Sparse shift-invariant NMF

Vamsi K. Potluru
MIND Research Network,
Dept. of Comp Science,
Univ of New Mexico
ismav@cs.unm.edu

Sergey M. Plis
Dept. of Comp Science,
Univ of New Mexico
pliz@cs.unm.edu

Vince D. Calhoun
MIND Research Network,
Dept. of Elec and Comp Engg,
Univ of New Mexico
vcalhoun@unm.edu

Abstract

Non-negative Matrix Factorization (NMF) has increasingly been used for efficiently decomposing multivariate data into a signal dictionary and corresponding activations. In this paper, we propose an algorithm called sparse shift-invariant NMF (ssiNMF) for learning possibly overcomplete shift-invariant features. This is done by incorporating a circulant property on the features and sparsity constraints on the activations. The circulant property allows us to capture shifts in the features and enables efficient computation by the Fast Fourier Transform. The ssiNMF algorithm turns out to be matrix-free for we need to store only a small number of features. We demonstrate this on a dataset generated from an overcomplete set of bars.

1 Introduction

Non-negative Matrix Factorization (NMF) is a tool to split the given data matrix into a product of two non-negative matrix factors. This process can be used to identify useful features in the dataset. The constraint of non-negativity results in a parts-based representation and is usually different from other factorization techniques which result in more holistic representations (e.g. Principal Component Analysis (PCA) and Vector Quantization (VQ)). Another tool used commonly to find features is Independent Component Analysis (ICA) [3]. ICA assumes that the features thus found are statistically independent [1].

NMF intrinsically enforces certain amount of sparsity in its representations. However, in the case of overcomplete representations, we would like to explicitly enforce a sparsity constraint. NMF with a sparsity constraint on the activations was introduced in [5]. Convolutive NMF had previously been studied in [8] with an application to audio data. It was extended with a sparsity constraint in [7].

In this paper, we combine convolutive dictionary with sparse activations. Unlike the previous approaches, we constrain the features to be circulant to model arbitrary shifts in the data. This property is useful for training datasets that have misaligned instances, such as datasets of images. We demonstrate the utility of our approach using the dataset of [5] by learning a parsimonious dictionary to represent it.

2 NMF

Given a non-negative $m \times n$ matrix $\mathbf{X}$, we want to represent it with a product of two non-negative matrices $\mathbf{W}, \mathbf{H}$ of sizes $m \times r$ and $r \times n$ respectively:

$$\mathbf{X} \approx \mathbf{WH}. \tag{1}$$

The non-negativity constraint corresponds to the intuitive notion of features adding up to give the resulting data.

Lee and Seung [6] describe two simple multiplicative updates which work well in practice. These correspond to two different cost functions representing the quality of approximation. Here, we use the Frobenius norm for the cost function. The cost function and the corresponding updates are:

$$\begin{aligned} E &= \|\mathbf{X} - \mathbf{WH}\|_F && (2)\\ \mathbf{W} &= \mathbf{W} \odot \frac{\mathbf{XH}^\top}{\mathbf{WHH}^\top} && (3)\\ \mathbf{H} &= \mathbf{H} \odot \frac{\mathbf{W}^\top\mathbf{X}}{\mathbf{W}^\top\mathbf{WH}}, && (4) \end{aligned}$$

where $\|.\|_F$ denotes the Frobenius norm and $\|.\|_1$ the L_1 norm. The operator $\odot$ represents element-wise multiplication. Division is also element-wise. It should be noted that the cost function to be minimized is convex in either $\mathbf{W}$ or $\mathbf{H}$ but not in both. As the algorithm iterates using the updates (3) and (4), $\mathbf{W}$ and $\mathbf{H}$ converge to a local minimum of the cost function. The value of r determines quality of approximation and it is usually based on prior knowledge of the data being modelled.

3 Sparse NMF

NMF with a sparsity constraint was introduced in [5]. It was shown that explicitly controlling sparsity gives better decompositions. Using L_1 norm for sparsity, sparse NMF is formulated as follows:

$$\min_{\mathbf{W},\mathbf{H}} \frac{1}{2}\|\mathbf{X}-\mathbf{WH}\|_F + \lambda\|\mathbf{H}\|_1 \tag{5}$$

The update equations for this objective are given by:

$$\mathbf{W} = \mathbf{W} - \eta[-\mathbf{XH}^\top + \mathbf{WHH}^\top] \tag{6}$$

$$\mathbf{H} = \mathbf{H} \odot \frac{\mathbf{W}^\top\mathbf{X}}{\mathbf{W}^\top\mathbf{WH} + \lambda\mathbf{1}} \tag{7}$$

The parameter η is the learning rate and has to be explicitly set. As has already been noted in [5], the objective function is not scale free. This can be seen by setting $\mathbf{W} \leftarrow \alpha\mathbf{A}$ and $\mathbf{H} \leftarrow \frac{1}{\alpha}\mathbf{H}$, with $\alpha > 1$. To overcome this problem, columns of $\mathbf{W}$ are normalized to have L_2 norm of 1. Since additive updates can be slow and require setting the learning rate they are undesirable. To improve the result, new multiplicative update rules were derived by Eggert and Körner [4]. The update for matrix $\mathbf{W}$ is given by

$$\mathbf{W} = \mathbf{W} \odot \frac{\mathbf{XH}^\top + \mathbf{W}\,\mathrm{diag}(\mathbf{1}(\mathbf{WHH}^\top \odot \mathbf{W}))}{\mathbf{WHH}^\top + \mathbf{W}\,\mathrm{diag}(\mathbf{1}(\mathbf{XH}^\top \odot \mathbf{W}))} \tag{8}$$

4 Matrix-free computations

If we allow circulant features, the feature matrix $\mathbf{W}$ for r features increases dimension from $m \times r$ to $m \times (m * r)$. it turns into a concatenation of r circulant matrices. However, this dimensionality increase does not harm the computation since in the case of circulant matrices, matrix-vector product can be efficiently computed by using Fast Fourier Transform (FFT). In addition, a circulant matrix of size $m \times m$ requires storage space of $O(m)$. In this section we give definitions which are necessary to construct the ssiNMF algorithm.

4.1 Circulant matrices

Let us introduce two operators that will be needed further:

- Circulant-shift operator $S^i(\mathbf{v})$: given a vector $\mathbf{v}$ and a shift size i, returns the right circularly-shifted vector shifted by i positions.
- Flip operator $\mathrm{FLIP}(\mathbf{v})$: returns a permuted vector with the i-th element replaced by the $n-i+1$-th element of the given vector.

The circulant matrix with the first column equal to the vector $\mathbf{c}$ is given by

$$\begin{aligned}\mathbf{C} &= \begin{bmatrix} S^0(\mathbf{c}) & S^1(\mathbf{c}) & \cdots & S^{n-1}(\mathbf{c})\end{bmatrix} \\ &= \mathrm{cm}(\mathbf{c})\end{aligned} \tag{9}$$

We note that even though it has $O(n^2)$ elements, it can be generated from $\mathbf{c}$, which has $O(n)$ elements.

If $\mathbf{f} = S^1(\mathrm{FLIP}(\mathbf{c}))$ then

$$\begin{aligned}\mathbf{C}^\top &= \begin{bmatrix} S^0(\mathbf{f}) & S^1(\mathbf{f}) & \cdots & S^{n-1}(\mathbf{f})\end{bmatrix} \\ &= \mathrm{cm}(\mathbf{f})\end{aligned} \tag{10}$$

4.2 Circulant matrix-vector product

Here, we outline an efficient routine to calculate the product of a circulant matrix $\mathbf{C}$ whose first column is $\mathbf{c}$ with an appropriately sized vector $\mathbf{r}$. Let us denote by FFT and iFFT the routines of Fast Fourier Transform and inverse Fast Fourier Transform respectively. We then have :

$$\begin{aligned}\mathbf{Cr} &= \mathrm{iFFT}(\mathrm{diag}(\mathrm{FFT}(\mathbf{c}))\,\mathrm{FFT}(\mathbf{r})) \\ &= \mathrm{iFFT}(\mathrm{FFT}(\mathbf{c}) \odot \mathrm{FFT}(\mathbf{r})) \\ &= \mathrm{mvc}(\mathbf{c}, \mathbf{r})\end{aligned} \tag{11}$$

4.3 Composite circulant products

Let us define the matrix $\mathbf{A}$ to be composite circulant if its elements are square circulant matrices. Matrix $\mathbf{A}$ is completely characterized by matrix $\mathbf{B}$ given the following relations:

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \ldots & \mathbf{b}_r \end{bmatrix} \tag{12}$$

$$\mathbf{A} = \begin{bmatrix} \mathrm{cm}(\mathbf{b}_1) & \mathrm{cm}(\mathbf{b}_2) & \ldots & \mathrm{cm}(\mathbf{b}_r) \end{bmatrix} \tag{13}$$

The matrix-vector products given an appropriately sized vector $\mathbf{y}$ with matrix $\mathbf{A}$ are given by:

$$
\begin{aligned}
\mathbf{A}\mathbf{y} &= \begin{bmatrix}\mathrm{cm}(\mathbf{b}_1) & \mathrm{cm}(\mathbf{b}_2) & \dots & \mathrm{cm}(\mathbf{b}_r)\end{bmatrix}\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \vdots\\ \mathbf{y}_r\end{bmatrix}\\
&= \sum_i \mathrm{mvc}(\mathbf{b}_i, \mathbf{y}_i)\\
&= \mathrm{fmvc}(\mathbf{B}, \mathbf{y}) \qquad (14)\\
\mathbf{A}^\top\mathbf{y} &= \begin{bmatrix}\mathrm{cm}(\mathbf{b}_1)^\top\\ \mathrm{cm}(\mathbf{g}_2)^\top\\ \vdots\\ \mathrm{cm}(\mathbf{b}_r)^\top\end{bmatrix}\mathbf{y}\\
&= \begin{bmatrix}\mathrm{mvc}(S^1(\mathrm{FLIP}(\mathbf{b}_1))), \mathbf{y})\\ \mathrm{mvc}(S^1(\mathrm{FLIP}(\mathbf{b}_2))), \mathbf{y})\\ \vdots\\ \mathrm{mvc}(S^1(\mathrm{FLIP}(\mathbf{b}_r))), \mathbf{y})\end{bmatrix}\\
&= \mathrm{tfmvc}(\mathbf{B}, \mathbf{y}) \qquad (15)
\end{aligned}
$$

5 Sparse shift-invariant NMF

In the ssiNMF framework, we model the dictionary $\mathbf{W}$ to be a set of circularly shifted features. This is captured by matrix $\mathbf{G}$ representing the features and matrix $\mathbf{W}$ - the set of all possible linear shifts. The relationship between the matrices is :

$$
\begin{aligned}
\mathbf{G} &= \begin{bmatrix}\mathbf{g}_1 & \mathbf{g}_2 & \dots & \mathbf{g}_r\end{bmatrix} \qquad (16)\\
\mathbf{W} &= \begin{bmatrix}\mathrm{cm}(\mathbf{g}_1) & \mathrm{cm}(\mathbf{g}_2) & \dots & \mathrm{cm}(\mathbf{g}_r)\end{bmatrix} \qquad (17)
\end{aligned}
$$

We need to store only the matrix $\mathbf{G}$ to generate the full matrix $\mathbf{W}$. This makes the algorithm matrix-free and computationally efficient for using FFT's to compute matrix-vector products.

Given the data matrix $\mathbf{X}$, we apply Algorithm 1 denoted by ssiNMF to obtain the features and their corresponding activations. We note that the vectors with superscripts and subscripts denote the row and column vectors of the corresponding matrices respectively.

Algorithm 1 *ssiNMF*

randomly initialize $\mathbf{G}$ and $\mathbf{H}$
normalize columns of $\mathbf{G}$ to unit L_2 norm
repeat
  update $\mathbf{G}$
  for each column $\mathbf{g}_i$ in $\mathbf{G}$ **do**
    $t \leftarrow \mathbf{0}$
    for each element j in $\mathbf{g}_i$ **do**
      $t \leftarrow t + \mathrm{fmvc}(\mathbf{G}, \mathbf{H}\mathbf{h}^{i*m+j}) - \mathbf{X}\mathbf{h}^{i*m+j}$
    end for
    $\mathbf{g}_i = \mathbf{g}_i - \eta t$
  end for
  update $\mathbf{H}$
  for each column $\mathbf{h}_i$ in $\mathbf{H}$ **do**
    $\mathbf{h}_i = \mathbf{h}_i \odot \frac{\mathrm{tfmvc}(\mathbf{G}, \mathbf{x}_i)}{(\mathrm{tfmvc}(\mathbf{G}, \mathrm{fmvc}(\mathbf{G}, \mathbf{h}_i)) + \lambda)}$
  end for
until convergence

6 Experiments

To test our algorithm we generate the bars dataset from [5]. As shown in Figure 1(a), its generating features are single and double bars aligned vertically and horizontally on a 3×3 grid. Since all double bars can be expressed in terms of the single bars, this feature basis is overcomplete. These features form a generating feature matrix W_{gen}. Initializing H_{gen} to a sparse random matrix, we construct the dataset as $X = W_{gen}H_{gen}$. 12 random samples from the dataset are shown in Figure 1(b).

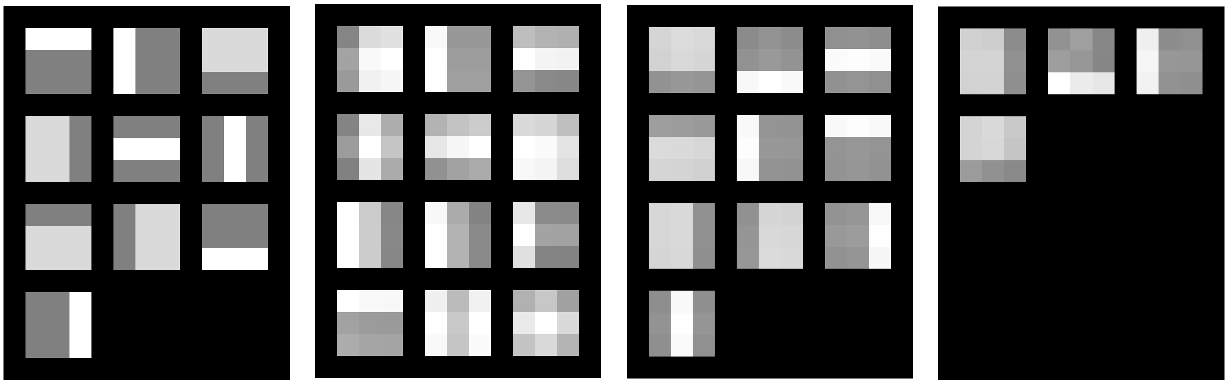
As previously demonstrated by Hoyer [5], the addition of sparsity assists in handling overcompleteness of the feature space. This is shown in the feature set learned by non-negative sparse coding in Figure 1(c). However, in the case of allowed translations the original overcomplete set can be represented by only 4 features: vertical single bar, horizontal single bar and corresponding double bars.

We applied ssiNMF by setting the number of features to 4 and $\lambda = 0.7$. Each feature in $\mathbf{G}$ was initialized by iid samples from the uniform distribution and normalized by its L_2 norm. Activations $\mathbf{H}$ were also randomly initialized from the uniform distribution.

Features identified by ssiNMF algorithm are shown in Figure 1(d). These features still represent an overcomplete basis since the double bar features can be represented in terms of the single bars. Shift-invariance leads to a smaller set of features while still enabling a sparse representation.

7 Discussion and future work

Circulant constraints make the computation of matrix-vector products fast and reduce storage space in case of dictionaries with shift-invariant features. The gradient descent rule for updating the dictionary matrix $\mathbf{W}$ is additive, however, it



(a) features (b) data (c) NNSC (10) (d) ssiNMF (4)

Figure 1. Experimental results on bars dataset. (a) The features used to generate training data. (b) A random sample from the dataset. (c) 10 features as learned by non-negative sparse coding of [5]. (d) 4 features that can represent the data in circulant case as learned by ssiNMF.

would be interesting to derive a suitable multiplicative rule.

The presented algorithm (ssiNMF) is also applicable for datasets which are misaligned. For example, ssiNMF could be applied to a dataset of fMRI images where the head is not stabilized. Shift-invariance in this case would compensate for the motion typically observed in fMRI experiments or for coregistration differences between subjects.

Our approach can also be extended to Non-negative Tensor Factorization (NTF) [2] which is a rich framework for modeling additional factors.

8 Acknowledgements

The first author would like to acknowledge the support from NIBIB grants 1 R01 EB 000840 and 1 R01 EB 005846. The second author was supported by NIMH grant 1 R01 MH076282-01. The latter two grants were funded as part of the NSF/NIH Collaborative Research in Computational Neuroscience Program. The authors would like to thank Barak Pearlmutter for the initial inspiration.

References

[1] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neu. Comp.*, 7(6):1129–59, 1995.

[2] A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S. Amari. Non-negative tensor factorization using alpha and beta divergences. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 3:III–1393–III–1396, Apr. 2007.

[3] P. Comon. Independent component analysis: A new concept. *Signal Processing*, 36:287–314, 1994.

[4] J. Eggert and E. Körner. Sparse coding and NMF. In *IEEE International Joint Conference on Neural Networks, 2004. Proceedings*, volume 4, pages 2529–33. IEEE, July 2004.

[5] P. O. Hoyer. Non-negative sparse coding. In *IEEE Workshop on Neural Networks for Signal Processing*, 2002.

[6] D. D. Lee and H. S. Seung. Learning the parts of objects with nonnegative matric factorization. *Nature*, 401:788–91, 1999.

[7] P. D. O'Grady and B. A. Pearlmutter. Convolutive non-negative matrix factorisation with sparseness constraint. In *International Workshop on Machine Learning for Signal Processing*, pages 427–432, Maynooth, Ireland, Sept. 6–8 2006. IEEE Press.

[8] P. Smaragdis. Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *Fifth International Conference on Independent Component Analysis*, LNCS 3195, pages 494–9, Granada, Spain, Sept. 22–24 2004. Springer-Verlag.