HAL
open science

Primal-Dual Algorithms for Non-negative Matrix Factorization with the Kullback-Leibler Divergence

Felipe Yanez, Francis Bach

To cite this version:

Felipe Yanez, Francis Bach. Primal-Dual Algorithms for Non-negative Matrix Factorization with the Kullback-Leibler Divergence. 2014. hal-01079229

HAL Id: hal-01079229

https://hal.science/hal-01079229

Preprint submitted on 31 Oct 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Primal-Dual Algorithms for Non-negative Matrix Factorization with the Kullback-Leibler Divergence

Felipe Yanez and Francis Bach

INRIA – SIERRA Project-Team
Département d'Informatique de l'École Normale Supérieure
Paris, France

November 29, 2014

Abstract

Non-negative matrix factorization (NMF) approximates a given matrix as a product of two non-negative matrices. Multiplicative algorithms deliver reliable results, but they show slow convergence for high-dimensional data and may be stuck away from local minima. Gradient descent methods have better behavior, but only apply to smooth losses such as the least-squares loss. In this article, we propose a first-order primal-dual algorithm for non-negative decomposition problems (where one factor is fixed) with the KL divergence, based on the Chambolle-Pock algorithm. All required computations may be obtained in closed form and we provide an efficient heuristic way to select step-sizes. By using alternating optimization, our algorithm readily extends to NMF and, on synthetic examples, face recognition or music source separation datasets, it is either faster than existing algorithms, or leads to improved local optima, or both.

1 Introduction

The current development of techniques for big data applications has been extremely useful in many fields including data analysis, bioinformatics and scientific computing. These techniques need to handle large amounts of data and often rely on dimensionality reduction; this is often cast as approximating a matrix with a low-rank element.

Non-negative matrix factorization (NMF) is a method that aims at finding part-based, linear representations of non-negative data by factorizing it as the product of two low-rank non-negative matrices (Paatero and Tapper, 1994; Lee and Seung, 1999). In 2000, two multiplicative algorithms for NMF were introduced by Lee and Seung, one that minimizes the conventional least-squares error, and other one that minimizes the generalized Kullback-Leibler (KL) divergence (Lee and Seung, 2000).

These algorithms extend to other losses and have been reported in different applications, e.g., face recognition (Wang et al., 2005), music analysis (Févotte et al., 2009), and text mining (Guduru, 2006). An important weakness of multiplicative algorithms is their slow convergence rate in high-dimensional data and their susceptibility to become trapped in poor local optima (Lin, 2007). Gradient descent methods for NMF provide additional flexibility and fast convergence (Lin, 2007; Kim et al., 2008; Gillis, 2011). These methods have been extensively studied for the minimization of the least-squares error (Lin, 2007; Kim et al., 2008).

The goal of this paper is to provide similar first-order methods for the KL divergence, with updates as cheap as multiplicative updates. Our method builds on the recent work of Sun and Févotte

(2014) which consider the alternating direction method of multipliers (ADMM) adapted to this problem. We instead rely on the Chambolle-Pock algorithm (Chambolle and Pock, 2011), which may be seen as a linearized version of ADMM, and thus we may reuse some of the tools developed by Sun and Févotte (2014) while having an empirically faster algorithm.

1.1 Contributions

The main contributions of this article are as follows:

- We propose a new primal-dual formulation for the convex KL decomposition problem in Section 3.1, and an extension to the non-convex problem of NMF by alternating minimization in Section 3.5.
- We provide a purely data-driven way to select all step-sizes of our algorithm in Section 3.3.
- In our simulations in Section 4 on synthetic examples, face recognition or music source separation datasets, our algorithm is either faster than existing algorithms, or leads to improved local optima, or both.
- We derive a cheap and efficient implementation (Algorithm 2). Matlab code is available online at: `anonymized website`

2 Problem Formulation

Let $\mathbf{V} \in \mathbb{R}_+^{n\times m}$ denote the $n \times m$ given matrix formed by m non-negative column vectors of dimensionality n. Considering $r \leq \min(n, m)$, let $\mathbf{W} \in \mathbb{R}_+^{m\times r}$ and $\mathbf{H} \in \mathbb{R}_+^{r\times n}$ be the matrix factors such that

$$\mathbf{V} \approx \mathbf{WH}.$$

Two widely used cost functions for NMF are the conventional least-squares error (not detailed herein), and the generalized KL divergence

$$\begin{aligned} D(\mathbf{V}\|\mathbf{WH}) \quad = \quad & -\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{V}_{ij}\left\{\log\left(\frac{(\mathbf{WH})_{ij}}{\mathbf{V}_{ij}}\right)+1\right\} \\ & +\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbf{WH})_{ij}. \end{aligned} \tag{1}$$

In this work, only the KL divergence is considered. Therefore, the optimization problem is as follows:

$$\underset{\mathbf{W},\mathbf{H}\,\geq\,0}{\text{minimize}} \quad D(\mathbf{V}\|\mathbf{WH}). \tag{2}$$

We recall that the previous problem is non-convex in both factors simultaneously, whereas convex in each factor separately, i.e., the non-negative decomposition (ND) problems,

$$\underset{\mathbf{W}\,\geq\,0}{\text{minimize}}\ D(\mathbf{V}\|\mathbf{WH}) \tag{3}$$

$$\text{and} \quad \underset{\mathbf{H}\,\geq\,0}{\text{minimize}}\ D(\mathbf{V}\|\mathbf{WH}), \tag{4}$$

are convex.

We now present two algorithms for NMF, multiplicative updates (Lee and Seung, 2000), and the ADMM-based approach (Sun and Févotte, 2014).

2.1 Multiplicative updates

Lee and Seung (2000) introduced two multiplicative updates algorithms for NMF. One minimizes the conventional least-squares error, and the other one minimizes the KL divergence.

The NMF problem, for both losses, is a non-convex problem in $\mathbf{W}$ and $\mathbf{H}$ simultaneously, but convex with respect to each variable taken separately; this make alternating optimization techniques, i.e., solving at each iteration two separate convex problems, very adapted: first fixing $\mathbf{H}$ to estimate $\mathbf{W}$, and then fixing $\mathbf{W}$ to estimate $\mathbf{H}$ (Lee and Seung, 2000; Févotte et al., 2009). The multiplicative updates algorithms (like ours) follow this approach.

For the KL divergence loss, the multiplicative update rule (Lee and Seung, 2000) for $\mathbf{W}$ and $\mathbf{H}$ is as follows and may be derived from expectation-maximization (EM) for a certain probabilistic model (Lee and Seung, 2000; Févotte and Cemgil, 2009):

$$\begin{aligned}
\mathbf{W}_{ia} &\leftarrow \mathbf{W}_{ia} \frac{\sum_{\mu=1}^{n} \mathbf{H}_{a\mu} \mathbf{V}_{i\mu} / (\mathbf{W}\mathbf{H})_{i\mu}}{\sum_{\nu=1}^{n} \mathbf{H}_{a\nu}}, \text{ and} \\
\mathbf{H}_{a\mu} &\leftarrow \mathbf{H}_{a\mu} \frac{\sum_{i=1}^{m} \mathbf{W}_{ia} \mathbf{V}_{i\mu} / (\mathbf{W}\mathbf{H})_{i\mu}}{\sum_{k=1}^{m} \mathbf{W}_{ka}}.
\end{aligned}$$

The complexity per iteration is $O(rmn)$.

2.2 Alternating direction method of multipliers (ADMM)

Sun and Févotte (2014) propose an ADMM technique to solve Problem (2) by reformulating it as

$$\begin{aligned}
\text{minimize} \quad & D(\mathbf{V} \| \mathbf{X}) \\
\text{subject to} \quad & \mathbf{X} = \mathbf{Y}\mathbf{Z} \\
& \mathbf{Y} = \mathbf{W}, \ \mathbf{Z} = \mathbf{H} \\
& \mathbf{W} \geq 0, \ \mathbf{H} \geq 0.
\end{aligned}$$

The updates for the primal variables $\mathbf{W}$, $\mathbf{H}$, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ are as follows and involve certain proximal operators for the KL loss which are the same as ours in Section 3.2:

$$\begin{aligned}
\mathbf{Y}^\top &\leftarrow \left(\mathbf{Z}\mathbf{Z}^\top + \mathbf{I}\right)^{-1} \left(\mathbf{Z}\mathbf{X}^\top + \mathbf{W}^\top + \tfrac{1}{\rho}\left(\mathbf{Z}\alpha_{\mathbf{X}}^\top - \alpha_{\mathbf{Y}}^\top\right)\right) \\
\mathbf{Z} &\leftarrow \left(\mathbf{Y}^\top\mathbf{Y} + \mathbf{I}\right)^{-1} \left(\mathbf{Y}^\top\mathbf{X} + \mathbf{H} + \tfrac{1}{\rho}\left(\mathbf{Y}^\top\alpha_{\mathbf{X}} - \alpha_{\mathbf{Z}}\right)\right) \\
\mathbf{X} &\leftarrow \frac{(\rho\mathbf{Y}\mathbf{Z} - \alpha_{\mathbf{X}} - \mathbf{1}) + \sqrt{(\rho\mathbf{Y}\mathbf{Z} - \alpha_{\mathbf{X}} - \mathbf{1})^2 + 4\rho\mathbf{V}}}{2\rho} \\
\mathbf{W} &\leftarrow \left(\mathbf{Y} + \tfrac{1}{\rho}\alpha_{\mathbf{Y}}\right)_+ \\
\mathbf{H} &\leftarrow \left(\mathbf{Z} + \tfrac{1}{\rho}\alpha_{\mathbf{Z}}\right)_+ .
\end{aligned}$$

Note that the primal updates require solving linear systems of sizes $r \times r$, but that the overal complexity remains $O(rmn)$ per iteration (the same as multiplicative updates).

The updates for the dual variables $\alpha_{\mathbf{X}}$, $\alpha_{\mathbf{Y}}$ and $\alpha_{\mathbf{Z}}$ are then:

$$\begin{aligned}
\alpha_{\mathbf{X}} &\leftarrow \alpha_{\mathbf{X}} + \rho\left(\mathbf{X} - \mathbf{Y}\mathbf{Z}\right) \\
\alpha_{\mathbf{Y}} &\leftarrow \alpha_{\mathbf{Y}} + \rho\left(\mathbf{Y} - \mathbf{W}\right) \\
\alpha_{\mathbf{Z}} &\leftarrow \alpha_{\mathbf{Z}} + \rho\left(\mathbf{Z} - \mathbf{H}\right).
\end{aligned}$$

This formulation introduces a regularization parameter, $\rho \in \mathbb{R}_+$, that needs to be tuned (in our experiments we choose the best performing one from several candidates).

Our approach has the following differences: (1) we aim at solving alternatively *convex* problems with a few steps of primal-dual algorithms for convex problems, as opposed to aiming at solving directly the non-convex problem with an iterative approach, (2) for the convex decomposition problem, we have certificates of guarantees, which can be of used for online methods for which decomposition problems are repeatedly solved (Lefèvre et al., 2011) and (3) we use a different splitting method, namely the one of Chambolle and Pock (2011), which does not require matrix inversions, and which allows us to compute all step-sizes in a data-driven way.

3 Proposed Method

In this section we present a formulation of the convex KL decomposition problem as a first-order primal-dual algorithm (FPA), followed by the proposed NMF technique.

3.1 Primal and dual computation

We consider a vector $a \in \mathbb{R}_+^p$ and a matrix $K \in \mathbb{R}_+^{p\times q}$ as known parameters, and $x \in \mathbb{R}_+^q$ as an unknown vector to be estimated, where the following expression holds,

$$a \approx Kx,$$

and we aim at minimizing the KL divergence between a and Kx.

This is equivalent to a ND problem as defined in Problems (3) and (4), considering a as a column of the given data, K as the fixed factor, and x as a column of the estimated factor, i.e., in Problem (3) a and x are column vectors of $\mathbf{V}^\top$ and $\mathbf{W}^\top$ with the same index and K is $\mathbf{H}^\top$, and in Problem (4) a and x are columns of $\mathbf{V}$ and $\mathbf{H}$ with the same index and K is $\mathbf{W}$.

The *convex* ND problem with KL divergence is thus

$$\underset{x\in\mathbb{R}_+^q}{\text{minimize}} \; -\sum_{i=1}^{p} a_i \left(\log(K_i x/a_i) + 1\right) + \sum_{i=1}^{p} K_i x, \tag{5}$$

which may be written as

$$\underset{x\in\mathcal{X}}{\text{minimize}} \quad F(Kx) + G(x), \tag{6}$$

with

$$\begin{aligned} F(z) &= -\sum_{i=1}^{p} a_i \left(\log(z_i/a_i) + 1\right) \\ G(x) &= 1_{x\succeq 0} + \sum_{i=1}^{p} K_i x. \end{aligned}$$

Following Pock et al. (2009); Chambolle and Pock (2011), we obtain the dual problem

$$\underset{y\in\mathcal{Y}}{\text{maximize}} \quad -F^*(y) - G^*(-K^* y),$$

with

$$\begin{aligned} F^*(y) &= \sup_z \left\{ y^\top z - F(z) \right\} = -\sum_{i=1}^{p} a_i \log\left(-y_i\right) \\ G^*(y) &= \sup_x \left\{ y^\top x - G(x) \right\} = 1_{y\preceq K^\top \mathbf{1}}. \end{aligned}$$

We then get the dual problem

$$\underset{K^\top(-y) \preceq K^\top \mathbf{1}}{\text{maximize}} \quad a^\top \log(-y)\,. \tag{7}$$

In order to provide a certificate of optimality, we have to make sure that the constraint $K^\top(-y) \preceq K^\top \mathbf{1}$ is satisfied. Therefore, when it is not satisfied, we project as follows:

$$y \leftarrow y / \max\{K^\top(-y) \oslash K^\top \mathbf{1}\},$$

where $\oslash$ represents the entry-wise division operator.

3.2 Primal-dual algorithm

The general FPA framework of the approach proposed by Chambolle and Pock for Problem (6) is presented in Algorithm 1.

Algorithm 1: First-order primal-dual algorithm.

Select $K \in \mathbb{R}_+^{p\times q}$, $x \in \mathbb{R}_+^q$, $\sigma > 0$, and $\tau > 0$;
Set $\bar{x} = x_{old} = x$, and $y = Kx$;

for N *iterations* **do**
  $y \leftarrow \mathbf{prox}_{\sigma F^*}(y - \sigma K\bar{x})$;
  $x \leftarrow \mathbf{prox}_{\tau G}(x - \tau K^* y)$;
  $\bar{x} \leftarrow 2x - x_{old}$;
  $x_{old} \leftarrow x$;
end

return $x^\star = x$.

Algorithm 1 requires the computation of proximal operators $\mathbf{prox}_{\sigma F^*}(y)$ and $\mathbf{prox}_{\tau G}(x)$. These are defined as follows:

$$\begin{aligned}
\mathbf{prox}_{\sigma F^*}(y) &= \arg\min_v \left\{ \frac{\|v-y\|^2}{2\sigma} + F^*(v) \right\}, \text{ and} \\
\mathbf{prox}_{\tau G}(x) &= \arg\min_u \left\{ \frac{\|u-x\|^2}{2\tau} + G(u) \right\}.
\end{aligned}$$

For further details, see (Boyd and Vandenberghe, 2004; Rockafellar, 1997).

Using the convex functions F^* and G, we can easily solve the problems for the proximal operators and derive the following closed-form solution operators

$$\begin{aligned}
\mathbf{prox}_{\sigma F^*}(y) &= \frac{1}{2}\left(y - \sqrt{y \circ y + 4\sigma a}\right), \text{ and} \\
\mathbf{prox}_{\tau G}(x) &= \left(x - \tau K^\top \mathbf{1}\right)_+ .
\end{aligned}$$

The detailed derivation of these operators may be found in the Appendix, the first one was already computed by Sun and Févotte (2014).

3.3 Automatic heuristic selection of σ and τ

In this section, we provide a heuristic way to select σ and τ without user intervention, based on the convergence result of Chambolle and Pock (2011, Theorem 1), which states that (a) the step-sizes have to satisfy $\tau\sigma\|K\|^2 \leq 1$, where $\|K\| = \max\{\|Kx\| : x \in \mathcal{X} \textit{ with } \|x\| \leq 1\}$ is the largest singular value of K; and (b) the convergence rate is controlled by the quantity

$$C = \frac{\|y_0 - y^\star\|^2}{2\sigma} + \frac{\|x_0 - x^\star\|^2}{2\tau},$$

where $(x^\star, y^\star)$ is an optimal primal/dual pair. If $(x^\star, y^\star)$ was known, we could thus consider the following minimization problem with the constraint $\tau\sigma\|K\|^2 \leq 1$:

$$\begin{aligned} & \min_{\sigma,\tau} \frac{\|y_0 - y^\star\|^2}{2\sigma} + \frac{\|x_0 - x^\star\|^2}{2\tau} \\ \iff \quad & \min_{\sigma} \frac{\|y_0 - y^\star\|^2}{2\sigma} + \frac{\|x_0 - x^\star\|^2}{2}\sigma\|K\|^2. \end{aligned}$$

Applying first order conditions, we find that

$$\sigma = \frac{\|y_0 - y^\star\|}{\|x_0 - x^\star\|}\frac{1}{\|K\|} \quad \text{and} \quad \tau = \frac{\|x_0 - x^\star\|}{\|y_0 - y^\star\|}\frac{1}{\|K\|}.$$

However, we do not know the optimal pair $(x^\star, y^\star)$ and we use heuristic replacements. That is, we consider the unknown constants α and β, and assume that $x^\star = \alpha\mathbf{1}$ and $y^\star = \beta\mathbf{1}$ solve Problems (5) and (7). Letting $(x_0, y_0) = (\mathbf{0}, \mathbf{0})$ we have

$$\|x_0 - x^\star\| = |\alpha|\sqrt{q} \quad \text{and} \quad \|y_0 - y^\star\| = |\beta|\sqrt{p}.$$

Plugging $x^\star$ to Problem (5), we are able to find that $\alpha = \frac{\mathbf{1}^\top a}{\mathbf{1}^\top K\mathbf{1}} > 0$. Now, using optimality conditions: $y^\star \circ (Kx^\star) = -a$, we obtain $\beta = -1$.

The updated version of the parameters is:

$$\sigma = \sqrt{\frac{p}{q}}\frac{1}{\alpha\|K\|} \quad \text{and} \quad \tau = \sqrt{\frac{q}{p}}\frac{\alpha}{\|K\|}.$$

Finally, an automatic heuristic selection of step sizes σ and τ is as follows:

$$\sigma = \frac{\sqrt{p}\sum_{i=1}^p K_i\mathbf{1}}{\sqrt{q}\|K\|\sum_{i=1}^p a_i} \quad \text{and} \quad \tau = \frac{\sqrt{q}\sum_{i=1}^p a_i}{\sqrt{p}\|K\|\sum_{i=1}^p K_i\mathbf{1}}.$$

Note the invariance by rescaling of a and K.

3.4 Implementation

The proposed method is based on Algorithm 1. The required parameters to solve each ND problem are

- Problem (3):
  - $a \leftarrow (\mathbf{V}^\top)_i$
  - $K \leftarrow \mathbf{H}^\top$
  - $x \leftarrow (\mathbf{W}^\top)_i$

- $\sigma \leftarrow \sqrt{\frac{m}{r}} \frac{\mathbf{1}^\top \mathbf{H} \mathbf{1}}{\mathbf{1}^\top \left(\mathbf{V}^\top\right)_i \|\mathbf{H}\|}$
- $\tau \leftarrow \sqrt{\frac{r}{m}} \frac{\mathbf{1}^\top \left(\mathbf{V}^\top\right)_i}{\mathbf{1}^\top \mathbf{H} \mathbf{1} \|\mathbf{H}\|}$

- Problem (4):
  - $a \leftarrow \mathbf{V}_i$
  - $K \leftarrow \mathbf{W}$
  - $x \leftarrow \mathbf{H}_i$
  - $\sigma \leftarrow \sqrt{\frac{n}{r}} \frac{\mathbf{1}^\top \mathbf{W} \mathbf{1}}{\mathbf{1}^\top \mathbf{V}_i \|\mathbf{W}\|}$
  - $\tau \leftarrow \sqrt{\frac{r}{n}} \frac{\mathbf{1}^\top \mathbf{V}_i}{\mathbf{1}^\top \mathbf{W} \mathbf{1} \|\mathbf{W}\|}$.

The previous summary treats each ND problem by columns. For algorithmic efficiency, we work directly with the matrices, e.g., $a \in \mathbb{R}_+^{n \times m}$ instead of $\mathbb{R}_+^n$. We also include normalization steps such that the columns of $\mathbf{W}$ have sums equal to 1. The stopping criteria is enabled for maximum number of iterations (access to data) and for duality gap tolerance.

3.5 Extension to NMF

A pseudo-code of the first-order primal-dual algorithm for non-negative matrix factorization can be found in Algorithm 2. It corresponds to alternating between minimizing with respect to $\mathbf{H}$ and minimizing with respect to $\mathbf{W}$. A key algorithmic choice is the number of inner iterations $\texttt{iter}_{ND}$ of the convex method, which we consider in Section 4.

The running-time complexity is still $O(rnm)$ for each inner iterations. Note moreover, that computing the largest singular value of $\mathbf{H}$ or $\mathbf{W}$ (required for the heuristic selection of step-sizes everytime we switch from one convex problem to the other) is of order $O(r \max\{m, n\})$ and is thus negligible compared to the iteration cost.

3.6 Extension to topic models

Probabilistic latent semantic analysis (Hofmann, 1999) or latent Dirichlet allocation (Blei et al., 2003), generative probabilistic models for collections of discrete data, have been extensively used in text analysis. Their formulations are related to ours in Problem (5), where we just need to include an additional constraint: $\mathbf{1}^\top x = 1$. In this sense, if we modify G, i.e., $G(x) = 1\{\mathbf{1}^\top x = 1;\ x \succeq 0\} + 1^\top K x$, we can use Algorithm 1 to find the latent topics. It is important to mention that herein $\mathbf{prox}_{\tau G}(x)$ does not have a closed solution, but can be efficiently solved with dedicated methods for orthogonal projections on the simplex (Maculan and de Paula, 1989).

4 Experimental Results

The proposed technique was tested on synthetic data, the CBCL face images database and a music excerpt from a recognized jazz song by Louis Armstrong & His Hot Five. The performance of the proposed first-order primal-dual algorithm (FPA) was compared against the traditional multiplicative updates algorithm (MUA) by Lee and Seung (2000) and the contemporary alternating direction method of multipliers (ADMM) by Sun and Févotte (2014). The three techniques were implemented in Matlab.

4.1 Synthetic data

A given matrix $\mathbf{V}$ of size $n = 200$ and $m = 1000$ is randomly generated from the uniform distribution $\mathcal{U}(0, 750)$. The low-rank element was set to $r = 15$. Initializations $\mathbf{W}_0$ and $\mathbf{H}_0$ are defined by the standard normal distribution's magnitude plus a small offset.

4.1.1 ND problem

To examine the accuracy of our method, we first apply Algorithm 2 to convex ND problems for fixed values of n, m and r, solving separately Problems (3) and (4). For both problems, we set

Algorithm 2: Proposed technique.

Select $\mathbf{V} \in \mathbb{R}_+^{n\times m}$, $\mathbf{W}_0 \in \mathbb{R}_+^{n\times r}$, and $\mathbf{H}_0 \in \mathbb{R}_+^{r\times m}$;

Set $\mathbf{W} = \bar{\mathbf{W}} = \mathbf{W}_{old} = \mathbf{W}_0$, $\mathbf{H} = \bar{\mathbf{H}} = \mathbf{H}_{old} = \mathbf{H}_0$, and $\chi = \mathbf{W}\mathbf{H}$;

while *stopping criteria not reached* **do**

  Normalize $\mathbf{W}$ and set $\sigma = \sqrt{\frac{m}{r}}\frac{\mathbf{1}^\top\mathbf{H}\mathbf{1}}{\mathbf{1}^\top\mathbf{V}^\top\|\mathbf{H}\|}\mathbf{1}$, $\tau = \sqrt{\frac{r}{m}}\frac{\mathbf{1}^\top\mathbf{V}^\top}{\mathbf{1}^\top\mathbf{H}\mathbf{1}\|\mathbf{H}\|}\mathbf{1}$, and $\mathbf{H}(-\chi^\top) \leq \mathbf{H}\mathbf{1}$;

  for $\texttt{iter}_{ND}$ *iterations* **do**

    $\chi^\top \leftarrow \chi^\top - \sigma \circ \left(\bar{\mathbf{W}}\mathbf{H}\right)^\top$;
    $\chi^\top \leftarrow \frac{1}{2}\left(\chi^\top - \sqrt{\chi^\top \circ \chi^\top + 4\sigma \circ \mathbf{V}^\top}\right)$;
    $\mathbf{W}^\top \leftarrow \left(\mathbf{W}^\top - \tau \circ \left(\mathbf{H}\left(\chi^\top + \mathbf{1}\right)\right)\right)_+$;
    $\bar{\mathbf{W}}^\top \leftarrow 2\mathbf{W}^\top - \mathbf{W}_{old}^\top$;
    $\mathbf{W}_{old}^\top \leftarrow \mathbf{W}^\top$;

  end

  Normalize $\mathbf{H}$ and set $\sigma = \sqrt{\frac{n}{r}}\frac{\mathbf{1}^\top\mathbf{W}\mathbf{1}}{\mathbf{1}^\top\mathbf{V}\|\mathbf{W}\|}\mathbf{1}$, $\tau = \sqrt{\frac{r}{n}}\frac{\mathbf{1}^\top\mathbf{V}}{\mathbf{1}^\top\mathbf{W}\mathbf{1}\|\mathbf{W}\|}\mathbf{1}$, and $\mathbf{W}^\top(-\chi) \leq \mathbf{W}^\top\mathbf{1}$;

  for $\texttt{iter}_{ND}$ *iterations* **do**

    $\chi \leftarrow \chi - \sigma \circ \left(\mathbf{W}\bar{\mathbf{H}}\right)$;
    $\chi \leftarrow \frac{1}{2}\left(\chi - \sqrt{\chi \circ \chi + 4\sigma \circ \mathbf{V}}\right)$;
    $\mathbf{H} \leftarrow \left(\mathbf{H} - \tau \circ \left(\mathbf{W}^\top\left(\chi + \mathbf{1}\right)\right)\right)_+$;
    $\bar{\mathbf{H}} \leftarrow 2\mathbf{H} - \mathbf{H}_{old}$;
    $\mathbf{H}_{old} \leftarrow \mathbf{H}$;

  end

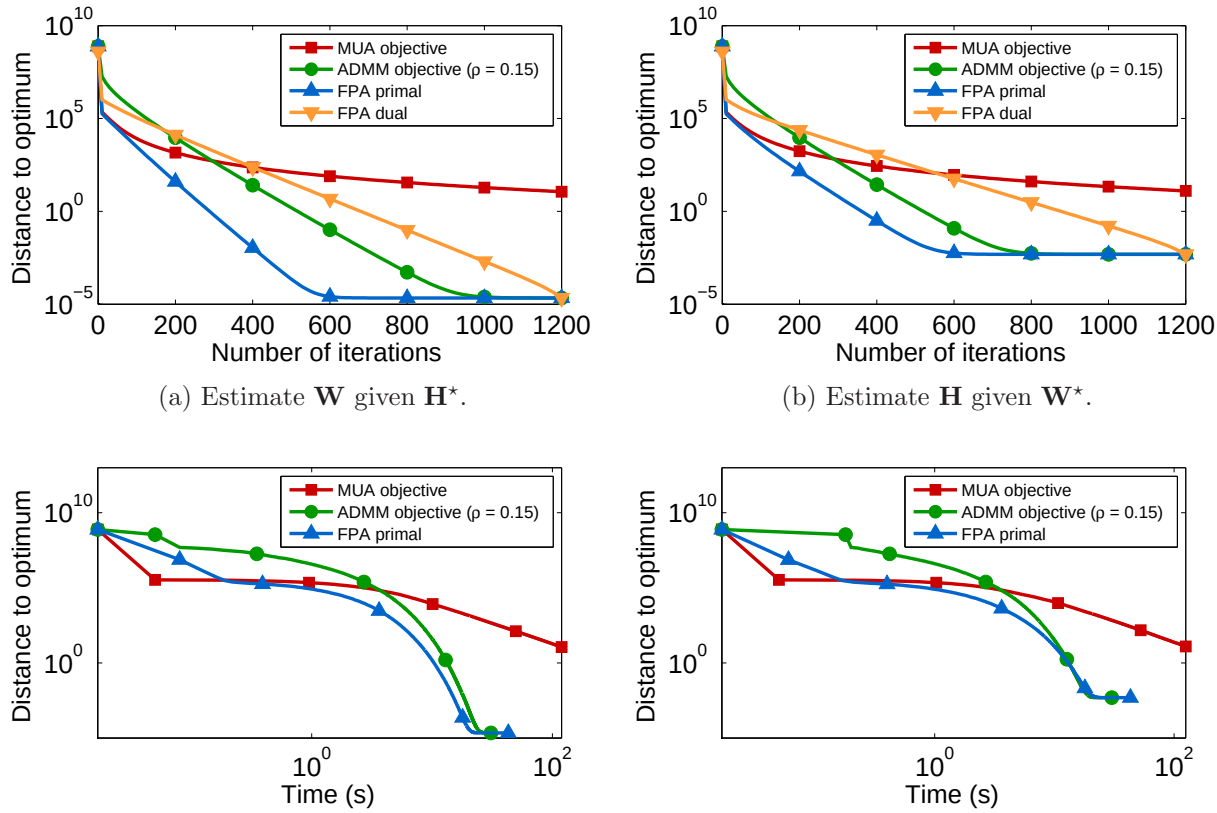
end

return $\mathbf{W}^\star = \mathbf{W}$, *and* $\mathbf{H}^\star = \mathbf{H}$.

the number of iterations of the traditional MUA and contemporary ADMM to 1200, as well as for the proposed FPA. Optimal factors $\mathbf{W}^\star$ and $\mathbf{H}^\star$ are obtained by running 5000 iterations of the MUA, starting from $\mathbf{W}_0$ and $\mathbf{H}_0$. For the first ND problem, we fix $\mathbf{H}$ to $\mathbf{H}^\star$ and estimate $\mathbf{W}$ starting from $\mathbf{W}_0$; for the second one, we fix $\mathbf{W}$ to $\mathbf{W}^\star$ and estimate $\mathbf{H}$ from $\mathbf{H}_0$. The optimal regularization parameter of ADMM, the tuning parameter that controls the convergence rate, is $\rho = 0.15$ (small values imply larger step sizes, which may result in faster convergence but also instability). Figure 1 (a-b) present us the distance to optimum of MUA and ADMM, as well as for the primal and dual of our technique that reveals strong duality. The FPA and ADMM algorithm converge to the same point, whereas the MUA exhibits slow convergence. Note moreover the significant advantage towards our algorithm FPA, together with the fact that we set automatically all step-sizes. In Figure 1 (c-d), we illustrate the distance to optimum versus time of the three methods.

4.1.2 NMF problem

The setting is slightly different as in the ND experiment, we increased the problem dimension to $n = 250$, $m = 2000$ and $r = 50$, and repeat both previously presented experiments. For all methods,



(a) Estimate $\mathbf{W}$ given $\mathbf{H}^\star$.

(b) Estimate $\mathbf{H}$ given $\mathbf{W}^\star$.

(c) Estimate $\mathbf{W}$ given $\mathbf{H}^\star$.

(d) Estimate $\mathbf{H}$ given $\mathbf{W}^\star$.

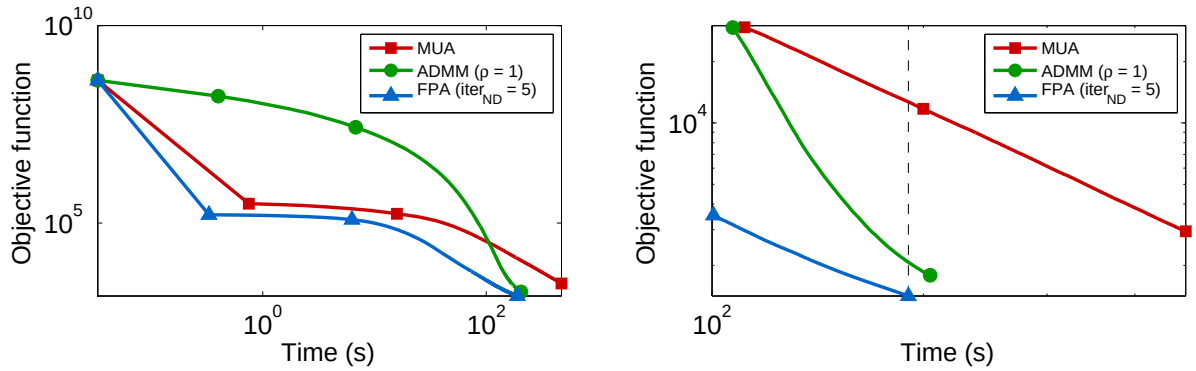
Figure 1: ND on synthetic data. (a-b) Distance to optimum versus iteration number. Distance to optimum reveals the difference between the values of the objective function and optimal point, $p^\star$. In the case of the dual function values, the distance to optimum is the difference between $p^\star$ and the dual points. (c-d) Distance to optimum versus time.

we set the number of iterations to 3000. The parameter $\texttt{iter}_{ND}$ indicates the number of iterations to solve each ND problem. We set $\texttt{iter}_{ND}$ to 5 iterations. To have a fair comparison between algorithms, for FPA, the number of iterations means access to data, i.e., we use 5 iterations to solve Problem (3) (as well as for Problem (4)), and repeat this 600 times. The optimal regularization parameter of the ADMM is here $\rho = 1$.

In Figure 2 we present the objective function of the three algorithm for the non-convex Problem (2). The MUA initially reports high decrement in the objective, but as time increases it exhibits evident slow tail convergence. The FPA primal objective decreases dramatically in only seconds (few iterations), and furthermore, it presents fast tail convergence achieving the lowest objective value. The ADMM has poor initial performance, but then achieves an optimal value similar to the one obtained by FPA. In order to show that FPA converges faster and with lower computational cost, we store the cost function values and computation times at each iteration. The total time required by the FPA was 190s, whereas 205s by the ADMM and 473s by the MUA. Then we analyze the ADMM and MUA for the same time 190s (the vertical dotted line in Figure 2 (b)), i.e., 2786 and 1211 iterations, respectively: the competitive algorithms have a significantly larger cost function for the same running time. The result of this experiment is illustrated in Figure 2 (b). The results considering the objective function versus iteration number may be found in the Appendix.

4.1.3 NMF with warm restarts

The problem dimension is $n = 150$, $m = 2000$ and $r_1 = 50$. We run 3000 iterations of each method using initializations $\mathbf{W}_0$ and $\mathbf{H}_0$; then we increase ten times the low-rank element, $r_2 = 100$; and



(a) Objective function versus time. (b) Objective function versus time (zoomed).

Figure 2: NMF on synthetic data. Recall that the dual function is not presented due to the non-convexity of the NMF problem.

finally run 2000 more iterations, producing $\mathbf{W}_2$ and $\mathbf{H}_2$. The idea is to use as initializations the estimations obtained after the first 3000 iterations, $\mathbf{W}_1$ and $\mathbf{H}_1$, considering that the low-rank element changed. A trivial solution could be to include random entries so that $\mathbf{W}_1$ and $\mathbf{H}_1$ have the proper dimensions, but that increases the objective value, diminishing the estimations. On the other hand, if we include zero entries so that $\mathbf{W}_1$ and $\mathbf{H}_1$ have the proper dimensions, we would be in a saddle-point where none of the algorithms could perform. However, if we set only one factor with zero entries, $[\mathbf{W}_1, c\mathbf{1}] \in \mathbb{R}^{n\times r_2}$ with $c = 0$, and the other one with non-zero values, $[\mathbf{H}_1; \nu] \in \mathbb{R}^{r_2\times m}$, we still maintain the same last objective value and perform FPA. In this situation, MUA cannot perform either (because of the absorbing of zeros), therefore we try some values of c to run the algorithm. Figure 3 illustrates the proposed experiment. Notice that as $c \to 0$, the MUA starts to get stuck in a poor local optima, i.e., the one obtained with $\mathbf{W}_1$ and $\mathbf{H}_1$. ADMM has a similar behavior as FPA, therefore, it is not displayed.

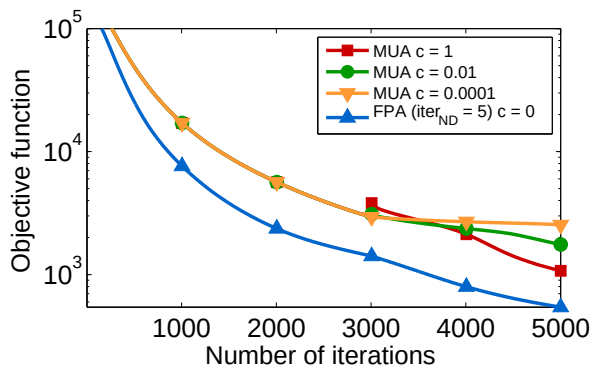


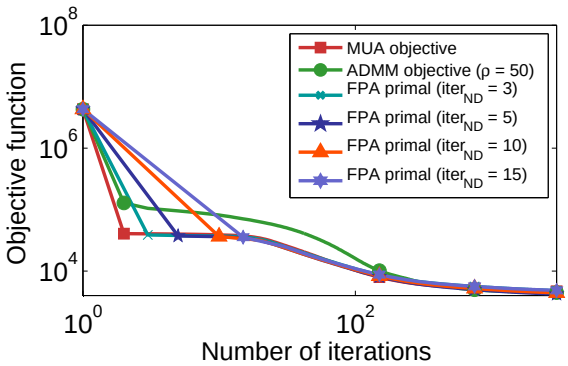
Figure 3: NMF with warm restarts on synthetic data. Value of the objective function at each iteration.

4.2 MIT-CBCL Face Database #1

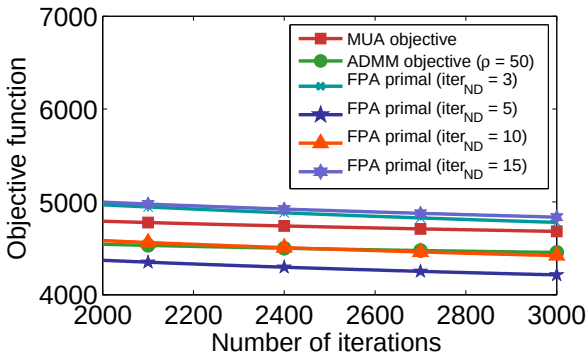
We use the CBCL face images database (Sung, 1996) composed of $m = 2429$ images of size $n = 361$ pixels. The low-rank element was set to $r = 49$. Figure 4 shows samples from the database.

Our next experiment is to determine the optimal the number of iterations for the current database. Therefore, we run 3000 iterations of FPA, using 3, 5, 10 and 15 iterations for the ND problem. The MUA and ADMM (ρ=50) algorithms are performing as well. Figure 5 illustrates the decay of the objective function of the FPA, MUA and ADMM algorithms.

Figure 4: MIT-CBCL Face Database #1 samples.



(a) Objective function versus iteration number.

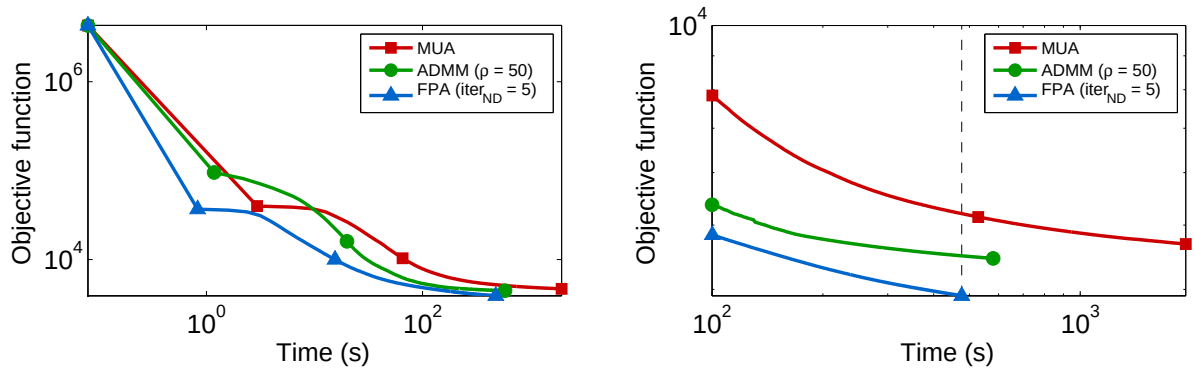


(b) Objective versus iteration number (zoomed).

Figure 5: NMF on the CBCL database. Value of the objective function at each iteration solving Problem (2) varying the number of iterations to solve each ND problem.

We appreciate that setting the number of iterations to 3 yield to over-alternation, whereas using 15 or even more iterations result in an under-alternating method. Using 10 iterations reveal good performance, but the best trade-off is obtained with 5 iterations. Therefore, we set $\texttt{iter}_{ND} = 5$, i.e., the number of iterations to solve Problem (3) and Problem (4). All following results in the MIT-CBCL Face Database #1 are with the same setting.

Finally, in Figure 6 (a) we present the objective function of the three algorithm for the non-convex Problem (2), where all algorithms perform similarly. However, in the zoomed Figure 6 (b) we can appreciate that the MUA presents the slowest convergence, whereas the proposed method the fastest one. The results considering the objective function versus iteration number may be found in the Appendix.



(a) Objective function versus time.

(b) Objective function versus time (zoomed).

Figure 6: NMF on the CBCL face image database.

4.3 Music excerpt from the song "My Heart (Will Always Lead Me Back to You)"

The last experiment is to decompose a 108-second-long music excerpt from "My Heart (Will Always Lead Me Back to You)" by Louis Armstrong & His Hot Five in the 1920s (Févotte et al., 2009). The time-domain recorded signal is illustrated in Figure 7.

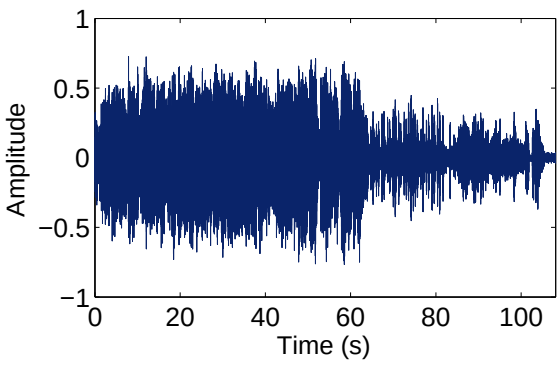


Figure 7: Time-domain recorded signal.

The recording consists of a trumpet, a double bass, a clarinet, a trombone, and a piano. The recorded signal is original unprocessed mono material contaminated with noise. The signal was downsampled to 11025 kHz, yielding $1{,}19\times10^6$ samples. The Fourier Transform of the recorded signal was computed using a sinebell analysis window of length 23 ms with 50% overlap between two frames, leading to $m = 9312$ frames and $n = 129$ frequency bins. Additionally, we set $r = 10$. Figure 8 illustrates the previously described spectrogram.

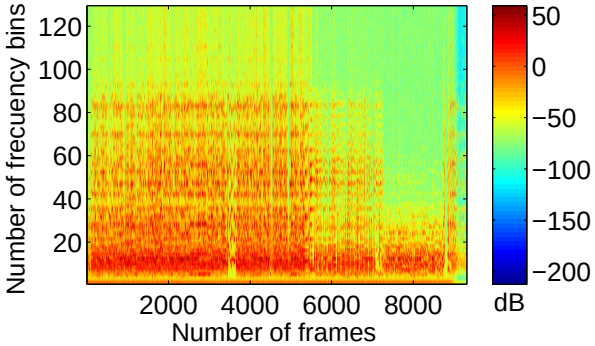
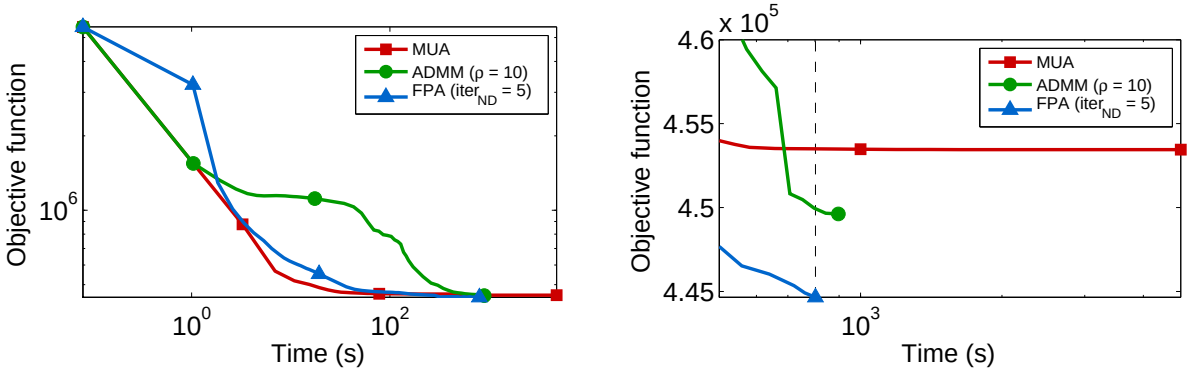


Figure 8: Log-power spectrogram.

The decomposition of the song is produced by the three algorithms. We initialize them with the same random values $\mathbf{W}_0$ and $\mathbf{H}_0$. For a fair competition, the number of iterations is set to 5000 for MUA and ADMM, and for our algorithm FPA we consider it as access to data, i.e., we use 5 iterations for the ND, repeating it 1000 times. For comparison, we measure the computation time of the three techniques. FPA has a run time of 13 min, whereas the ADMM ($\rho = 10$) one of 15 min and the MUA one of 80 min.



(a) Objective function versus time.

(b) Objective function versus time (zoomed).

Figure 9: NMF on an excerpt of Armstrong's song.

In this experiment, Figure 9 illustrates the evolution of the objective of the three techniques along *time*. Initially the MUA obtained the lowest objective value, but as previously discussed, as the number of iterations increases the MUA starts exhibiting evident slow tail convergence and since approximately 100s it is reached by the FPA and shows no further substantial decrement, i.e., it gets stuck in a worse local optima. FPA converges to a slight lower cost value, overpassing MUA. Finally, ADMM reveals a slow initial performance on this dataset, but later converges to a similar point as the previous algorithms. The results considering the objective function versus iteration number may be found in the Appendix.

5 Conclusion

We have presented an alternating projected gradient descent technique for NMF that minimizes the KL divergence loss; this approach solves convex ND problems with the FPA. Our approach demonstrated faster convergence than the traditional MUA by Lee and Seung (2000) and contemporary ADMM by Sun and Févotte (2014). The FPA introduces a new parameter, the number of iterations for each convex ND problem. Experiments reveal that the number of iterations is mostly bounded between 3 and 10 iterations, which leads to a trivial tuning by the user. Therefore, our

algorithm affords reasonable simplicity, where further user manipulation is not required. Finally, an extension to latent Dirichlet allocation and probabilistic latent semantic indexing can be easily implemented using our proposed method, thus allowing to go beyond the potential slowness of the expectation-maximization (EM) algorithm.

Acknowledgements

This work was partially supported by a grant from the European Research Council (ERC SIERRA 239993).

References

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40:120–145, 2011.

C. Févotte and A. T. Cemgil. Nonnegative matrix factorizations as probabilistic inference in composite models. In *Proceedings of the 17th European Signal Processing Conference*, 2009.

C. Févotte, N. Bertin, and J. L. Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural Computation*, 21:793–830, 2009.

N. Gillis. *Nonnegative Matrix Factorization: Complexity, Algorithms and Applications*. PhD thesis, Université Catholique de Louvain, 2011.

N. Guduru. Text mining with support vector machines and non-negative matrix factorization algorithms. Master's thesis, University of Rhode Island, 2006.

T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, 1999.

D. Kim, S. Sra, and I. S. Dhillon. Fast projection-based methods for the least squares nonnegative matrix approximation problem. *Statistical Analysis and Data Mining*, 1:38–51, 2008.

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

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, 2000.

A. Lefèvre, F. Bach, and C. Févotte. Online algorithms for nonnegative matrix factorization with the Itakura-Saito divergence. In *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2011.

C. J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19:2756–2779, 2007.

N. Maculan and G. Galdino de Paula. A linear-time median-finding algorithm for projecting a vector on the simplex of $\mathbb{R}^n$. *Operations research letters*, 8:219–222, 1989.

P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error. *Environmetrics*, 5:111–126, 1994.

T. Pock, D. Cremers, H. Bischof, and A. Chambolle. An algorithm for minimizing the mumford-shah functional. In *Proceedings of the 12th International Conference on Computer Vision*, 2009.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1997.

D. L. Sun and C. Févotte. Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. In *Proceedings of the 39th International Conference on Acoustic, Speech and Signal Processing*, 2014.

K. K. Sung. *Learning and Example Selection for Object and Pattern Recognition*. PhD thesis, Massachusetts Institute of Technology, 1996.

Y. Wang, Y. Jia, C. Hu, and M. Turk. Non-negative matrix factorization framework for face recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 19:495–511, 2005.

Appendix

Derivation of proximal operators

The definition of the proximal operator of F^* and G, i.e., $(I+\sigma\partial F^*)^{-1}(y)$ and $(I+\tau\partial G)^{-1}(x)$, respectively, is as follows (Pock et al., 2009; Chambolle and Pock, 2011):

$$\begin{aligned}(I+\sigma\partial F^*)^{-1}(y) &= \arg\min_v \left\{\frac{\|v-y\|^2}{2\sigma}+F^*(v)\right\}, \text{ and}\\ (I+\tau\partial G)^{-1}(x) &= \arg\min_u \left\{\frac{\|u-x\|^2}{2\tau}+G(u)\right\},\end{aligned}$$

where ∂F^* and ∂G are the subgradients of the convex functions F^* and G.

To facilitate the computation of $(I+\sigma\partial F^*)^{-1}(y)$, we consider Moreau's identity

$$\begin{aligned}y &= (I+\tau\partial F^*)^{-1}(y)+\sigma\left(I+\frac{1}{\sigma}\partial F\right)^{-1}\left(\frac{y}{\sigma}\right)\\ &= \mathbf{prox}_{\sigma F^*}(y)+\sigma\,\mathbf{prox}_{F/\sigma}(y/\sigma).\end{aligned}$$

Let us consider the variable $v\in\mathcal{Y}$, and using Moreau's identity, we can compute

$$\begin{aligned}\mathbf{prox}_{\sigma F^*}(y) &= y-\sigma\,\mathbf{prox}_{F/\sigma}(y/\sigma)\\ &= y-\sigma\arg\min_v\left\{\frac{\sigma}{2}\left\|v-\frac{y}{\sigma}\right\|^2+F(v)\right\}\\ &= y-\sigma\arg\min_v\left\{\frac{\sigma}{2}\left\|v-\frac{y}{\sigma}\right\|^2-\sum_{i=1}^n a_i\left(\log\left(\frac{v_i}{a_i}\right)+1\right)\right\}\\ &= y-\sigma\arg\min_v\left\{\sum_{i=1}^n\frac{\sigma}{2}\left(v_i^2-\frac{2v_iy_i}{\sigma}+\frac{y_i^2}{\sigma^2}\right)-a_i\left(\log\left(\frac{v_i}{a_i}\right)+1\right)\right\}\\ &= y-\sigma\arg\min_v\left\{\sum_{i=1}^n\frac{\sigma}{2}v_i^2-y_iv_i-a_i\log(v_i)\right\}.\end{aligned}$$

Applying first order conditions to obtain the minimum:

$$\begin{aligned}\frac{d}{dv_i}\left\{\frac{\sigma}{2}v_i^2-y_iv_i-a_i\log(v_i)\right\}=0 \quad &\Longrightarrow \quad \sigma v_i-y_i-\frac{a_i}{v_i}=0\\ &\Longrightarrow \quad \sigma v_i^2-y_iv_i-a_i=0\\ &\Longrightarrow \quad v_i=\frac{y_i\pm\sqrt{y_i^2+4\sigma a_i}}{2\sigma}\\ &\Longrightarrow \quad v=\frac{y+\sqrt{y\circ y+4\sigma a}}{2\sigma}, \text{ as } v\succ 0.\end{aligned}$$

Finally, the proximal operator is as follows:

$$\mathbf{prox}_{\sigma F^*}(y)=\frac{1}{2}\left(y-\sqrt{y\circ y+4\sigma a}\right).$$

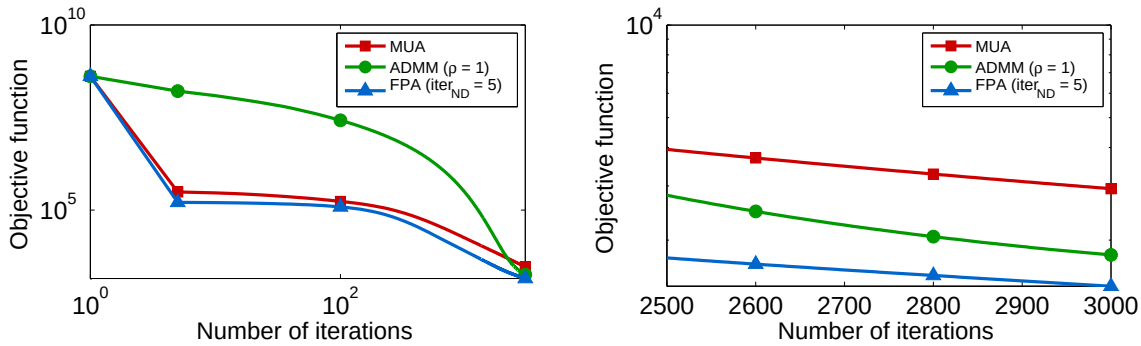
For the second proximal operator, we consider $u\in\mathcal{X}$ and compute $\mathbf{prox}_{\tau G}(x)$ as

$$
\begin{aligned}
\mathbf{prox}_{\tau G}(x) &= \arg\min_{u \succeq 0} \left\{ \frac{\|u - x\|^2}{2\tau} + G(u) \right\} \\
&= \arg\min_{u \succeq 0} \left\{ \frac{\|u - x\|^2}{2\tau} + \sum_{i=1}^{n} K_i u \right\} \\
&= \left(x - \tau K^\top \mathbf{1} \right)_+ .
\end{aligned}
$$

Synthetic data: additional results

NMF problem

A given matrix $\mathbf{V}$ of size $n = 250$ and $m = 2000$ is randomly generated from the uniform distribution $\mathcal{U}(0, 750)$. The low-rank element was set to $r = 50$. For the three methods, we set the number of iterations to 3000. We set $\texttt{iter}_{ND}$ to 5 iterations. The optimal tuning parameter of the ADMM is here $\rho = 1$. In Figure 10 we present the objective function versus iteration number of the three algorithms for the non-convex NMF problem.



(a) Objective versus iteration number. (b) Zoomed version.

Figure 10: NMF on synthetic data. It is important to recall that the dual function is not presented due to the non-convexity of the NMF problem.

MIT-CBCL Face Database #1: additional results

ND problem

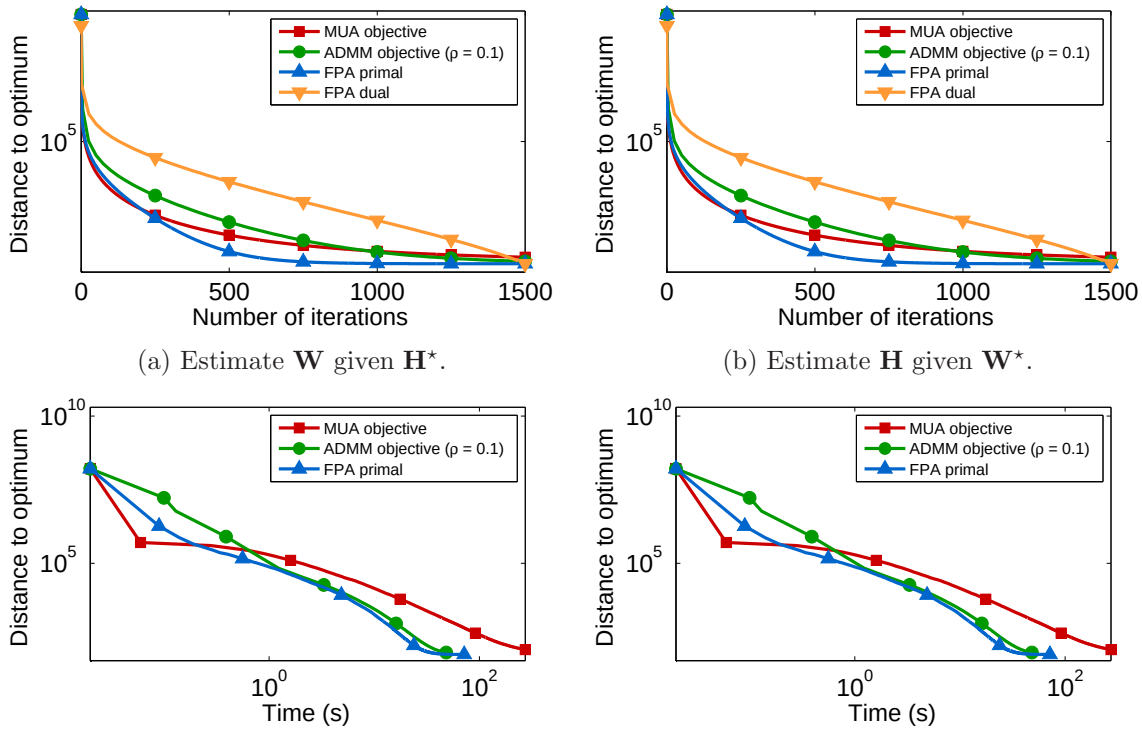
We solve convex ND problems for fixed values of n, m and r, setting the number of iterations of all algorithms to 1500. Optimal factors $\mathbf{W}^\star$ and $\mathbf{H}^\star$ are obtained by running 5000 iterations of the MUA. The optimal tuning parameter of the ADMM is here $\rho = 0.1$. Figure 11 (a-b) presents us the distance to optimum of the MUA and ADMM, as well as for the primal and dual of our technique that reveals strong duality in all experiments. In Figure 11 (c-d), we illustrate the distance to optimum versus time of the three methods.

NMF problem

For all methods, we set the number of iterations to 3000. We set $\texttt{iter}_{ND}$ to 5 iterations. The optimal tuning parameter of the ADMM is here $\rho = 50$. In Figure 12 we present the objective function versus iteration number of the three algorithms.

Features learned from the CBCL face image database

The features learned from the CBCL face image database obtained with the three algorithms is presented in Figure 13. The figure reveals the parts-based learned by the algorithm, i.e. $\mathbf{W}$.



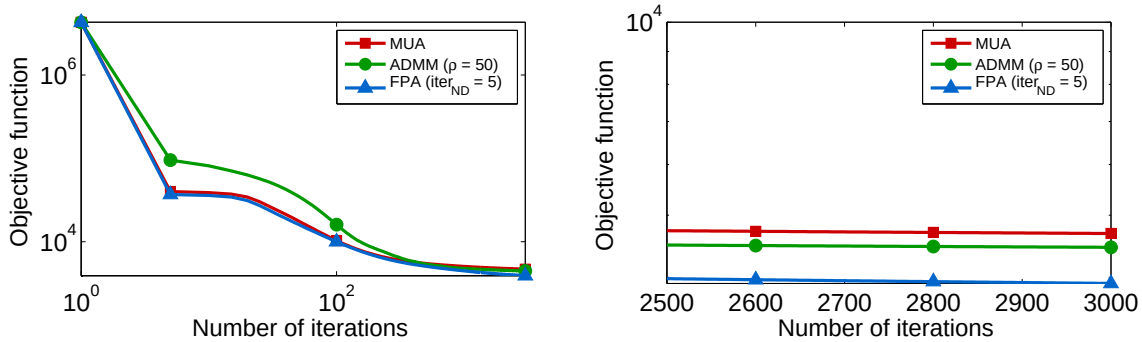
(a) Estimate $\mathbf{W}$ given $\mathbf{H}^\star$.

(b) Estimate $\mathbf{H}$ given $\mathbf{W}^\star$.

(c) Estimate $\mathbf{W}$ given $\mathbf{H}^\star$.

(d) Estimate $\mathbf{H}$ given $\mathbf{W}^\star$.

Figure 11: Distance to optimum versus (a-b) iteration number, and (c-d) time.



(a) Objective versus iteration number.

(b) Zoomed version.

Figure 12: NMF on the CBCL face image database.

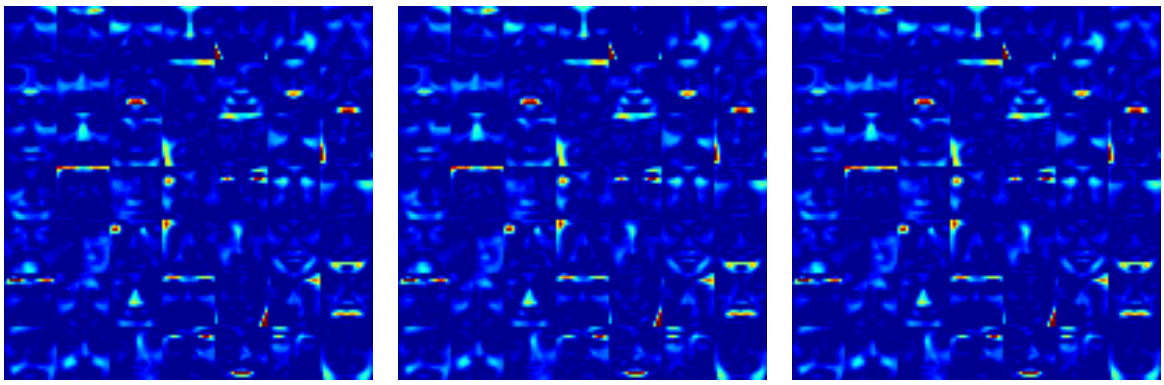
"My Heart (Will Always Lead Me Back to You)": additional results

ND problem

We solve convex ND problems for fixed values of n, m and r, setting the number of iterations of all algorithms to 2000. Optimal factors $\mathbf{W}^\star$ and $\mathbf{H}^\star$ are obtained by running 5000 iterations of the MUA. The optimal tuning parameter of the ADMM is here $\rho = 0.5$. Figure 14 (a-b) presents us the distance to optimum of the MUA and ADMM, as well as for the primal and dual of our technique that reveals strong duality in all experiments. In Figure 14 (c-d), we illustrate the distance to optimum versus time of the three methods.

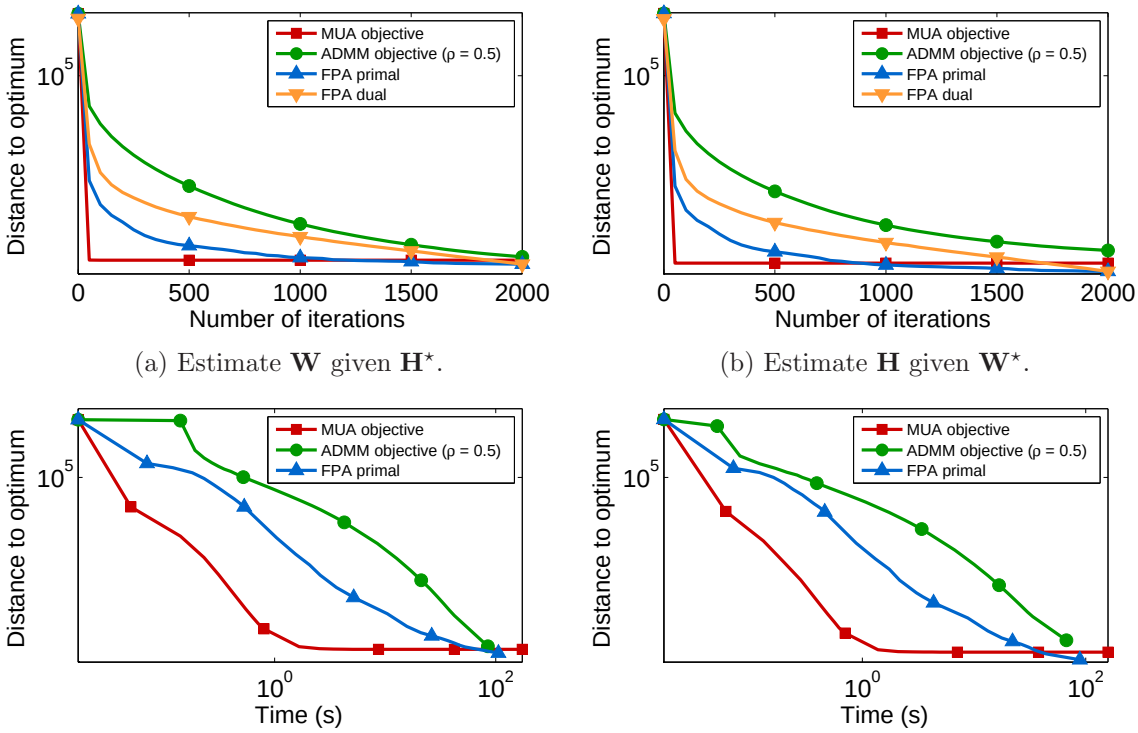
NMF problem

For all methods, we set the number of iterations to 5000. We set $\texttt{iter}_{ND}$ to 5 iterations. The optimal tuning parameter of the ADMM is here $\rho = 10$. In Figure 15 we present the objective



(a) MUA. (b) ADMM. (c) FPA.

Figure 13: Features learned from the CBCL face image database.

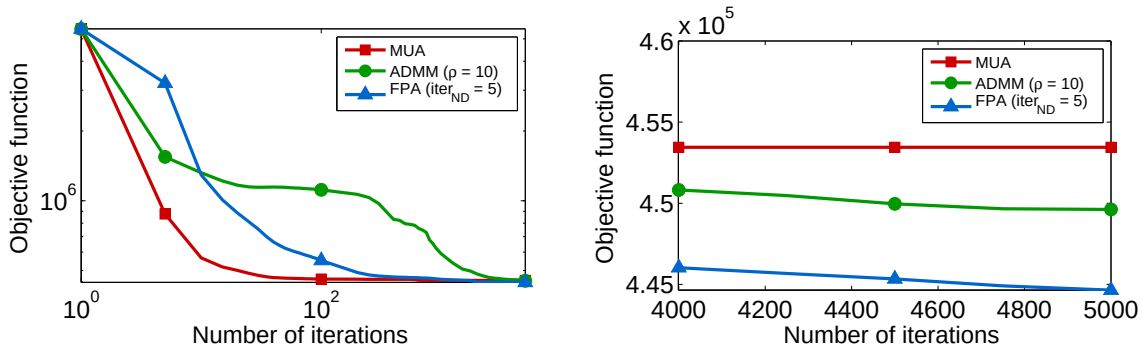


(a) Estimate $\mathbf{W}$ given $\mathbf{H}^\star$. (b) Estimate $\mathbf{H}$ given $\mathbf{W}^\star$.

(c) Estimate $\mathbf{W}$ given $\mathbf{H}^\star$. (d) Estimate $\mathbf{H}$ given $\mathbf{W}^\star$.

Figure 14: Distance to optimum versus (a-b) iteration number, and (c-d) time.

function versus iteration number of the three algorithms.



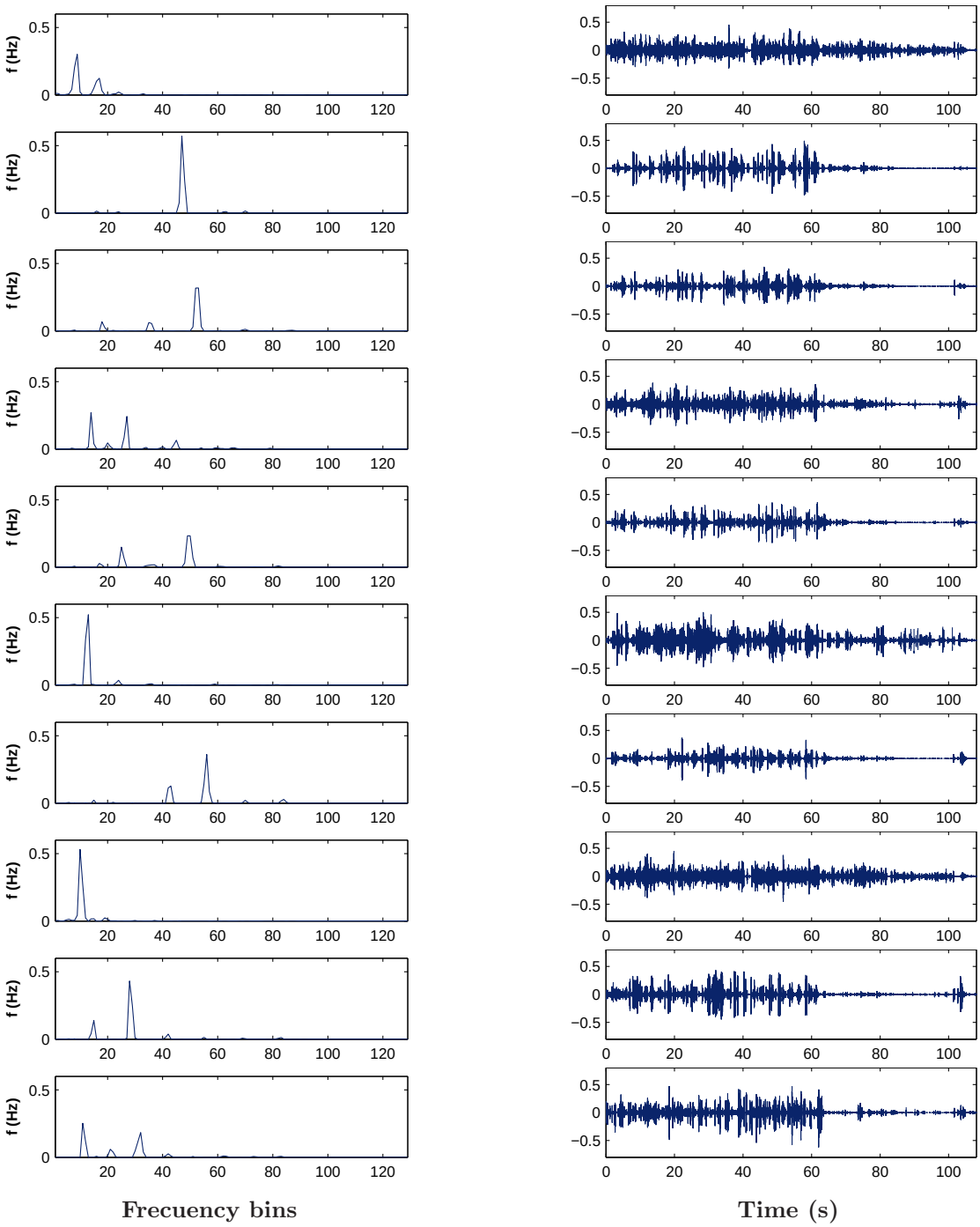
(a) Objective versus iteration number.

(b) Zoomed version.

Figure 15: NMF on an excerpt of Armstrongs song.

Features learned from the song

The decomposition of the song by Louis Armstrong and band obtained with the proposed FPA is presented in Figure 16, revealing the parts-based learned by the algorithm, i.e., $\mathbf{W}$. The time-domain signal is recovered from Wiener filtering (Févotte et al., 2009).



W, parts-based learned by the FPA.

Time-domain recovered signal.

Figure 16: Decomposition of Louis Armstrong and band song.