# Accelerated Search for Non–Negative Greedy Sparse Decomposition via Dimensionality Reduction

Konstantinos A. Voulgaris Mike E. Davies Mehrdad Yaghoobi
Institute for Digital Communications, the University of Edinburgh, EH9 3JL, UK
{Konstantinos.Voulgaris, Mike.Davies, myvaigha}@ed.ac.uk

***Abstract*— Non–negative signals form an important class of sparse signals. Many algorithms have already been proposed to recover such non-negative representations, where greedy and convex relaxed algorithms are among the most popular methods. One fast implementation is the FNNOMP algorithm that updates the non–negative coefficients in an iterative manner. Even though FNNOMP is a good approach when working on libraries of small size, the operational time of the algorithm grows significantly when the size of the library is large. This is mainly due to the selection step of the algorithm that relies on matrix vector multiplications. We here introduce the Embedded Nearest Neighbor (E–NN) algorithm which accelerates the search over large datasets while it is guaranteed to find the most correlated atoms. We then replace the selection step of FNNOMP by E–NN. Furthermore we introduce the Update Nearest Neighbor (U–NN) at the look up table of FNNOMP in order to assure the non–negativity criteria of FNNOMP. The results indicate that the proposed methodology can accelerate FNNOMP with a factor 4 on a real dataset of *Raman* Spectra and with a factor of 22 on a synthetic dataset.**

**Index Terms: Matching Pursuit, Orthogonal Matching Pursuit, Non-negative Sparse Approximations, Non-negative Least Square and Spectral Decomposition, Scalable algorithms, Dimensionality Reduction, Linear Embedding, Raman Spectroscopy**

## I. INTRODUCTION

Let the signal of interest be $y \in R^M$ and a dictionary of elements $\Phi \in R^{M \times N}$ be given. The linear sparse approximation can be formulated as finding the sparsest $x \in R^N$, $M < N$, i.e having the minimum number of non–zero elements, as follows:

$$y \approx \Phi x \tag{1}$$

The greedy sparse approximation algorithms are in general characterized by a low computational cost, suitable for real–time and large scale sparse approximations. The Orthogonal Matching Pursuit (OMP) [1],[2] algorithm is introduced, to find the best representation using selected atoms and approximate the sparse solution of the following problem:

$$\tilde{x} := \text{argmin}_{x_s} ||y - \Phi_s x_s||_2 \tag{2}$$

There are many applications for which the coefficient vectors are not only sparse, but they are also non–negative. Spectral and multi–spectral unmixing, [3],[4], microarray analysis [5] and Raman spectral deconvolution [6] are a few examples.

The original implementation of OMP has been modified in order to adopt the algorithm to the non–negativity setting for coefficients. Essentially the original minimization problem introduced in (2) is reformulated by adding a constraint that guarantees the non–negativity of the coefficients and takes the following form:

$$\begin{aligned} \tilde{x} := \quad & \text{argmin}_{x_s \geq 0} ||y - \Phi_s x_s||_2 \\ & ||x_s||_0 \leq j \end{aligned} \tag{3}$$

The authors in [7] introduced the Fast Non–Negative Orthogonal Matching Pursuit algorithm which is a greedy technique based on OMP suitable for real–time applications. Even though the implementation of this strategy may be straightforward when considering a dictionary with a relatively small number of atoms, this is not the case when working with a library that contains a significant number of atoms (i.e thousands). This is mainly due to the selection step of the algorithm which has a computational complexity of $\mathcal{O}(MN)$. Consequently the executional time of the algorithm will scale linearly along with the number of atoms in $\Phi$.

Essentially the selection of the best possible candidate within a normalized dictionary can be represented as the Nearest Neighbor Search (NNS): Given a set of points $P = \{p_1, p_2, \cdots, p_n\}$ in a metric space $X$ with distance function $d$, NNS is to efficiently answer queries for finding the closest point in $P$ to $q \in X$. There exist several data structures proposed to fulfill such task such as the *kd–tree* [8] or the *cover tree* [9]. Due to the curse of dimensionality, it is unlikely that there exist a general efficient solution to the exact *k*–NN problem [10]. Approximate algorithms have been proposed to overcome these impracticalities such as the *locality–sensitivity hashing* [11]. Although these algorithms do not guarantee the acquisition of the exact nearest neighbor they are fast and scalable.

Within this paper we introduce an Embedded Nearest Neighbor (E–NN) in order to reduce the computational cost of the selection step in the algorithm. Considering the dictionary $\Phi$, this means that we shrink the size of $M$ via dimensionality reduction. Given that the data are typically characterized by an intrinsic dimensionality, we are addressing E–NN as a practical framework that exploits the benefits of conducting the brute force search on the $K$–dimensional subspace compared to the $M$–dimensional original domain. Since it is expected that a mismatch in between the closest point from one domain to the other

978-1-7281-0503-1/19/$31.00 ©2019 IEEE

will occur, we introduce an update step on the algorithm to compensate the error and eventually acquire the exact NN. In that sense, we are considering E–NN as a bridge between the approximate to the exact solution for the brute force search.

## II. Linear Embeddings

In this section we introduce the guidelines for reducing the size of problem via a Linear Embedding. The standard notion regarding dimensionality reduction is that by having an input signal $y \in R^M$, the dimension of the signal is reduced via a linear operator $Q : R^M \rightarrow R^K$, with $K < M$, that embeds the input signal into the lower dimensional space. The projection of the signal $\hat{y}$ in $R^K$ is then computed as follows: $\hat{y} = Qy$.

Linear embedding is a standard approach in many applications where we seek for a low–dimensional representation of data living on a high–dimensional space. There exist different methods to perform the embedding, i.e principal component analysis (PCA) [12], random projections [13] etc. A common characteristic of these embeddings is that the relevant position between library elements is changed when the points are embedded from $R^M$ to $R^K$. In that sense, given a pair of elements $\phi_i, \phi_c \in R^M$ and their representations $\hat{\phi}_i, \hat{\phi}_c \in R^K$, we usually have: $d(\phi_i, \phi_c) \neq d(\hat{\phi}_i, \hat{\phi}_c)$. For an algorithm that searches for the Nearest Neighbor (NN) of $y$ in $\Phi$, this may lead to a situation in which $\text{NN}_M \neq \text{NN}_K$ where NN is the abbreviation for the Nearest Neighbor and $M, K$ corresponds to the dimensions for each domain. At this section we introduce the Embedded Nearest Neighbor (E–NN) algorithm that under a specific condition the search in the lower dimensional space eventually yields the nearest neighbor in the original domain. In that sense we are seeking an embedding that yields a minimum distortion from $R^M \rightarrow R^K$. This aspect of the problem can be addressed in terms of a reformulation of the Constructive Johnson–Lindenstrauss [14] introduced in (4) where $d(b,t) = ||b - t||_2$, where $b, t \in \mathcal{A} \subset R^M$. Let $Q$ distorts the distance for at most $\epsilon_{b,t}$. We then have:

$$\begin{aligned} (1 - \epsilon_{bt}) d(b,t) \leq d(\hat{b}, \hat{t}) &\leq (1 + \epsilon_{bt}) d(b,t) \\ d(z,t) - \epsilon_{bt} d(b,t) \leq d(\hat{b}, \hat{t}) &\leq d(b,t) + \epsilon_{bt} d(b,t) \\ d(b,t) - \delta \leq d(\hat{b}, \hat{t}) &\leq d(b,t) + \delta \end{aligned} \tag{4}$$

where,

$$\delta = \max_{b,t \in \mathcal{A} \subset R^M} \epsilon_{bt} d(b,t), \tag{5}$$

***Lemma*** 1.$\forall b, t \in \mathcal{A}$ with a $\delta$ coming from (5) and $\forall y \notin \mathcal{A}$ with $\max\ \epsilon_{yt} d(y,t) \leq \delta$, the E–NN introduced in Algorithm 1 guarantees the acquisition of the exact NN.

*Proof:* Considering three points $y, b, t$ where $d(y,b) \leq d(y,t)$. Then there exist 4 characteristic cases for pairwise distances.

- Both distances shrink: $d(\hat{y}, \hat{b}) \leq d(y,b)$, $d(\hat{y}, \hat{t}) \leq d(y,t)$. Then by incorporating (4) :
  $d(\hat{y}, \hat{b}) \leq d(y,b) \leq d(y,t) + \delta$.
- Both distances stretch. Then from (4) we have:
  $d(\hat{y}, \hat{b}) - \delta \leq d(y,b) \leq d(y,t)$
  $\Rightarrow d(\hat{y}, \hat{b}) \leq d(y,t) + \delta$
- $d(\hat{y}, \hat{b})$ stretches: $d(y,b) \leq d(\hat{y}, \hat{b})$, $d(\hat{y}, \hat{t})$ shrinks: $d(y,t) \leq d(\hat{y}, \hat{t}) + \delta$. Then it follows:
  $d(\hat{y}, \hat{b}) \leq d(y,b) + \delta \leq d(y,t) + \delta$.
- $d(\hat{y}, \hat{b})$ shrinks: $d(\hat{y}, \hat{b}) \leq d(y,b)$, $d(\hat{y}, \hat{t})$ stretches: $d(y,t) \leq d(\hat{y}, \hat{t})$. Then:
  $d(\hat{y}, \hat{b}) \leq d(y,b) \leq d(y,t)$. □

The analysis provided by proof of the *Lemma* simply states that in cases where $NN_M \neq NN_K$, assuming that $b = NN_M$ and $t = NN_K$ , then $d(\hat{y}, \hat{NN}_M) \leq d(y, NN_K) + \delta$.

The complexity of the E–NN introduced in **Algorithm 1** varies over steps 2–4 of the algorithm. At step 2 the input signal $y \in R^M$ is embedded in $R^K$ via the linear operator $Q \in R^{K \times M}$. Hence the complexity of step 2 is $\mathcal{O}(KM)$. At step 3 we conduct a number of $N$ distance computations over $M$–dimensional vectors. The computational cost of the corresponding operations is $\mathcal{O}(KN)$. Finally, at the last of the the algorithm we perform a number of $|S|$ distance computations on the original space $R^M$. The computational cost of the step is $\mathcal{O}(|S|M)$.

As it can be derived from the analysis there are two critical parameters to benefit from the brute force search in the lower dimensional space. The intrinsic dimensionality of the dataset expressed by $K$ and the cardinality of $S$ on the update step which depends on $\delta$.

Essentially we are seeking for an embedding $Q$:

$$\delta = \min_{Q} \max_{i,c} \epsilon_{ic} d(\phi_i, \phi_c), \forall i, c \in \Phi. \tag{6}$$

The most common approach to construct a dimension reduction is principal component analysis (PCA). A key advantage of PCA is that it is computationally efficient. The embedding to the $K$–dimensional space is simply performed by taking the $K$ dominant eigenvectors of the data covariance matrix. The main drawback of PCA though is that it distorts pairwise distances arbitrarily. In that sense the distance distortion may be significantly larger from the one pair of points to the other.

An alternative to PCA is the approach of *random projections*. According to the Johnson–Lindenstrauss lemma , given any point cloud $\Omega$ in $R^M$, there exists an embedding $Q$ of dimension $K = \mathcal{O}(\log|\Omega|)$ with minimal distortion of the $\binom{|\Omega|}{2}$ pairwise distances between the $|\Omega|$ points. This linear embedding is easy to implement in practice. We simply construct a matrix $Q \in R^{K \times M}$ with elements drawn randomly from a certain probability distribution. The authors in [15], introduced a deterministic framework, called *NuMax*,

**Algorithm 1** Embedded NN (E–NN)

```
1: Input: $\Phi, \hat{\Phi}, Q, y$.
2: $\hat{y} = Qy$.
3: Form set S=$\{i : d(\hat{\phi}_i, \hat{y}) \leq d(y, \text{NN}_K) + \delta\}, \forall \hat{\phi}_i \in \hat{\Phi}$.
4: return $arg\ min_{i \in S}\ d(y, \Phi_i)$ .
```

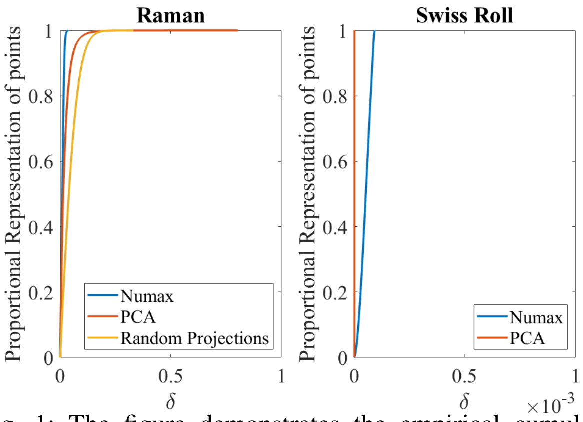


Fig. 1: The figure demonstrates the empirical cumulative distribution function (CDF) of $\delta$ over $\Phi$. The distortion on $\Phi$ introduced by random embeddings into Swiss Roll is much larger than *Numax* and PCA hence it is not demonstrated.

that constructs *linear*, *near*–isometric embeddings for data that live in a high–dimensional space. Given a set of training points $\Phi \in R^M$, the authors consider a *secant set* $S(\Phi)$ consisting of all pairwise difference vectors of $\Phi$ that lie on the unit sphere. The problem is formulated as an affine rank minimization problem to construct $Q$ such that the norms of all vectors in $S(\Phi)$ are preserved up to a distortion parameter.

We aim to solve the problem introduced in Equation (6) empirically for library *Raman* spectra with $M = 1507$ and $N = 4041$ [16] and a library of Swiss Roll data [17] which is a synthetic machine Learning dataset of points that lie on a 2–D manifold but embedded in $R^{1507}$. We found that the minimization problem introduced by the *NuMax* algorithm yields a matrix $Q \in R^{K,M}$ with $K = 172$ for *Raman* while for the Swiss Roll case $K = 3$. Then we construct $Q$ for PCA and *random projections* by setting $K = 172$ and $K = 3$ accordingly such that we can investigate which method serves the purpose for $R^K$.

The performance for each method is evaluated with respect to the error distortion function $\delta(\phi_i, \phi_c)$ as follows:

$$\delta(\phi_i, \phi_c) = |d(\phi_i, \phi_c) - d(\hat{\phi}_i, \hat{\phi}_c)|. \qquad (7)$$

The obtained results are demonstrated in figure 1.

### A. The case of mixture $y$

Within our framework we set $\delta$ with respect to the knowledge derived from elements that belong to an available library $\Phi$. The case of mixtures $y$ is slightly different. In particular, each $y$ with sparsity (number of contributing atoms) up to $j$ is formulated as a linear combination of $\phi_i \in \Phi$ as follows: $y = \sum_{w=1}^{j} a_w \phi_w$.

This essentially means that there is not any particular knowledge regarding $\delta(\phi_i, y)$. Hence, an obvious question is whether $y$ is consistent with the choice of $\delta$. Given that according to the results introduced in figure 1 the $Q$ obtained by the *NuMax* algorithm yields the best results we perform a simulation study for $y$ over a sparsity level up to 5 which is the maximum sparsity of the signals for the applications we focus on. The distortion is then evaluated according to the error distortion function introduced in (7) with $y$ taking the place of $\phi_i$ and $\hat{y}$ the place of $\hat{\phi}_i$ accordingly. For each $j$ we generate a set of mixtures $\mathcal{Y} = \{y_m\}_{m=1}^{L}$ via 10000 (denoted as $L$) Monte Carlo simulations. The obtained results are demonstrated in figure 2. Note that $a_w \sim U[0, 1]$ and $||y||_2 = ||\hat{y}||_2 = 1$.

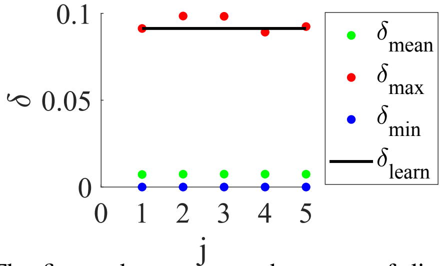


Fig. 2: The figure demonstrates the range of distortion over sparsity.$\delta_{mean}(j) = \frac{1}{|\mathcal{Y}||\Phi|}(\sum_{y_m \in \mathcal{Y}} \sum_{\phi_i \in \Phi} \delta(\phi_i, y_m))$, $\delta_{max}(j) = \max\ \delta(\phi_i, y_m)$, $\delta_{min}(j) = \min\ \delta(\phi_i, y_m)$

The results indicate that $\delta$ flunctuates around $\delta_{learn}$. We empirically observe that the maximum pairwise distortion $\forall \phi_i \in \Phi$ (denoted as $\delta_{learn}$) exceeded only 0.003% over $L$. Even in these cases, the algorithm acquires the exact NN. This is happening due to the fact that the pairwise distortion is on average much lower than $\delta_{max}$ and a lower $\delta$ hence serves the purpose.

## III. Search FNNOMP

In this section we introduce an update on the structure of FNNOMP, as introduced in Algorithm 2, with respect to the algorithm introduced in Algorithm 1. The first change in the structure takes place in the selection step of FNNOMP [6, pp2] where we place E–NN. A common phenomenon in sparse non–negative decomposition is that a selected atom may be rejected by the non–negativity criteria introduced in Table I and with respect to equation (8). Consequently, we need to modify the content in Table I compared to the original FNNOMP version. A key aspect of the changes is the insertion of the U–NN algorithm, as introduced in Algorithm 3, such that E–NN adopts on the non–negativity setting. All the changes in the overall structure of FNNOMP are highlighted with red.

In practice Update NN can be addressed as a next NN Algorithm. In that sense anytime that the NN acquired by E–NN and indexed by $\mu$ is rejected by the criteria introduced

| if | then |
|---|---|
| $0 < z \leq z^t, z > z^c$ | $z_{j+1} \leftarrow z$, Terminate |
| $0 < z \leq z^t, z \leq z^c$ | $z_{j+1} \leftarrow z^c, p \leftarrow p^c$, Terminate |
| $z > z^c \geq z^t$ | $p = p + 1$, $\mu \leftarrow$U–NN |
| $z \geq z^c > z^t$ | $z_{j+1} \leftarrow z^c, p \leftarrow p^c$, Terminate |
| $z > z^t > z^c$ | $z^c \leftarrow z^t, p^c \leftarrow p$, $\mu \leftarrow$U–NN |
| $z < 0$ | Terminate |

TABLE I

$$z_{j+1} \leq z^t = \begin{cases} \min\limits_{\gamma_i<0} \frac{|x_i|}{|\gamma_i|} & \exists i, \gamma_i \leq 0 \\ \infty, \ \text{otherwise} & \end{cases} \tag{8}$$

**Algorithm 2** E–NN on FNNOMP

```
1: Initialization: $s = z_0 = \emptyset, j = 0, r_0 = y$.
2: while $j < K\&$ max$(\Phi^T r_k > 0)$.
     i   $\mu \leftarrow$ Embedded–NN.
     ii  $p \leftarrow 1$.
     iii $p^c \leftarrow \mu$.
     iv  $z^c = 0$
     v   while $\sim$ Terminate & $p < N$
     vi      $z_t$ from (8).
     vii     z$\leftarrow \psi_\mu^T r_k$: $\psi_\mu = \frac{q}{||q||_2}$, $q = (I - \Psi\Psi^T)\phi_\mu$
     viii    Update based on Table I
     ix  end while
     x   $s = s \cup \mu$.
     xi  Update $\Psi$ and R$^{-1}$
     xii $z_{j+1} \leftarrow [z_j, z_{j+1}]$
     xiii $r_{j+1} =\leftarrow r_j - z_{j+1}\psi_{j+1}$
     xiv $j \leftarrow j + 1$
3: end while .
4: output: $x|_s \leftarrow R^{-1} z_j$
```

**Algorithm 3** Update NN

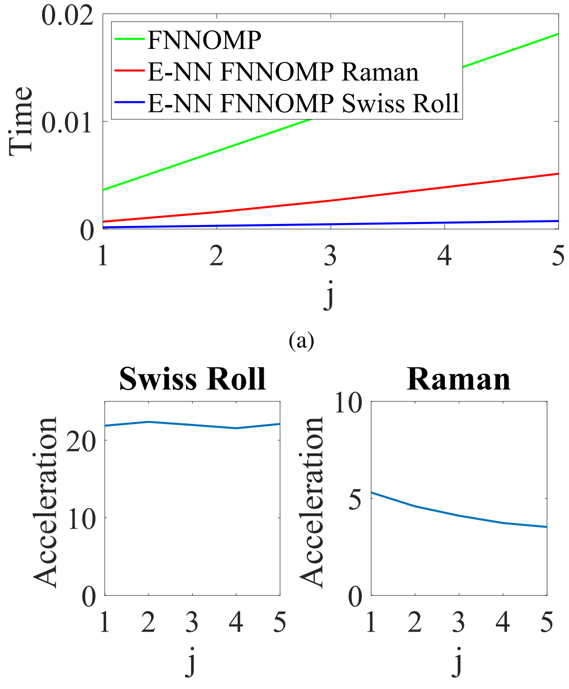
```
1: Input: $\Phi, \hat{\Phi}, y, \mu, S$.
2: $S = S - \mu$.
3: Form set $S' = \{i : \ d(\hat{\phi}_i, \hat{y}) \leq \min d(y, NN_K) + \delta\}$.
4: Form set $S'' = S' - S$.
5: return $arg\ min_{\phi_i \in S \cup S''}\ d(y, S) \cup d(y, S'')$ .
```

in Table II, the task of U–NN is the acquisition of the next closest point to $y$. To do as such we need to reject $\mu$ from $S$. This is done in step 2 of the algorithm.

The implementation of E–NN provides U–NN with the full set of distance measurements in $R^K$ and a number of distance measurements equal to $|S| - 1$ in $R^K$ since $\mu$ is rejected in Step 2. Hence no additional distance computation is conducted in Step 3 of U–NN but a simple logical comparison that yields a new set of indexes. Given though that for some of these indexes the distance in $R^M$ is already available from E–NN we introduce Step 4 in order to avoid the recomputation. We then compute the distances for $\phi_i \in S''$ and then we perform a comparison with the measurements of $\phi_i \in S$ in order to find the next NN in $R^M$.

## IV. Results

In this section we evaluate the performance of the proposed algorithm with respect to FNNOMP. Based on the results introduced in figure 1 we select the $Q$ obtained by the *NuMax* algorithm as the linear operator that projects offline the dictionary $\Phi$ and online the mixture $y$ in $R^K$ while for the Swiss Roll we select the $Q$ obtained by PCA. We set $\delta = 0.09$ for the *Raman* library and $\delta = 0$ for the *Swiss Roll*. We then generate signal mixtures of varying sparsity $j$ from the elements in $\Phi$.



(a)

(b)

Fig. 3: Top of the figure: Elapsed time for each of the algorithms. Bottom: Acceleration over sparsity. Where Acceleration$(j) = \frac{\text{Time FNNOMP}(j)}{\text{Time E-NN FNNOMP}(j)}$.

The obtained results demonstrated in figure 3 show that E–NN FNNOMP is generally faster than FNNOMP. The overall performance of the algorithm though decays over sparsity for the *Raman* spectra. Given that the computational cost at steps 1 and 2 of E–NN, the only parameter related to the complexity that may vary over $j$ is $|S|$. In order to obtain a better understanding regarding that issue we demonstrate the average number of points per iteration of the algorithm in figure 4. As can be seen from the results, the task of signal decomposition in the lower dimensional space becomes more difficult while sparsity increases. This is obviously not the case for the search in Swiss Roll. Essentially the acceleration factor remains constant. This happens because $\delta = 0$ hence

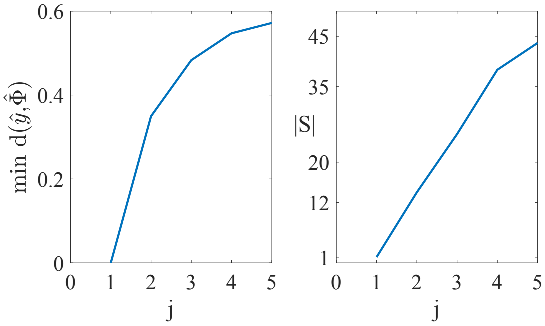


Fig. 4: Average points in Step 4 of E–NN over sparsity.

the update step of E–NN is unnecessary. This means that in practice we compare the implementation of FNNOMP into different domains. This phenomenon may occur when all of the points that lie in $R^M$ in reality they lie in the same subspace $R^K$. As it can be seen from the *Raman* library though this is not something to be expected in a realistic setting.

## V. CONCLUSIONS

We here presented E–NN which is a novel algorithm aiming to accelerate the NN sparse decomposition using a big library. The obtained results indicate that the E–NN FNNOMP outperforms FNNOMP. The current approach of E–NN leverages the underlying sparsity of $\Phi$ via a linear embedding of $\Phi$ on $R^K$. However many datasets contain essential nonlinear structures that are invisible to linear techniques [18]. For example, the *Swiss Roll* dataset consists of 3D points that form a 2D manifold. PCA and *NuMax* reveal the underlying linear subspace that our artificial dataset lives, but they cannot benefit from the underlying geometrical structure of this space. Exploring the nonlinear dimensionality reduction for acceleration of nonnegative sparse approximations has been left for the future work.

## Acknowledgment

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant numbers EP/S000631/1 and EP/K014277/1 and the MOD University Defence Research Collaboration (UDRC) in Signal Processing.

## References

[1] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,“ in *Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40–44.

[2] B.K. Natarajan, "Sparse approximate solutions to linear systems,“ *SIAM Journal of Comput*, vol. 24, no. 2, pp. 227–234, 1995.

[3] MD Iordache, JM Bioucas-Dias, and A Plaza, "Sparse unmixing of hyperspectral data,“ *Geoscience and Remote Sensing, IEEE Transactions. on*, vol. 49, no. 6, pp. 2014–2039, 2011.

[4] Y Qian, S Jia, J Zhou, and A Robles-Kelly, "Hyperspectral unmixing via sparsity-constrained nonnegative matrix factorization,“ *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 49, no. 11, pp. 4282–4297, 2011.

[5] H Kim and H Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis,“ *Bioinformatics*, vol. 23, no. 12, pp. 14951502, 2007.

[6] D Wu, M Yaghoobi, SI Kelly, ME Davies, and R Clewes, "A sparse regularized model for raman spectral analysis,“ in *Sensor Signal Processing for Defence*, Edinburgh, 2014.

[7] Yaghoobi Vaighan, M, Wu, D & Davies, M 2015, "Fast Non-Negative Orthogonal Matching Pursuit“ *IEEE Signal Processing Letters*, vol. 22,no. 9, 2015.

[8] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time“, *ACM Transactions on Mathematical Software*, 3(3):209-226, September 1977.

[9] Alina Beygelzimer, Sham Kakade and John Langford, "Cover Trees for nearest Neighbor“,in *Proc. International Conference on Machine Learning (ICML)*, 2006.

[10] Leonid Boytsov, "Efficient and Accurate Non-Metric k-NN Search with Applications to Text Matching“, *PhD Thesis*, 2018.

[11] Piotr Indyk, Rajeev Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality“, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing* ,(1998).

[12] B. Moore "Principal component analysis in linear systems: Controllability, observability, and model reduction“ *IEEE Trans. Autom. Control*. vol 26 no.1 pp. 17-32,1981.

[13] D. Achlioptas,"Database–friendly random projections“ in *Proc. Symp. Principles of Database Syst. (PODS)*, Santa Barbara, CA,USA, May 2001.

[14] Jiri Matusek "On variants of the Johnson–Lindenstrauss lemma“ *Random Structures and Algorithms, 33, 142156.*.

[15] Chinmay Hegde, Aswin C. Sankaranarayanan, Wotao Yin, and Richard G. Baraniuk, "NuMax: A Convex Approach for Learning Near-Isometric Linear Embeddings“, *IEEE Transactions on Signal Processing* vol. 63, 2015.

[16] `https://www.stjapan.de/`

[17] S. Marshland " Machine Learning: An Algorithmic Perspective“, Chapter 10, 2009.

[18] J.B Tenenbaum, V. de Silva, J.C Langford "A Global Geometric Framework for Nonlinear Dimensionality Reduction“, *Science 290 (5500): 2319-2323*,22 December 2000.