nabla g \rangle \nabla g / \|\nabla g\|_2^2. \tag{6}$$

Here, the explicit gradients are given by $\nabla f(\mathbf{y}) = \mathbf{y} - \mathbf{x}$ and $\nabla g(\mathbf{y}) = p \ \operatorname{sgn}(\mathbf{y}) \, |\mathbf{y}|^{\odot(p-1)}$, where $\operatorname{sgn}(\mathbf{y})$ denotes the vector of the componentwise signs of $\mathbf{y}$, and $|\mathbf{y}| := \operatorname{sgn}(\mathbf{y})\mathbf{y}$ the componentwise absolute value. The projection algorithm is summarized in algorithm 1. Iteratively, after calculating the constrained gradient (lines 2 and 3), it performs a gradient-descent update step (line 4) followed by a projection onto S_p^{n-1} (line 5) to guarantee that the algorithm stays on the submanifold.

The method performs well, however as most gradient-descent-based algorithms, without further optimization it takes quite a few iterations to achieve acceptable convergence, and the choice of an optimal learning rate $\eta(i)$ is non-trivial. We therefore propose a second projection method employing a fixed-point optimization strategy. Its idea is based on the fact that at local minima $\mathbf{y}$ of $f(\mathbf{y})$ on S_p^{n-1}, the gradient $\nabla f(\mathbf{y})$ is orthogonal to S_p^{n-1}, so $\nabla f(\mathbf{y}) \propto \nabla g(\mathbf{y})$. Ignoring signs for illustrative purposes, this means that $\mathbf{y} - \mathbf{x} \propto p \, \mathbf{y}^{\odot(p-1)}$, so $\mathbf{y}$ can be calculated from the fixed-point iteration $\mathbf{y} \leftarrow \lambda p \, \mathbf{y}^{\odot(p-1)} + \mathbf{x}$ with additional normalization. Indeed, this can be equivalently derived from the previous Lagrange equations (5), also yielding equations for the proportionality factor λ: we can simply determine it from one component of equation (5), or to increase numerical robustness, as mean from the total set. Taking into account the signs of the gradient (which we ignored in equation (5)), this yields an estimate $\hat{\lambda} := \frac{1}{n} \sum_{i=1}^n (y_i - x_i)/(p \ \operatorname{sgn}(y_i) \, |y_i|^{p-1})$. Altogether, we get the fixed-point algorithm 2, which in experiments turns out to have a considerably higher convergence rate than algorithm 1.

In table 1, we compare the algorithms 1 and 2, namely with respect to the number of iterations they need to achieve convergence below some given threshold. As expected, the fixed-point algorithm outperforms gradient-descent always except for the case of higher dimensions and $p > 2$ (non-sparse case). In the following we will therefore use the fixed-point algorithm for projection onto S_1^{n-1}.

2.3. Projection onto convex sets

If M is a convex set, then the Euclidean projection $\boldsymbol{\pi}_M(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^n$ is already unique, so $\mathcal{X}(M) = \emptyset$ and the operator $\boldsymbol{\pi}_M$ is called

Algorithm 2: Projection onto S_p^{n-1} via fixed-point iteration. Again, the iteration is to be stopped after only sufficiently small updates are taken.

Input: vector $\mathbf{x} \in \mathbb{R}^n$
Output: Euclidean projection $\mathbf{y} = \boldsymbol{\pi}_{S_p^{n-1}}(\mathbf{x})$

Initialize $\mathbf{y} \in S_p^{n-1}$ randomly.
for $i \leftarrow 1, 2, \ldots$ **do**
  $\lambda \leftarrow \sum_{i=1}^n (y_i - x_i) \,/\, \left(n \operatorname{sgn}(y_i) |y_i|^{p-1}\right)$
  $\mathbf{y} \leftarrow \mathbf{x} + \lambda \operatorname{sgn}(\mathbf{y}) |\mathbf{y}|^{\circledcirc(p-1)}$
  $\mathbf{y} \leftarrow \mathbf{y}/\|\mathbf{y}\|_p$
end

Table 1. Comparison of the gradient- and fixed-point-based projection algorithms 1 and 2 for finding the Euclidean projection onto cS_p^{n-1} for varying parameters; mean was taken over 100 iterations with $\mathbf{x} \in [-1, 1]^n$ sampled uniformly. Here $\#\text{its}_{\text{gd}}$ and $\#\text{its}_{\text{fp}}$ denote the numbers of iterations the algorithm took to achieve update steps of size smaller than $\varepsilon = 0.0001$, and $\|\mathbf{y}_{\text{gd}} - \mathbf{y}_{\text{fp}}\|$ equals the norm of the difference of the found minima.

n	p	c	$\#\text{its}_{\text{gd}}$	$\#\text{its}_{\text{fp}}$	$\|\mathbf{y}_{\text{gd}} - \mathbf{y}_{\text{fp}}\|$
2	0.9	1.2	6.7 ± 4.7	3.7 ± 1.0	0.0 ± 0.0
2	0.9	2.0	10.9 ± 6.9	4.1 ± 1.0	0.0 ± 0.0
2	2.2	0.9	13.0 ± 21.0	5.5 ± 4.2	0.0 ± 0.0
3	0.9	3	13.7 ± 6.9	4.4 ± 1.0	0.0 ± 0.0
3	2.2	0.9	9.6 ± 16.6	7.2 ± 10.2	0.0 ± 0.0
4	0.9	3	9.8 ± 6.8	4.4 ± 1.1	0.0 ± 0.0
4	2.2	0.9	6.0 ± 5.0	9.2 ± 8.1	0.0 ± 0.0

convex projector, see e.g. [3], lemma 2.4 and [4]. The theory of *projection onto convex sets (POCS)* [4, 5] is a well-known technique in signal processing; given N convex sets $M_1, \ldots, M_N \subset \mathbb{R}^n$ and an operator defined by $\boldsymbol{\pi} = \boldsymbol{\pi}_{M_N} \cdots \boldsymbol{\pi}_{M_1}$, POCS can be formulated as the recursion defined by $\mathbf{y}_{i+1} = \boldsymbol{\pi}(\mathbf{y}_i)$. It always approaches the intersection of the convex sets that is $\mathbf{y}_i \to M^* = \bigcap_{i=1}^N M_i$.

Note that POCS only finds an *arbitrary* point in $\bigcap_{i=1}^N M_i$, which not necessarily coincides with its Euclidean projection: for example if $M_1 := \{\mathbf{x} \in \mathbb{R}^n | \ \|\mathbf{x}\|_2 \leq 1\}$ is the unit disc, and $M_2 := \{\mathbf{x} \in \mathbb{R}^n | \ x_1 \leq 0\}$ the lower first half-plane, then the Euclidean projection from $\mathbf{x} := (1, 1, 0, \ldots, 0)$ onto $M_1 \cap M_2$ equals $\boldsymbol{\pi}_{M_1 \cap M_2}(\mathbf{x}) = (0, 1, 0, \ldots, 0)$, but application of POCS yields $(0, 1/\sqrt{2}, 0, \ldots, 0)$.

3. SPARSE PROJECTION

In this section, we combine the notions from the previous sections to search for sparse projections. Given a signal $\mathbf{x} \in \mathbb{R}^n$, our goal is to find the closest signal $\mathbf{y} \in \mathbb{R}^n$ of fixed sparseness $\sigma_p(\mathbf{y}) = c$. Hence, we search for $\mathbf{y} \in \mathbb{R}^n$ with

$$\mathbf{y} = \operatorname{argmin}_{\sigma_p(\mathbf{y})=c} \|\mathbf{x} - \mathbf{y}\|_2. \tag{7}$$

Due to the scale-invariance of σ, the problem (7) is equivalent to finding

$$\mathbf{y} = \operatorname{argmin}_{\|\mathbf{y}\|_2=1, \|\mathbf{y}\|_p=c} \|\mathbf{x} - \mathbf{y}\|_2. \tag{8}$$

In other words, we are looking for the Euclidean projection $\mathbf{y} = \boldsymbol{\pi}_M(\mathbf{x})$ onto $M := S_2^{n-1} \cap cS_p^{n-1}$. Note that due to theorem 2.1, this solution to (8) exists if $M \neq \emptyset$ and is almost always unique.

Algorithm 3: Projection onto spheres (POSH). In practice, some abort criterion has to be implemented. Often $q = 2$.

Input: vector $\mathbf{x} \in \mathbb{R}^n \setminus \mathcal{X}(S_p^{n-1} \cap S_q^{n-1})$ and $p, q > 0$
Output: $\mathbf{y} = \boldsymbol{\pi}_{S_p^{n-1} \cap S_q^{n-1}}(\mathbf{x})$

Set $\mathbf{y} \leftarrow \mathbf{x}$.
while $\mathbf{y} \notin S_p^{n-1} \cap S_q^{n-1}$ **do**
  $\mathbf{y} \leftarrow \boldsymbol{\pi}_{S_q^{n-1}}(\boldsymbol{\pi}_{S_p^{n-1}}(\mathbf{y}))$
end

3.1. Projection onto spheres (POSH)

In the special case of $p = 1$ and nonnegative $\mathbf{x}$, Hoyer has proposed an efficient algorithm for finding the projection [2], simply by using the explicit formulas for the p-sphere projection; such formulas do not exist for $p \neq 1, 2$, so a more general algorithm for this situation is proposed in the following.

Its idea is a direct generalization of POCS: we alternately project first on S_2^{n-1} then on S_p^{n-1}, using the Euclidean projection operators from section 2.2. However, the difference is that the spheres are clearly non-convex (if $p \neq 1$), so in contrast to POCS, convergence is not obvious. We denote this projection algorithm by *projection onto spheres (POSH)*, see algorithm 3.

First note that POSH obviously has the desired solution as fixed-point. In experiments, we observe that indeed POSH converges, and moreover it converges to the closest solution i.e. to the Euclidean projection (which does not hold for POCS in general)! Finally we see that in higher dimensions, all update vectors together with the starting point $\mathbf{x}$ lie in a single two-dimensional plane, so theoretically we can reduce proofs to two-dimensional cases as well as build algorithms using this fact.

In the following section, we will prove the above claims for the case of $p = 1$, where an explicit projection formula (3) is known. In the case of arbitrary p, so far we are only able to give experimental validation of the astonishing facts of convergence and convergence to the Euclidean projection.

3.2. Convergence

The proof needs the following simple convergence lemma, which somewhat extends on a special case treated by the more general Banach fixed-point theorem.

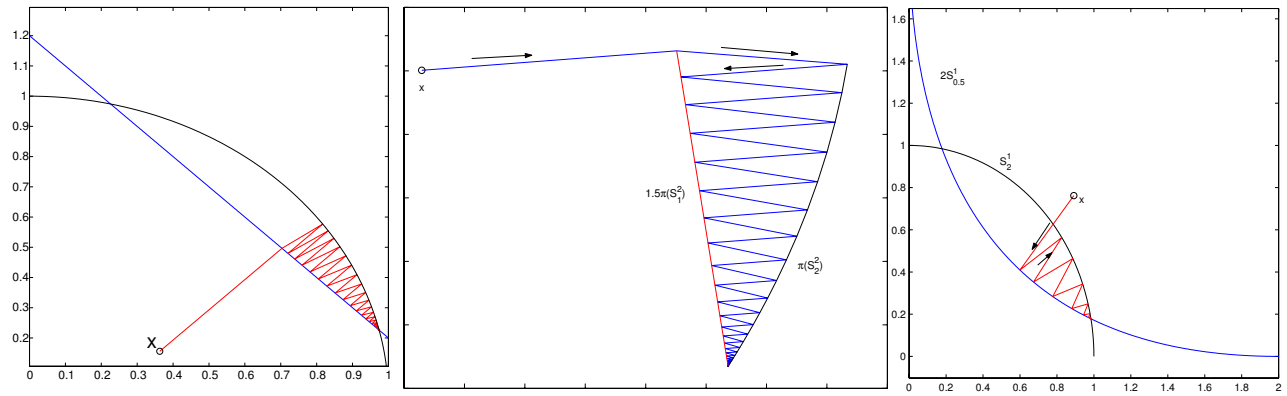
Lemma 3.1. *Let $f : \mathbb{R} \to \mathbb{R}$ be continuously-differentiable with $f(0) = 0$, $f'(0) > 1$ and let f' be positive and strictly decreasing. If $f'(x) < 1$ for some $x > 0$, then there exists a single positive fixed-point $\hat{x}$ of f, and $f^i(x)$ converges to $\hat{x}$ for $i \to \infty$ and any $x > 0$.*

Theorem 3.2. *Let $n \geq 2$, $p > 0$ and $\mathbf{x} \in \mathbb{R}^n \setminus \mathcal{X}(M)$. If $\mathbf{y}^1 := \boldsymbol{\pi}_{S_2^{n-1}}(\mathbf{x})$ and iteratively $\mathbf{y}^i := \boldsymbol{\pi}_{S_2^{n-1}}(\boldsymbol{\pi}_{S_1^{n-1}}(\mathbf{y}^{i-1}))$ according to the POSH algorithm, then $\mathbf{y}^i$ converges to some $\mathbf{y}^\infty \in S_2^{n-1}$, and $\mathbf{y}^\infty = \boldsymbol{\pi}_M(\mathbf{x})$.*

Using lemma 3.1, we can prove the convergence theorem in the case of $p = 1$, but omit the proof due to lack of space.

3.3. Simulations

At first, we confirm the convergence results from theorem 3.2 for $p = 1$ by applying POSH with 100 iterations in 1000 runs onto vectors $\mathbf{x} \in \mathbb{R}^6$ sampled uniformly from $[0, 1]^6$; c was chosen to be sufficiently large ($c = 2.4$). We always get convergence. We also calculate the correct projection (using Hoyer's projection algorithm



(a) POSH for $n = 2$, $p = 1$ (b) POSH for $n = 3$, $p = 1$ (c) POSH for $n = 2$, $p = 0.5$

Fig. 2. Starting from $\mathbf{x}_0$ ($\circ$), we alternately project onto cS_1 and S_2. POSH performance is illustrated for $p = 1$ in dimensions 2 (a) and 3 (b), were a projection via PCA is displayed — no information is lost hence the sequence of points lies in a plane as shown in the proof theorem 3.2. Figure (c) shows application of POCS for $n = 2$ and $p = 0.5$.

Table 2. Performance of the POSH algorithm 3 for varying parameters. See text for details.

n	p	c	$\|\mathbf{y}_{\text{POSH}} - \mathbf{y}_{\text{scan}}\|_2$
2	0.8	1.2	0.005 ± 0.0008
2	4	0.9	0.02 ± 0.005
3	0.8	1.2	0.02 ± 0.009
3	4	0.9	0.04 ± 0.03

[2]). The distance between his and our solution was calculated to give a mean value of $5 \cdot 10^{-13} \pm 5 \cdot 10^{-12}$ i.e. we get virtually always the same solution.

In figures 2(a) and (b), we show application for $p = 1$; we visualize the performance in 3 dimensions by projecting the data via PCA — which by the way throws away virtually no information (confirmed by experiment) indicating the validness of theorem 3.2 also in higher dimensions. In figure 2(c) a projection for $p = 0.5$ is shown.

Now, we perform batch-simulations for varying p. For this, we uniformly sample the starting vector $\mathbf{x} \in [0, 1]^n$ in 100 runs, and compare the POSH algorithm result with the true projection. POSH is performed starting with the p-norm projection using algorithm 1 and 100 iterations. As the true projection $\boldsymbol{\pi}_M(\mathbf{x})$ cannot be determined in closed form, we scan $[0, 1]^{n-1}$ using the stepsize $\varepsilon = 0.01$ to give the first $(n-1)$ coordinates of our estimate $\mathbf{y}$ of $\boldsymbol{\pi}_M(\mathbf{x})$; its n-th coordinate is then constructed to guarantee $\mathbf{y} \in S_p^{n-1}$ (for $p < 1$) or $\mathbf{y} \in S_2^{n-1}$ (for $p > 1$) respectively. Using Taylor-approximation of $(y + \varepsilon)^p$, it can easily be shown that two adjacent grid points have maximal difference $\left| \|(y_1 + \varepsilon, \ldots, y_n + \varepsilon)\|_p^p - \|\mathbf{y}\|_p^p \right| \leq pn\varepsilon + O(\varepsilon^2)$ if $\mathbf{y} \in [0, 1]^n$ and $p \geq 1$. Hence by taking only vectors $\mathbf{y}$ as approximation of $\boldsymbol{\pi}_M(\mathbf{x})$ with $\left| \|\mathbf{y}\|_2^2 - 1 \right| < n\varepsilon$ (for $p < 1$) or $\left| \|\mathbf{y}\|_p^p - c^p \right| < pn\varepsilon$ moreover guarantees that $\mathbf{y}$ approximately lies in M. We then choose $\mathbf{y}$ within this set with minimal distance to the original $\mathbf{x}$ as approximate of $\boldsymbol{\pi}_M(\mathbf{x})$. Table 2 shows the performance of POSH for varying dimension and sparseness-parameter p. Clearly the differences between the POSH result $\mathbf{y}_{\text{POSH}}$ and the approximated true projection $\mathbf{y}_{\text{scan}}$ is in the order of ε, which confirms that POSH converges to the correct projection also for $p \neq 1$.

4. CONCLUSION

We have shown how to reach signal sparseness, defined simply by the p-norm after normalization. As a modification of the traditional POCS, we have proposed a new projection algorithm named POSH in order to find projections for $p \neq 1$. We have theoretically justified this idea in some cases, and experimental results support our claim. Our theory and algorithm now has a wide range of applications, since measuring sparseness by norms with $p < 1$ models the 0-pseudo-norm closer than the common 1-norm approach, analyzed for example in [1, 2]. Since we have modeled the algorithm after POCS, many of its extensions (see e.g. [4]) may possibly translated to the non-convex POSH model. Although this paper mostly focuses on the theoretical aspects, we can apply the sparse projection to matrix factorization problems similar to [2]. Also, by imposing additional constraint such as non-negativity to the model — which is easily possible in POSH by simply adding an additional (possibly convex) projection step — various applications to image processing are possible, where non-negativity naturally occurs. The approach can for example be applied to feature extraction for pattern recognition, blind image deconvolution, sparse coding of images, etc. These applications are open problems for future work.

5. REFERENCES

[1] D. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l^1 minimization," *Proc. Nat. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, 2003.

[2] P.O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[3] F.J. Theis, K. Stadlthanner, and T. Tanaka, "First results on uniqueness of sparse non-negative matrix factorization," in *Proc. EUSIPCO 2005*, Antalya, Turkey, 2005.

[4] D.C. Youla and H. Webb, "Image restoration by the methods of convex projections. part i — theory," *IEEE Trans. Med. Imaging*, vol. MI, no. I, pp. 81–94, 1982.

[5] P.L. Combettes, "The foundations of set theoretic estimation," *Proc. IEEE*, vol. 81, pp. 182–208, 1993.