$UoI\text{-}NMF_{cluster}$: A Robust Nonnegative Matrix Factorization Algorithm for Improved Parts-Based Decomposition and Reconstruction of Noisy Data

Shashanka Ubaru* Kesheng Wu† Kristofer E. Bouchard ‡

October 3, 2017

Abstract

With the ever growing collection of large volumes of scientific data, development of interpretable machine learning tools to analyze such data is becoming more important. However, robust, interpretable machine learning tools are lacking, threatening extraction of scientific insight and discovery. Nonnegative Matrix Factorization (NMF) algorithms decompose an $m \times n$ nonnegative data matrix A into a $k \times n$ basis matrix H and an $m \times k$ weight matrix W, such that $A \approx WH$, where k is the desired rank. In this paper, we present a novel two stage algorithm, $UoI\text{-}NMF_{cluster}$ for NMF, which is based on three innovations: (i) completely separate bases (H) learning from weight (W) estimation, (ii) learn bases (H) by clustering NMF results across bootstrap resamples of the data, and (iii) use the recently introduced Union of Intersections (UoI) framework to estimate ultra-sparse weights (W) that maximize data reconstruction accuracy. We deploy our algorithm on various synthetic and scientific data to illustrate its performance, with a focus on neuroscience data. Compared to other NMF algorithms, $UoI\text{-}NMF_{cluster}$ yields: a) more accurate parts-based decompositions of noisy data, b) a sparse and accurate weight matrix, and c) high-accuracy reconstructions of the de-noised data. Together, these improvements enhance the performance and interpretability of NMF application to noisy data, and suggest similar approaches may benefit other matrix decomposition algorithms.

1 Introduction

In many scientific fields, the development of new sensing and imaging technologies has resulted in generation of large volumes of data. These large datasets bring with them opportunities of new discoveries and insights into the fundamentals of nature. In order to realize such opportunities, development of novel machine learning and statistical data analysis methods is necessary. Statistical-machine learning algorithms for scientific data should satisfy the bi-criteria of returning results that are simultaneously predictive and interpretable. By predictive, we mean that it can predict (e.g., reconstruct) the data with high accuracy; by interpretable, we mean that the results give insight into the (bio)-physical processes that generated the data. Interpretability usually entails the sparse selection and accurate estimation of a small number of physically meaningful features of the data. However, these bi-criteria are often at odds, and methods that robustly (few assumptions on the data/noise) achieve both are lacking. Such methods could provide insights into natural phenomena through the extraction of physically or biologically interpretable models.

Dimensionality reduction and low rank approximations/ decompositions are popular tools used in many applications to analyze high dimensional data. However, methods such as principal component analysis (PCA) often yield uninterpretable results, as the eigenvectors can be additive combinations of up to all the data features. Alternate matrix decomposition methods such as Nonnegative Matrix Factorization (NMF) and CUR decomposition have been shown

*Department of Computer Science and Engineering, University of Minnesota at Twin Cities, MN USA. Email: ubaru001@umn.edu

†Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA, USA Email: kwu@lbl.gov

‡Biological Systems and Engineering Division, Lawrence Berkeley National Laboratory; Computational Research Division, Lawrence Berkeley National Laboratory; Redwood Center for Theoretical Neuroscience, University of California at Berkeley; Kavli Institute for Fundamental Neuroscience, University of California at San Francisco; Email: kebouchard@lbl.gov

to perform well in some scientific applications [25, 26]. In this work, we develop a novel NMF algorithm and deploy it on scientific data from neuroscience and analytical chemistry to extract interpretable features.

Since its popularization, NMF [17] has been used in many applications for obtaining interpretable decompositions of data. Given a matrix $A \in \mathbb{R}_+^{m\times n}$ ($\mathbb{R}_+$ represents the positive orthant), where each row of A corresponds to a data point in $\mathbb{R}_+^n$, and a rank k, the problem of NMF is to compute the matrices $W \in \mathbb{R}_+^{m\times k}$ and $H \in \mathbb{R}_+^{k\times n}$, such that $A \approx WH$. This problem is generally posed as a non-convex optimization problem,

$$\min_{W\geq 0, H\geq 0} \|A - WH\|_F. \tag{1}$$

Here, the rows of H form the basis of the objects (say images), and the rows of W are the encoding of the basis in A. Since both W and H are nonnegative, NMF sometimes gives more interpretable parts based decompositions, with the intuitive notion of "combining parts to form a whole" [17].

Several algorithms to solve the NMF problem have been developed to achieve various objectives, such as more interpretable results, sparser solutions, unique solutions, etc. Sparse NMF [12, 13, 14] and convolutional NMF [2, 22] are two popular variants of NMF. It has been claimed that NMF implicitly yields a sparse representation of the data. However, in order to obtain explicit sparse NMF solutions, the following objective function is popularly used:

$$\min_{W\geq 0, H\geq 0} \frac{1}{2}\left[\|A - WH\|_F^2 + \lambda_1\|W\|_F^2 + \lambda_2 \sum_{j=1}^{k} \|H(j,:)\|_p^2\right]$$

with $p = \{0, 1\}$ (promotes sparsity), λ_1 and λ_2 are regularization parameters. Algorithms to solve this problem are discussed in [12, 13, 14].

The uniqueness of the solutions obtained by NMF was first discussed in [7], using a geometric interpretation of NMF with simplicial cones. The separability of data was shown to be the key required property for unique solutions for NMF. Uniqueness of NMF is also discussed in [15, 10, 1] and many other works. Article [9] showed that subset separability of data is sufficient for obtaining unique solutions for NMF. We discuss more on the uniqueness of NMF in sec. 3.1. Recently, [4] discussed NMF under heavy noise for topic modeling. That article proposes a new NMF algorithm for noisy data and shows uniqueness of the solution obtained.

However, almost all NMF algorithms with theoretical guarantees make strong assumptions on the type of input data (separable and subset separable conditions) and on the noise. Such assumptions are hard to check and these algorithms may not be practically useful for scientific data, where the data need not be separable and the noise distribution is likely unknown. Moreover, most existing algorithms fail to give stable interpretable bases when the input data has high noise. In particular, when the data has noise, solving the non-convex optimization is problematic, as different starting points yield different results, and thus unstable bases. So, while these methods minimize an objective function related to the reconstruction error, and hence sometimes give good prediction quality, they generally do not yield accurate parts based decompositions of the data generation process. This clearly has negative consequences for interpretability, an admittedly not completely-well-defined property of algorithms that is crucial in many scientific applications. Thus, alternate approaches need to be explored.

Our Contribution In this paper, we present a novel, noise-robust NMF algorithm ($UoI\text{-}NMF_{cluster}$) that gives more accurate parts based decompositions and sparser weight matrices with improved reconstruction of denoised data. $UoI\text{-}NMF_{cluster}$ is inspired by the Union of Intersections (UoI) framework [5], and incorporates three innovations: (i) completely separate bases (H) learning from weight (W) estimation, (ii) learn bases (H) by clustering NMF results across bootstrap resamples of the data, and (iii) use UoI to estimate ultra-sparse weights (W) to maximize data reconstruction accuracy.

$UoI\text{-}NMF_{cluster}$ is a two stage algorithm which computes sets of bases over bootstrap resamples of the data using a standard NMF algorithm, and clusters the bases to learn the best stable and uncorrelated set of k bases. The algorithm then directly uses UoI applied to the non-negative least squares problem to compute a sparse weight matrix that best reconstructs the original input data given the selected bases. Using this two stage process, our method ensembles different models (bases), selects the stable bases using clustering, and achieves sparse, low-variance solutions (weights W in our case) without imposing a prior, see [5] for the discussion.

The goal of $UoI\text{-}NMF_{cluster}$ is not to solve a single optimization problem to obtain a single NMF, but to extract stable bases and learn sparse weights that map these bases to the data with high accuracy. Our algorithm has some resemblance to popular *ensemble* methods [6], which improve prediction accuracy by combining different models (parameter estimates). However, ensemble methods often include more features (expand the feature space) to predict the response variables, which make them hard to interpret [5]. $UoI\text{-}NMF_{cluster}$ generates a bootstrapped ensemble of potential bases, and uses clustering to extract uncorrelated bases that are stable to the bootstrap procedure. It then selects few bases (i.e., sparse model selection) and uses bagging to estimate their contributions (W) without imposing an explicit prior on the weight distribution, thus giving low-bias and low-variance estimates. Through this approach, we find that $UoI\text{-}NMF_{cluster}$ learns interpretable and predictive structure from complex, noisy data.

In the following, we discuss the uniqueness of the bases learned by clustering NMF bases utilizing geometric interpretations, see section 3.1. We show how a stability criterion proposed in [25] can be naturally incorporated into our bootstrap resampling approach to select the best nonnegative rank k. Numerical experiments with various synthetic and scientific data illustrate the performance of the proposed algorithm relative to other approaches.

2 Preliminaries

In this section, we discuss the conceptual framework that lays the foundation for $UoI\text{-}NMF_{cluster}$.

Union of Intersections Union of Intersections (UoI) is a recently introduced flexible, modular, and scalable framework for statistical-machine learning problems [5]. The core concept of the UoI framework is to separate feature selection from feature estimation, and use bootstrap resampling to determine stable features and estimate the parameter values for those features to maximize predictive accuracy. In UoI-based methods, model selection is first performed through intersection (compressive) operations which induce sparsity, followed by model estimation through union (expansive) operations which reduces the variance of estimates.

For example, consider the regression problem with ℓ_1 regularization: Given the data $(Y_1, X_1), \ldots, (Y_n, X_n)$, with univariate response Y and p-dimensional predictor variable X, we wish to minimize

$$L(\beta, \lambda) = \|Y - X\beta\|_2^2 + \lambda\|\beta\|_1.$$

In the UoI_{Lasso} algorithm, we (1) calculate model supports (S_j) (location of nonzero entries of β) using an intersection operation across different bootstrap resamples of the data for a range of regularization parameters (λ: increases in λ shrink all values of β towards 0), constructing a family of model supports [$\mathbf{S} : S_j \subset S_{j-k}$ for $\Delta\lambda = \lambda_j - \lambda_{j-k}$ sufficiently large]; (2) combine the pure model selection (obtained from the intersection operation) with model estimation using a union operation to obtain better selection, estimation and prediction accuracy. Further details on the UoI_{Lasso} algorithm can be found in [5]. Here, we adapt this framework to the NMF problem by separating the feature (i.e., bases: H) learning/selection from the weight estimation (W). Additionally, we use UoI for solving a non-negative least-squares problem to determine sparse weight matrix W once the best basis H is computed.

NMF algorithms Several optimization algorithms have been proposed for solving the NMF problem (1). For example, the multiplicative algorithm [18], alternating least squares (ALS) [3], Projected gradient [19], and alternating direction method of multipliers (ADMM) [24] methods. Polynomial time algorithms with provable error bounds have also been developed, but require separability assumption and certain assumptions on noise [1, 9, 4]. For a comprehensive survey on NMF algorithms and applications, we refer to [11]. In almost all of these NMF algorithms, the basis matrix H and the coefficient matrix W are typically updated in an alternating fashion. This contrasts with our approach to first learn the bases, and then estimate the weights to maximize reconstruction accuracy (note that our goal is not to solve the optimization problem, but to extract stable interpretable representations).

The Euclidean distance minimization problem is unnatural for nonnegative data, since Euclidean distance assumes Gaussian type distribution, which is unlikely to be the case with nonnegative data. Hence, a KL-divergence type error metric, such as relative entropy $D(A\|B)$, where $B = WH$, given by,

$$D(A\|B) = \sum_{ij} \left(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij} \right)$$

has been proposed by [18], where a multiplicative first order method to solve the new KL-divergence optimization problem is provided (and which we will use). The new optimization problem can be viewed as the minimization of entropy instead of energy, which enhances the sparsity of the solution.

Stability Criterion Similar to approaches for clustering (*'Consensus Clustering'*), a stability-driven model selection criterion was recently proposed in [25] for NMF. For stability measurement and model selection, the following dissimilarity measure based on the cross correlation matrix C between two basis matrix H and H' is used:

$$diss(H,H') = \frac{1}{2k}\left(2k - \sum_{j=1}^{k}\max_i C_{ij} - \sum_{i=1}^{k}\max_j C_{ij}\right). \tag{2}$$

The idea in [25] is to compute multiple sets of bases H using NMF with different initial conditions and a range of rank k. Then compute the above dissimilarity measure for each pair of bases (H, H') and each rank k, and choose the rank k that achieves lowest instability (sum of $diss$) as the optimal rank. Thus, [25] innovated the selection of k based on the stability of bases from different initializations of a basic NMF algorithm, but returns the results of the basic NMF with best reconstruction accuracy for the selected k. However, as discussed below, when the data is noisy, it is unlikely that any single NMF result will contain the actual parts-based decomposition of the data (see Supplement for examples). In contrast, $UoI\text{-}NMF_{cluster}$ uses bootstrap resampling to form a large set of potential NMF bases, and cluster these bases so as to find the stable decomposition (bases) of the data, and then use UoI to estimate their contributions in data reconstruction.

3 UoI-NMFcluster

Notation The input matrix A is assumed to be of size $m \times n$ with m data points in $\mathbb{R}^n$, is decomposed into a basis matrix $H \in \mathbb{R}^{k\times n}$ with k rows and a weight matrix $W \in \mathbb{R}^{m\times k}$. We denote the output of $UoI\text{-}NMF_{cluster}$ by $\hat{H}, \hat{W}$ of best rank $\hat{k}$. The different sets of NMFs obtained for different bootstraps are denoted by the pairs $\{W_i, H_i\}_{i=1}^{B_1}$, where B_1 is the number of bootstrap samples used. The set of integers $1, \ldots, n$ is denoted by $[n]$. A matrix that contains the indices of the nonzero entries of W is denoted by W_{idx}. For a given rank k_1, the matrix where the basis matrices $\{H_i\}_{i=1}^{B_1}$ are stacked up is denoted by $\tilde{H}^{(k_1)}$.

The $UoI\text{-}NMF_{cluster}$ algorithm is as follows:

Bases Learning We compute the matrices H_i and W_i for different bootstrap samples of the data $i = 1, \ldots, B_1$, and for different ranks k using a standard NMF algorithm. The multiplicative update algorithm [18] for the KL divergence error metric gave us the best results (any other NMF algorithm can also be used). The next step is to learn the best basis matrix $\hat{H}$ from all the sets of bases $\{H_i\}_{i=1}^{B_1}$ learned over different bootstraps. The objective is to learn a set of bases that are stable parts based decomposition of the data.

We make the observation that bases which are stable, i.e., similar bases (near duplicates) that appear from different bootstrap samples are individual parts of the data, and are close to each other spatially (are dense points in the spatial distributions). Also, different parts of the data are dissimilar and must be apart from each other spatially. Intuitively, one part should be different than other parts. That is, the dense clusters formed by similar bases must be well separated. This is indeed true for separable data/NMF where the bases are separable (see section 3.1 for details). The noisy or spurious bases learned will be different for each bootstrap samples and these are typically spread out spatially. Hence, in order to learn a stable parts based decomposition of the data, i.e., extract the stable (similar) bases from the set of bases learned over different bootstraps, and ignore the noisy and spurious bases, we employ the popular robust density based clustering algorithm called DBSCAN (Density Based Spatial Clustering of Applications with Noise) [8].

We cluster the $k \cdot B_1$ bases learned over different bootstraps using the DBSCAN algorithm. The DBSCAN algorithm has two parameters, namely, the threshold Eps and the least minimum number of points per cluster ($MinPts$). We choose $MinPts \approx B_1/2$ because the stable bases should be learned for at-least half of the bootstrap samples. The threshold Eps can be chosen using the strategy proposed in [8]. The algorithm naturally clusters spatially dense points into individual clusters, hence, similar (stable) parts based bases which are spatially dense, are grouped into

Algorithm 1 UoI-$NMF_{cluster}$

```
Input: Data $A \in \mathbb{R}_+^{m \times n}$, minimum and maximum ranks $k_{\min}$, $k_{\max}$, and number of bootstrap resamples, $B_1, B_2$.
Output: $\hat{W} \in \mathbb{R}_+^{m \times \hat{k}}$ and $\hat{H} \in \mathbb{R}_+^{\hat{k} \times n}$.
1. Bases Learning and Selection
for $k_1 = k_{\min}$ to $k_{\max}$. do
  for $i = 1$ to $B_1$. do
    i) Generate $r_{id} \in [n]$ random indices.
    ii) $[H_i, W_i] = NMF_{KL}(A(r_{id}, :), k_1)$.
    iii) $R_{id}(:, i) = r_{id}$; $\tilde{H}^{(k_1)} = [\tilde{H}^{(k_1)}; H_i]$.
  end for
end for
If $(k_{\max} - k_{\min}) > 0$, Compute $diss(H_i^{(k_1)}, H_j^{(k_1)})$ and $\Gamma(k_1)$ and choose the best rank $\hat{k}$.
1a) Choose the best set of bases
1. Cluster the stacked matrix $\tilde{H}^{(\hat{k})}$ using DBSCAN.
2. Set centers of the clusters as new best bases set $\hat{H}$.
1b) Update weights and intersection of supports
Recompute $\{W_i\}_{i=1}^{B_1}$ wrt. $\hat{H}$ using NNLS.
for $l = 1$ to $n$ do
  $[r, c] = find(R_{id} = l)$.
  For all $r$, intersect the support of rows of $W_{[r]}$ and save in $W_{idx}$.
end for
2. Weight Estimation
for $i = 1$ to $B_2$ do
  Generate new $r'_{id} \in [n]$ random indices.
  for $l = 1$ to $LEN(r'_{id})$. do
    $w_{idx} = W_{idx}(r'_{id}(l))$;
    Compute $w_{idx}$ entries of $\{W_i\}_{i=1}^{B_1}$ using NNLS.
  end for
end for
$\hat{W} = $ entrywise-mean$(\{W_i\}_{i=1}^{B_1})$.
```

different clusters. The algorithm leaves out all noisy points (points not within the Eps-neighborhood of a cluster) without assigning them to a group. Therefore, the clusters we obtain for DBSCAN have only similar (stable) parts based bases. We choose the centers of these clusters as the best (stable) bases $\hat{H}$.

The best rank k can be computed using the dissimilarity measure given in (2) proposed in [25]. For each rank k, we can compute $diss(H_i, H_j)$ for each pair of bases learned over different bootstraps. Next, we compute the discrepancy of all B_1 bases as:

$$\Gamma(k) = \frac{2}{B_1(B_1 - 1)} \sum_{1 \le i \le j \le s} diss(H_i, H_j).$$

Then, select the best k that achieves a small $\Gamma(k)$.

Bases Selection and Weight Estimation Once the best bases $\hat{H}$ is learned, we next update/recompute the weight matrices $\{W_i\}_{i=1}^{B_1}$ based on $\hat{H}$. We use the same UoI strategy as in UoI_{Lasso} (UoI for ℓ_1 regression described above) for intersecting the supports (intersect the location of nonzeros) of each rows of the weights based on new $\{W_i\}_{i=1}^{B_1}$'s estimated over bootstrap samples.

First, for selection, we compute a new weight matrix W_i for each bootstrap sample $i = 1, \ldots, B_1$, using $\hat{H}$ and the nonnegative least squares (NNLS) or nonnegative LASSO regression method. For each row of weights, we then compute the intersection of the support over all bootstrap samples in which this row was considered. This gives us a sparse index matrix W_{idx} with the intersected support of each row of $\{W_i\}_{i=1}^{B_1}$ over different bootstraps.

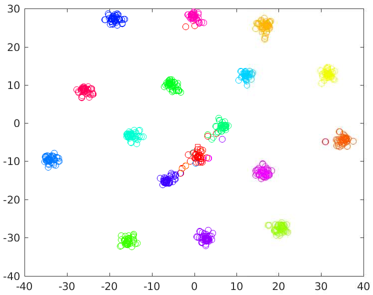


Figure 1: 2D visualization of clusters of bases

Next, for estimation, we consider a new set of bootstrap samples (B_2) and use bagging to comput the weights of the output coefficient matrix $\hat{W}$. We use the sparse coefficient index matrix W_{idx}, the best basis $\hat{H}$, and NNLS to compute the rows of new W_i over B_2 bootstrap samples. For each row of W_i, for all bootstrap samples in which this row is considered, we employ NNLS to compute the weights of the coefficient whose indices are in the rows of W_{idx}. The mean of the weights the across bootstrap samples is chosen as the optimal estimate of the weights, $\hat{W}$, which will be sparse (because W_{idx} is sparse), and have low-bias (no explicit regularization) and low-variance (from bagging).

Algorithm 1 describes our *UoI-NMFcluster* algorithm. The algorithm can be modified for other NMF variants and other methods for clustering (see supplementary).

3.1 Geometric interpretation and uniqueness

In this section, we present the theoretical intuition to use clustering across bases learned from bootstrap samples to obtain more stable parts based decompositions. The uniqueness of the solutions to the NMF problem was discussed in [7], using a geometric interpretation of NMF with simplicial cones.

Geometric Interpretation: There is an unknown H-simplex whose vertices are the rows of $H \in \mathbb{R}_+^{k\times n}$. We observe m points $A \in \mathbb{R}_+^{m\times n}$ that lie in the H-simplex. The goal is to identify the vertices of the H-simplex.

An important observation in [7, 9] is that, if the input data points come from a simplex (without loss of generality), the bases learned by an NMF algorithm will be the vertices of this simplex (non-overlapping bases with separated supports). In this case, the data is called "separable".

Separability: A NMF is separable if all the vertices $H(j,:)$'s appear in the observed points $A(i,:)$'s.

The separability of data was shown to be the key required property for unique solutions for NMF. Polynomial time algorithms have been proposed to find these vertices [1]. Article [9] showed that subset separability of data (a milder condition of separability) is sufficient for obtaining unique solutions.

A NMF algorithm will return a unique solution (learn the vertices) when the data are separable (uniqueness guarantees are shown in the literature only when the NMF is separable or subset separable). However, such separability conditions are hard to test, and are unlikely to hold when the data are noisy. If the data are from a simplex with added noise, the NMF algorithms may learn some of the simplex vertices as bases, along with the noise learned as one or more additional bases, because NMF is an additive model. We show in our numerical experiments that basic NMF algorithms indeed learn a few of these pure bases (vertices) and other bases are related to noise.

Generating bases from multiple bootstrap samples makes it likely that all the simplical vertices will be present with in the superset of bases, i.e., rows of $\tilde{H}$ will likely have many points near the vertices (the 'parts' bases) which are dense spatially, and a few other noisy points related to noisy bases. The noisy bases are widespread and unlikely to be near the vertices. Hence, density based clustering across all the points and using the centroids will give us the vertices of the simplex. DBSCAN ignores the few noisy points that are spread out. Figure 1 gives a 2D visualization obtained by tSNE algorithm [20], of the clustered (spatial) distribution of bases learned over different bootstrap samples for the Swimmer dataset, described in the next section. The colors indicate the clusters assigned by the DBSCAN algorithm with red depicting noise points. Therefore, for separable data with noise, using density based clustering and extracting the cluster centroids will likely return the vertices of the simplex, i.e., the stable part based bases.

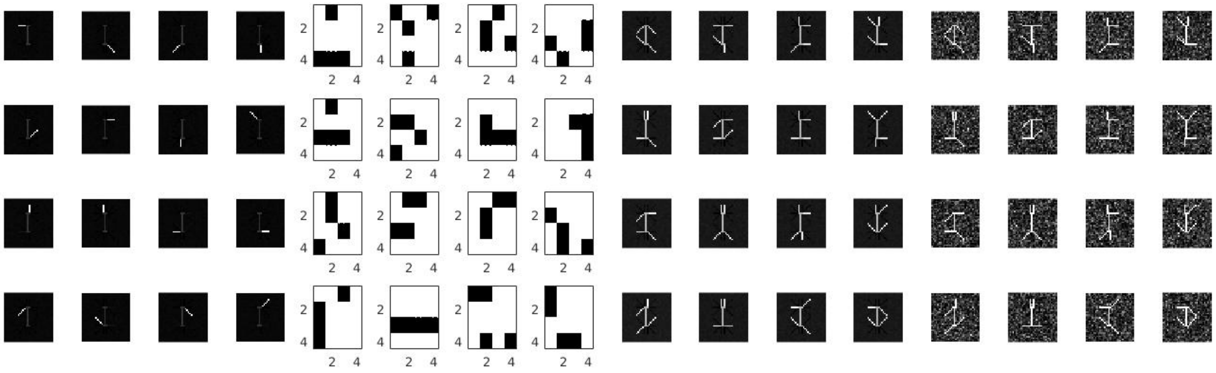


Figure 2: *UoI-NMFcluster for noisy Swimmer data.* First 4×4 images: 16 bases learned by UoI-NMFcluster for the high noise Swimmer data. Second 4×4 images: The nonzero pattern of the sparse weights learned for randomly chosen 16 images. Third 4×4 images: The recovered images. Bottom 4×4 images: The original noisy input images.

4 Numerical Experiments on Synthetic Data

In this section, we illustrate the performance of our proposed algorithm on various synthetic datasets. We compare $UoI\text{-}NMF_{cluster}$ (with $B_1 = 20$ and $B_2 = 10$, see Supplement for varying B_1) to basic NMF (using ALS with multiple initial conditions), sparse NMF (as implemented by SPAMS library), and TSVDNMF. For basic NMF, we used multiple starting matrices $[H, W]$, and for sparse NMF, we used different parameters λ incrementally, and reported the best results. We find that $UoI\text{-}NMF_{cluster}$ yields parts-based, noise-free bases, and thus reconstructions obtained are also noiseless (denoised data).

Swimmer Dataset In the first experiment, we consider the swimmer dataset [7], the canonical example of separable data: each image can be reconstructed from a subset of non-overlapping bases. We compared the performance of $UoI\text{-}NMF_{cluster}$ against other NMF algorithms with multiple random initial conditions for the swimmer dataset corrupted by heavy additive noise (Absolute Gaussian noise, $|\mathcal{N}(0, 0.25)|$, see Supplement for varying noise levels). The dataset contains 256 images of size 32×32 each. We concatenated 10 noisy sets of these 256 images (2560 in total) as the input matrix (of size 2560×1024).

Figure 6 illustrates the performance of $UoI\text{-}NMF_{cluster}$ algorithm on this noisy data. The first 4×4 images of the figure show the 16 bases (parts) learned by $UoI\text{-}NMF_{cluster}$. The second set (4×4 images) displays the sparse weights estimated to reconstruct 16 randomly chosen images. The third set depicts the recovered images $\hat{A} = \hat{W}\hat{H}$, and the last 4×4 image set gives the original noisy input images A. The bases learned by other NMF algorithms (basic NMF, sparse NMF and TSVDNMF [4]) are given in the supplementary.

The resulting bases from $UoI\text{-}NMF_{cluster}$ are remarkably good parts based decompositions of the denoised data, even though the input matrix had very high noise. We see that $UoI\text{-}NMF_{cluster}$ learns all the 16 bases (parts) almost exactly. Thus, for data generated from bases that are vertices of a simplex, our algorithm yields the unique solution that exists, even when the observed data is highly noisy, and hence not separable. The 2D visualization of the spatial distribution of the bases and how DBSCAN clusters the 16 parts based bases of the data in Fig. 1. We also observe that the weights learned are sparse due to the intersection operation of UoI, resulting in the algorithm choosing only bases that are relevant for the reconstruction of the original data. The nonzero patterns of the weights given in Fig. 6 shows that exactly four bases are chosen for reconstruction for all the images. We clearly see that the recovered images are denoised versions of the noise corrupted input images. When the input data are noisy, most basic NMF algorithms tend to learn the noise as a separate bases (due to the additive nature of the factorization). The median nonzeros per row in $\hat{W}$ for $UoI\text{-}NMF_{cluster}$ was 4 and in $\hat{H}$ was 22. The average mean squared error (MSE) between the exact and the learned bases was just 0.0015, and the reconstruction errors $\|A - \hat{W}\hat{H}\|_F$ for noisy data was 195.1 and for noiseless data was 16.8. Table 2 summarizes these results for different NMF algorithms and on different datasets (also see supplementary).

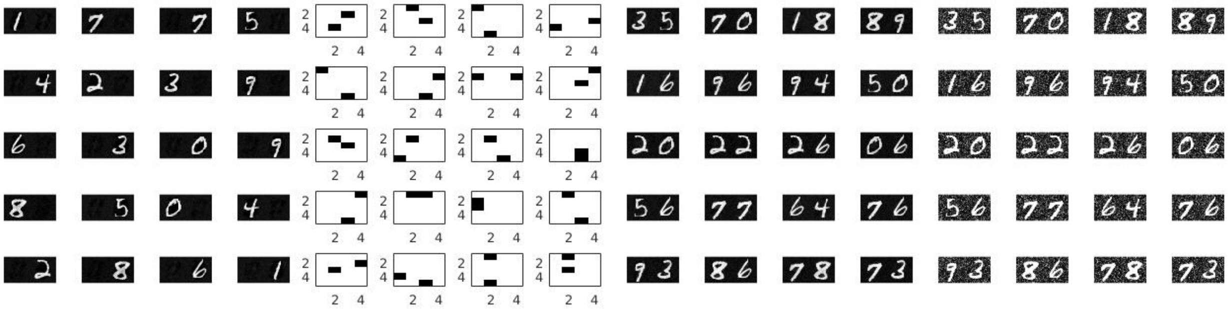


Figure 3: *UoI-NMFcluster for noisy MNIST two digits data.* First 5×4 images: 20 bases learned by UoI-NMFcluster for the high noise MNIST two digits data. Second 5×4 images: The nonzero pattern of the sparse weights learned for ten randomly chosen images. Third 5×4 images: The recovered images. Last 5×4 images: The original noisy input images.

Table 1: Reconstruction error for noiseless and noisy data, avg. MSE between exact and learned bases and median nnz per row in W & H.

Methods	Data	Error (n.less)	MSE bases	Error (noisy)	$nnz(\hat{W})$	$nnz(\hat{H})$
$UoI\text{-}NMF_{cluster}$ (KL. metric)	Swimmer	16.8	0.0015	195.1	4	22
basic NMF (KL. metric)	Swimmer	54.2	0.0052	202.6	3	41.5
sparse NMF	Swimmer	60.4	0.0055	206.2	5	60
TSVDNMF	Swimmer	71.3	0.0236	240.5	3	80
$UoI\text{-}NMF_{cluster}$ (KL. metric)	MNIST	36.61	0.0029	194.3	2	105.5
basic NMF (KL. metric)	MNIST	48.69	0.0102	153.07	3	144.5
sparse NMF	MNIST	59.06	0.0268	192.02	2	149
TSVDNMF	MNIST	78.83	0.0580	256.77	3	156

MNIST 2-digit data Next, we use the popular handwritten digit images from the MNIST dataset [16]. The dataset contains different sets of handwritten digits from 0 to 9 (by different individuals), and in this experiment we select one such set and concatenate two of these images to form 2-digit handwritten numbers (00 to 99). We have 100 such concatenated images. We consider noise ($|N(0, 0.2)|$) corrupted images (10 repetitions, hence we have 1000 images) for training the NMF algorithms. The goal is to learn the individual digits (at units and tens place).

Figure 7 shows the results. The first 5×4 images show the bases learned by our algorithm. The estimated weights to reconstruct 20 randomly chosen images are given along with the reconstructions and the original noisy images. Results for other NMF algorithms are given in the supplementary. $UoI\text{-}NMF_{cluster}$ gives better single digit bases than basic NMF, as well as sparse NMF (see supplementary for results). We also observe the weights learned are quite sparse, exactly two in most cases. Note that, in contrast to the swimmer data set, in this example, the bases (digits) are not quite vertices of a simplex, and hence even noiseless data is not quite separable. Thus, the learned bases are not perfectly decorrelated (e.g., a nine and a one and a seven are all highly correlated). Yet, $UoI\text{-}NMF_{cluster}$ learns these 20 bases quite accurately with average MSE between exact and learned bases just 0.0034.

Table 2 summarizes the results obtained by the different NMF algorithms (first column) on different datasets (second column). The algorithms are : basic NMF (KL. metric) is the basic NMF algorithm with KL divergence error metric, $UoI\text{-}NMF_{cluster}$ (KL. metric) is the proposed algorithm with basic NMF-KL as the inside algorithm, sparse NMF and TSVDNMF.

We give the error $\|A - WH\|_F$ for the reconstruction of original noiseless data the third column and list the average mean squared error (MSE) between the exact and the learned bases in the fourth. The reconstruction error $\|A - WH\|_F$ when the noisy training data was recovered are listed in the fifth column. We also give the median nonzeros per row in the weights W and the bases H learned by the different algorithms in the last two columns respectively. Several additional results, observations and details are provided in the supplementary.

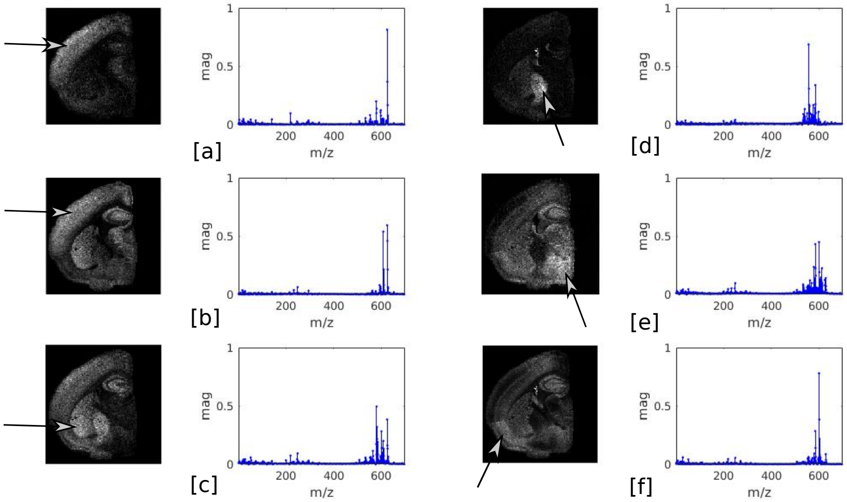


Figure 4: Mouse brain MSI data: The six bases learned by $UoI\text{-}NMF_{cluster}$ and the corresponding weight distribution learned for the respective bases.

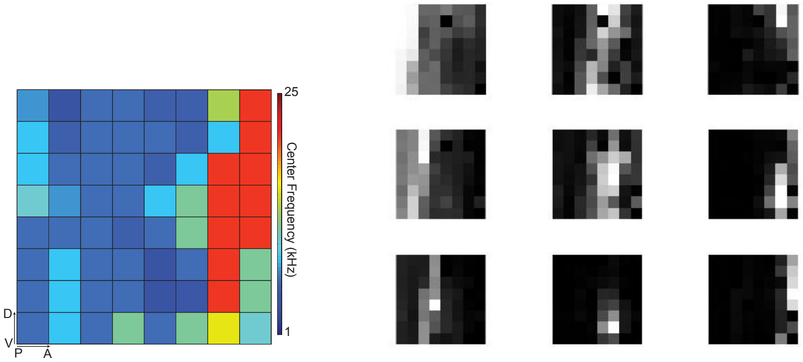


Figure 5: Rat ECoG recordings: Left - ToneMap of the auditory cortex. Right - $UoI\text{-}NMF_{cluster}$ bases.

5 Experiments on Scientific Data

In the following experiments, we demonstrate the performance of $UoI\text{-}NMF_{cluster}$ in two scientific applications.

Mass Spectrometry Imaging of Mouse Brains Mass spectrometry imaging (MSI) is popularly used for label-free, high-resolution spatial mapping of the chemical composition of complex biological samples [21]. MSI acquires one or more mass spectra at each location. Each spectrum is digitized into 10^4 to 10^6 frequency bins (m/z).

Here we present the results obtained when $UoI\text{-}NMF_{cluster}$ was used on a MSI coronal section of mouse brain (NIMS) available in OpenMSI [23] (`https://openmsi.nersc.gov/`). We processed the data as described in [26], which results in 697 images (each of size 122×120). Figure 10 shows the six bases learned by $UoI\text{-}NMF_{cluster}$ and the corresponding weight distribution learned for the respective bases.

We found that many of the bases learned by $UoI\text{-}NMF_{cluster}$ corresponded to spatially localized, anatomically defined parts of the mouse brain (e.g., sensory cortex [a,b], hippocampus/putamen [c], and globus paladus [d], hypothalamus [e] and piriform cortex [f], pointed by arrows). We also observe from the weight distributions that, these parts (bases) appear at different frequency m/z bins, in line with the notion that these different regions of the brain have different chemical compositions. These results were not found by other NMF algorithms (see Supplement).

Electrophysiological Data from Rat Cortex In our final experiment, we employ our $UoI\text{-}NMF_{cluster}$ algorithm to electrophysiological data collected from the rat brain. A 64-channel (8×8) electrocorticography (ECoG) array is placed on the primary auditory cortex of an anesthetized rat, and neural responses (extracted from the time of peak response) to an auditory stimuli consisting of 210 different sounds (30 frequencies and 7 amplitudes) with 20 repetitions (trials) each was collected, thus data is of size 64×4200.

From these data, for each recording channel ('pixel'), we can determine the sound frequency which gave the largest response across amplitudes (the center frequency). In Figure 9 (left), we color code each pixel in the array according to its center frequency (color bar on right). Here, we see a general posterior-to-anterior (left-to-right) progression of center frequencies going from low center frequencies to high center frequencies, with relative isotonic representations along the dorsal-ventral (top-to-bottom) axis. In neuroscience, this spatial organization of frequency representations is known as tonotopy. We note that, while we can summarize the responses in this way, the underlying data is more complex, with each electrode giving a graded response as a function of both amplitude and frequency, and the data on single-trials are noisy: thus, these data are not separable (see Supplement).

Figure 9(right) gives the bases learned by $UoI\text{-}NMF_{cluster}$ on this data. Bases are plotted as 8×8 grid to represent the ECoG grid for visualization as per the channel grid location, and are ordered according to the location of large values. Here, we see that the different bases reflect the tonopic organization of the underlying cortical tissue. That is, the different bases are constrained in the anterior-posterior axis (i.e., across columns) while being extensive in the dorsal-ventral axis (i.e., across rows), and generally tile the grid across the anterior-posterior axis. These results were not observed with other NMF algorithms (see Supplement).

6 Conclusion

Our approach has three key innovations: (i) completely separate bases learning from weight estimation; (ii) cluster bases learned over multiple bootstrap samples to obtain improved parts based decompositions; and (iii) use the UoI-framework to solve the non-negative least squares problem for weight estimation of the learned bases. Separating bases learning from weight estimation helps us in learning a best set of bases and eliminate noisy and spurious bases at the fist stage. We then estimate the sparse weights separately to optimally reconstruct the original data using the best set of bases. This naturally helps in denoising and obtain low errors. The clustered bases learning over multiple bootstraps helps us obtain improved parts by selecting stable bases and eliminating noisy bases. The UoI-framework helps us select the right set of bases and estimate the weights separately to reconstruct the data optimally and avoid the reconstruction of the noise.

Thus, together these innovations gave rise to improved parts-based decompositions with sparse weights, which greatly improves the interpretability of the results and is critical for scientific applications. Furthermore, we find that the learned bases and sparse weights effectively denoise the reconstructed images. Finally, our results suggest improved performance on noisy data sets with less pre-processing and relaxed assumptions. We note that our algorithm has lots of natural parallelism (e.g., over bootstrap resamples), and NNLS computations can be computed in parallel using the alternating directions method of multiplies (ADMM), thus allowing scalability to large data sets. Together, these results suggest that a similar approach may result in improved results from other data decomposition algorithms (e.g., CUR or sparse coding).

References

[1] S. Arora, R. Ge, R. Kannan, and A. Moitra. Computing a nonnegative matrix factorization–provably. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 145–162. ACM, 2012.

[2] S. Behnke. Discovering hierarchical speech features using convolutional non-negative matrix factorization. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 4, pages 2758–2763. IEEE, 2003.

[3] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.

[4] C. Bhattacharyya, N. Goyal, R. Kannan, and J. Pani. Nonnegative matrix factorization under heavy noise. In *Proceedings of the 33nd International Conference on Machine Learning*, pages 1426–1434, 2016.

[5] K. E. Bouchard, A. F. Bujan, F. Roosta-Khorasani, S. Ubaru, Prabhat, A. M. Snijders, J.-H. Mao, E. F. Chang, M. W. Mahoney, and S. Bhattacharyya. Union of Intersections (UoI) Method for Interpretable Data Driven Discovery and Prediction. In *Advances in neural information processing systems (NIPS)*, 2017.

[6] T. G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[7] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in neural information processing systems*, 2003.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

[9] R. Ge and J. Zou. Intersecting faces: Non-negative matrix factorization with new guarantees. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2295–2303, 2015.

[10] N. Gillis. Sparse and unique nonnegative matrix factorization through data preprocessing. *Journal of Machine Learning Research*, 13(Nov):3349–3386, 2012.

[11] N. Gillis. The why and how of nonnegative matrix factorization. *Regularization, Optimization, Kernels, and Support Vector Machines*, 12(257), 2014.

[12] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.

[13] H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007.

[14] J. Kim and H. Park. Sparse nonnegative matrix factorization for clustering. Technical report, Georgia Institute of Technology, 2008.

[15] H. Laurberg, M. G. Christensen, M. D. Plumbley, L. K. Hansen, and S. H. Jensen. Theorems on positive data: On the uniqueness of nmf. *Computational intelligence and neuroscience*, 2008, 2008.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[18] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[19] C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[20] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[21] L. A. McDonnell and R. Heeren. Imaging mass spectrometry. *Mass spectrometry reviews*, 26(4):606–643, 2007.

[22] P. D. O'grady and B. A. Pearlmutter. Convolutive non-negative matrix factorisation with a sparseness constraint. In *2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, pages 427–432. IEEE, 2006.

[23] O. Rübel, A. Greiner, S. Cholia, K. Louie, E. W. Bethel, T. R. Northen, and B. P. Bowen. OpenMSI: A high-performance web-based platform for mass spectrometry imaging. *Analytical Chemistry*, 85(21):10354–10361, 2013.

[24] D. L. Sun and C. Fevotte. Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6201–6205. IEEE, 2014.

[25] S. Wu, A. Joseph, A. S. Hammonds, S. E. Celniker, B. Yu, and E. Frise. Stability-driven nonnegative matrix factorization to interpret spatial gene expression and build local gene networks. *Proceedings of the National Academy of Sciences*, 113(16):4290–4295, 2016.

[26] J. Yang, O. Rubel, M. W. Mahoney, and B. P. Bowen. Identifying important ions and positions in mass spectrometry imaging data using cur matrix decompositions. *Analytical chemistry*, 87(9):4658–4666, 2015.

A Additional Details and Experiments

In this supplementary material, we give additional details of the proposed NMF algorithm and present several additional experimental results. We also give additional details about the scientific datasets used in this work. First, we describe an alternate approach to select the best k set of bases from the B_1 different sets of bases learned over different bootstrap samples.

A.1 Best set of bases by correlation thresholding

Recall that, in our main algorithm, we use a clustering based idea, particularly density based clustering (DBSCAN) to select the best set of k set of bases $\hat{H}$ from the B_1 different sets of bases learned over different bootstrap samples. Other clustering methods such as kmeans/kmedian clustering can also be used, but we found that DBSCAN gave the best results in most cases. In the next section, we give an example where the DBSCAN algorithm gives poor result. Here we present an alternate approach based on correlation thresholding. Suppose we have 'B_1' sets of 'k' bases learned (k bases learned over B_1 bootstrap samples), and these bases are stacked up in $\tilde{H}$ with $k \cdot B_1$ rows. We consider a $(k \cdot B_1 \times k \cdot B_1)$ cross-correlation matrix, given by $C = \tilde{H}\tilde{H}^T - diag(diag(\tilde{H}\tilde{H}^T))$ (if the rows have unit norm, else $C = abs(corr(\tilde{H}^T) - I)$), where $\tilde{H}$ contains all $k \cdot B_1$ bases stacked up as rows.

Next, recall that our objective is to combine similar (near duplicate) bases that come from different bootstrap samples into one bases and eliminate all noisy and spurious bases. Hence, in order to group similar bases, we use the cross-correlation matrix. We consider all pairs of bases with cross-correlation between them greater than a certain preselected threshold, for example, $C_{ij} \geq 0.92$, and combine these bases by averaging them. This is because, similar bases should have high cross-correlation between them. Once all such similar bases are combined, we can then choose the best set using the least sum of correlation idea presented in the main paper. The best k set of bases is then chosen as the k bases that have the lowest sum of pairwise correlations. For this, we sum the rows of the new cross-correlation matrix C (after similar bases are combined), and choose the best bases $\hat{H}$ as the k bases with the smallest sum of correlations.

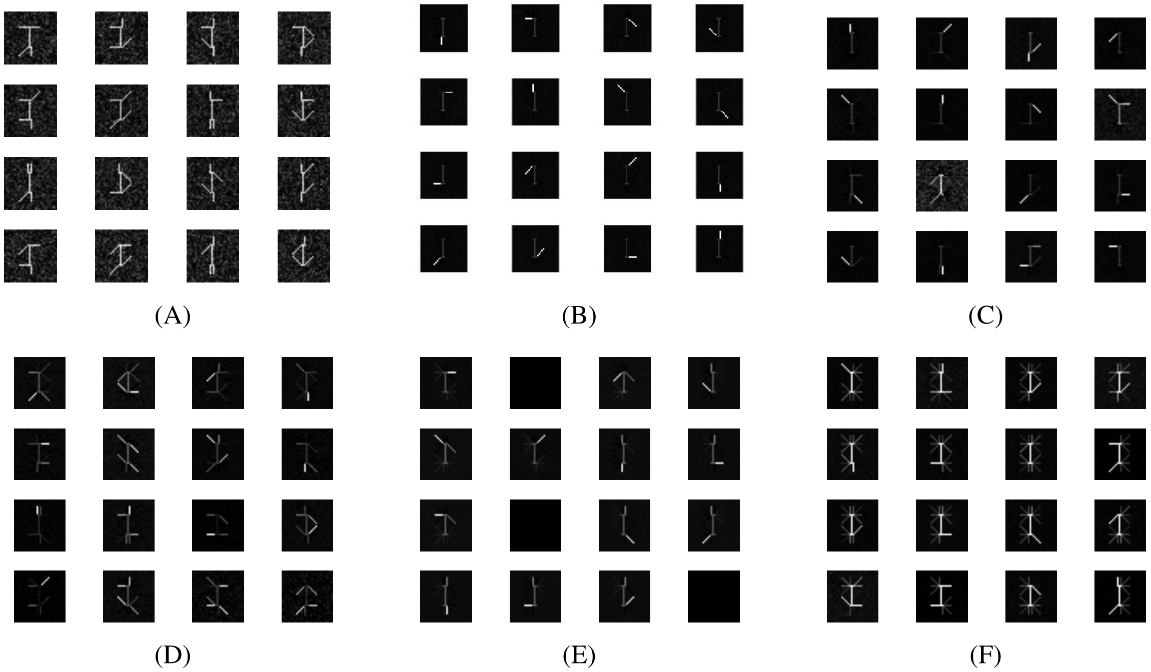
A drawback with this approach is that, we need to choose a right threshold such that the similar bases are grouped together. A smaller than ideal threshold might result in dissimilar bases being combined. In the experimental results presented in latter in this supplementary, we see that this correlation thresholding method performs well sometimes.

A.2 Additional experimental results

In this section, we present several additional experimental results illustrating the performance of $UoI\text{-}NMF_{cluster}$ in comparison with various NMF algorithms.

Comparisons First, we consider the Swimmer and the two-digit MNIST datasets (ten copies of noise corrupted images). The sizes of these matrices are 2560×1024 and 1000×1568, respectively. We reported the bases H and the weights W learned (for a few randomly chosen images) by our proposed algorithm for these two datasets in the main paper. Here, we give the bases learned from other popular NMF algorithms for these two datasets. The number of bootstraps used for $UoI\text{-}NMF_{cluster}$ in all the experiments here and in the main paper was $B_1 = 20$ for bases learning and selection, and $B_2 = 10$ for estimation of weights. We show how varying the number of bootstraps B_1 affects the performance of $UoI\text{-}NMF_{cluster}$ in the latter part of this section (see Figure 11).

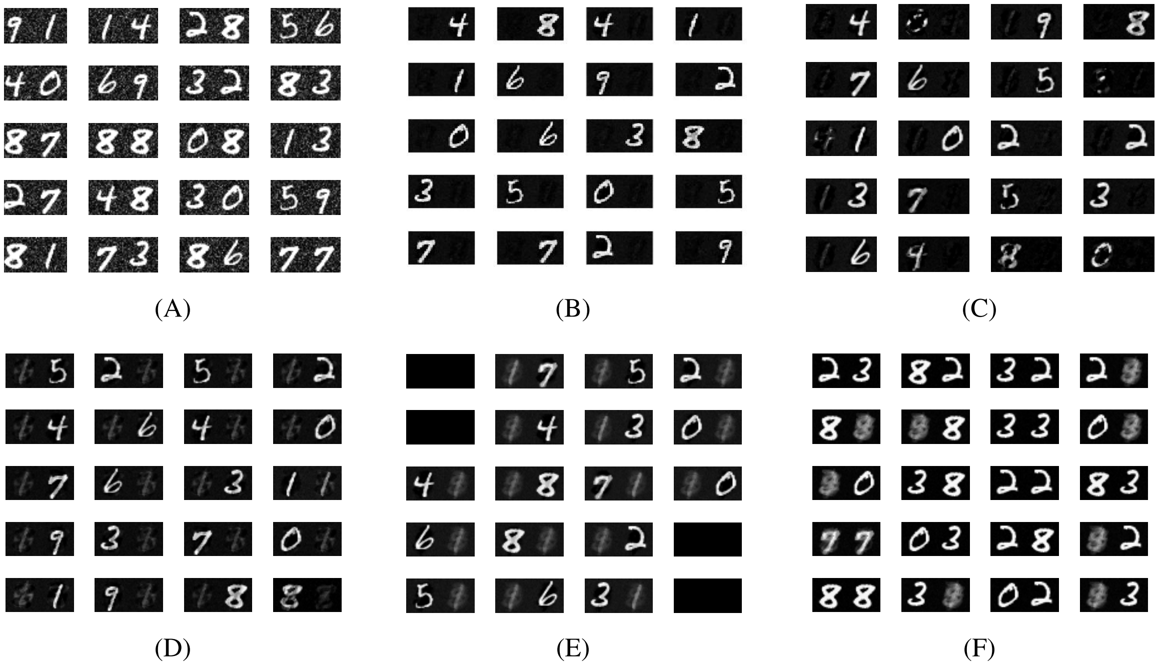
Figure 6 gives the 16 bases learned by various NMF algorithms from the noisy Swimmer dataset. The first 4×4 set of images (fig. 6 (A)) shows the noise corrupted images (randomly chosen) on which the algorithms were trained. The second set (B) is the 16 bases learned by $UoI\text{-}NMF_{cluster}$, which we also saw in the main paper. The third set (C) corresponds to the bases learned by basic NMF algorithm with the KL metric. The algorithm was run for different starting random matrices W and H and the set of bases that gave the least error is presented. We see how the basic NMF algorithm learns few of the individual exact parts as bases and the noise is learned as one or two separate bases (images [3,2] and [2,4]). $UoI\text{-}NMF_{cluster}$ learns such bases over different bootstrap resamples, clusters the superset of bases and selects the best k bases using the least sum of correlation, yielding all the exact bases. Clearly, the noisy bases have higher correlation than the individual exact parts.



(A) (B) (C)

(D) (E) (F)

Figure 6: Bases learned by various algorithms: (A) The noise corrupted images (randomly chosen); (B) $UoI\text{-}NMF_{cluster}$ (C) basic NMF-KL metric; (D)basic NMF-Euclidean metric;(E) sparse NMF; and (F) TSVDNMF.



(A) (B) (C)

(D) (E) (F)

Figure 7: Bases learned by various algorithms: (A) The noise corrupted images (randomly chosen); (B) $UoI\text{-}NMF_{cluster}$ (C) basic NMF-KL metric; (D)basic NMF-Euclidean metric;(E) sparse NMF; and (F) TSVDNMF.

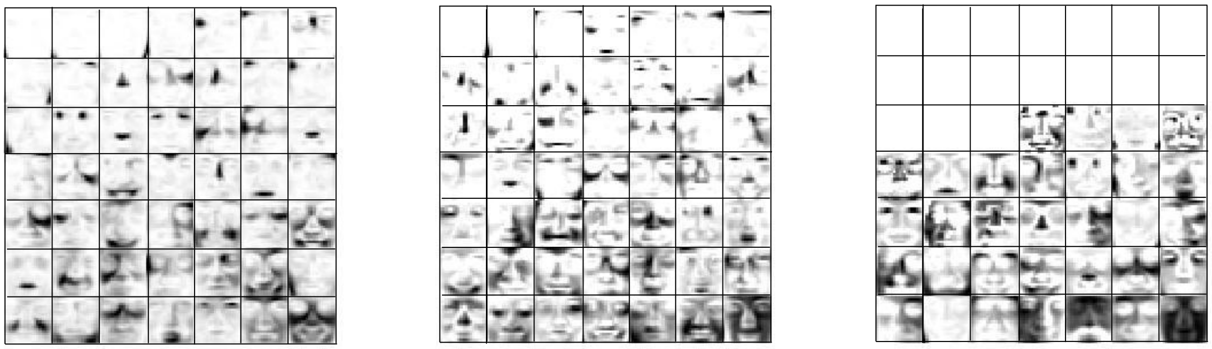


Figure 8: The 49 bases learned for CBCL face images using different NMF algorithms. Left: UoI-NMFcluster; Middle: basic NMF; and Right: sparse NMF.

For all compared algorithms other than $UoI\text{-}NMF_{cluster}$, we report the best results obtained, throughout the paper. This we feel is the best representative results for the remaining algorithms. Note that without clustering bases, averaging is non-sensical, so there are no real ways to include averaging type results with base NMF algorithms. Further discussion and results on the quality of the bases learned by the algorithms are given in the latter part of this section.

Figure 6(D) gives the bases learned by basic NMF algorithm with the Euclidean metric (with multiple starting matrices). The next set (E) is of the bases learned by sparse NMF algorithm. The algorithm was run over a range of parameter λ_1 and the set of bases that gave the least error is presented. We see that sparse NMF tends to give some empty (all zero) bases. Tuning the regularization parameters λ_1 and λ_2 only increased or decreased the sparsity, and did not improve the quality of the bases learned. The last set (F) is the bases learned by the TSVDNMF algorithm [4] (as implemented by the authors). This algorithm is mainly tailored for topic modeling and what is known as dominant NMF. The algorithm tends to learn the trunk in all bases (which is the dominant part of all images). The parameters within the algorithm are automatically set for optimal performance as implemented by the authors and we have not attempted to tune them.

Figure 7 presents the 20 bases learned by the various NMF algorithms from the noisy MNIST dataset. The six sets of images are ordered in the same fashion as in fig. 6. That is, (A) is the noise corrupted images (randomly chosen); (B) bases learned by $UoI\text{-}NMF_{cluster}$ (C) by basic NMF-KL metric; (D) by basic NMF-Euclidean metric; (E) by sparse NMF; and (F) by TSVDNMF algorithm. We again observe the basic NMF-KL algorithm learns some of the exact bases and other bases are spurious. The basic NMF-Euclidean metric algorithm learned the parts (digits in units and tens place) almost correctly, but contains noise on the other part. Sparse NMF learns a few empty bases and the TSVDNMF bases contain many eight and zero (which are dominant).

CBCL face images. Next we consider another 'synthetic' experiment with the CBCL face images database, used in the seminal paper [17]. The dataset consists of 2429 face images of size 19×19. We use these images are input matrix A (without any alteration), and learn 49 bases.

Figure 8 displays the 49 bases learned by the three NMF algorithms, namely $UoI\text{-}NMF_{cluster}$, basic NMF and sparse NMF. This dataset is interesting for multiple reasons. First, for $UoI\text{-}NMF_{cluster}$, we use kmeans clustering with cosine distance metric for this particular dataset. This is because, DBSCAN fails to give good results for this dataset because the clusters of the bases learned are spatially overlapping. A correlation based clustering such as kmeans with cosine metric performs better than density based clustering in such cases. Next, we make the following observations from these results: a) The $UoI\text{-}NMF_{cluster}$ (Left) algorithm performs much better than the other two algorithms in terms of yielding better parts based decompositions of the faces. We clearly observe that many of the bases learned by $UoI\text{-}NMF_{cluster}$ are parts of a face such as nose, eyes, eyebrows, mouth, mustache, and cheek, rather than whole face like features. Basic NMF algorithm in the middle and the sparse NMF algorithm in the right give bases that look more like faces than parts of faces. b) The sparse NMF seems to perform poorly and either yields bases that are complete faces or empty (all zeros) bases. c) The basic NMF algorithm does not seem to replicate the

Table 2: Median nonzeros per row in W and H, reconstruction error for noisy and noiseless data, and average MSE between exact and learned bases.

Methods	Data	Error (n.less)	MSE bases	Error (noisy)	$nnz(\hat{W})$	$nnz(\hat{H})$
$UoI\text{-}NMF_{cluster}$ (KL. metric)	Swimmer	16.8	0.0015	195.1	4	22
basic NMF (KL. metric)	Swimmer	54.2	0.0052	202.6	3	41.5
sparse NMF	Swimmer	60.4	0.0055	206.2	5	60
TSVDNMF	Swimmer	71.3	0.0236	240.5	3	80
$UoI\text{-}NMF_{cluster}$ (Euc. metric)	Swimmer	40.5	0.0047	246.2	4	42
basic NMF (Euc. metric)	Swimmer	58.2	0.0076	309.7	10	45
$UoI\text{-}NMF$ threshold 0.92 (KL)	Swimmer	25.3	0.0036	219.1	3	34
$UoI\text{-}NMF_{cluster}$ (KL. metric)	MNIST	36.61	0.0029	194.3	2	105.5
basic NMF (KL. metric)	MNIST	48.69	0.0102	153.07	3	144.5
sparse NMF	MNIST	59.06	0.0268	192.02	2	149
TSVDNMF	MNIST	78.83	0.0580	256.77	3	156
$UoI\text{-}NMF_{cluster}$ (Euc. metric)	MNIST	44.89	0.0138	178.73	2	108.5
basic NMF (Euc. metric)	MNIST	45.80	0.0160	185.73	4	168
$UoI\text{-}NMF$ threshold 0.93 (KL)	MNIST	80.33	0.0192	272.66	3	124
$UoI\text{-}NMF_{cluster}$ (Euc. metric)	CBCL (no PP)	74.85	-	-	21	124
basic NMF (Euc. metric)	CBCL (no PP)	161.48	-	-	26	153
sparse NMF	CBCL (no PP)	68.67	-	-	11	235
$UoI\text{-}NMF_{cluster}$ (Euc. metric)	CBCL (with PP)	83.40	-	-	6	70
basic NMF (Euc. metric)	CBCL (with PP)	80.82	-	-	8	68
sparse NMF	CBCL (with PP)	65.24	-	-	11	180

results presented in the seminal paper [17].

Our algorithm returns better parts based decomposition because the algorithm clusters all the face like bases into fewer groups/bases. The parts bases (which are sparse and less correlated with the other bases) are naturally separated into different clusters, hence, yielding better parts based decompositions of the faces. The reason why we are unable to replicate the results presented in the paper [17] for basic NMF algorithm is because there the face images are heavily preprocessed. The bases learned for such a preprocessed face images by the three algorithms are given latter in this section. This experiment shows that, compared to other algorithms, $UoI\text{-}NMF_{cluster}$ yields better parts based decompositions of data that are clearly not separable, and does so with no data preprocessing.

Table 2 summarizes the results obtained by the different NMF algorithms (first column) on different datasets (second column). The algorithms are : basic NMF (KL. metric) is the basic NMF algorithm with KL divergence error metric, and basic NMF (Euc. metric) is the basic NMF algorithm with Euclidean distance error metric. $UoI\text{-}NMF_{cluster}$ (KL. metric) is the proposed algorithm with basic NMF-KL as the inside algorithm. $UoI\text{-}NMF_{cluster}$ (Euc. metric) is the proposed algorithm with basic NMF-Euc inside. $UoI\text{-}NMF$ threshold 0.92 (KL) is the variant of our proposed algorithm where the bases are selected using the correlation thresholding idea presented earlier. The threshold selected was 0.92 and NMF-KL was the inside algorithm.

We give the error $\|A-WH\|_F$ for the reconstruction of original noiseless data the third column and list the average mean squared error (MSE) between the exact and the learned bases in the fourth. The reconstruction error $\|A-WH\|_F$ when the noisy training data was recovered are listed in the fifth column. We also give the average nonzeros per row in the weights W and the bases H learned by the different algorithms in the last two columns respectively. For the CBCL face images, we do not consider a noisy version. Instead we compare the performance when faces are preprocessed (CBCL (with PP)) as in [17] and are unaltered (CBCL (no PP)). Details on the preprocessing and the bases learned for the preprocessed data are given in the latter part of this section. We see that, the reconstruction error for face dataset, is lowest for sparseNMF. This is because, the bases learned by sparseNMF are more like faces than parts.

Bases quality Next, in order to analyze how good the learned bases are, for the different NMF algorithms, we look at the quality of learned bases. For this, we consider the exact bases for the swimmer dataset, which are the 16 possible

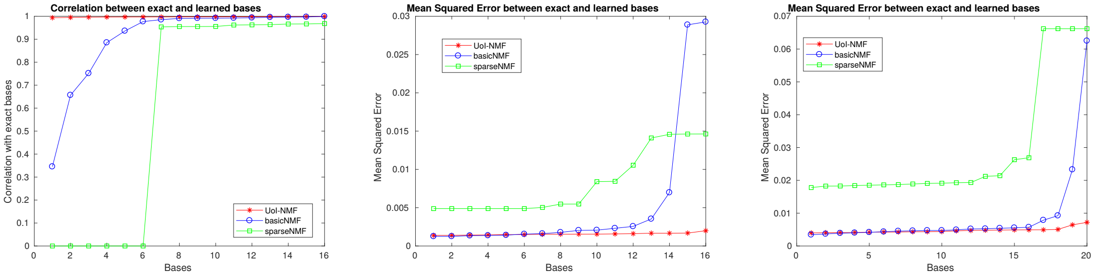


Figure 9: Correlation and Mean Squared Error (MSE) between the exact 16 parts of the swimmer dataset and the 16 bases learned by the three different methods, viz., basic NMF, sparse NMF and UoI-basicNMF. (plotted in ascending order).

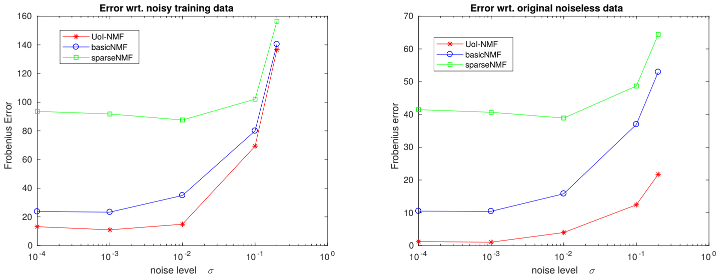


Figure 10: Error v/s Noise level (σ): For noisy image recovery (left) and original image recovery (right) with absolute Gaussian noise with noise level σ.

hand and leg positions of the swimmer, and compare the the learned bases against these exact bases. We use two measures to analyze the quality, namely, the pairwise correlation and the mean squared error (MSE) between the exact and learned bases (average of which is listed in the last column of Table 2 for the different algorithms). Figure. 9 plots the pairwise correlation (left) and the MSE (middle) between the exact and learned bases for the swimmer dataset. To check which exact basis is the closest to a given learned basis, we compute the pairwise correlation with all 16 exact bases and choose the one with maximum correlation.

We observe that for $UoI\text{-}NMF_{cluster}$, for all 16 bases, the pairwise correlation is almost one and MSE is very close to zero indicating the bases learned are very close to the exact ones (which can be verified visually in Figure 6 here and in the main paper). For basic NMF (with KL metric), we see that some of the bases have pairwise correlation almost one (very low MSE) indicating basic NMF learns some of the exact bases very well. Other bases are poor (have high MSE). We observed this result in Figure 6(C) as well. Results for sparse NMF are also plotted. The correlation with the empty bases are depicted by zeros.

Figure 9 also plots the Mean Squared Error (MSE) between the exact 20 digits (units and tens place) and the 20 bases learned by the three different methods. We again observe that all 20 bases learned by $UoI\text{-}NMF_{cluster}$ are very good, and many of the bases learned by basic NMF were exact parts too, but it also learns few spurious bases due to noise.

Errors v/s noise level Next, we analyze the performances of the three NMF methods, viz., basic NMF, sparseNMF and $UoI\text{-}NMF_{cluster}$ in recovering the noisy and original images as a function of the noise level for the swimmer dataset. We consider five noisy sets of the swimmer dataset with absolute Gaussian noise with different noise levels σ. Figure 10 plots the Frobenius error $\|A - WH\|_F$ as a function of the noise level (σ). In the left, we have the errors when reconstructing A the noisy training data when the weights W and bases H were learned by the algorithms over this A which was corrupted by noise of the corresponding noise level σ. The errors obtained when these bases (learned

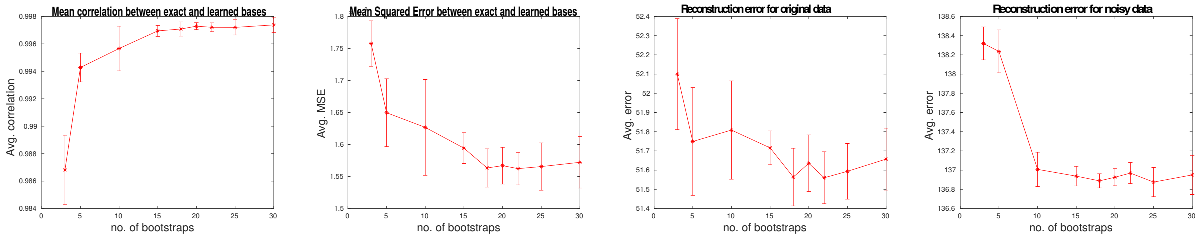


Figure 11: Number of bootstraps: Mean Correlation and MSE between exact and learned bases (first two plots); Reconstruction errors for noisy and the original data (last two plots). Error bars over 5 trials.

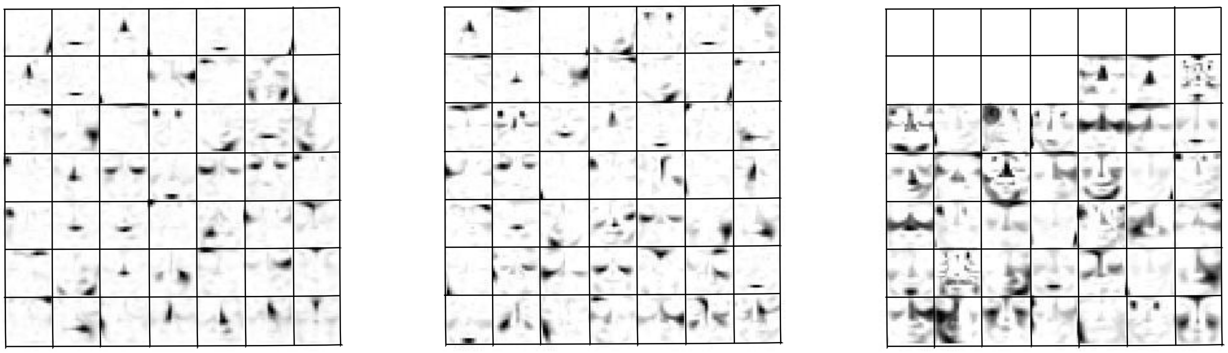


Figure 12: The 49 bases learned for preprocessed CBCL face images using different NMF algorithms. Left: $UoI\text{-}NMF_{cluster}$; Middle: basic NMF; and Right: sparse NMF.

from the different noisy data) were used to reconstruct the original noiseless data X are given in the right plot. The results plotted are averages over five trials.

We observe that, all the reconstruction errors for recovering noisy images increase as the noise level increases. This is expected since the reconstructions are expected to be noise free. For lower noise levels, the basic NMF algorithm learns bases that are close to the exact ones, and both basic NMF and $UoI\text{-}NMF_{cluster}$ learn similar bases. However, due to the intersection operation in weight W estimation, $UoI\text{-}NMF_{cluster}$ learns better weights and yields lower error for both noisy and noiseless cases. As the noise level increases, the basic NMF learns the noise as one or two bases and hence, the reconstruction of the original noiseless images becomes poor. Whereas, $UoI\text{-}NMF_{cluster}$ NMF still learns exact bases, therefore the reconstruction error of the original noiseless images does not increase much. This plot clearly illustrates the noise tolerance of $UoI\text{-}NMF_{cluster}$. Other noise distributions were also tried, for e.g., Poisson noise (normalized to $[0, 1]$) and we obtained similar results.

Number of bootstraps We know that the number of bootstraps B_1 and B_2 used in $UoI\text{-}NMF_{cluster}$ are parameters which we can tune. In this experiment, we try to understand the influence of the number of bootstrap samples used on the quality of results obtained. We apply $UoI\text{-}NMF_{cluster}$ with different number of bootstraps B_1 on a noisy Swimmer dataset (five noisy sets with $\sigma^2 = 0.2$), and plot the results.

Figure 11 plots the mean pairwise correlation and the average MSE between the exact and the learned bases as a function of the number of bootstraps B_1 in the first two plots. The reconstruction errors for noisy and original data using the bases learned by the $UoI\text{-}NMF_{cluster}$ for the different number of bootstraps used are plotted in the last two plots. All plots show the mean and the error bars over 5 trials. We see that, as the number of bootstraps increases, the quality of bases learned improves up to a certain number and then for large number of bootstraps, the quality remains the same. This is because, as the number of bootstraps increase, the density of the clusters increase and the DBSCAN

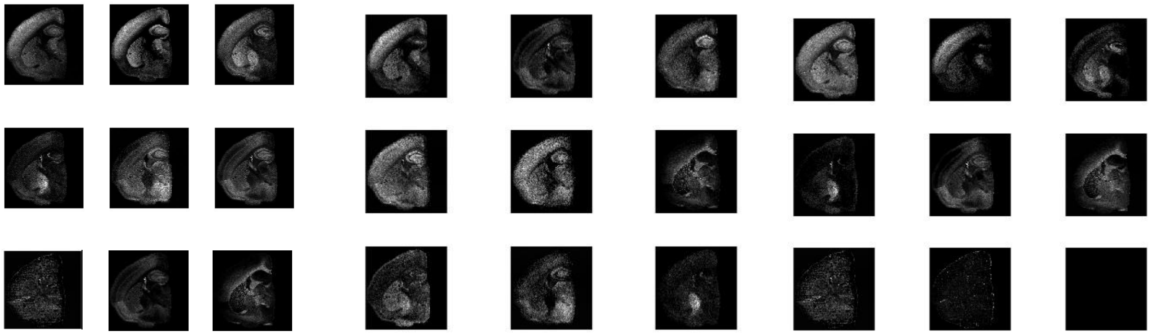


Figure 13: Mouse brain MSI data: The top nine bases learned by $UoI\text{-}NMF_{cluster}$ (left), basic NMF (middle) and sparse NMF (right).

algorithm performance improves. There seems to be a number (between 15-20) beyond which, DBSCAN is able to select all 16 bases exactly. Increasing the number of bootstraps beyond will not have much effect on the quality of bases learned.

The reconstruction errors for the noisy and original data also decrease initially as the number of bootstraps increases due to improvement in the basis quality. The improvement in the results are also due to improved weight learning by the UoI framework. But, for larger B_1 (> 25), the error slightly increases for recovering noisy bases because the weights learned become very sparse due to the intersection operation. We see the peak performance occurs for around 20-22 bootstrap resamples. Hence, we chose $B_1 = 20$ in all our experiments. The effect of the number of bootstraps B_2 in the weight estimation stage will be similar to effect of the number of bootstraps in the feature estimation stage of UoI_{LASSO}. For discussion related to this, see [5].

Preprocessed CBCL images We saw earlier that, the bases learned for the CBCL face data by basic NMF were not similar to the results presented in the paper [17]. This is because, in that work, the face images are preprocessed. The face images are processed with mean variance normalization and thresholding. The grayscale intensities are first linearly scaled so that the pixel mean and standard deviation are equal to 0.25, and then clipped to the range [0,1]. The bases learned for such preprocessed face images by $UoI\text{-}NMF_{cluster}$, basic NMF, and sparse NMF are given figure 12. We observed that the basic NMF performance improves significantly and the results replicate the ones presented in [17]. However, we see that $UoI\text{-}NMF_{cluster}$ also gives very good parts based bases. This experiment shows that, compared to other NMF algorithms, $UoI\text{-}NMF_{cluster}$ yields better parts based decompositions of data that are clearly not separable, and does so without requiring data preprocessing.

A.3 Additional details on the scientific data

Mass Spectrometry Imaging of Mouse Brains Mass Spectrometry Imaging (MSI) is a modern chemical imaging technique that has enabled investigation of metabolic processes at very high resolution (subcellular to centimeter range). In MSI, a laser is raster scanned across a surface and molecules are desorbed from the surface at each location. These ions/molecules are then collected and analyzed by mass spectrometry, which yields a large number of spectral images. MSI may contain spectral images with up to a million pixels and are typically collected over 10^4 to 10^6 frequency bins (m/z). Hence, such MSI data present many analysis and interpretation challenges due to the size and complexity of the data. The objective of using NMF (or any other dimensionality reduction techniques) on MSI data is to reduce the large volume of measured data into easier to interpret smaller blocks. The goal is to identify important locations/pixel positions and the corresponding chemical composition.

The MSI data (NIMS) of the mouse brain which we reported in the main paper was obtained from OpenMSI[1] [23].

[1] https://openmsi.nersc.gov/

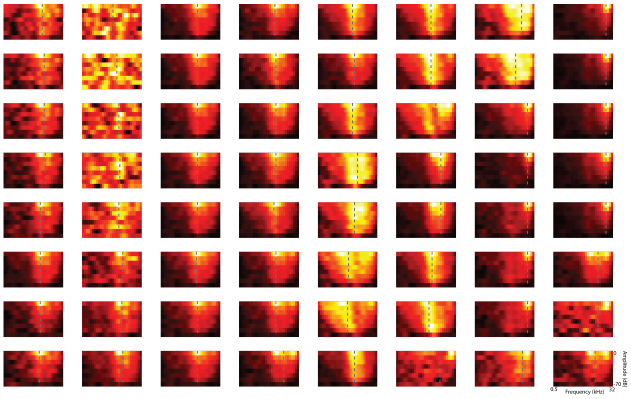


Figure 14: Rat ECoG recordings: Recording patterns.

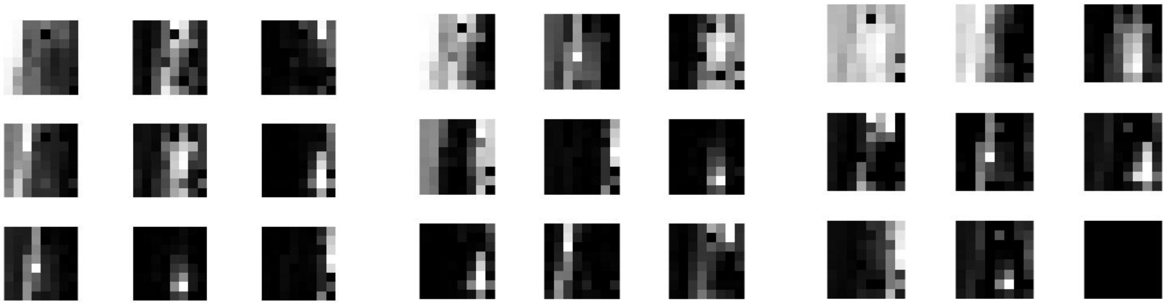


Figure 15: Rat ECoG recordings: The top nine bases learned by $UoI\text{-}NMF_{cluster}$ (left), basic NMF (middle) and sparse NMF (right).

The data contains 120 × 122 size spectral images, computed at 80339 frequency m/z bins. The data is very noisy and certain preprocessing is necessary. The data were preprocessed by using background subtraction, smoothing and peak picking as mentioned in [26]. This reduced the number of data points from 80339 to 697, after peak picking. We saw that the results obtained from $UoI\text{-}NMF_{cluster}$ corresponded to spatially localized, anatomically defined parts of the mouse brain. The weight distributions indicated the dominant chemical compositions for the localized regions.

We saw the important six (meaningful from neuroscience perspective) bases learned by $UoI\text{-}NMF_{cluster}$ for this MSI data in the main paper. These bases were spatially localized and were anatomic parts of the mouse brain. Figure 13 gives the top nine bases from $UoI\text{-}NMF_{cluster}$ (left), basic NMF (middle) and sparse NMF (right) algorithms. We clearly note that, not all bases for the latter two algorithms are parts of the brain (parts based bases) with some bases highlighting all regions of the brain. The top bases learned by $UoI\text{-}NMF_{cluster}$ are better parts based representations of the mouse brain compared to the bases learned by basic and sparse NMF, which do not yield all parts for a given decomposition. $UoI\text{-}NMF_{cluster}$ ensembles all the bases learned and chooses the best parts based bases. Table 3 gives the average cross correlation between these nine bases and their median sparsities (nnz of bases). $UoI\text{-}NMF_{cluster}$ bases are less correlated and are sparser, indicating that $UoI\text{-}NMF_{cluster}$ bases are better parts based representations (individual parts are likely to be uncorrelated from each other and are sparse).

Electrophysiological Data from Rat Cortex The objective of this experiment is to learn a set of bases (channel responses/ neuron firing patterns) from the ECoG recordings for certain stimuli. With the ever increasing number of simultaneously recorded neural signals, neuroscience has seen a resurgence in the application of dimensionality reduction algorithms to summarize high-dimensional data. However, the primary method used in the field, PCA, has made the interpretation of the physical meaning of the derived axes opaque (a common critique of PCA). Here, our goal was to determine if $UoI\text{-}NMF_{cluster}$ could extract a physically meaningful bases directly from neural recordings

Table 3: Average cross correlation between bases and median nnz per row in H and W.

Methods	Data	Corr.	$nnz(\hat{H})$	$nnz(\hat{W})$
$UoI\text{-}NMF_{cluster}$	MSI	0.3960	955	13 (of 20)
basic NMF	MSI	0.4440	1311	17 (of 20)
sparse NMF	MSI	0.3915	1647	3 (of 20)
$UoI\text{-}NMF_{cluster}$	ECoG	0.1545	22.5	3 (of 12)
basic NMF	ECoG	0.1670	24.5	9 (of 12)
sparse NMF	ECoG	0.1674	43.5	6 (of 12)

when there is a known spatial organization of neural response properties (otherwise, how would we know what success would look like?). To this end, we applied $UoI\text{-}NMF_{cluster}$ to neural recordings taken from the auditory cortex of a rat, which has a well characterized spatial organization of frequency representations across the cortical surface (i.e., tonotopy).

In this experiment, we used the neural response recordings collected from the primary auditory cortex of an anesthetized rat using a 64 channel μECoG. This is novel data collected by the authors (us) and has not been used in any prior literature. Following standard procedures in the field, at each electrode, we determined the neural response by extracting the analytic amplitude from the 'high-gamma band' [70-150Hz], which correlates well with multi-unit spiking activity. These responses were z-scored relative to the baseline statistics for each channel individually. The response was for auditory stimuli consisting of 210 different sounds (30 frequencies and 7 amplitudes) with 20 repetitions (trials) each. The number of time steps used was 101. Hence, the data was of size $64 \times 4200 \times 101$. This data was preprocessed by doing peak response picking. For each stimuli, the peak response after the stimulus starts (after 40 time steps), for each channel was chosen as that channel's output for that particular stimulus. Hence, the data was reduced to 64×4200 (with each 20 set of columns corresponding to the 210 stimuli each). A heat-map of each electrodes (as they are laid out on the grid) neural response to each frequency × amplitude pairing can be seen in Figure 14. This shows how the underlying data is more complex, with each electrode giving a graded response as a function of both amplitude and frequency, and the data on single-trials are noisy.

The NMF bases learned by $UoI\text{-}NMF_{cluster}$ were reported in the main paper. These bases were plotted as 8×8 grid to represent the ECoG grid for visualization as per the channel grid location. We saw the bases learned by $UoI\text{-}NMF_{cluster}$ for this MSI data had meaningful columnar structure which corresponded to the tonopic organization of the underlying cortical tissue. Figure 15 gives the top nine bases from $UoI\text{-}NMF_{cluster}$ (left), basic NMF (middle) and sparse NMF (right) algorithms. We again note that, the latter two algorithms fail to learn all parts for a given decomposition, and some of the bases learned do not have the columnar structure that we are looking for. $UoI\text{-}NMF_{cluster}$ yields better parts based representation since it ensembles the bases learned over different bootstrap samples and chooses the best parts based bases.

Table 3 gives the average cross correlation between the top bases and their median sparsities H and W (nnz of bases and weights) for the two scientific datasets. $nnz(W)$ was computed for the original number of bases used (as indicated in the parantheses). $nnz(H)$ reported are for the top nine bases shown in figures 13 and 15. For computing the cross correlation, we ignored the zero bases obtained by sparse NMF in both cases. We saw how $UoI\text{-}NMF_{cluster}$ gives more meaningful bases from the scientific applications viewpoint. The above table shows that, the top $UoI\text{-}NMF_{cluster}$ bases are less correlated and are sparser compared to the top bases of basic and sparse NMF, indicating that $UoI\text{-}NMF_{cluster}$ bases are better parts based representations. Our method reconstructs the data with fewer number of bases as well (sparser W), which helps interpretability.