Learning Overcomplete Dictionaries with $\ell_0$-Sparse Non-negative Matrix Factorisation

Ken O’Hanlon, Mark D. Plumbley
Centre for Digital Music
Queen Mary University of London
London, UK
{keno, Mark.Plumbley}@eecs.qmul.ac.uk

Abstract—Non-negative Matrix Factorisation (NMF) is a popular tool in which a ‘parts-based’ representation of a non-negative matrix is sought. NMF tends to produce sparse decompositions. This sparsity is a desirable property in many applications, and Sparse NMF (S-NMF) methods have been proposed to enhance this feature. Typically these enforce sparsity through use of a penalty term, and a $\ell_1$ norm penalty term is often used. However an $\ell_1$ penalty term may not be appropriate in a non-negative framework. In this paper the use of a $\ell_0$ norm penalty for NMF is proposed, approximated using backwards elimination from an initial NNLS decomposition. Dictionary recovery experiments using overcomplete dictionaries show that this method outperforms both NMF and a state of the art S-NMF method, in particular when the dictionary to be learnt is dense.

Index Terms—sparse, non-negative, dictionary learning, NMF

I. INTRODUCTION

Non-negative Matrix Factorisation (NMF) is a learning algorithm which seeks the approximation

$$M = DX$$

(1)

where $M \in \mathbb{R}^{M \times N}$ is the matrix for which the factorisation is sought, $D \in \mathbb{R}^{M \times K}$ and $X \in \mathbb{R}^{K \times N}$ are a dictionary matrix containing template atoms and a coefficient matrix with each row containing the activations of the corresponding dictionary atom, respectively, and $M, D, X \geq 0$. NMF was originally proposed by Paatero and Tapper [1], in which it was proposed to minimise a Euclidean distance cost function:

$$C_E = \|M - DX\|_F^2$$

(2)

using Alternating Non-negative Least Squares (ANLS) projections. NMF was popularised by Lee and Seung [2] who proposed using fast multiplicative gradient descent updates instead of the ALS methodology. While NMF algorithms have been proposed for many different cost functions [3], the Euclidean distance NMF is popular for many applications, and several proposals have been proposed for performing fast ANLS by taking various approaches to the Non-Negative Least Squares (NNLS) subproblems [4] [5] [6] [7].

A noted feature of NMF factorisations is that they tend to be sparse, a desirable property that several authors have proposed to augment by introducing a sparse penalty term:

$$C_S = \frac{1}{2} \|M - DX\|_F^2 + \lambda \sum_{n=1}^{N} \|x_n\|_p$$

(3)

where $\lambda$ is a parameter that controls the sparsity and $\|\cdot\|_p$ is an $\ell_p$ vector norm. Typically in the NMF literature, a $\ell_1$ norm is used as the penalty term on the activation matrix, which can be seen as a non-negative matrix variant of the LASSO [8], or Basis Pursuit Denoising (BPDN) [9]. This was first proposed by Hoyer [10] in the multiplicative update framework. In the ALS framework, Kim and Park [6] proposed to apply a squared $\ell_1$-penalty term, $\lambda \|x\|_2^2$, that can be considered a $\ell_1$-penalty that scales the signal. An $\ell_1$ penalty term is effected using an Iterative Soft Thresholding (IST) [11] approach that is known to converge for LASSO / BPDN, in [12], where the authors also propose starting with a large value of $\lambda$ that is gradually decreased. However, the use of an $\ell_1$ penalty may not be optimal in a non-negative framework. It has been shown recently that Thresholded NNLS outperforms non-negative $\ell_1$-minimisation, such as LASSO/BPDN due to the innate regularisation of the non-negative constraint [13]. Indeed, in [12] an iterative strategy using hard thresholding was often seen to perform better than the IST approach, the authors noting that the hard thresholding strategy tends towards an $\ell_0$ penalty, where $\|x\|_0 = |x| \neq 0$.

Other Sparse NMF approaches which seek to approximate a $\ell_0$ norm include NMF-\$\ell_0$ [14] and Non-negative K-SVD [15], characterised by using a different approach to the dictionary update step, such as a modified K-SVD algorithm [15], or repeated NMF updates [14]. In both cases, pursuit algorithms are used, with an emphasis on matching pursuits, with stopping conditions determined by a predetermined number of atoms, [15] [14], or a relative error measure [15].

The focus of this paper is on the use of an $\ell_0$ penalty for sparse NMF. In particular a backwards elimination approach using a modified sparse cost function that we have seen previously to be effective in non-negative sparse decompositions [16] is employed for the sparse approximation step. While an optimal solution to the problem may not be guaranteed, the backward elimination step is locally optimal, and does not require the use of a predetermined number of atoms, or relative error condition as a stopping condition. In the rest of this paper, the relevant background material is first briefly described before introducing the proposed methodology. Experimental results on synthetic data are offered, which validate the approach taken before concluding with pointers to future work.

(c) 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works for resale or redistribution to servers or lists, or reuse of any copyrighted components of this work in other works. Published in: Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing (GlobalSIP), December 3-5, 2013, Austin, Texas, USA, pp. 977-980. DOI:10.1109/GlobalSIP.2013.6737056
II. BACKGROUND

A. NMF with ANLS

While NMF was popularised using multiplicative updates, the ANLS approach is generally considered to perform better converging to a lower value of the cost function. The ANLS approach performs alternating NNLS projections:

\[ X \leftarrow \min_X \| M - DX \|_F^2 \quad s.t. \quad X \geq 0 \]  \hfill (4)

to update the coefficient matrix and

\[ D \leftarrow \min_D \| M^T - X^T D^T \|_F^2 \quad s.t. \quad D \geq 0 \]  \hfill (5)

to update the dictionary. The computational load of using NNLS projections for the subproblems has been noted and variants of the ANLS algorithm have been proposed using projected gradient [5], optimised active set [4], block pivoting [6] and coordinate descent [7] methods in order to counter this computational load. A noted ability of ANLS methods relative to multiplicative update methods is their ability to handle overcomplete dictionaries [7].

B. Backward Elimination

Backwards elimination is a stepwise strategy that starts with an initial set, \( \Gamma \), of indices of supported atoms, and eliminates an atom with index \( \hat{k} \), such that \( \Gamma \leftarrow \Gamma \setminus \hat{k} \) at each iteration such that

\[ \hat{k} = \arg \min_k \Delta^k r \]  \hfill (6)

where

\[ \Delta^k r = \| \bar{r}^k \|_2^2 - \| r^k \|_2^2 \]  \hfill (7)

where \( r^i \) is the residual at the current matrix and \( \bar{r}^k \) is the hypothetical residual given the sparse support \( \Gamma^k = \Gamma \setminus \hat{k} \). A fast elimination criteria is proposed as part of the Greedy Sparse Least Squares (GSLS) algorithm [17], derived through using the block matrix inversion formulae:

\[ \Delta r = \hat{x}^T \otimes \text{diag}(|D_k^T D_k|^{-1}) \]  \hfill (8)

which calculates \( \Delta^k r \) for all \( k \) simultaneously, where \( \hat{x} \) is the least squares solution vector given \( \Gamma \), the current support, \( a^0 \) denotes elementwise power of \( a \) and \( \otimes \) denotes elementwise division.

III. METHOD

A. Modified Sparse Cost Function

In the sparse representations literature an \( \ell_0 \) penalised least squares solution is considered optimal. In an orthonormal basis [18], hard thresholding using a threshold \( \lambda \) is equivalent to

\[ C_T = \| m - D x \|_F^2 + \lambda^2 \| x \|_0. \]  \hfill (9)

We have observed empirically found that in a non-negative framework, that better sparse approximation, using the backwards elimination framework, occurs with a slight modification to this cost function [16] using the residual norm

\[ C_{mod} = \| m - D x \|_2 + \lambda \| x \|_0 \]  \hfill (10)

for which the motivation comes from the observation that the backwards elimination criteria (8) scales to the square of \( \hat{x} \).

The backwards elimination step (6) can be seen to locally optimise the sparse cost function (9) when \( \Delta^k r < \lambda^2 \). In order to optimise the modified sparse cost function the measure

\[ \Delta^k r = \sqrt{\| r \|_2^2 + \Delta^k r} - \| r \|_2 \]  \hfill (11)

can be simply calculated, noting that \( \hat{k} \), the index of the selected atom is the same regardless of the cost function applied. The stopping condition for the backwards elimination criteria is then given similarly as \( \Delta^k r < \lambda \).

B. \( \ell_0 \)-Sparse NMF

A sparse NMF algorithm is proposed which uses the modified sparse cost (10), an approach referred to as \( \ell_0 \)-Sparse NMF, (\( \ell_0 \)-S-NMF) outlined in Fig. 1. \( \ell_0 \)-Sparse NMF, as outlined here follows the ANLS approach. After initialisation of the dictionary \( D \), the \( \ell_0 \)-S-NMF enters an iterative loop. First NNLS is performed to estimate the coefficient matrix \( X \). Approximation of the minimum of (10) with a non-negative constraint is then performed using backwards elimination at each column of the initial NNLS decomposition.

\[ \text{Input } M \in \mathbb{R}^{M \times N}, \Gamma, \lambda \]

Initialise \( D \in \mathbb{R}^{M \times K} \)

\begin{verbatim}
repeat
    Calculate \( X \) using (4)
    for \( n = 1:N \) do
        \( \Gamma_n = \{ k | [X]_{k,n} > 0 \} \)
        \begin{verbatim}
            repeat
                Select \( \hat{k}_n \) using (6) (8)
                \( \Gamma_n \leftarrow \Gamma_n \setminus \hat{k}_n \)
            until \( \Delta^k r_n < \lambda \)
        \end{verbatim}
        \( x_n = \arg \min_x \| m_n - D_{\Gamma_n} x \|_2^2 \quad s.t. \quad x \geq 0 \)
    end for
    Update \( D \) using (5)
until stopping condition
\end{verbatim}

Fig. 1: \( \ell_0 \) S-NMF Algorithm

After the elimination process is completed \( X \) is recalculated using NNLS constrained to the new sparse support, \( \Gamma \). The final step of the iteration consists of re-estimating \( D \), using the transposed NNLS projection (5).

In some cases it may be useful to enforce sparsity in the dictionary, using a cost function such as

\[ C_{mod} = \sum_n \left( \| m_n - D x_n \|_2 + \lambda \| x_n \|_0 \right) + \eta \| D \|_0 \]  \hfill (12)

where \( \| D \|_0 \) is the number of non-zero elements in the dictionary. The reason that the ANLS approach is considered is that sparsity of the dictionary can also be enforced using the backwards elimination approach, applied to the transposed NNLS problem (5) with a stopping condition \( \Delta^m r < \eta \).
IV. EXPERIMENTS

Some synthetic dictionary recovery experiments were designed to test the proposed approach. Random, twice-overcomplete non-negative dictionaries \( \mathbf{D} \) of dimension 200 \( \times \) 400 were generated, using a flat equal probability distribution in the range \([0,1]\), and all dictionary columns were normalised to unit \( \ell_2 \) norm. A coefficient matrix \( \mathbf{X} \) of dimension 200 \( \times \) 800 was synthesised using an equal distribution in \([0.021]\). Between 5 and 10 entries of \( \mathbf{X} \) were randomly selected to be active in each column for all experiments, and all other entries of \( \mathbf{X} \) were set to zero. Experiments were performed using different sparsity levels in the dictionary, with \( \{10, 25, 50\} \% \) of entries set as non-zero.

The matrix \( \mathbf{M} = \mathbf{D} \mathbf{X} \) was synthesised. Subsequent factorisation was performed using different approaches. All approaches use the transposed ANLS approach (5) to perform the dictionary update, while different algorithms are used to estimate the coefficient matrix \( \mathbf{X} \) at each iteration. Each approach was run for 50 iterations of the alternating projection. The proposed \( \ell_0 \)-S-NMF is used with \( \lambda = 0.02 \), the minimum value of an activation in the synthesised dictionary. NMF was performed using the ANLS approach. OMP was used as a sparse approximation step, with non-negative constraints applied. OMP stopped iterating when either 15 atoms were selected, or the relative error \( \|r_s\|_2 \|m_s\|_2 < 0.05 \). Thresholded NNLS (T-NNLS) was performed using two different values of the threshold \( \lambda = 0.02 \) and \( \lambda = \sqrt{0.02} \). An \( \ell_1 \)-S-NMF approach was also performed, with \( \lambda = 0.02 \), and also with \( \lambda = 0.04 \) (\( \ell_1 \)-SNMF (2\( \lambda \)). For all NNLS calculations the active set Fast-NNLS [19] method was used, considering each column of \( \mathbf{M} \) separately.

The goal of the experiments was to reproduce a similar dictionary using the described NMF techniques. In order to measure the similarity between the original and estimated dictionaries, the simple measure:

\[
P = \min \left\{ \frac{\sum_{k=1}^{K} \max g_k}{\sum_{k=1}^{K} \max g^k} \right\}
\]

(13)

where \( \mathbf{G} = \mathbf{D}^T \mathbf{D} \), is used. \( P_{\text{max}} \) is recorded as the final value of \( P \). A value of \( P = 0.95 \) is considered success in these experiments, and an additional measure \( I \), relates the number of iterations taken to achieve \( P = 0.95 \) is also tabulated. This threshold of 0.95 is considered success in these experiments.

V. RESULTS

The results for the experiments are shown in Table 1, while Fig. 2 plots the evolution of the average of \( P \) for all experiments of given dictionary sparsity. It is observed that NMF performs poorly in all cases, being unsuccessful for all dictionaries, with the average correlation falling relative to the initialised dictionary. While a stated advantage of the ANLS approach to NMF is that overcomplete dictionaries can be used [7], it would appear not be viable without a sparsity-based approach. The proposed \( \ell_0 \)-SNMF approach is seen to be the only algorithm successful in all experiments. However, a small drop-off in performance is observed in the case of the sparsest dictionary, where \( P_{\text{max}} \) is reached after around 15 iterations, and not subsequently improved. A sparsity constraint on the dictionary, such as in (12), may improve this performance. The use of OMP was seen to be reasonable, with success seen with both of the sparser dictionaries, and a relatively high value of \( P_{\text{max}} \) in the case of the densest dictionary. When a high value of \( \lambda \) was used, the T-NNLS approach was seen to perform well for the denser dictionaries, while failing in the sparsest dictionary. With a lower threshold, \( \lambda = 0.02 \), the T-NNLS approach was seen to perform similar to NMF, and the results are not recorded. Using an \( \ell_1 \)-SNMF approach was seen to be relatively unsuccessful, using the considered parameters. Only when the higher threshold (2\( \$ \lambda \)) was used was success seen in any of the experiments. In this case, however, \( P_{\text{max}} \) was seen to be higher than for all other algorithms.

The effect of the use of a higher threshold is obvious in the results, as seen in the case of T-NNLS, with a high threshold. Some experiments were run with the \( \ell_2 \)-penalty norm suggested by [6]. The \( \ell_2 \)-norm can be considered a scaled \( \ell_1 \)-norm, and seen to perform well using a value of \( \lambda = 0.02 \). Considering the synthesis of the coefficient matrices used here, the \( \ell_2 \) squared approach is similar to \( \ell_1 \) with a higher value of \( \lambda \). Other initial experiments performed using high values of \( \lambda \) with the \( \ell_1 \)-SNMF approach were seen to bring similar improvements in the dictionary recovery to the \( \ell_2 \) approach. These observations would seem to validate the approach taken in [12] where a large value of \( \lambda \), used at initial iterations, was gradually decreased. However, it is noted that in all cases the proposed \( \ell_0 \) approach performs similarly, without requiring any scaling.

VI. CONCLUSIONS

A new variant of Sparse NMF has been presented using a modified sparse cost function with an \( \ell_0 \) penalty with a backwards elimination strategy proposed to perform the approximation. The use of this approach was seen to improve dictionary recovery results in synthetic experiments, and real-world data will be considered in future. The improvement in other approaches by scaling up the sparsity parameter, \( \lambda \) was noted, however, the proposed approach was seen to perform adequately while using the value of \( \lambda \) suggested by the experimental setup.

An extension of this approach to enforce sparsity of the dictionary itself was also proposed, although not implemented.
Future work will incorporate this approach. It was found that the use of accelerated NNLS algorithms such as [6] was problematic in these experiments, possibly due to the lack of structure between individual matrix columns which might be expected in real-world experiments and also to ill-condition introduced by overcompleteness of the dictionary. Repeated use of the F-NNLS algorithm was seen to be computationally expensive. OMP was seen to perform reasonably well, and use of a bi-directional stepwise optimal pursuit should improve these results, possibly close to that of the backwards elimination approach employed, while reducing the computational load. Such an approach could also be used to solve the transposed problem (5). The backwards elimination approach has been seen to perform well in the case of block sparsity, and an investigation of the use of block sparsity in NMF will be undertaken.

References