Ghostbusters: A Parts-based NMF algorithm

Ruairí de Fréin

Telecommunications Software and Systems Group, WIT, Ireland

E-mail: rdefrein@gmail.com

Abstract — **An exact nonnegative matrix decomposition algorithm is proposed. This is achieved by 1) Taking a nonlinear approximation of a sparse real-valued dataset at a given tolerance-to-error constraint, ϵ; Choosing an arbitrary lectic ordering on the rows or column entries; And, then systematically applying a closure operator, so that all closures are selected. Assuming a nonnegative hierarchical closure structure (a Galois lattice) ensures the data has a *unique* ordered overcomplete dictionary representation. Parts-based constraints on these closures can then be used to specify and supervise the form of the solution. We illustrate that this approach outperforms NMF on two standard NMF datasets: it exhibits the properties described above; It is correct and exact.**

Keywords — **Nonnegative Matrix Factorization, Lectic Orderings, Unique Solutions**

I Introduction

In a seminal paper by Lee and Seung, the notion of Non-negative Matrix Factorization (NMF) was proposed as a way to find a set of basis functions for representing non-negative data [1]. NMF, claimed Lee and Seung, is useful for image articulation libraries made up of combinations of articulations and poses. They claimed NMF found the intrinsic "parts" of these images. This idea of decomposing images, financial time-series [2], or word corpuses into significant parts (basis functions) and activations of these parts (indicator functions) in the data ensemble has been enthusiastically applied; The application of NMF to Blind Source Separation (cf. [3]) related tasks has been frequently reported in these proceedings [4, 5, 6].

More recently, however, Donoho and Stodden posed the two following fundamental questions [7]:
- "Under what assumptions is the notion of NMF well-defined, for example is the factorization in some sense unique?"
- "Under what assumptions is the factorization correct, recovering the right answer?"

To begin to address these questions Donoho and Stodden developed a geometric view of the NMF generative model, and derived geometric conditions under which the factorization was *essentially* unique. They also formulated a class of images which looked to satisfy these conditions. This class of images was created by an NMF-style generative model, where all different parts –building blocks of the images– were exhaustively sampled. They named this class of images a *Separable Factorial Articulation Family* (SFAF). They claimed that NMF of images from this family produced factorizations which were *effectively* unique.

By introducing one factor which did not obey the conditions of a SFAF –into their ensemble of images that had the SFAF properties– Donoho and Stodden were only able to produce an approximately correct NMF. The reason that this solution was only approximately correct was that this pathological factor appeared as a *ghost function* in each factor in subsequent decompositions. It is this ghost that we aim to eliminate in this paper.

Contribution 1: We show how to generate a truly unique solution to the pathological SFAF problem proposed by the Donoho and Stodden, and call this approach *Ghostbusters*. This is an important result: NMF is widely used. Having the ability –even off-line– to determine the correct solution is useful for retrospective analysis of NMF.

Contribution 2: We show for binary matrices, a unique binary NMF can always be learned, irrespective of the properties embodied by the class of problems in the SFAF. NMF is often applied to binary datasets by adding a small amount of additive noise to ensure the dataset is in the nonnegative orthant, more recently, by leveraging some nonlinearity in a heuristic approach to generate the factorization [8, 9, 10, 11]. We contribute a binary NMF that operates directly on binary data.

Most humans typically out-perform NMF when decomposing Donoho and Stodden's dataset into parts. There is no framework which incorporates the considerable amount of information available to the user into an NMF decomposition. We introduce an intuitive framework (which goes beyond introducing sparse priors) for incorporating prior information (and representation selectivity)

into the decomposition via *parts-based rules*.

Contribution 3: We then contribute an algorithm –suited to nonnegative datasets– to learn a unique NMF subject to some target tolerance-to-error, which is practitioner specified. This NMF problem formulation has its interest, as it gives an approximate (or exact) decomposition where the approximation's quality is user specified. Traditionally, the practitioner has little control over the quality of the NMF approximation save for running the NMF routine (sometimes) exhaustively until the approximation quality criteria is met. *Ghostbusters* is slow, admittedly; However, we point to a related paper which indicates how the underpinning routine may be significantly sped-up by parallelizing the decomposition without communication between the different computational resources [12] –a common failing of MapReduce implementations of NMF [13, 14].

II Nonnegative Matrix Factorization

This paper deals with both binary-relational (association) and nonnegative (intensity) matrices which are denoted by $\boldsymbol{X} \in \Re_{01}^{M\times N}$ and $\boldsymbol{X}^+ \in \Re_+^{M\times N}$ respectively. Applications where an overcomplete dictionary –a set of linearly dependent vectors– which is tuned to a stimulus ensemble $\boldsymbol{X}^+$, so that signals drawn from the ensemble have sparse representations in the dictionary, arise in source separation [15], finance and semantic and sentiment analysis. Given the matrix $\boldsymbol{X}^+$, NMF decomposes $\boldsymbol{X}^+$ into the product of two matrices, $\boldsymbol{W}^+ \in \Re_+^{M\times R}$ and $\boldsymbol{H}^+ \in \Re_+^{R\times N}$ where all matrices have exclusively nonnegative elements ($M > R, N > R$). NMF-Frobenius' objective is the squared-ℓ_2 norm:

$$D_F(\boldsymbol{X}^+||\boldsymbol{W}^+\boldsymbol{H}^+) = \frac{1}{2}\sum_{m,n}|\boldsymbol{X}^+_{m,n} - [\boldsymbol{W}^+\boldsymbol{H}^+]_{m,n}|^2. \quad (1)$$

A suitable step-size parameter, proposed by Lee and Seung in [16], results in two alternating, multiplicative, gradient descent updating algorithms (the datatype qualifier is leftout as it is clear from the context):

$$\boldsymbol{W} \leftarrow \boldsymbol{W} \odot \boldsymbol{X}\boldsymbol{H}^T \oslash \boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T, \quad (2)$$

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \odot \boldsymbol{W}^T\boldsymbol{X} \oslash \boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H} \quad (3)$$

where $\odot$ represents element-wise multiplication, and $\oslash$ is element-wise division. The NMF solution is generally not unique, or exact. For every invertible $\boldsymbol{A}$ we have a potential factorization [17, 18],

$$\boldsymbol{X} \approx (\boldsymbol{W}\boldsymbol{A})(\boldsymbol{A}^{-1}\boldsymbol{H}). \quad (4)$$

In the present paper, we aim to solve the *uniqueness* problem and to address the *inexactness* of NMF, and also for good measure, the permutation/scaling problem.

When $\boldsymbol{A} = \boldsymbol{P}\boldsymbol{D}$, a permutation matrix $\boldsymbol{P}$ and a diagonal matrix $\boldsymbol{D}$, the elements or $\boldsymbol{A}^{-1}$ are the reciprocal of the non-zero elements in $\boldsymbol{A}$ or zero; Thus $\boldsymbol{A}^{-1}$ is also a permutation times a diagonal matrix. The condition that $\boldsymbol{W}$ and $\boldsymbol{H}$ are nonnegative is not sufficient to eliminate the permutation/scaling degree of freedom of NMF. In short, we aim to learn $\boldsymbol{W}$ and $\boldsymbol{H}$ so that the matrix $\boldsymbol{A} = \boldsymbol{I}$, is the canonical basis, and the solution is exact, $\boldsymbol{X} = \boldsymbol{W}\boldsymbol{H}$. This problem is expressed concisely as follows:

Problem 1 *(Unique, Exact, Permutation, Scaling-free NMF): Given $\boldsymbol{X} \geq 0$, decompose $\boldsymbol{X}$ into the factorization $\boldsymbol{W}\boldsymbol{H}$ such that $\boldsymbol{X} = \boldsymbol{W}\boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{H}$, $\boldsymbol{W}\boldsymbol{A} \geq 0$, $\boldsymbol{A}^{-1}\boldsymbol{H} \geq 0$ and the matrix $\boldsymbol{A}$ has the form $\boldsymbol{A} = \boldsymbol{I}$.*

III Introducing the Ghostly spectre: NMF's short-comings

We consider a binary image dataset in order to introduce the uniqueness problem first. Donoho and Stodden constructed a library of images showing a stick figure with four limbs going through a range of motions to illustrate the ideas of a SFAF [7]. We will focus on this example, namely the *swimmers data-set* as our touch-stone example. Some of these swimmers are illustrated in Fig. 1. White denotes zero, black denotes ones in all figures. They consist of four body parts (limbs); Each of these parts has four possible articulations, (horizontal left/right, up/down, diagonal up/down). The limbs and articulations are illustrated in Fig. 2. We posit that in addition, this set of parts should include a torso –the **I**-shaped body part. It is this torso which causes incorrect NMF decompositions.

In the swimmers dataset there are 256, 32×32 images. Each image contains a torso of 17 pixels in the centre and four 5 pixel limbs. Using all combinations of the four limbs gives the 256 image dataset we use to illustrate our approach.

Remark: The torso component is present in each swimmer in Fig. 1. However, this presence causes a ghostly version of the **I**-shaped body part to appear in NMF decompositions (cf. Fig. 3).

The NMF generative model

$$\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H} \quad (5)$$

is a good candidate mixing model for these simple image settings: each scene in Fig. 1 is composed of standard limbs (in Fig. 2) in various articulations. The rows of the matrix $\boldsymbol{H}$ should hold various articulations of the limbs. The images in rows of $\boldsymbol{X} \in \Re_+^{256\times 1024}$ consist of superpositions of the parts, weighted by the values in the columns of $\boldsymbol{W}$. When a part is present it has a positive activation; When it is not, it has a zero activation.

Example: To illustrate the problem we wish to solve, we run NMF for 5000 alternating iterations on the swimmers datset and plot the parts learned in the decomposition in Fig. 3. The problems with

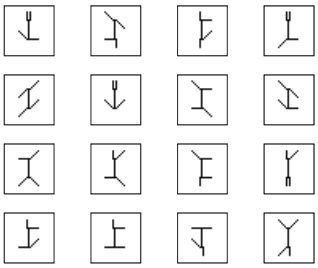

Fig. 1: Swimmers dataset: A sample of the images taken from Donoho and Stodden's library of swimmers.

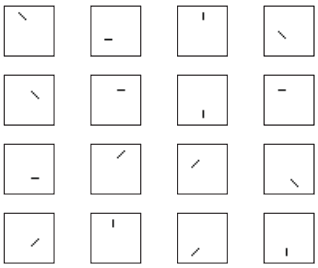

Fig. 2: Limbs of the swimmers dataset: All four limbs in all four articulations that are used to construct the swimmers are plotted. The torso is not plotted: The torso is a separate part –A 17^{th} part.

the derived solution are listed as follows: **1:** Each part is mixed with the torso (Donoho and Stodden label this torso a *ghost*); **2:** Only 16 and not 17 parts are learned. Increasing the rank of the decomposition does not improve the situation (by demixing the torso); **3:** The solution is not unique; The solution is not exact; **4:** The parts are permuted (randomly); **5:** The dataset is binary (not necessarily zeros and ones). Small random values must be added to each matrix entry before decomposition in order to ensure that the data is in fact nonnegative (there are no zero values): the NMF update rules are guaranteed to improve the objective monotonically if the data, and factors are initialized to be nonnegative. **6:** There exists no NMF formulation which allows the user to add information about the torso into the decomposition.

IV Banishing the Ghostly Spectre: Towards Ordered Closures

A unique NMF decomposition for the swimmers exemplar problem is discussed. First we show that NMF learns a mixture of rank-one approximations of $\boldsymbol{X}$; then we show that closures are rank-one approximations with appealing properties.

What is meant by unique here, is that any time NMF is run, all four limbs in all four articulations are learned (cf. Fig. 2) in the rows of the matrix $\boldsymbol{H}$. Moreover, the matrix $\boldsymbol{H}$ should be element-wise binary, not element-wise nonnegative. In addition, the torso should also be learned as a basis function

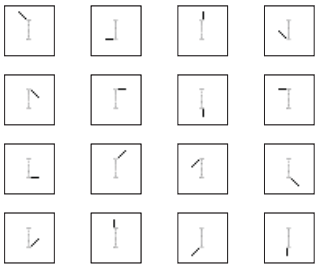

Fig. 3: NMF decomposition of the swimmers dataset: We run NMF 5000 times on the swimmers datset and plot the parts learned in the decomposition. Each part is mixed with the torso, Donoho and Stodden's ghost torso.

for this dataset in a row of $\boldsymbol{H}$. This is the *correct* solution; The underpinning criteria for correctness here is that the decomposition is parts-based –the parts are learned exactly.

To start, let's view the binary matrix $\boldsymbol{X}$ considered by Donoho and Stodden as a binary relationship between two sets of labels, which identify the rows and columns of the matrix $\boldsymbol{X}$. By definition, $\mathcal{R} = \{\mathfrak{r}_1, \mathfrak{r}_2, \ldots, \mathfrak{r}_M\}$ and $\mathcal{C} = \{\mathfrak{c}_1, \mathfrak{c}_2, \ldots, \mathfrak{c}_N\}$ are the sets of labels assigned to the rows and columns of $\boldsymbol{X}$. In words, the swimmer $\mathfrak{r}_1$ activates the following sets of pixels $\{\mathfrak{c}_n\}$. The torso's presence in the dataset is described by the set of pairs of labels

$$\{\{\mathfrak{r}_m, \mathfrak{c}_n\} | \mathfrak{r}_{10} \leq \mathfrak{r}_m \leq \mathfrak{r}_{22}, \mathfrak{r}_m \in \mathbb{Z}, \mathfrak{c}_n = \mathfrak{r}_{16}\}. \quad (6)$$

Each pair gives the position of a "one" in the matrix $\boldsymbol{X}$ which forms part of the torso. A similar set of pairs-of-labels describes each limb in each articulation. Seventeen sets (of sets of pairs of labels) describe the parts of the swimmers dataset.

Proposition 1 *One interpretation of the NMF mixing model is that it is the sum of an ensemble of rank-one approximations of the dataset.*

Example: There are many possible rank-one matrices which may be used to represent $\boldsymbol{X}$. The torso and each swimmer part (in each articulation) –denoted by the row vectors $\boldsymbol{H}_{r,:}$– times their corresponding activation vector –the column vectors $\boldsymbol{W}_{:,r}$– have the property that they are rank-one approximations of the entire dataset. The swimmers dataset is made up of $R = 17$ *parts-based* rank-one matrices. The problem lies in detecting the correct ones.

Let's be more selective. We defer mention of the mechanism for incorporating selectivity via parts-based rules until later. We are interested in the set of rank-one matrices which are also closures.

Proposition 2 *Each limb in each articulation plus the torso, is a rank-one binary matrix, e.g.* $(\boldsymbol{H}_{torso,:} + \boldsymbol{H}_{leg\ left\ 1,:})$. *Each of these rank-one binary matrices is a closure. Moreover, the torso is also a separate closure.*

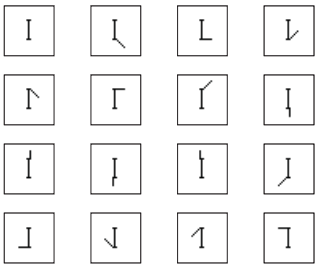

Fig. 4: Some closures (1 - 16) from the swimmers dataset.

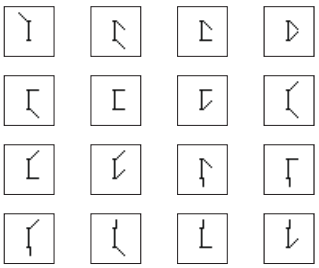

Fig. 5: Some closures (17 - 32) from the swimmers dataset.

To fix ideas, we give some examples of these rank-one matrices ($\boldsymbol{W}_{:,\text{activation}}$ denotes the corresponding activation vector): $\boldsymbol{W}_{:,\text{torso}}\boldsymbol{H}_{\text{torso},:}$, $\boldsymbol{W}_{:,\text{activation}}\left(\boldsymbol{H}_{\text{torso},:}+\boldsymbol{H}_{\text{leg left 1},:}\right)$, $\boldsymbol{W}_{:,\text{activation}}\left(\boldsymbol{H}_{\text{torso},:}+\boldsymbol{H}_{\text{leg left 2},:}\right)$. These matrices describe some of the closures illustrated in Figures. 4 and 5.

Remark: It is significant is that the torso is present in this set of closures as a separate closure. Moreover, there is an ordering on the closures. The closures are plotted here using a permuted ordering for ease of illustration, starting with the torso in the upper left hand plot, row-wise (the plot in row one, column two is next). Each subsequent closure includes a *full-bodied* torso –the torso is no longer ghostly. By set subtraction we can remove the effect of the torso from all closures save the torso closure –the ghost is eliminated. It is this *parts-based rule* that gives its name to this paper. This subtraction is possible due to the properties of closures.

To develop the relationship between the rank-one approximations, closures and a framework for parts-based rules we introduce some notation.

V The Ghostbusters Algorithm

Lectic ordering is defined *ab initio* by arranging the labels $\{\mathfrak{c}_1, \mathfrak{c}_2, \ldots\}$ of $\boldsymbol{X}$: the ordering may be arbitrary. The default ordering is $\mathfrak{c}_1 < \mathfrak{c}_2 \ldots < \mathfrak{c}_N$.

We can apply the *derivation* operator on subsets of $\mathcal{R}$ and $\mathcal{C}$, respectively, $\mathcal{X}_R$ and $\mathcal{X}_C$:

$$\mathcal{X}_R' = \{\mathfrak{c} \in \mathcal{C} \mid \forall \mathfrak{r} \in \mathcal{R} : (\mathfrak{r}, \mathfrak{c}) \in \boldsymbol{X}\} \quad (7)$$

$$\mathcal{X}_C' = \{\mathfrak{r} \in \mathcal{R} \mid \forall \mathfrak{c} \in \mathcal{C} : (\mathfrak{r}, \mathfrak{c}) \in \boldsymbol{X}\}. \quad (8)$$

We generate a closure by applying these derivation operators twice, the mappings,

$$\mathcal{X}_R \mapsto \mathcal{X}_R,'' \text{ and } \mathcal{X}_C \mapsto \mathcal{X}_C''. \quad (9)$$

These mappings have the properties that:

$$\mathcal{X}_C \subseteq \mathcal{C}, \quad \mathcal{X}_R \subseteq \mathcal{R}, \quad \mathcal{X}_R' = \mathcal{X}_C, \quad \mathcal{X}_C' = \mathcal{X}_R. \quad (10)$$

Example: Application of the closure operator on the rank-one swimmers' limbs, yield closures which consist of the limbs plus the torso.

All that is required now is a systematic method to search for all possible closures in the swimmers dataset. This method should preferably yield a unique ordered set of closures.

Algorithm 1 AllClosure

Input: $\{\{\}, \mathcal{C}\}, \boldsymbol{X}, \mathcal{C}, \mathcal{R}$: starting/stopping closures
Output: $\mathcal{D}$: derived closures.

```
Initiate process: D = {}, {}_a X_C = {}.
while {}_a X_C is not the last closure C do
    [{}_a X_C] = NextClosure(R, C, X, {}_a X_C);
    D ← D ∪ {}_a X_C;
end while
```

Algorithm 2 NextClosure

Input: $\mathcal{R}, \mathcal{C}, \boldsymbol{X}, {}_a\mathcal{X}_C$
Output: ${}_a\mathcal{X}_C$.

```
for c_n from c_N down to c_1 do
    if c_n ∉ {}_a X_C then
        {}_b X_C ← {}_a X_C ⊕ c_n;
        if {}_a X_C ≤_{c_n} {}_b X_C then
            {}_a X_C ← {}_b X_C;
            break;
        end if
    end if
end for
```

We appeal to a procedure called *NextClosure*, a well-known application of lattice and order theory [19], to build the Galois lattice of $\boldsymbol{X}$, using an algorithm proposed in [19, 20] and made more efficient by distribution in [13] and again by parallelization [12]. We then convert the lattice $\mathcal{D}$ into an ordered ensemble-tuned dictionary $\boldsymbol{H}$. This lattice has the property that it is unique and complete. Starting from the empty set $\mathcal{X}_C = \{\}$, given $\mathcal{X}_C \subseteq \mathcal{C}$, and $\mathfrak{c}_n \subset \mathcal{C}$, we may generate all closures by systematically applying the rule,

$${}_b\mathcal{X}_C = {}_a\mathcal{X}_C \oplus \mathfrak{c}_n := (({}_a\mathcal{X}_C \cap \{\mathfrak{c}_1, \ldots, \mathfrak{c}_{n-1}\}) \cup \{\mathfrak{c}_n\})'', \quad (11)$$

which augments the current closure, ${}_a\mathcal{X}_C$, by concatenating successive elements of $\mathcal{C}$ (largest-to-smallest) and keeping the resulting set, ${}_b\mathcal{X}_C$, if it is a closure and it is lectically smaller than any closure already mined, a condition which is verified by checking:

$${}_a\mathcal{X}_C \leq_{\mathfrak{c}_n} {}_b\mathcal{X}_C :\Longleftrightarrow \exists_{\mathfrak{c}_n}(\mathfrak{c}_n \in {}_b\mathcal{X}_C, \mathfrak{c}_n \notin {}_a\mathcal{X}_C, \forall_{\mathfrak{c}_j < \mathfrak{c}_n}(\mathfrak{c}_j \in {}_a\mathcal{X}_C \Longleftrightarrow \mathfrak{c}_j \in {}_b\mathcal{X}_C)). \quad (12)$$

Algorithm 3 Ghostbusters

Input: $\boldsymbol{X}$

Output: $\mathcal{D}, \boldsymbol{W}, \boldsymbol{H}$: derived closures, parts and activations matrices.

1: $[\mathcal{D}] = \text{AllClosure}(\mathcal{R}, \mathcal{C}, \boldsymbol{X}, \{,\}, \mathcal{C})$;
2: Initialization process: generate a dictionary of atoms $\boldsymbol{H} \in \Re_{01}^{|C| \times N}$ by stacking row-vectors, which are constructed by inserting ones in the entries given by the sets $\mathcal{D}$, and zeros elsewhere.
3: Check for ghosts: Construct the row vector $\boldsymbol{j}_n = \sum_{i=1}^{|C|} \boldsymbol{H}_{i,n}$.
4: De-ghost the ensemble tuned dictionary:

$$\boldsymbol{H}_{i,n} = \begin{cases} 0, & \text{if } \boldsymbol{j}_n = |C|, \text{ and } i > 1, \\ \boldsymbol{H}_{i,n}, & \text{if } \boldsymbol{j}_n \neq 0, \text{ and } i > 1. \end{cases} \tag{14}$$

5: Regularization via ***Parts-based rules***. Encode parts-based rules using encoding matrix $\boldsymbol{H} = \boldsymbol{E}\boldsymbol{H}$.
6: **if** Sufficient rank $\boldsymbol{H}$ **then**
7: An exact binary decomposition of $\boldsymbol{X}^+$ is obtained by solving for the matrix $\boldsymbol{W} \in \Re_+^N$:

$$\begin{aligned} &\text{minimize}_{\boldsymbol{W}(p,:)} \boldsymbol{W}(p,:)\mathbf{1}, \\ &\quad \text{subject to } \boldsymbol{X}(p,:)^T = \boldsymbol{H}^T \boldsymbol{W}(p,:)^T, \\ &\qquad \boldsymbol{W}(p,:) \geq 0. \end{aligned} \tag{15}$$

8: **else**
9: Lee-Seung activation update for low-rank $\boldsymbol{W}$.
10: Project $\boldsymbol{X}, \boldsymbol{H}$ into nonnegative orthant if necessary: $\boldsymbol{W} \rightarrow \boldsymbol{W} \odot \boldsymbol{X}\boldsymbol{H}^T \oslash \boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T$.
11: **end if**

A property of NextClosure is that the closure set is unique and complete [19, 20], and is indexed in lectically increasing order using, ${}_r\mathcal{X}_C$, where $r = 1, 2 \ldots, R$,

$${}_1\mathcal{X}_C \leq {}_2\mathcal{X}_C \leq \cdots {}_r\mathcal{X}_C \leq \cdots {}_R\mathcal{X}_C. \tag{13}$$

This algorithm is described in Alg. 1 and 2. What is appealing is that it is as simple to implement as the NMF procedure.

The Ghostbusters algorithm is described in Alg. 3. Eqn. 14 describes the removal of the omnipresent torso. It is one example of a *parts-based rule* which is a general framework embodying the parts-based selectivity mentioned above. We encode prior information to the solver (7-10 in Alg. 3) using $\boldsymbol{E} \in \Re_{01}^{|C| \times |C|}$. We now solve

$$\boldsymbol{X} = (\boldsymbol{W}\boldsymbol{A})\left(\boldsymbol{A}^{-1}\boldsymbol{H}\right) = \boldsymbol{W}\left(\boldsymbol{E}\boldsymbol{H}\right). \tag{16}$$

To recover the swimmer parts we introduce additional constraints on the complexity of the parts that are used by the activations, specificially, restrictions on the length of the maximum vector norm, ℓ_0-norm of the closures, e.g $\ell_0(\boldsymbol{H}_{i,:}) = \#\{n | \boldsymbol{H}_{i,n} \neq 0\}$. In effect we are imposing Occam's razor –the principle of parsimony– on the closures and the activations using $\boldsymbol{E}$ by introducing ones on the diagonal for closures that satisfy the constraints, and a zeros for the rest.

Once an ordered ensemble-tuned dictionary is learned (steps 1-4 in Alg. 3), the corresponding activations must be determined: this problem is addressed (in this paper) by solving 1) a nonnegative least-squares, or 2) a nonnegative linear programming problem depending on the type of solution desired. The statistical interpretation of the first approach is maximum likelihood estimation, given linear measurements corrupted by noise – regularization may also be considered. Regularization is often used by the NMF community to encourage a parts-based decomposition [21]. For this swimmers dataset regularization is required. Appealingly, an ℓ_0-norm regularization contraint may be introduced here due to the binary form of the closures. In the latter case, the activations have a Laplacian prior and the observation model is noise-free; This is interpreted as MAP estimation. In both cases the matrix $\boldsymbol{H}$ generally has a sparse structure which speeds up the computation of $\boldsymbol{W}$. Nonnegative and parts-based datasets are frequently sparse. Solving for $\boldsymbol{W}$ is convex.

VI Empirical Evaluation and Discussion

We demonstrate a subtractive and a "complexity-reducing" parts-based rule using the removal of the omnipresent torso region in Donoho and Stodden's swimmers as a first test case. We then demonstrate how complexity-reducing constraints may be applied more generally to Hoyer's dataset (which has no omnipresent factor): We apply bars-like constraints on neighbouring pixels. In both cases, the incorporation of simple parts-based constraint yield the exact solution. The first dataset is binary, and the second is nonnegative. The motivation is to show that 1) a range of prior information can be incorporated into the solution and 2) NMF disregards this information.

a) Unique and Correct Swimmers Decomposition

We run the Ghostbusters algorithm on Donoho and Stodden's swimmers dataset to demonstrate that a unique and correct swimmers decomposition is achieved for the pathological SFAF problem.

By inspection we have prior information: 1) There are four limbs in four articulations (16 parts in total); 2) The torso is omnipresent; 3) The limbs vectors have low ℓ_0-norm. The principle of Occam's razor is incorporated when determining the solution by selecting the 16 closures with the smallest complexity, measured here using the ℓ_0-norm, and also by removing omnipresent features (encoded using $\boldsymbol{E}$). We argue that this choice is akin to selecting the parameter R for NMF, and therefore justifiable: prior information about the rank of the desired solution is incorporated by both methods, it is done by Ghostbusters using a complexity constraint. NMF, however, has no facility for extracting an omnipresent feature because: 1) The ghost learned in each NMF feature is not uniform in value in each parts vector or across each parts vector (subtracting the average ghost may

cause the parts to become negative). See for example the parts in Fig. 3; 2) Omnipresent vectors are typically not uniform valued in the dataset (similar problems to averaging and subtracting the ghost torso arise). Ghostbusters generates a closure, which is the torso, which is easily removed using *parts-based rules*, as illustrated above.

Complexity penalty parts-based rules are needed as some closures are linear combinations of other closures (cf. Fig. 6 and 7), the desired solution has low *parts* complexity. We aim to learn a sparse decomposition, but not decompositions which only activate the torso and a single other closure (one with four limbs in the correct particular articulation –a closure which is a linear combination of the limbs and articulations required). The total ordered ensemble-tuned dictionary is highly overcomplete: what is required is a low-rank solution.

Comparison: The activations of the swimmers dataset are generated using the NMF-Frobenius update $\boldsymbol{W} \rightarrow \boldsymbol{W} \odot \boldsymbol{X}\boldsymbol{H}^T \oslash \boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T$ and 16 vectors, preserved by $\boldsymbol{E}$, are plotted in Fig. 8. We run the NMF-Frobenius update for 5000 iterations, noise (of machine error order) is added to the dataset to ensure nonnegativity. Table 1 summarizes a comparison between Ghostbusters and NMF. Runtimes are given for an octave implementation on a 2.6GHz personal computer – they are guideline figures. NMF is run for 5000 iterations; Using an alternative measure of convergence, NMF may be stopped earlier. NMF suffers from the scaling problem: the sparsity measure of the $\boldsymbol{W}$ is therefore subject to this ambiguity. However, NMF consistently leads to poorer sparsity measures in these experiments (the parts it learns are mixed). NMF has a shorter runtime, yet the torso is mixed with each part –the swimmers problem is not solved. NMF has considerable approximation error, which is measured using (Eqn. 1), compared to Ghostbusters. The activations are approximately binary for Ghostbusters, whereas the activations for NMF are nonnegative. Note the ℓ_1-norm of $\boldsymbol{W}$ for NMF is uninformative as NMF suffers from the scaling ambiguity (which may be addressed by row-normalizing $\boldsymbol{H}$). In short, Ghostbusters is slower but the solution is correct and parts-based. Ghostbusters solves the problem; NMF does not.

Discussion: The significance of this result is explained.

1) The set of closures is unique, therefore the matrix $\boldsymbol{H}$ is unique. Arbitrary parts-based constraints may be applied in an ordered manner in Ghostbusters, this information is not (and cannot be) incorporated into the NMF solution.

2) Ghostbusters does not suffer from the scaling and permutation ambiguity; The matrix $\boldsymbol{H}$ is binary and unique. In addition, solving for $\boldsymbol{W}$ is a convex optimization problem. NMF performs an alternating minimization optimization which is convex in $\boldsymbol{W}$ or $\boldsymbol{H}$, but not in both factors.

3) The rank is encoded into the activation matrix

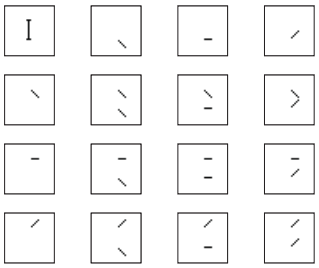

Fig. 6: Ghostbusters: Ordered closures (1 - 16) swimmers dataset. Not all closures are linearly independent.

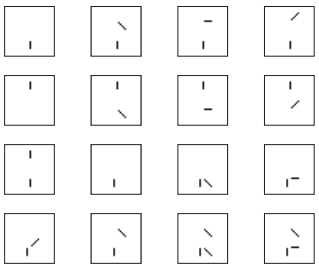

Fig. 7: Ghostbusters: Ordered closures (17 - 32) from the swimmers dataset.

solver based on user constraints on the complexity of the parts (using $\boldsymbol{E}$). We can think of the rank of the problem as the cardinality of the *active set* of the optimization problem (the diagonal of $\boldsymbol{E}$).

b) Binary Matrices and Nonnegative Matrices

The "bars" dataset was proposed by Hoyer in [15] in order to motivate sparse nonnegative coding problems. It is challenging for NMF as an overcomplete dictionary is required from a mixture of parts. Regularization is used to address this challenge: the choice of a suitable weighting term for the regularization parameter is difficult [21].

A second challenge lies in the fact that whilst the parts are binary, they are mixed synthetically with nonnegative values. However, parts-based constraints recommend themselves to this problem because the underlying parts are binary, and so we address the problem of learning binary parts from nonnegative matrices using Ghostbusters.

Nonnegative mixtures $\boldsymbol{X}^+ \in \Re^{100\times 9}$ are mixed (in Fig. 10) using the ground-truth *bars*, $\boldsymbol{H} \in \Re^{10\times 9}$ (reformulated as 3×3 matrices in Fig. 9) and uniformly distributed nonnegative activations $\boldsymbol{W} \sim \mathcal{U}(0,1)$. Noise is uniformly distributed.

$$\boldsymbol{X}^+ = \boldsymbol{W}\boldsymbol{H} + \boldsymbol{N}, \quad \boldsymbol{N}_{m,n} \sim \mathcal{U}(0, 0.1). \tag{17}$$

There is no omnipresent part (similar to the torso) in this dataset –this rule is not needed. The closures (36 excluding the empty closure) are computed from the bars for comparison purposes and

Table 1: Comparison: Ghostbusters with NMF

Ghostbusters	Swimmers	Hoyer
Closures	626 (separate torso)	10/69/212
Run time	700.3s (closures)	2.38s
$\boldsymbol{W}$-update	20s ($5000its$)	0.113s
Error	2×10^{-13}	0
	Convex in $\boldsymbol{W}$	–
Perm.-free	✔	✔
Scaling-free	✔	✔
Correctness	✔	✔
$\ell_1(\boldsymbol{W})$	1280 ($\boldsymbol{W} \in \Re^{256\times 17}$)	193.40
NMF	Swimmers	Hoyer
No. parts	16 (no separate torso)	10
Run time	59.924s (5000its)	15.6s (5000its)
Error	3.68×10^{-4}	4.1×10^{-4}
Perm.-free	×	×
Scaling-free	×	×
Correctness	Mixed	Mixed
$\ell_1(\boldsymbol{W})$	2014.2 ($\boldsymbol{W} \in \Re^{256\times 16}$)	492

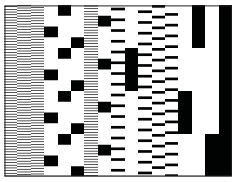

Fig. 8: Activations $\boldsymbol{W} \in \Re^{256\times 17}$ for the swimmers computed using parts-based encoding so that $\boldsymbol{X} \approx \boldsymbol{W}(\boldsymbol{EH})$. The torso is omnipresent (rightmost col.)

plotted in Fig. 11 in order to show the ground-truth closures. Parts-based rules are encoded in $\boldsymbol{E}$ to consider the following observations which are drawn by inspection of Fig. 10. This information is disregarded by NMF.
- There are no omnipresent closures.
- The ℓ_0-norm of the salient parts is 3 or 6.
- There are 10 parts. These parts are bar-like (parts have triplets-of-pixels that are horizontal or vertical neighbours).

Problem 2 *Given the matrix $\boldsymbol{X}^+$, decompose $\boldsymbol{X}^+$ into $\boldsymbol{W}^+ \in \Re^{M\times R}$ times $\boldsymbol{H}^+ \in \Re^{R\times N}$, subject to $D_F(\boldsymbol{X}^+||\boldsymbol{W}^+\boldsymbol{H}^+) = 0$, such that there is no permutation or scaling, and the rows of $\boldsymbol{H}$ yield a sparse signal representation.*

To deal with the noise, a NonLinear Approximation (NLA) [22] of the sparse matrix $\boldsymbol{X}^+$ produces its binary-relational counterpart, $\boldsymbol{X}$, with approximation error $D_F(\boldsymbol{0}||(\neg\boldsymbol{X}) \odot \boldsymbol{X}^+) < \epsilon$, where $\neg$ is element-wise negation,

$$\boldsymbol{X}_{m,n} = \begin{cases} 1, & \text{if } \boldsymbol{X}^+_{m,n} > T, \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

In this paper, the threshold parameter T is found quickly by application of the Armijo Rule or the Bisection method.
Comparison: In total, 212 closures are obtained from the NLA (Eqn. 18) of the mixture (Eqn. 17). They are illustrated in Fig. 12 and maybe directly compared with the closures computed using the raw bars (in Fig. 11), as both sets of closures are plotted in lectic ordering. What is clear is that

Fig. 9: Set of bars used to generate nonnegative mixtures.

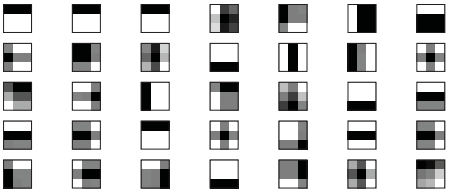

Fig. 10: A sample of the mixed noisy bars data.

the closures in Fig. 12 have higher ℓ_0-norm. Sixty-nine closures (in Fig. 12) satisfy the constraints that $\ell_o(\boldsymbol{H}_{i,:}) = 3$ or $\ell_o(\boldsymbol{H}_{i,:}) = 6$; However we may also encode bar-like constraints, $\boldsymbol{E}_{i,i} =$

$$\begin{cases} 1, & \text{if } (\boldsymbol{H}_{i,n} = 1 \& \boldsymbol{H}_{i,n+1} = 1 \& \boldsymbol{H}_{i,n+2} = 1) \\ & \text{and } \ell_0(\boldsymbol{H}_{i,:}) = 3 \text{ or } 6, \\ & \text{or } (\boldsymbol{H}_{i,n} = 1 \& \boldsymbol{H}_{i,n+3} = 1 \& \boldsymbol{H}_{i,n+2\times 3} = 1) \\ & \text{and } \ell_0(\boldsymbol{H}_{i,:}) = 3 \text{ or } 6 \\ 0, & \text{otherwise.} \end{cases}$$

Discussion: Only 10 closures satisfy the ℓ_0 and bar-like constraints. Ghostbusters is once again compared with Lee-Seung NMF in Table. 1 for this dataset with no prior information, save the rank $R = 10$. NMF's sparsity regularization involves trial-and-error tuning and is not performed here. NMF does not determine the correct solution; Ghostbusters does. During Monte Carlo trials, a different set of stimulus-tuned closures is learned by Ghostbusters for a given ϵ. Typically the ground-truth closures are amongst this set.

This paper raises the question of how to include a range of prior information into an NMF so that NMF generates the *appropriate* parts-based representation. Frequently, the solution can almost be "picked-out-by-eye" and yet NMF frustrates by learning a good parts-based solution, but not the solution the user wants. The swimmers dataset is a case-in-point. Allowing the user to encode a set of criteria into the solver, in order to specify the type of solution that is interesting to him, is a powerful concept; It raises a fundamental question. Is this technique supervised or unsupervised?

Decomposing the magnitude spectrogram of speech is one problem NMF has been applied to. The relationship between the inter-formant frequency distances, though well-understood by the community, is not used in NMF decompositions [21]. As a result speech phones are sometimes separated into high frequency features and low frequency features, as the appropriate rank is not known, and the solution is not parts-based. We

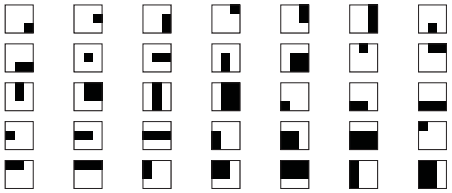

Fig. 11: Sample of closures for the unmixed bars.

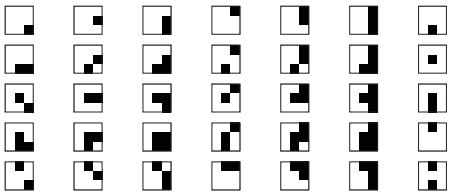

Fig. 12: Sample of closures for the mixed bars.

will investigate if improvement is speech representations may be achieved by incorporating this domain-expertise. By exploiting the binary datatype of closures we have proposed a framework which allows the user to be prescriptive when formulating the desired form of the NMF solution.

VII Conclusions

The label "parts-based" implies that the data is composed of simple building blocks, which may often be identified by eye. Encoding parts-based constraints in NMF algorithms is difficult: NMF does not allow for this level of direct specificity. We introduce a framework based on closure-finding. Once a set of suitable closures has been identified, parts-based based constraints may be easily incorporated into the optimization routine. Ghostbusters has a number of advantages over NMF: 1) It learns overcomplete representations; 2) It allows for the encoding of arbitrary constraints; 3) It is unique, correct, permutation and scaling free; 4) It can learn sparser solutions which are exact.

This work was supported by SFI via 08/SRC/I1403 FAME SRC and 11/TIDA/I2024.

References

[1] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. In *Nature*, pages 788–91, 1999.

[2] R. de Fréin, K. Drakakis, and Scott Rickard. Portfolio diversification using subspace factorizations. In *Inf. Scien. Sys., 42nd Ann. Conf.*, pages 1075–80, 2008.

[3] P.D. O'Grady and B.A. Pearlmutter. Hard-lost: Modified k-means for oriented lines. *Proc. Irish Sig. and Sys. Conf.*, pages 247–52, 2004.

[4] D. FitzGerald. User assisted source separation using non-negative matrix factorisation. *22nd IET Irish Sig. and Sys. Conf.*, 2011. Dublin.

[5] R. Jaiswal, D. FitzGerald, E. Coyle, and S.T. Rickard. Shifted nmf using an efficient constant-q transform for monaural sound source separation. *22nd IET Irish Sig. and Sys. Conf.*, 2011. Dublin.

[6] P.D. O'Grady and S.T. Rickard. Automatic hexaphonic guitar transcription using non-negative constraints. *Proc. Irish Sig. and Sys. Conf.*, June 2009. Dublin.

[7] Donoho D. and V. Stodden. When does non-negative matrix factorization give correct decomposition into parts? In *Proc. Neur. Inf. Proc. Sys.* MIT Press, 2003.

[8] Z. Zhang, Li T., C. Ding, and X. Zhang. Binary matrix factorization with applications. *Proc. 17th IEEE Int. Conf. on Data Min.*, pages 391–400, 2007. Washington, DC, USA.

[9] P. Miettinen, T. Mielikainen, A. Gionis, G. Das, and H.; Mannila. The discrete basis problem. *Knowl. and Data Eng., IEEE Trans.*, 20(10):1348–62, Oct. 2008.

[10] P. Miettinen. Sparse boolean matrix factorizations. *IEEE 10th Int. Conf. on Data Min.*, pages 935–40, Dec. 2010.

[11] T. Li. A general model for clustering binary data. *In Proc. 11th ACM SIGKDD int. conf. on Knowl. Disc. in Data Mining*, pages 188–97, 2005.

[12] R. de Fréin. Formal Concept Analysis via Atomic Priming. In *Formal Concept Analysis*, volume 7880 of *LNAI*, pages 92–108. Springer, 2013.

[13] B. Xu, R. de Fréin, E. Robson, and M. Ó Foghlú. Distributed Formal Concept Analysis Algorithms Based on an Iterative MapReduce Framework. In *Formal Concept Analysis*, volume 7278 of *LNCS*, pages 292–308. Springer, 2012.

[14] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Commun. ACM*, 51(1):107–13, 2008.

[15] P.O. Hoyer. Non-negative sparse coding. In *Proc. IEEE Workshop on Neur. Nets for Sig. Proc.*, pages 557–65, 2002.

[16] D.D. Lee and S.H. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–62. MIT Press, 2000.

[17] S. Rickard and A. Cichocki. When is non-negative matrix decomposition unique? In *IEEE 42nd Ann, Conf. Inf. Scien. Sys.*, pages 1091–2, 2008.

[18] H. Laurberg, M.G. Christensen, M.D. Plumbley, L.K. Hansen, and S.H. Jensen. Theorems on positive data: On the uniqueness of nmf. *Comput Intell Neurosci*, 2008, 2008.

[19] R. Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In *Formal Concept Analysis*, volume 5548 of *LNCS*, pages 314–39. Springer Berlin Heidelberg, 2009. Originally published: Ordered Sets, 1982.

[20] B. Ganter. Two Basic Algorithms in Concept Analysis. volume 5986 of *LNCS*, pages 312–40. Springer Berlin Heidelberg, 2010. Originally published: 1984.

[21] Paul D. Ogrady and Barak A. Pearlmutter. Discovering speech phones using convolutive non-negative matrix factorisation with a sparseness constraint. *Neurocomp.*, pages 88–101, 2008.

[22] C. Weidmann and M. Vetterli. Rate distortion behavior of sparse sources. *Inf. Th., IEEE Trans.*, 58(8):4969–92, Aug. 2012.