Abstract—This work introduces a new framework for nonnegative matrix factorization (NMF) in multisensor or multimodal data configurations, where taking into account the mutual dependence that exists between the related parallel streams of data is expected to improve performance. In contrast with previous works that focused on co-factorization methods — where some factors are shared by the different modalities — we propose a soft co-factorization scheme which accounts for possible local discrepancies across modalities or channels. This objective is formalized as an optimization problem where concurrent factorizations are jointly performed while being tied by a coupling term that penalizes differences between the related factor matrices associated with different modalities. We provide majorization-minimization (MM) algorithms for three common measures of fit — the squared Euclidean norm, the Kullback-Leibler divergence and the Itakura-Saito divergence — and two possible coupling variants, using either the \( \ell_1 \) or the squared Euclidean norm of differences. The approach is shown to achieve promising performance in two audio-related tasks: multimodal speaker diarization using audiovisual data and audio source separation using stereo data.

Index Terms—Nonnegative matrix factorization, co-factorization, multimodal data, segmentation, source separation.

I. INTRODUCTION

FACTOR models have gained a lot of attention in the machine learning and signal processing communities. In particular, nonnegative matrix factorization (NMF) is a powerful technique for nonnegative data analysis. For example, it has shown excellent performance in tasks such as movie ratings prediction [1] or spectrogram decomposition for source separation [2]. Given a nonnegative matrix \( V \), where every column represents a data point, NMF consists in finding the approximation

\[
V \approx WH
\]  

(1)

where \( W \) and \( H \) are nonnegative matrices as well. In this formulation, \( W \) acts as a dictionary of patterns representative of the data while \( H \) contains the activation coefficients of these patterns in \( V \).

In some cases, data can be available in several modalities. Consider for example the audio and visual streams of a video or the images and accompanying text from Internet photo databases. By extension, we can also use the different channels of multichannel data, such as the left and right signals of a stereo recording, and analyze them in a multimodal fashion. In such cases, it is desirable to exploit the mutual information shared by the different modalities. As such, in the setting of factor models, it has been assumed that the different modalities share a common factor, usually the activation matrix \( H \). In other words, if \( V_1 \) and \( V_2 \) represent the two data modalities, a co-factorization model of the form \( V_1 \approx W_1 H \), \( V_2 \approx W_2 H \) may be appropriate. This is in essence the approach taken in various settings by previous papers [3]–[10]. The assumption that the two modalities share the exact same activation matrix \( H \) may be a too strong one. As such, in this article we propose to relax this hard constraint and instead produce a “soft” co-factorization such that \( V_1 \approx W_1 H_1 \), \( V_2 \approx W_2 H_2 \) and \( H_1 \approx H_2 \). This new form of factorization will be shown to model more adequately the data in two applications, audiovisual speaker diarization and stereo audio source separation. A related idea has been proposed in [11], that uses a coupling penalty to identify dictionary entries that are shared across the EEG recordings of distinct subjects. This paper considers only the situation where the coupling penalty and all the measures of fit are \( \ell_2 \) norms. We consider here more general problems (Sections V and VI) that require the use of different divergences as measures of fit and a \( \ell_1 \) norm as a coupling term, and propose generic optimization algorithms to minimize the resulting cost functionals.

The paper is organized as follows. The general framework is presented in Section II and the algorithmic contribution is detailed in Section III. The remaining sections are devoted to evaluation with synthetic data experiments reported in Section IV, while Section V and Section VI are devoted to, respectively, multimodal speaker diarization using audiovisual data and audio source separation using stereo data. This paper significantly extends our previous conference contribution in the following way: while [12] introduced a specific version of soft co-factorization, results are presented here for distinct measures of fit and the characteristics of produced algorithms are precisely analyzed. The generality of the method is also illustrated on a new problem — source separation.

II. OBJECTIVE CRITERION FOR SOFT CO-FACTORIZATION

A. Notations

\( V_1 \) and \( V_2 \) denote the matrices associated with respectively the first and the second modality. \( V_1 \in \mathbb{R}^{F_1 \times N} \) and \( V_2 \in \mathbb{R}^{F_2 \times N} \). \( F_1 \) and \( F_2 \) are possibly different, they correspond to the dimensionality of the observations constituting the matrices \( V_1 \) and \( V_2 \). \( N \) corresponds to the number of observations. \( V_1 \) is factorized as the product of nonnegative matrices \( W_1 \in \mathbb{R}^{F_1 \times K_1} \) and \( H_1 \in \mathbb{R}^{K_1 \times N} \), whereas \( V_2 \) is factorized as the product of \( W_2 \in \mathbb{R}^{F_2 \times K_2} \) and \( H_2 \in \mathbb{R}^{K_2 \times N} \).
where \( K_1 \) and \( K_2 \) correspond to the ranks of the factorizations for respectively the first and the second modality.

**B. Nonnegative co-factorization**

The first solution to solve jointly two NMF problems with a shared factor (e.g.) \( H \) is to follow an alternate optimization scheme. As a first step, the matrix \( H \) is updated considering only the first problem, then used to update the matrix \( W_2 \) in the second problem. In a second step, problems are swapped: \( H \) is updated considering the second problem, then used to update \( W_1 \). This kind of approach is widespread and used in [9], [13], [14].

Another simple nonnegative matrix co-factorization scheme, used in [15], consists in stacking vertically two matrices \( V_1 \in \mathbb{R}_{+}^{F_1 \times N} \) and \( V_2 \in \mathbb{R}_{+}^{F_2 \times N} \) to form a new matrix \( V \in \mathbb{R}_{+}^{(F_1 + F_2) \times N} \). The matrix \( V \) is then factorized into the product \( WH \). Denoting by \( W_1 \) the \( F_1 \) first rows of \( W \), and \( W_2 \) the \( F_2 \) last rows of \( W \), \( W_1 H \) and \( W_2 H \) are factorizations for respectively \( V_1 \) and \( V_2 \), that share the same “activation” matrix \( H \).

But it is also possible to link the two factorizations using a coupling term. This gives the possibility to relax the “common factor” hypothesis and to weight the relative importance of the factorizations of the two modalities and the closeness between the resulting factors.

**C. Soft nonnegative co-factorization**

Indeed, two NMF problems can be independently solved by adding up the measures of fit corresponding to separate problems and solving a similar optimization program. It is then natural, if one knows that there exists some dependency between \( H_1 \) and \( H_2 \), to take this dependency into account, possibly by introducing an adapted penalization function \( P \), hence the formulation of our co-factorization problem via the following program:

\[
\min_{W_1, W_2, H_1, H_2} C(W_1, H_1, W_2, H_2) = D_1(V \mid W_1 H_1) + \gamma D_2(V \mid W_2 H_2) + \delta P(H_1, H_2),
\]

\[
\text{s.t. } W_1 \geq 0, \quad H_1 \geq 0, \quad W_2 \geq 0, \quad H_2 \geq 0,
\]

where \( D_1 \) and \( D_2 \) are measures of fit (also called divergences) for the two modalities—which may differ in this framework—and \( \gamma \) and \( \delta \) are scalars, corresponding to weighting hyper-parameters. We consider in the following only the situations where \( H_1 \) and \( H_2 \) have the same dimensions, and where the penalty \( P \) is used to account for a dependency between \( H_1 \) and \( H_2 \). Yet, this goes without loss of generality:

- the penalty could easily rather target a dependency between \( W_1 \) and \( W_2 \) for all data fitting measures that satisfy
  \[ D(V \mid WH) = D(V^T \mid H^T W^T) \]
  (as is the case for the three divergences considered below);
- if \( H_1 \) and \( H_2 \) have different dimensions, the penalizing function can readily ignore rows and columns of \( H_1 \) that have no match in \( H_2 \). We will thus equally use the notations \( K, K_1 \) and \( K_2 \) in what follows.

**D. Choices for \( D_1, D_2, P \)**

We will consider here three distinct situations, where the measures of fit \( D_1 \) and \( D_2 \) correspond to the sum over the coefficients of compared matrices of the scalar Euclidean distance, generalized Kullback-Leibler divergence and Itakura-Saito divergence (see Table I). These three measures of fit are the three most-used members of the \( \beta \)-divergence family; they correspond respectively to \( \beta = 2, \beta = 1 \) and \( \beta = 0 \). They are widely used in the NMF field [16], [17], and are well-suited to address problems with distinct statistical properties: the Euclidean distance can cope with Gaussian additive noise, while the Kullback-Leibler divergence is appropriate for multinomial distributions or Poisson noise and the Itakura-Saito divergence fits Gamma multiplicative noise [18], [19].

**E. Numerically stable objective function**

In this section, we modify (2) so as to preserve its main characteristics while avoiding spurious undesirable optima. To illustrate the drawbacks of (2), we focus on the case where \( P(H_1, H_2) = | |H_1 - H_2| | \) where \( | | \) corresponds either to the \( \ell_1 \) or \( \ell_2 \)-squared norm. Firstly, there is a scale ambiguity between matrices \( W_1 \) and \( W_2 \) on the one hand, and matrices \( H_1 \) and \( H_2 \) on the other hand. Given \( \alpha \) such that \( 0 < \alpha < 1 \) we have

\[
C(W_1 / \alpha, \alpha H_1, W_2 / \alpha, \alpha H_2) < C(W_1, H_1, W_2, H_2).
\]

Any unconstrained algorithm will therefore lead to degenerate solutions, where matrices \( H_1 \) and \( H_2 \) tend towards 0 while matrices \( W_1 \) and \( W_2 \) grow infinitely in norm. Secondly, we made no assumptions on the scales of matrices \( H_1 \) and \( H_2 \). We only supposed that they were related, which means that we need to rescale them prior to any comparison: the similarity between \( H_1 \) and \( H_2 \) is here considered in terms of their shape regardless of their scale.

The scale ambiguity can be solved by multiplying \( H_1 \) and \( H_2 \) respectively by the diagonal matrices \( \Lambda_1 \) and \( \Lambda_2 \in \mathbb{R}_{+}^{K \times K} \) in the penalty part [20]. Their \( k \)-th diagonal coefficients are

\[
\lambda_{1,k} = \sum_f w_{1,fk} \quad \text{and} \quad \lambda_{2,k} = \sum_f w_{2,fk},
\]

where \( w_{1,fk} \) and \( w_{2,fk} \) denote the coefficients of \( W_1 \) and \( W_2 \). This in fact amounts to constraining the \( \ell_1 \)-norm of the columns of \( W_1 \) and \( W_2 \) to the unit, using a simple substitution: in the modified program (3) proposed below, we have

\[
C(W_1, H_1, W_2, H_2) = C(\Lambda_1^{-1} W_1, H_1, \Lambda_2^{-1} W_2, H_2),
\]

where by construction \( \Lambda_1^{-1} W_1 \) and \( \Lambda_2^{-1} W_2 \) have normalized columns. The scale mismatch between \( H_1 \) and \( H_2 \) can also be solved using a diagonal matrix \( S \in \mathbb{R}_{+}^{K \times K}, \text{diag}(S) = (s_1, \ldots, s_K) \), according to:

\[
\min_{W_1, W_2, H_1, H_2, S} C(W_1, H_1, W_2, H_2, S) = D_1(V \mid W_1 H_1)
\]

\[
\quad + \gamma D_2(V \mid W_2 H_2) + \delta \| \Lambda_1 H_1 - S \Lambda_2 H_2 \| ,
\]

\[
\text{s.t. } W_1 \geq 0, \quad H_1 \geq 0, \quad W_2 \geq 0, \quad H_2 \geq 0, \quad \text{diag}(S) \geq 0.
\]

\( S \) is thus to be estimated along with the other factors. It will be easily updated since its coefficients can be obtained in closed form given \( H_1 \) and \( H_2 \) (see Section III). The obtained program (3) is numerically stable, and we will propose an algorithm to solve it.
III. ALGORITHMS

A. General architecture

The objective function (3) has no known closed-form optimum, hence we must resort to an iterative algorithm. We follow a block-coordinate descent approach where matrices $H_1, H_2, W_1, W_2$ and $S$ are sequentially updated. The update rules for $H_1, H_2, W_1$ and $W_2$ are very similar while the updates for $S$ is different and somewhat simpler. We will therefore give details here only for the updates of $H_1$ and $S$. A complete algorithm is given as an example in appendix for the Kullback-Leibler divergence used with $\ell_2$ coupling, the complete derivations for the various combinations of divergences and coupling penalty are given in the supplementary material available online.1

The updates for $S$ only depend on the soft co-factorization penalty and not on the data fit terms. The update rule for $S$ can thus be obtained in closed-form given the other matrices, with formulas that depend on the choice of the coupling norm (we consider below the case of the squared $\ell_2$ and $\ell_1$ norms).

In contrast, updating $H_1$ requires to optimize a criterion that depends jointly on the divergence $D_1$ and on the soft co-factorization penalty. We chose to update $H_1$ using a majorization-minimization (MM) approach [17], [21]. The main idea of MM algorithms is the following: i) first build an upper bound of the original objective function, easier to minimize and tight to the original objective function at the current iterate; ii) then minimize it. Minimizing this upper bound will bring a decrease in the original objective function: MM algorithms have not necessarily the best convergence speed, but they ensure that the objective function is strictly decreasing. With either a squared $\ell_2$ norm or a $\ell_1$ norm, the coupling term $\|A_1 H_1 - S A_2 H_2\|$ is both convex and separable with respect to the coefficients $h_{1,kn}$ of $H_1$. The upper bound of (3), required by the MM algorithm, is therefore built as follows:

- majorize the data-fitting term $D_1(V_1 | W_1 H_1)$ with a convex separable auxiliary function;
- add the coupling term as is;
- the resulting functional is convex and separable, and turns out to be amenable to optimization in closed-form.

B. Auxiliary function for the data-fitting term

The auxiliary function for the data-fitting term $D_1$ is denoted by $G(H_1|\tilde{H}_1)$. $\tilde{H}_1$ is the current iterate, where the auxiliary function is built. This function is minimized with respect to $H_1$, its minimizer gives the next iterate. $G(H_1|\tilde{H}_1)$ is specific for each data-fitting term, but can be built in a systematic way, majorizing convex parts using Jensen’s inequality and concave parts with their tangents [17]. The majorants for the considered measures of fit are given in Table I.

\[ G(H_1|\tilde{H}_1) = \sum_{k=1}^{K_1} \sum_{n=1}^{N} \left( \psi_{1,kn} h_{1,kn} + \phi_{1,kn} h_{1,kn} \right) + \text{cst.} \]

\[ \psi_{1,kn} = \sum_{f=1}^{F_1} \left( h_{1,kn} w_{1,fn} h_{1,fn} \right) \]

\[ \phi_{1,kn} = \sum_{f=1}^{F_1} \left( \frac{w_{1,fn} h_{1,kn}}{h_{1,fn}} \right) \]

\[ G(H_1|\tilde{H}_1) = \sum_{k=1}^{K_1} \sum_{n=1}^{N} \left( \psi_{1,kn} \log h_{1,kn} + \phi_{1,kn} h_{1,kn} \right) + \text{cst.} \]

\[ \psi_{1,kn} = \sum_{f=1}^{F_1} \left( h_{1,kn} w_{1,fn} h_{1,fn} \right) \]

\[ \phi_{1,kn} = \sum_{f=1}^{F_1} \left( \frac{w_{1,fn} h_{1,kn}}{h_{1,fn}} \right) \]

\[ G(H_1|\tilde{H}_1) = \sum_{k=1}^{K_1} \sum_{n=1}^{N} \left( \psi_{1,kn} \log h_{1,kn} + \phi_{1,kn} h_{1,kn} \right) + \text{cst.} \]

\[ \psi_{1,kn} = \sum_{f=1}^{F_1} \left( h_{1,kn} w_{1,fn} h_{1,fn} \right) \]

\[ \phi_{1,kn} = \sum_{f=1}^{F_1} \left( \frac{w_{1,fn} h_{1,kn}}{h_{1,fn}} \right) \]

\[ G(H_1|\tilde{H}_1) = \sum_{k=1}^{K_1} \sum_{n=1}^{N} \left( \psi_{1,kn} \log h_{1,kn} + \phi_{1,kn} h_{1,kn} \right) + \text{cst.} \]

\[ \psi_{1,kn} = \sum_{f=1}^{F_1} \left( h_{1,kn} w_{1,fn} h_{1,fn} \right) \]

\[ \phi_{1,kn} = \sum_{f=1}^{F_1} \left( \frac{w_{1,fn} h_{1,kn}}{h_{1,fn}} \right) \]

\[ G(H_1|\tilde{H}_1) = \sum_{k=1}^{K_1} \sum_{n=1}^{N} \left( \psi_{1,kn} \log h_{1,kn} + \phi_{1,kn} h_{1,kn} \right) + \text{cst.} \]

\[ \psi_{1,kn} = \sum_{f=1}^{F_1} \left( h_{1,kn} w_{1,fn} h_{1,fn} \right) \]

\[ \phi_{1,kn} = \sum_{f=1}^{F_1} \left( \frac{w_{1,fn} h_{1,kn}}{h_{1,fn}} \right) \]

\[ G(H_1|\tilde{H}_1) = \sum_{k=1}^{K_1} \sum_{n=1}^{N} \left( \psi_{1,kn} \log h_{1,kn} + \phi_{1,kn} h_{1,kn} \right) + \text{cst.} \]

\[ \psi_{1,kn} = \sum_{f=1}^{F_1} \left( h_{1,kn} w_{1,fn} h_{1,fn} \right) \]

\[ \phi_{1,kn} = \sum_{f=1}^{F_1} \left( \frac{w_{1,fn} h_{1,kn}}{h_{1,fn}} \right) \]

The following updates for the coefficients of $S$:

\[ s_k = \frac{\lambda_1,k \sum_{n=1}^{N} h_{1,kn} h_{2,kn}}{\lambda_2,k \sum_{n=2}^{N} h_{2,kn}} \]  

2) Update rule for $H_1$: In this case the penalized MM majorant is smooth and convex hence minimizing the auxiliary function for the whole objective functional (auxiliary function for the data-fitting term plus coupling term) is easy. For each measure of fit, one can show, using the strict convexity of the penalized majorant and the limits of its derivative towards $0^+$ and $+\infty$, that the derivative on $F_2^+\alpha$ admits a unique zero. From a practical viewpoint, finding this zero amounts to solving algebraic equations, of degree one for the Euclidean distance, degree two for the Kullback-Leibler divergence and degree three for the Itakura-Saito divergence. Details are given in supplementary materials.

D. $\ell_1$ case

1) Update rule for $S$: The coupling term corresponds here to $\|A_1 H_1 - S A_2 H_2\|_1$; we must therefore minimize a convex continuous piecewise linear function w.r.t. $S$. Since the function is piecewise linear, two situations can occur: first, the function is identically minimized on a nontrivial segment, and each value of $s_k$ chosen on this segment cancels the derivative; we can decide to retain an extremity of this segment

\[ \text{http://plato-tsi.telecom-paristech.fr/publi/26108/} \]
—therefore a point of change of slope of the function—as an update rule. Second, the minimum is reached precisely at a point of change of slope. Consequently, finding an optimal value of the function can be done as follows:

- compute critical values \( \tilde{r}_j = \frac{\lambda_1,k h_{1,k} - \lambda_2,k h_{2,k}}{h_{2,k} \delta} \) that correspond to the points of change of slope; sort them and store the permutation \( \mathcal{S} \) associated with the sorting process;
- look for \( n_e = \min \left\{ n, 2 \sum_{j=1}^{n} h_{2,k} \delta \right\} \);
- the new value of \( s_k \) is \( f_{\mathcal{S}(n_e)} \).

2) Update rule for \( H_1 \): It is no more possible to consider the derivative of the auxiliary function for the whole objective function, as an absolute value \( |\lambda_1,k h_{1,k} - \lambda_2,k h_{2,k}| \) is used in the coupling term. However, for all considered measures of fit, one can still show that \( 0 \) belongs to the subdifferential of the auxiliary function for a unique value of \( \mathbb{R}_+^* \), with the same arguments. From a practical viewpoint, a logical disjunction must be considered to find this value. The absolute value results in a term sign \( (\lambda_1,k h_{1,k} - \lambda_2,k h_{2,k}) \) in the subdifferential. The change of sign occurs for \( h_1 = h_c = \frac{s_k \lambda_1,k h_{2,k}}{\lambda_1,k} \) and we have:

- if the values of the subdifferential are negative at the left of \( h_c \) and positive at the right of \( h_c \), \( h_c \) is the solution no matter what measure of fit is used;
- otherwise, replace in the canceling equation associated with the considered measure of fit the expression \( (\lambda_1,k h_{1,k} - \lambda_2,k h_{2,k}) \) by \( 1 \) if the subdifferential has positive values in \( h_c^+ \), by \( -1 \) if the subdifferential has negative values in \( h_c^- \), and solve it accordingly.

The above results in a simple algebraic equation, of degree one for the Euclidean distance or the Kullback-Leibler divergence, and degree two for the Itakura-Saito divergence, hence no extra difficulties are to be expected after the logical disjunction. It should be noted that this logical disjunction corresponds to a conditional projection, meaning that if the values of \( H_1 \) and \( H_2 \) are “close enough”, the algorithm will naturally make them equal.

E. Discussion of the implementation

1) Permutation insensitivity: Given two matrices \( W_1 \) and \( H_1 \) and a permutation \( \mathcal{S} \) of the set \( \{1, \ldots, K\} \) we can build matrices \( \bar{W}_1 \) and \( \bar{H}_1 \) by applying \( \mathcal{S} \) to the rows of \( H_1 \) and the columns of \( W_1 \). We have \( W_1 H_1 = \bar{W}_1 \bar{H}_1 \), so that for any measure of fit \( F \) and a matrix \( V_1 \) we have

\[
D(V_1 | W_1 H_1) = D(V_1 | \bar{W}_1 \bar{H}_1).
\]

However, given another matrix \( H_2 \), we generally have \( \|H_1 - H_2\| \neq \|\bar{H}_1 - \bar{H}_2\| \). Therefore, if a “natural” matching order exists between matrices \( H_1 \) and \( H_2 \), the optimization of the objective function (3) will naturally return solutions where the rows of matrices \( H_1 \) and \( H_2 \) are in the rightful order: permutations of the rows do not affect the measures of fit, but increase the penalty term if they match rows that are not alike. Consequently, it is not necessary to have specific knowledge of the structure of the addressed problem, and random initializations are perfectly feasible.

2) Numerical concerns: one iteration of the algorithm consists in updating the different factor matrices. If \( V_1 \in \mathbb{R}^{F_1 \times N} \) and \( V_2 \in \mathbb{R}^{F_2 \times N} \), updating \( H_1 \) or \( H_2 \) costs respectively \( O(F_1 K N) \) or \( O(F_2 K N) \). The same analysis stands for the updates of \( W_1 \) and \( W_2 \). However, the updates of \( S \) are different: although they cost only \( O(K N) \) while using a \( \ell_2 \) penalty, the use of a \( \ell_1 \) norm involves the sorting of an array of size \( K \times N \) along its second dimension, therefore resulting in a complexity of \( O(K N \log N) \). For standard problems, we clearly have \( \log N \leq \min(F_1, F_2) \) and we can thus consider that the global complexity of one iteration is \( O(\max(F_1, F_2) K N) \).

As a hint, a Matlab implementation on a modern computer (Core 2 Duo, 6 Gb of RAM) requires 100 seconds to perform 1000 iterations —generally enough to reach convergence—with \( K = 5 \), \( F_1 = F_2 = 200 \) and \( N = 10^4 \).

3) Hyper-parameters adjustment: modeling the rightful link between two modalities, well-suited for a particular problem, implies to chose the hyper-parameters \( \gamma \) and \( \delta \). These parameters should in principle only depend on the nature of the considered problem, not on the dimensions \( F_1, F_2, K \) and \( N \). If there is primarily no other choice than using a training set to adjust these parameters, it has been found that it is easy to develop efficient heuristics, that consist in conserving the balance between the distinct terms of the objective function, to adjust these parameters when changing the dimensions of the problem [22].

IV. Experiments with synthetic data

To illustrate the potential of the approach, we first consider experiments on synthetic data. These data are generated as follows: the first step consists in building \( H_1 \) and \( H_2 \in \mathbb{R}^{2 \times 240} \). Their rows are made of irregular patterns of zeros and ones, artificially chosen to illustrate similar global behaviors in both modalities with some local discrepancies; the first row of \( H_1 \) and the first row of \( H_2 \) can respectively be observed in the first and the second column of Figure 1 (dashed lines). To generate \( W_1 \) and \( W_2 \in \mathbb{R}^{20 \times 2} \), each coefficient has been chosen following a uniform distribution on \([1, 11] \). In order to focus on the effects induced by the coupling term, the values of \( W_1 \) and \( W_2 \) are considered to be known in this experiment and are thus not updated during the training.

We first illustrate the impact of the hyper-parameter \( \delta \) while using a \( \ell_1 \) penalty. We can see in Figure 1 that activation patterns for both modalities quickly become identical while using a \( \ell_1 \) penalty, the use of a \( \ell_1 \) norm involves the sorting of an array of size \( K \times N \) along its second dimension, therefore resulting in a complexity of \( O(K N \log N) \). For standard problems, we clearly have \( \log N \leq \min(F_1, F_2) \) and we can thus consider that the global complexity of one iteration is \( O(\max(F_1, F_2) K N) \).

As a hint, a Matlab implementation on a modern computer (Core 2 Duo, 6 Gb of RAM) requires 100 seconds to perform 1000 iterations —generally enough to reach convergence—with \( K = 5 \), \( F_1 = F_2 = 200 \) and \( N = 10^4 \).

3) Hyper-parameters adjustment: modeling the rightful link between two modalities, well-suited for a particular problem, implies to chose the hyper-parameters \( \gamma \) and \( \delta \). These parameters should in principle only depend on the nature of the considered problem, not on the dimensions \( F_1, F_2, K \) and \( N \). If there is primarily no other choice than using a training set to adjust these parameters, it has been found that it is easy to develop efficient heuristics, that consist in conserving the balance between the distinct terms of the objective function, to adjust these parameters when changing the dimensions of the problem [22].

The same observations can be made while using a \( \ell_2 \) penalty in Figure 2. However, the behavior with respect to the values of \( \delta \) is different: the observed distortion is far more progressive;
changes can be observed even for low values of $\delta$, and even high values of $\delta$ do not result in numerical equality of $H_1$ and $H_2$.

We finally observe the impact of the weighting hyper-parameter $\gamma$ in Figure 3. With a constant weight for the coupling term ($\ell_1$ norm here), lower values for $\gamma$ are expected to put more emphasis on the data fitting term for the first modality. And indeed, although activation patterns $H_1$ and $H_2$ returned by the algorithm stay equal, one can observe that the common value moves closer to the ground truth values of the first modality when $\gamma$ decreases.

It is possible, with these properties, to chose objective functions that will, given some prior knowledge, embody arbitrary links between $H_1$ and $H_2$.

![Figure 1](image1.png)

Figure 1. Influence of a $\ell_1$ coupling: left and right columns correspond respectively to the first and the second modality. Continuous lines are the activation patterns returned by the algorithm, while dashed lines correspond to the ground truth. For simplicity, only one row per modality is displayed.

![Figure 2](image2.png)

Figure 2. Influence of a $\ell_2$ coupling (same display conventions as in Figure 1).

V. APPLICATION TO MULTIMODAL SPEAKER DIARIZATION

A. Introduction

Given an audio or video recording, the speaker diarization task consists in finding “who speaks when” in a non-supervised fashion [23]. This amounts to grouping homogeneous content segments, where a given person speaks without being interrupted. Beyond the audio modality, speaker diarization systems can take advantage of other streams of data that may be available at the same time, such as subtitles or an associated visual track. The joint exploitation of two or more modalities corresponds to multimodal speaker diarization [24], [25].

We consider here the specific case of professionally edited videos, where the intervention of a director (or an editor) results in a relationship between the final audio and video tracks: the editor will select among available multiview video tracks, at each time, the images that best illustrate the audio content. This generally results in showing onscreen the current speaker. If an audio feature matrix $V_{audio}$ is available, where each column is characteristic of the voice of a given speaker, and a video feature matrix $V_{video}$ where each column is characteristic of the onscreen appearance of a given person, it will be possible to use the soft coupling algorithm presented in Section III to perform the multimodal speaker diarization task, exploiting the relationship between audio and video tracks.

B. Co-factorization to perform multimodal speaker diarization

Interpreting NMF as a part-based representations [26], a factorization of $V_{audio}$ will return a matrix $W_a$ formed with the characteristic audio templates of the speakers, and a matrix $H_a$ where each row corresponds to the activations (interventions) of a given speaker. In a similar way, a factorization of $V_{video}$ will return a matrix $W_v$ formed with the characteristic video templates of the speakers, and a matrix $H_v$ where each row corresponds to the activations (onscreen appearances) of a given speaker.
The matrices $W_a$ and $W_v$ have no reason to be similar, and can have different dimensions, depending on the building process of $V_{audio}$ and $V_{video}$. But each row of $H_a$ should have a match in $H_v$, since onscreen appearances and verbal interventions have been supposed to be related. The soft coupling is particularly appropriate here since the relationship between rows of $H_a$ and rows in $H_v$ simply consists in a correlation: no equality can be expected, as the director might also choose from time to time to show onscreen a person who is not the current speaker.

To build representative matrices $V_{audio}$ and $V_{video}$, we use the approach of [20], with some slight modifications for the audio matrix $V_{audio}$: histograms are built using an aggregation window of 2 seconds, and hidden Markov models are trained with 50 states per speaker. In the end, each column of $V_{audio}$ consists of histograms of audio states, inferred from short-term Mel Frequency Cepstral Coefficients. The idea is similar for $V_{video}$, where each column corresponds to a histogram of visual words. The visual words are PHOW features [27] extracted within bounding boxes on faces and clothing areas in the frames.

The Kullback-Leibler divergence is chosen as a measure of fit: $V_{audio}$ and $V_{video}$ are, by construction, histogram data, and the Kullback-Leibler divergence is well-suited for multinomial distributions [18], [19]. We also choose to use a $\ell_1$ norm as a penalty, since it favors sparsity of $H_1 - H_2$ (see Section IV) that corresponds well to our hypothesis: rows of $H_a$ and $H_v$ are generally equal and sometimes very different. The multimodal speaker diarization is achieved directly by optimizing the following objective:

$$
\min_{W_a, H_a, W_v, H_v, S} C(W_a, H_a, W_v, H_v, S) =
D_{KL}(V_{video} | W_v H_v) + \gamma D_{KL}(V_{audio} | W_a H_a) + \delta \| \Lambda_a H_a - S \Lambda_v H_v \|_1,
\text{s.t. } W_a \geq 0, H_a \geq 0, W_v \geq 0, H_v \geq 0, \text{diag}(S) \geq 0.
$$

\textbf{C. Data}

To evaluate the performance of this approach we considered the Canal9 political debates database video database [28]. Each video tallies with a broadcast where a moderator and different guests debate a political question. Both the guests and the background vary over different broadcasts. We consider the 33 first available videos, and test the diarization on 8-minute long video segments. One segment is used per video, that starts at 3 minutes and 30 seconds after the beginning of the show to avoid the opening credits.

Using our soft coupling algorithm implies identifying correct settings for $\gamma$, $\delta$ and $K$. We consider that the number of speakers $Q$ is known, which is often the case for TV contents, which include subtitles or teletext, and we use $K = Q$ for $V_{audio}$ and $K = Q + 1$ for $V_{video}$. This leaves a component available for wide shots that do not clearly feature a single person. Using $K = Q + 1$ has also been tested for $V_{audio}$ but has proven to be harmful: in practice, the algorithm tends to use the additional component to “split” a speaker with a varied tone rather than to model background noise. Hence, only the $Q$ first components of $H_v$ are coupled with the components of $H_a$. There is also no need to use a training set to decide the value of $\gamma$, and we immediately decide to use $\gamma = 5$. This choice is consistent with both the impact of this parameter observed in IV and the intention to give priority to the audio track: we want to obtain speaker diarization results supported by the video track rather than an onscreen person spotting factorization supported by the audio track. However, the parameter $\delta$ must be trained with a separate dataset, and we have randomly selected for this purpose 10 development videos among the 33 available. We retained here the value $\delta = 0.1$.

\textbf{D. Results}

The tests are made as follows: for each video, we run the algorithm with 15 random initializations. Each initialization will lead to a different result, and we keep only the result associated with the lowest final value of the objective function. We now have one solution to the speaker diarization problem for each video, which is given by the rows of $H_a$. Each row is rescaled to the interval $[0, 1]$, then thresholded using a limit of 0.5: we consider that a person speaks at a given time if the associated activation coefficient is strong enough. The resulting diarization can now be compared to the ground truth, using the NIST scoring script for speaker diarization evaluation [29]. This returns a Diarization Error Rate (DER), which approximately corresponds to the fraction of speaker time that is not attributed to the right speaker. Lower is therefore better.

The comparison is made:

1) with an algorithm that performs a NMF of only the audio track (“Audio only”);
2) with an alternative algorithm using a hard coupling of audio and video tracks (“Hard”);
3) with the LIUM algorithm [2], [30] algorithm using a completely different approach (“LIUM”).

Note that, because of the additive form of the objective function, the hard coupling is strictly equivalent to performing a NMF of a matrix $V_{stacked}$, formed by the vertical concatenation of $V_{video}$ and $V_{audio}$, which has been done here.

In contrast to our soft co-factorization method, the hard coupling does not allow to use a number of components that differs for the audio and video terms, and we then do the test twice, using first $K = Q$ then $K = Q + 1$. The LIUM algorithm is extensively described in [30]; it relies on Gaussian mixture models and hierarchical agglomerative clustering. To ensure a fair comparison, we slightly modified this algorithm to take into account the fact that the number of speakers is known, and stop the clustering at the right step. Results are presented in Table II for the 23 test videos.

The following conclusions can be drawn: firstly, a hard coupling gives better results if the factorization is made using $Q + 1$ components. Secondly, using the video brings useful

\textsuperscript{2}Videos 06-11-15, 06-06-07, 05-11-23, 05-10-12, 06-04-19, 06-02-08, 06-10-18, 06-11-29, 05-12-07 and 06-10-04.
information; this information is better exploited with the proposed soft-coupling algorithm. Finally, the LIUM algorithm, using classical methods, clearly yields the best results.

It is however possible to introduce more prior knowledge into the objective function (5). Thus, we introduced supplementary the $\ell_1$ smoothing penalties

$$
\beta_{sa} \sum_{k=1}^{K} \sum_{n=2}^{N} |\lambda_k h_{a,kn} - \lambda_k h_{a,k(n-1)}|
$$

and

$$
\beta_{sv} \sum_{k=1}^{K} \sum_{n=2}^{N} |\lambda_k h_{v,kn} - \lambda_k h_{v,k(n-1)}|
$$

for, respectively, the audio and video activation patterns. These penalties support a temporal regularity of the activations in matrices $H_a$ and $H_v$, essentially by encouraging activation coefficients that are consecutive in time to be close to each other (possibly equal [31]). We used $\beta_{sv} = 0.01$ and $\beta_{sa} = 0.2$ after tests on the training set. The results using this soft coupling, smoothed algorithm are presented in Table III. We observe a noticeable improvement, even though the performance of the LIUM algorithm is not reached. We also mention the number of videos for which each algorithm gives the best solution: this illustrates that the differences between the LIUM algorithm and the proposed one mainly comes from some odd situations where the soft coupling totally fails. Indeed, the soft coupling method gives the best results for nearly half of the tested videos, but the average DER is still distinct from the average DER for the LIUM algorithm. This comes from videos where the speaking time is shared very unequally between the distinct speakers: the soft coupling methods tends to ignore the speaker with the lowest speaking time, and to “reuse” the freed component to “split” another speaker with a varied tone over two components, yielding very high DER. Still, regardless of further improvements that could be made to the objective function, the soft coupling is a method that gives good results the major part of the time.

VI. APPLICATION TO STEREO SOURCE SEPARATION

As a second illustration of the proposed soft co-factorization paradigm, we now turn to a stereo source separation example.

<table>
<thead>
<tr>
<th>Method</th>
<th>Audio only</th>
<th>Hard $K = Q$</th>
<th>Hard $K = Q + 1$</th>
<th>Soft coupling</th>
<th>LIUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean DER</td>
<td>21.4</td>
<td>25.1</td>
<td>18.9</td>
<td>16.8</td>
<td>8.0</td>
</tr>
</tbody>
</table>

**Table II**

MEAN DER OF THE DIFFERENT METHODS (TEST SET). LOWER VALUES INDICATE BETTER PERFORMANCE.

A. Methodology

We consider a two by two problem in which two sources have to be estimated from two sensors in a reverberant environment. Our approach may however be extended to an arbitrary number of sources and sensors, including overdetermined or underdetermined scenarios. In the time domain the mixing model is

$$
x_l(t) = s_1(t) + s_2(t)
$$

$$
x_r(t) = a_1 * s_1(t) + a_2 * s_2(t)
$$

where the indices $l$ and $r$ refer to left and right channels. We assume that we can include the room responses from the sources to the first channel in the source signals themselves in order to lift ambiguities. Under the narrow-band assumption (i.e., assuming that reverberation times are smaller than the length of analysis window), the mixing model may be written in the short-time Fourier transform (STFT) domain as

$$
x_{lf,n} = s_{1f,n} + s_{2f,n}
$$

$$
x_{rf,n} = a_{1f} s_{1f,n} + a_{2f} s_{2f,n}
$$

where the indices $f$ and $n$ refer to frequency and time frame, respectively. In this model, the convolution is essentially represented by frequency-dependent linear instantaneous mixing models. In [32], [33], the sources are modeled by a Gaussian composite model (GCM) that writes

$$
s_{jf,n} \sim N_c(0, |W_j H_j| f_n).
$$

This is a latent factor model that has proven very efficient in audio settings [32], [33]. Under this assumption, the negative log-likelihood of the sources writes

$$
- \log p(|S_j|^2 | W_j H_j) = D_{IS}(|S_j|^2 | W_j H_j) + \text{ cst.}
$$

where $|S_j|^2$ denotes the power spectrogram of source $s_j$.

In this paper, we propose to relax the assumption that the source activation matrices $H_j$ are the same in both channels. Our hope is that this increased flexibility in the model may mitigate the point-source and narrow-band assumptions that may be erroneous in some real-world settings. As such, we assume instead that the contribution of source $j$ to channel $l$ is $H_{l,j}$ and that its contribution to channel $r$ is $H_{r,j}$, and we will enforce that $H_{l,j} \approx H_{r,j}$. With these assumptions, the negative log-likelihood of the left channel writes

$$
- \log p(X_l | W_1, W_2, H_{l1}, H_{l2}) = D_{IS}(V_l | W_1 H_{l1} + W_2 H_{l2}) + \text{ cst.}
$$

where $X_l$ is the STFT of $x_l$ and $V_l = |X_l|^2$ is its power spectrogram. The negative log-likelihood of the right channel writes

$$
- \log p(X_r | A_1, A_2, W_1, W_2, H_{r1}, H_{r2}) = D_{IS}(V_r | A_1 W_1 H_{r1} + A_2 W_2 H_{r2}) + \text{ cst.}
$$

where $A_1$ and $A_2$ are the diagonal matrices with coefficients $\{ |a_{1f}|^2 \}_f$ and $\{ |a_{2f}|^2 \}_f$, respectively. We propose in this article to estimate the parameters $\theta = \{ A_1, A_2, W_1, W_2, H_{r1}, H_{r2}, H_{l1}, H_{l2} \}$ by optimizing the sum of the log-likelihoods of both channels. This is suboptimal.
as compared to optimizing the joint likelihood but was found to be an easier and still viable approach in [33]. Denoting \( W_l = [W_1, W_2] \), \( W_r = [A_1 W_1, A_2 W_2] \), \( H_l = [H_{l1}^T, H_{l2}^T]^T \) and \( H_r = [H_{r1}^T, H_{r2}^T]^T \), our approach boils down to minimizing the following objective function

\[
C(\theta) = D_{IS}(V_l | W_l H_l) + D_{IS}(V_r | W_r H_r) \tag{13}
\]

subject to \( H_l \approx H_r \). As such, it defines a soft nonnegative matrix co-factorization problem. A particularity is that \( W_l \) and \( W_r \) are not independent and share a specific structure, since they both depend, by construction, on \( W_1 \) and \( W_2 \). The MM framework can readily handle this specificity, and the details are skipped for brevity.

B. Experimental setup

In this section we report evaluation results for both the soft coupling algorithm, corresponding to Equation (13), and the hard coupling algorithm obtained by enforcing the constraint \( H_l = H_r \).

We consider the publicly available Signal Separation Evaluation Campaign (SiSEC) 2011 noisy speech development dataset [34]. This dataset is made up of mixtures of speech and real-world background noise; the background noise is recorded in three different environments (subway, cafeteria and square), and recordings are made with varying positions of the microphones. We use only the stereophonic mixtures, made up of two sources (speech and noise), and recorded with two sensors, which leaves us with ten recordings\(^3\). Though the hyperparameter \( \delta \) and the number of components \( K \) are optimized with the available data, the use of the development dataset ensures a fair comparison with other algorithms.

After the source separation, the sources are reconstructed with a standard Wiener filtering, and we compute the resulting Signal to Distortion Ratio (SDR) for each source, using the evaluation script given for the 2011 Signal Separation Evaluation Campaign [35]. This gives 20 scores that can be compared\(^4\) to the algorithms that processed exactly the same data during the campaign. Three algorithms are available, these algorithms will simply be denoted by algorithm 5, algorithm 6 and algorithm 8, to keep the notations used online.

C. Results

The tests are made as follows: the algorithms corresponding to soft and hard coupling are randomly initialized 25 times. Among these 25 runs, we keep the “best” run, corresponding to the initialization leading to the lowest value of the optimized objective function. This process is repeated for different values of \( K \) (8, 15, 25) both for the soft and the hard coupling algorithms. It is also repeated for different values of \( \delta \) (0.01, 0.1, 1 and 10) for the soft coupling algorithm. We then retain the parameters that give the best results in terms of average SDR, which is possible because we use the development set.\(^5\)

\(^3\)dev_Ca1_Ce_A, dev_Ca1_Ce_B, dev_Ca1_Co_A, dev_Ca1_Co_B, dev_Sq1_Ce_A, dev_Sq1_Ce_B, dev_Sq1_Co_A, dev_Sq1_Co_B, dev_Sul1_Ce_A and dev_Sul1_Ce_B.

\(^4\)Results are available online: http://www.irisa.fr/metiss/SiSEC11/noise/results_dev.html.

This way, we obtain \( K = 15 \) and \( \delta = 1 \). The results for these values are summarized in Table IV: the SDR are averaged over the 20 computed scores, and the third column refers to the proportion of results for which a given algorithm offers the best solution.

We can immediately draw two conclusions: firstly, both algorithms perform reasonably well compared to existing solutions, even though the best results are achieved by algorithm 8. Secondly, the soft coupling algorithm yields a slight improvement over the hard coupling algorithm. We can interpret this improvement as follows: the flexibility of the soft coupling algorithm has been used to relax an hypothesis made by the hard coupling algorithm. With the right parameters, the soft coupling consequently models more accurately the behavior of the sources in each channel, and leads to a better separation.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Average SDR (dB)</th>
<th>Best result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard coupling</td>
<td>3.2</td>
<td>5/20</td>
</tr>
<tr>
<td>Soft coupling</td>
<td>3.7</td>
<td>5/20</td>
</tr>
<tr>
<td>Algorithm 5</td>
<td>3.3</td>
<td>4/20</td>
</tr>
<tr>
<td>Algorithm 6</td>
<td>3.2</td>
<td>1/20</td>
</tr>
<tr>
<td>Algorithm 8</td>
<td>4.2</td>
<td>5/20</td>
</tr>
</tbody>
</table>

Table IV

Results for an “unassisted” source-separation task.

It should be noted that the flexibility of the soft coupling algorithm is not optimally exploited while fixing once and for all the hyper-parameter \( \delta \). When it comes to source separation, we can imagine situations where a sound engineer can select the best value specifically for each recording, or where an automatic setting is used: although the dataset is not wide enough to conclude definitely, it seems that low values of \( \delta \) are better-suited for large reverberant recording environments. This would not be possible with the hard coupling algorithm, since this algorithm offers no way to control the coupling.

VII. Conclusions

The soft co-factorization paradigm introduced in this paper is suitable to address the numerous problems where two (or more) modalities or data channels are assumed to be related through similar factors. It is able to leverage the dependencies that exist among the related parallel streams of data being analyzed while accounting for possible local discrepancies, essentially by relaxing the too rigid constraint that common factors are imposed. This paradigm can be instantiated using many different measures of fit and coupling penalties, and algorithms can be readily built for the most classic ones following the process presented in Section III. Depending on the nature of the problem, the setting of one or two hyper-parameters is required, which is a difficulty but proves to be useful when it comes to adapting the model to a particular configuration.

The multiple possible choices offer full control over the relationship that one wants to model (as was illustrated using synthetic data). The soft co-factorization paradigm is therefore very generic, and can easily apply to every situation where the common factor assumption cannot be made and where an explicit modeling of the differences between related
factors would be complex. To illustrate this potential, we have addressed two challenging applications, namely source separation and multimodal speaker diarization. In both cases, we rapidly obtained reasonable results using our soft coupling paradigm.

Further work could consist in examining to what extent the proposed method could be adapted in a probabilistic framework and brought closer to [36], [37], which modeled other type of dependencies between modalities in a probabilistic fashion.

APPENDIX

We present here the equations and the algorithm to solve (3) with a Kullback-Leibler divergence and a $\ell_2$ coupling term.

A. Minimization w.r.t. $H_1$ and $H_2$

Adding the coupling term to the auxiliary function given in Table I for the Kullback-Leibler divergence, and taking derivatives w.r.t. $h_{q, kn}$ $q \in \{1, 2\}$, for $h_{q, kn} > 0$, we get the following system of equations, canceling the derivatives:

$$\begin{aligned}
\frac{-\psi_{q,1}}{h_{1, kn}} + \lambda_{1, k} - 2\delta s_k \lambda_{1, k} \lambda_{2, k} h_{2, kn} + 2\delta \lambda_{1, k}^2 h_{1, kn} &= 0 \\
\frac{-\psi_{q,2}}{h_{2, kn}} + \lambda_{2, k} - 2\delta s_k \lambda_{1, k} \lambda_{2, k} h_{1, kn} + 2\delta \lambda_{1, k}^2 h_{2, kn} &= 0
\end{aligned}
$$

These quadratic equations can be solved in closed form w.r.t. $h_{1, kn}$ (resp. $h_{2, kn}$); the unique positive solution corresponds to the update rule.

B. Minimization w.r.t. $W_1$ and $W_2$

Auxiliary functions can be found w.r.t. $W_1$ and $W_2$ as w.r.t. $H_1$ and $H_2$. We use the notation $\Theta_q$ to designate the matrix with coefficients $\theta_{q, fk}$, where:

$$\begin{aligned}
\theta_{q, fk} &= w_{q, fk} \left( \sum_n h_{q, kn} \left( \frac{w_{q, pk}}{\sum_n w_{q, pk} h_{q, kn}} \right) \right) \\
\sigma_{q, k} &= \sum_n h_{q, kn}
\end{aligned}
$$

With the coupling term added to the auxiliary function, we get the following system of equations, canceling the derivatives:

$$\begin{aligned}
\frac{-\psi_{1, f k}}{w_{1, f k}} + \sigma_{1, k} + 2\delta \sum_n h_{1, kn} \left( \sum_p w_{1, pk} h_{1, kn} - s_k \lambda_{1, k} h_{2, kn} \right) &= 0 \\
\frac{-\psi_{2, f k}}{w_{2, f k}} + \sigma_{2, k} + 2\delta \sum_n s_k h_{2, kn} \left( \sum_p w_{2, pk} s_k h_{2, kn} - \lambda_{1, k} h_{1, kn} \right) &= 0
\end{aligned}
$$

Considering these equations for $f_1 \neq f_2$, one obtains:

$$w_{q, f k} = \frac{w_{q, f k} \theta_{q, f k}}{\theta_{q, f k}}, \quad q \in \{1, 2\}. \tag{15}$$

We can therefore rewrite the system as:

$$\begin{aligned}
\frac{-\psi_{1, f k}}{w_{1, f k}} + \sigma_{1, k} - 2\delta s_k \lambda_{2, k} \sum_n h_{1, kn} h_{2, kn} + 2\delta \lambda_{1, k}^2 h_{1, kn} &= 0 \\
\frac{-\psi_{2, f k}}{w_{2, f k}} + \sigma_{2, k} - 2\delta s_k \lambda_{1, k} \sum_n h_{1, kn} h_{2, kn} + 2\delta \lambda_{1, k}^2 h_{2, kn} &= 0
\end{aligned} \tag{16}$$

These separable quadratic equations can be solved in closed form w.r.t. $w_{1, f k}$ (resp. $w_{2, f k}$); the unique positive solution corresponds to the update rule.

C. Algorithm

Algorithm 1 KL-NMF with soft $\ell_2$-coupling

```
\begin{algorithm}
\begin{algorithmic}
\State \textbf{do}
\State \textbf{update} $S$ \textbf{according to equation (4)}
\State \textbf{update} $H_1$ and $H_2$ \textbf{according to equation (14)}
\State \textbf{compute} one row of $W_1$ \textbf{according to equation (16)}
\State \textbf{deduce} other rows using $\Theta_1$ and equation (15)
\State \textbf{do} the same for $W_2$ and $\Theta_2$
\State until convergence
\end{algorithmic}
```

REFERENCES


