A unifying framework and comparison of algorithms for non-negative matrix factorization

Asger Hobolth[1,⋆], Qianyun Guo[1], Astrid Kousholt[1] and Jens Ledet Jensen[2]

1. Bioinformatics Research Center, Aarhus University

2. Department of Mathematics, DIGIT, Aarhus University

⋆. Corresponding author. Email: asger@birc.au.dk

August 13, 2018

Abstract

Non-negative matrix factorization (NMF) is an increasingly popular unsupervised learning method. However, parameter estimation in the NMF model is a difficult high-dimensional optimization problem. We consider algorithms of the alternating least square type. Solutions to the least square problem fall in two categories. The first category are iterative algorithms, and include algorithms such as the majorize-minimize (MM) algorithm, coordinate descent, gradient descent, and the Févotte-Cemgil Expectation-Maximization (FC-EM) algorithm. We introduce a new family of iterative updates based on a generalization of the FC-EM algorithm. The coordinate descent, gradient descent and FC-EM algorithms are special cases of this new EM family of iterative procedures. Curiously, we show that the MM algorithm is *never* a member of our general EM algorithm. The second category is based on convex analysis and in particular cone projection. We describe and prove a cone projection algorithm tailored to the non-negative least square problem. We compare the algorithms on a test case and on the problem of identifying mutational signatures in human cancer. We generally find that cone projection is an attractive choice. Furthermore, in the cancer application we find that a mix-and-match strategy performs better than running each algorithm in isolation.

Key words

Cone projection, EM-algorithm, mutational signatures, non-negative least squares (NLS), non-negative matrix factorization (NMF).

1 Introduction

Non-negative matrix factorization (NMF) is a popular tool for unsupervised learning. Lee and Seung (1999) use NMF for feature extraction in images, and Févotte et al. (2009) apply NMF for denoising a recording of Louis Armstrong and His Hot Five. An important recent application is in cancer genomics where the model is used to learn and understand the somatic mutations in a cancer tumour. In particular common so-called signatures are identified across hundreds of cancer patients. These signatures can often be attributed to exposures such as UV-light, asbestos or smoking; we refer to Baez-Ortega and Gori (2017) for a recent review on mutational signatures in cancer.

In NMF a data matrix V with non-negative entries of dimension $M \times N$ is factorized into a matrix W of dimension $M \times K$ and H of dimension $K \times N$ such that WH approximately equals V. The entries in W and H must be non-negative and we assume K is small relative to M and N. The aim is to find W and H such that the Frobenius norm

$$\|V - WH\|_{\mathrm{F}} = \left(\sum_{m=1}^{M} \sum_{n=1}^{N} \left(V_{mn} - (WH)_{mn} \right)^2 \right)^{\frac{1}{2}} \tag{1}$$

is minimized (in Section 7 we discuss other cost functions). We consider algorithms of the alternating non-negative least square (NLS) type. This means that $\|V - WH\|_{\mathrm{F}}$ is decreased using an iterative procedure where either a column of H or a row of W is updated in each iteration. This task is a NLS problem. The alternating NLS algorithm is a special case of the more general class of iterative partial maximization algorithms (sometimes also called block coordinate descent algorithms). Convergence properties for these types of algorithms are described in e.g. Appendix A4 in Lauritzen (1996) and Appendix A in Drton (2004).

In the application of mutational signatures in cancer the data consists of the number of mutations V_{mn} that cancer patient m has of mutation type n. We have $V_{mn} \approx \sum_{k=1}^{K} W_{mk} H_{kn}$, and the vector $(H_{k1}, \ldots, H_{kN})$ is signature k, $k = 1, \ldots, K$, and the vector $(W_{m1}, \ldots, W_{mK})$ are the weights (sometimes also called loadings) for each signature for cancer patient m, $m = 1, \ldots, M$.

The purpose of this paper is to develop and compare various optimization algorithms for both the NLS problem and non-negative matrix factorization. Algorithms for the NLS problem can be divided into two categories. The first category are iterative algorithms and include well-known algorithms such as the Majorize-Minimize algorithm (MM; see Lee and Seung (2000)), projected coordinate descent (PCD; see Lange et al. (2014)), projected gradient descent (PGD; see Lange et al. (2014)), and the Févotte-Cemgil Expectation-Maximization algorithm (FC-EM; see Févotte and Cemgil (2009)). In this paper we introduce a new family of iterative updates based on a generalization of the EM-algorithm in Févotte and Cemgil (2009). The PGD, PCD and FC-EM algorithms are special cases of this new EM family of iterative procedures for the NLS problem. However, and perhaps somewhat surprisingly, we show that the MM algorithm is *never* a member of our general EM algorithm.

The second category of algorithms for the NLS problem is based on convex analysis and in particular cone projection. We prove and describe in detail a cone projection algorithm tailored to the NLS problem. The algorithm is based on recent work by Meyer (2013), and is based on results from linear models. As described in Meyer (2013), other algorithms that exploit the edges of the cone in a similar fashion as cone projection are available. Most notable is perhaps the NLS algorithm of Lawson and Hanson (1974), which is based on the active set method and implemented in the R package 'NNLS' by Mollun and Stokkum (2015). From this second category of algorithms we have decided to focus on cone projection.

We compare and apply the various algorithms on a test problem and on a signature identification problem from cancer genomics. In our applications we demonstrate several important and intriguing points for optimization in complex statistical learning problems. We find that convergence issues are easier to detect when running different algorithms on the same problem, and we find that cone projection is an attractive choice. Furthermore, we find that the mix-and-match strategy advocated by Lange et al. (2014) is certainly advantageous.

The remaining part of the paper is organized as follows. In Section 2 we illustrate the iterative algorithms for the non-negative least square problem on a simple example. In particular we visualize the updates for the traditional algorithms and the updates for the family of EM algorithms. In Section 3 we describe in detail the traditional algorithms. In Section 4 we introduce the general family of EM algorithms, and relate the update rules for the traditional algorithms to the update rules for the EM family. Section 5 is concerned with convex analysis. We describe the non-negative least square problem in terms of cone projection, and provide and prove an algorithm for solving the problem. In Section 6 we compare the algorithms on a test problem and on mutation data from breast cancer patients. In Section 7 we provide recommendation and a discussion of the various algorithms and methods. Finally in Section 8 we provide a description of the accompanying supplementary material. The Supplementary Material contains computer code (written in R) and data sets for the reproduction of our algorithms and analyses.

2 Overview: The traditional algorithms and the EM family

Suppose we want to update column n of H for fixed W and where the remaining columns of H are also fixed. The term in (1) that we must minimize is then given by

$$\sum_{m=1}^{M} \big(V_{mn} - (WH)_{mn}\big)^2 = \|v - Wh\|^2. \tag{2}$$

Here $\|\cdot\|$ denotes the usual Euclidean norm and to simplify notation we write h for column n of H ($h_k = H_{kn}$) and v for column n of V ($v_m = V_{mn}$). The basic problem is therefore the non-negative least square (NLS) problem:

$$\text{minimize} \;\; f(h) = \|v - Wh\|^2 \;\; \text{subject to} \;\; h \geq 0. \tag{3}$$

We remark that if the unconstrained solution $(W'W)^{-1}W'v$ has non-negative entries, then the problem is solved. The distance (2) can be written as

$$\|v - Wh\|^2 = h'W'Wh - 2v'Wh + v'v = \frac{1}{2}h'Ah + b'h + c,$$

where $A = 2W'W$, $b = -2W'v$ and $c = v'v$. The solution to the basic problem is therefore identical to the solution of the problem:

$$\text{minimize} \;\; f(h) = \frac{1}{2}h'Ah + b'h \;\; \text{subject to} \;\; h \geq 0, \tag{4}$$

where A and b are given as above. In this formulation the NLS problem is often referred to as a non-negative quadratic programming problem.

Since

$$\|V' - H'W'\|_{\mathrm{F}} = \|V - WH\|_{\mathrm{F}},$$

update rules for the columns of H can easily be modified into update rules for the rows of W. Therefore, it is sufficient to derive update rules for the columns of H. Note also that the size of the problem does not change when M and N increase since the dimension of A and b only depends on K.

In Figure 1 we illustrate the iterations for the Majorize-Minimize (upper left plot), projected coordinate descent (upper right plot) and projected gradient descent (lower left plot) algorithms for a NLS problem with

$$W = \begin{bmatrix} 10 & 1 \\ 5 & 2 \end{bmatrix} \quad \text{and} \quad v = \begin{bmatrix} 1 \\ 8 \end{bmatrix}. \tag{5}$$

The initial value of h is $h^0 = (2, 2)'$ and the unconstrained minimum is at $(-0.4, 5)'$. These three traditional methods for solving the NLS problem are summarized in Section 3. In Section 4 we describe the new family of EM updates. The updating scheme in this family is illustrated in the lower right plot of Figure 1. Updates in the gray area are possible steps in the first iteration in the EM family. Projected coordinate descent is a member of the EM family, projected gradient descent with a small step size is a member of the EM family, and the update proposed in Févotte and Cemgil (2009) is a member of the family. Curiously, the Majorize-Minimize algorithm is *not* a member of the EM family.

3 Traditional algorithms for non-negative least square regression

In this section we describe and prove the Majorize-Minimize (MM), Projected Coordinate Descent (PCD) and Projected Gradient Descent (PGD) algorithms in detail. We focus on the non-negative least square problem in the formulation (4).

3.1 Majorize-Minimize (MM) algorithm

As described in Lange et al. (2014), the idea behind the majorize-minimize (MM) algorithm is that the objective function f in each iteration is *majorized* by a function g that depends on the current estimated minimizer h^t of f. Instead of directly minimizing f, the function g is *minimized.* The surrogate function g should satisfy that $f(h) \leq g(h \mid h^t)$ and $f(h) = g(h \mid h)$ for any non-negative vector $h \in \mathbb{R}^K$. Now, the update of h^t is defined as

$$h^{t+1} = \arg\min_{h \geq 0} g(h \mid h^t).$$

Then we have the sandwich inequality

$$f(h^{t+1}) \leq g(h^{t+1} \mid h^t) \leq g(h^t \mid h^t) = f(h^t), \tag{6}$$

so the update h^{t+1} decreases the value of f. The MM algorithm also comes in a minorize-maximize version, where the function g should satisfy $f(h) \geq g(h \mid h^t)$ and $f(h) = g(h|h)$. The update defined as

$$h^{t+1} = \arg\max_{h \geq 0} g(h \mid h^t)$$

then increases the value f. A prominent example of a minorize-maximize algorithm is the Expectation-Maximization algorithm that we consider in Section 4.

Theorem 1 (Lee and Seung (2000))**.** *The iterative procedure*

$$h_k^{t+1} = \frac{-b_k}{(Ah^t)_k} h_k^t, \quad k = 1, \ldots, K, \tag{7}$$

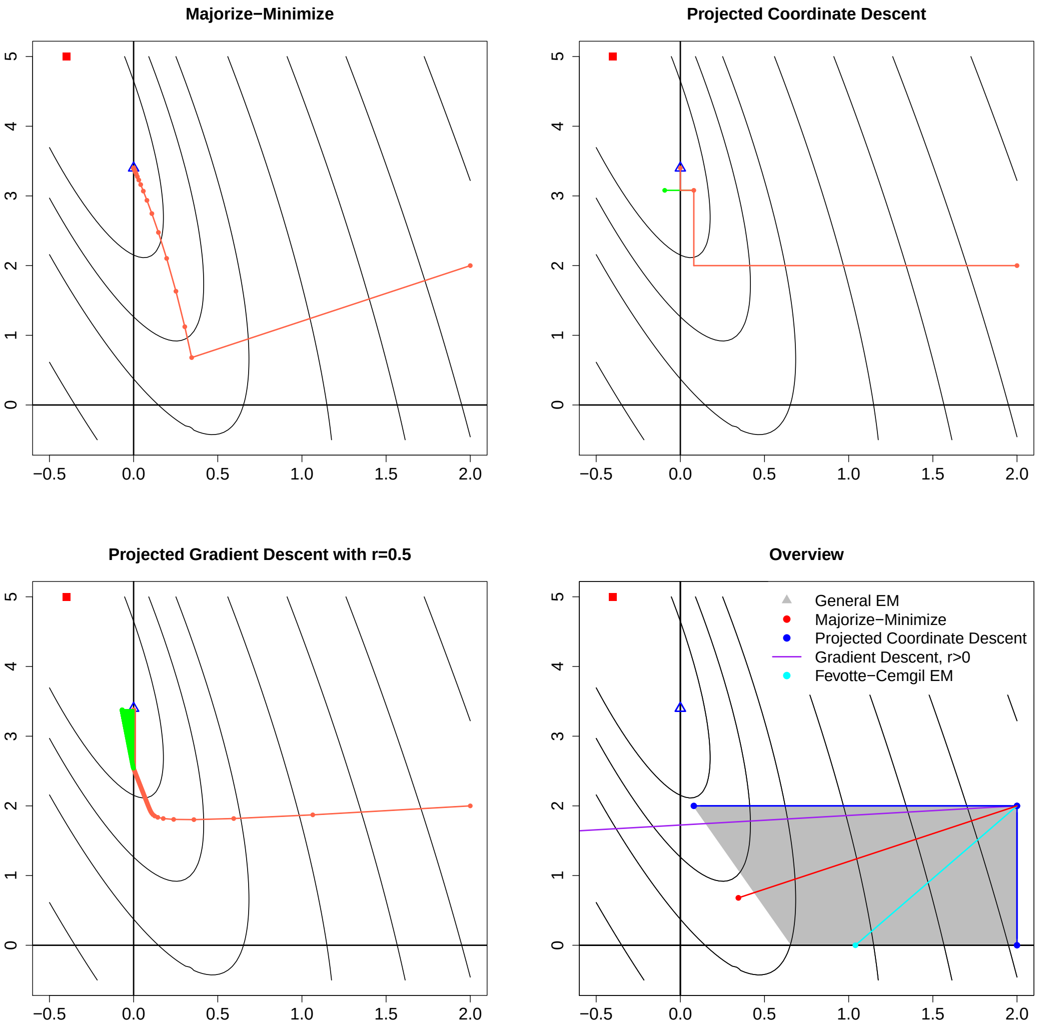

Figure 1: Visualization of the NLS problem (3) with parameters given by (5). The contour levels are for the quadratic form (4). The red square is the unconstrained minimum of the quadratic form, and the blue triangle is the constrained minimum. Upper left: Trajectory for the majorize-minimize algorithm. Upper right: Trajectory for the projected coordinate descent (the red points are the actual steps in the algorithm and the green line shows the projection part of a step). Lower left: Trajectory for the projected gradient descent with step size parameter $r = 0.5$. Lower right: Overview of the algorithms discussed in this paper and their relationship. The gray area are possible steps in the first iteration of the EM family.

converges to the solution of the NLS problem (3). We assume the initial vector h^0 has positive entries.

Proof. We follow Lange et al. (2014) and show that the iterative procedure (7) is an MM algorithm. We take advantage of the inequality

$$xy \leq \frac{1}{2}\Big(\alpha x^2 + \frac{1}{\alpha}y^2\Big), \tag{8}$$

which is easily derived from $0 \leq (\sqrt{\alpha}x - y/\sqrt{\alpha})^2$. Applying (8) with $x = h_k$, $y = h_l$ and $\alpha = h_l^t/h_k^t$ we get

$$h'Ah = \sum_{k=1}^{K}\sum_{l=1}^{K} A_{kl}h_kh_l \leq \sum_{k=1}^{K}\sum_{l=1}^{K} A_{kl}\frac{1}{2}\Big(\frac{h_l^t}{h_k^t}h_k^2 + \frac{h_k^t}{h_l^t}h_l^2\Big) = \sum_{k,l=1}^{K} A_{kl}\frac{h_l^t}{h_k^t}h_k^2, \tag{9}$$

where in the last equation we used that A is symmetric. Note that $(h_k^t, h_l^t) = (h_k, h_l)$ implies equality in (9). Now letting

$$g_0(h \mid h^t) = \sum_{k=1}^{K}\sum_{l=1}^{K} A_{kl}\frac{h_l^t}{h_k^t}h_k^2 \quad \text{and} \quad g(h \mid h^t) = \frac{1}{2}g_0(h \mid h^t) + b'h,$$

we have that $f(h) \leq g(h \mid h^t)$ and $g(h \mid h) = f(h)$ for all $h \geq 0$. Hence, it follows from (6) that $h^{t+1} = \arg\min g(h \mid h^t)$ is a valid update. Since

$$\frac{\partial g(h \mid h^t)}{\partial h_k} = b_k + \frac{h_k}{h_k^t}\sum_{l=1}^{K} A_{kl}h_l^t = b_k + \frac{h_k}{h_k^t}(Ah^t)_k,$$

the desired expression for h^{t+1} is obtained by setting the derivative equal to zero. □

We emphasize that a very advantageous feature of the MM update (7) is that the update is always non-negative because A and h^t has positive entries and b has negative entries.

Inserting the expressions for A and b in (7), the update rules for H and W given by Theorem 1 become

$$h_{kn}^{t+1} = h_{kn}^t \frac{((W^t)'V)_{kn}}{((W^t)'W^tH^t)_{kn}} \quad \text{and} \quad w_{mk}^{t+1} = w_{mk}^t \frac{(V(H^{t+1})')_{mk}}{(W^tH^{t+1}(H^{t+1})')_{mk}}.$$

These update rules are exactly the update rules given in Lee and Seung (2000, Equation (4)). Essentially, the proof of Theorem 1 is a simplified version of the proof of Lemma 2 in Lee and Seung (2000). Lee and Seung (2000) derive the update rules by minimizing the function

$$F(h) = \frac{1}{2}\|v - Wh\|^2 = \frac{1}{2}\big(f(h) + v'v\big)$$

using an MM algorithm. They use a majorizing function G that imitates the second-order Taylor expansion of F around h^t. The majorizing function is given as

$$G(h \mid h^t) = F(h^t) + (h - h^t)'\nabla F(h^t) + \frac{1}{2}(h - h^t)'K(h^t)(h - h^t),$$

where $K(h^t)$ is a $K \times K$ diagonal matrix with diagnonal elements $\{(W'Wh)_k/h_k\}_k$. Noting that $K(h^t)h^t = W'Wh^t$ and $h'K(h^t)h = 2g_0(h \mid h^t)$, basic calculations show the relation

$$G(h \mid h^t) = \frac{1}{2}\Big(g(h \mid h^t) + v'v\Big).$$

To ensure that G is a majorizing function of F, Lee and Seung (2000) show that the matrix $K(h^t) - W'W$ is positive semidefinite by showing that the scaled version $\{h_k(K(h^t) - W'W)h_l\}_{kl}$ is positive semidefinite. As seen from the method of proof in Theorem 1, this procedure can be greatly simplified.

3.2 Projected Coordinate Descent (PCD) algorithm

The projected coordinate descent algorithm is an iterative method for optimization problems. In each iteration, one coordinate is updated, while the other coordinates remain fixed. More precisely, we iterate according to

$$h_k^{t+1} = \arg\min_{h_k \geq 0} f(h_1^t, \ldots, h_k, \ldots, h_K^t) \quad \text{and} \quad h_l^{t+1} = h_l^t \text{ for } l \neq k.$$

Theorem 2. *Porjected coordinate descent is given by*

$$h_k^{t+1} = \lfloor h_k^t - \frac{1}{A_{kk}} \nabla f(h^t)_k \rfloor_+. \tag{10}$$

Proof. The gradient ∇f of the objective function f is given by

$$\nabla f(h) = Ah + b = Ah + b + \nabla f(h^t) - Ah^t - b = \nabla f(h^t) + A(h - h^t).$$

Setting the partial derivative with respect to the k-th coordinate h_k equal to zero implies that the unconstrained update for h_k becomes

$$h_k^{t+1} = h_k^t - \frac{1}{A_{kk}} \nabla f(h^t)_k.$$

By adding the non-negativity constraint, we obtain the update rule given by (10). □

Note that the projected coordinate descent algorithm ensures that the objective function is monotonically decreasing, i.e. $f(h^t) \leq f(h^{t+1})$.

3.3 Projected Gradient Descent (PGD) algorithm

The projected gradient method updates all coordinates at the same time and project the resulting value of h to the admissible parameter region. We iterate according to the updating rule

$$h^{t+1} = h^t - s\nabla f(h^t)$$

for some scalar $s > 0$. The updated value h^{t+1} of h is the projection of h^{t+1} to the permissible parameter region. The projection h^{t+1} is obtained by setting negative coordinates of h^{t+1} to 0. Note that h^{t+1} might not satisfy $f(h^{t+1}) < f(h^t)$, so in contrast to all other algorithms described in this paper, the projected gradient algorithm is not monotonically decreasing.

Lange et al. (2014) suggest the stepsize $s = r/L$, where $r \in (0, 2)$ and L is the largest eigenvalue of A. In the lower left plot in Figure 1 we show an example of PGD with $r = 0.5$, and in the left plot in Figure 2 we show an example of PGD with $r = 2$.

Allowing s to depend on the current value h^t of h, another option is to perform exact line search and use the stepsize s^t that gives the maximum decrease in the objective function value, i.e.

$$s^t = \underset{s>0}{\operatorname{argmin}}\, f(h^t - s\nabla f(h^t)).$$

Theorem 3. *The stepsize in exact line search is given by*

$$s^t = \frac{\|\nabla f(h^t)\|^2}{\nabla f(h^t)' A \nabla f(h^t)}. \tag{11}$$

Proof. First notice that

$$
\begin{aligned}
f(h^t - s\nabla f(h^t)) &= \frac{1}{2}(h^t - s\nabla f(h^t))'A(h^t - s\nabla f(h^t)) + b'(h^t - s\nabla f(h^t)) \\
&= \Big(\frac{1}{2}\nabla f(h^t)'A\nabla f(h^t)\Big)s^2 - \|\nabla f(h^t)\|^2 s + \frac{1}{2}h^t A h^t + b'h^t,
\end{aligned}
$$

so that $f(h^t + s\nabla f(h^t))$ is a second order polynomial in s. Since A is positive definite, the leading coefficient is positive, so the polynomial attains its minimum value at s^t given by (11). □

Since the stepsize of the exact line search (11) satisfies $s^t \leq 1/L$ the stepsize r/L suggested by Lange et al. (2014) is always smaller when $r \in (0, 1]$. Note that exact line search is not necessarily the optimal choice as it might yield large negative entries of h_0^{t+1} that in turn might cause the projection h^{t+1} to be in a completely different direction than $-\nabla f(h^t)$. In the right plot in Figure 2 we show an example of PGD with exact line search.

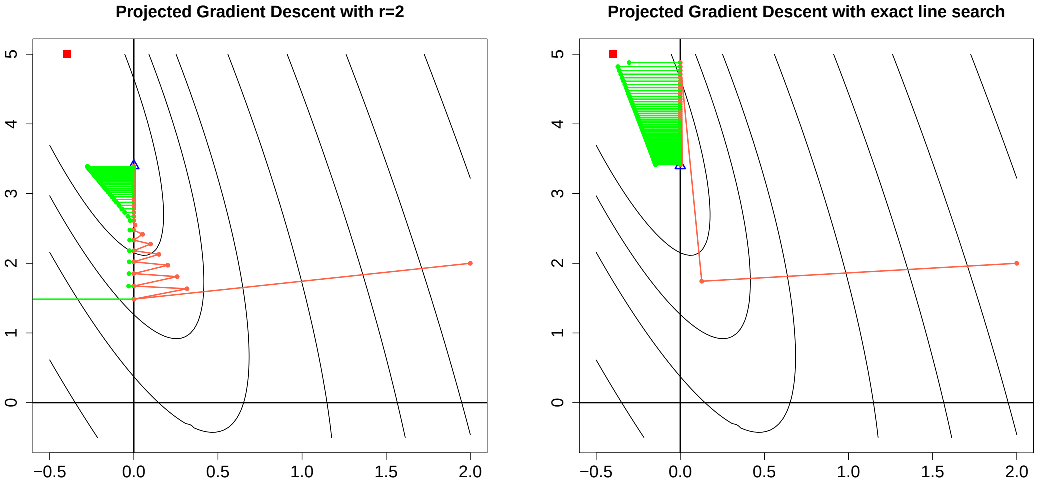

Figure 2: Left: Projected Gradient Descent (PGD) with step size $r = 2$. Right: Projected Gradient Descent (PGD) with exact line search.

4 The family of Expectation-Maximization algorithms

The NLS problem (3) can be interpreted in a probabilistic framework, and this identification naturally leads to the application of the EM algorithm (Dempster et al., 1977) to solve the minimization problem. The key idea is to view the non-negative least square problem as a missing data problem.

Recall the non-negative quadratic programming problem in the formulation (3). Consider the model of independent random variables

$$
y \sim N(0, \tau^2) \;\text{ and }\; x_{mk} \sim N\big(W_{mk}h_k, \sigma_k^2\big).
$$

for $m = 1, \ldots, M$ and $k = 1, \ldots, K$. The variables y and $X = \{x_{mk}\}$ are latent, and rather we observe the variables $v = \{v_m\}$ where

$$v_m = y + \sum_{k=1}^{K} x_{mk} \sim N\Big((Wh)_m, \tau^2 + \sigma^2\Big)$$

and $\sigma^2 = \sum_{k=1}^{K} \sigma_k^2$. Maximum likelihood inference for the parameter h in this Gaussian model is identical to solving (3) because the log-likelihood function

$$\ell(h; v) = \frac{M}{2} \log\Big(2\pi(\tau^2 + \sigma^2)\Big) - \frac{1}{2(\tau^2 + \sigma^2)} \sum_{m=1}^{M} \Big(v_m - (Wh)_m\Big)^2$$

is, up to an additive and multiplicative constant, identical to $-\|v - Wh\|^2$. Due to the interpretation as a missing data problem, the maximization of the log-likelihood can be carried out by means of the EM algorithm.

4.1 Derivation of update rules

The EM algorithm is a Minorize-Maximize algorithm (e.g. Hunter and Lange (2004)), where the minorizing function g is given by the expected log-likelihood for the full data X conditional on the current parameter h^t and the observed data v. Up to an additive constant, the log-likelihood for the full data X is given by

$$\ell(h; X) = -\sum_{m=1}^{M} \sum_{k=1}^{K} \frac{1}{\sigma_k^2} (x_{mk} - W_{mk} h_k)^2,$$

and it follows that the minorizing function g is given by

$$g(h \mid h^t) = \mathrm{E}_{h^t}\left[\, \ell(h; X) \mid v \,\right] = -\sum_{m=1}^{M} \sum_{k=1}^{K} \frac{1}{\sigma_k^2} \mathrm{E}_{h^t}\Big[\, (x_{mk} - W_{mk} h_k)^2 \mid v_m \,\Big].$$

The distribution of (x_{mk}, v_m) at the parameter h^t is given by

$$\begin{pmatrix} x_{mk} \\ v_m \end{pmatrix} \sim N_2\left(\begin{pmatrix} W_{mk} h_k^t \\ (Wh^t)_m \end{pmatrix}, \begin{pmatrix} \sigma_k^2 & \sigma_k^2 \\ \sigma_k^2 & \tau^2 + \sigma^2 \end{pmatrix} \right),$$

and we therefore have that

$$x_{mk} \mid v_m \sim N\Big(W_{mk} h_k^t + \frac{\sigma_k^2}{\tau^2 + \sigma^2}\big(v_m - (Wh^t)_m\big), \sigma_k^2 - \frac{\sigma_k^4}{\tau^2 + \sigma^2}\Big).$$

We then obtain, up to an additive constant, that

$$g(h \mid h^t) = -\sum_{m=1}^{M} \sum_{k=1}^{K} \frac{1}{\sigma_k^2} \Big(W_{mk} h_k^t - W_{mk} h_k + \frac{\sigma_k^2}{\tau^2 + \sigma^2}\big(v_m - (Wh^t)_m\big)\Big)^2.$$

This function has to be maximized simultaneously in all K variables $(h_1, \ldots, h_k)$, but fortunately the function is completely additively separable, and we can thus treat each variable independently of all the others. Differentiating with respect to h_k gives

$$\frac{\partial g(h \mid h^t)}{\partial h_k} = \frac{2}{\sigma_k^2}\sum_{m=1}^{M}\Big(W_{mk}h_k^t - W_{mk}h_k + \frac{\sigma_k^2}{\tau^2+\sigma^2}\big(v_m - (Wh^t)_m\big)\Big)W_{mk},$$

and setting equal to zero, we obtain the update rule

$$h_k^{t+1} = h_k^t + \frac{\sigma_k^2}{\tau^2+\sigma^2}\frac{(W'v - W'Wh^t)_k}{(W'W)_{kk}} = h_k^t - \frac{\sigma_k^2}{\tau^2+\sigma^2}\frac{(\nabla f(h^t))_k}{A_{kk}}, \quad k = 1,\dots,K. \tag{12}$$

The function $g(h \mid h^t)$ is a second-order polynomial in each variable, and the leading coefficients are always negative. Therefore the non-negativity constraint can be incorporated by setting negative coordinates equal to zero. This update ensures that the data log-likelihood is increasing and that the updates are always non-negative. Allowing the variances to vary between iterations gives generalized update rules that preserves the monotone decrease of the objective function.

4.2 The Fevotte-Cemgil EM update rules

Févotte and Cemgil (2009) make the choice $\sigma_k^2 = 1/K$ (implying $\sigma^2 = 1$) and $\tau^2 = 0$. Then the update rule (12) becomes

$$h_k^{t+1} = h_k^t - \frac{1}{K}\frac{(\nabla f(h^t))_k}{A_{kk}} = h_k^t + \frac{1}{K}\frac{(W'v - W'Wh^t)_k}{(W'W)_{kk}}. \tag{13}$$

The non-negativity constraint is incorporated by setting negative coordinates equal to zero. In the left plot in Figure 3 we show the Fevotte-Cemgil update rules for the example from Section 2.

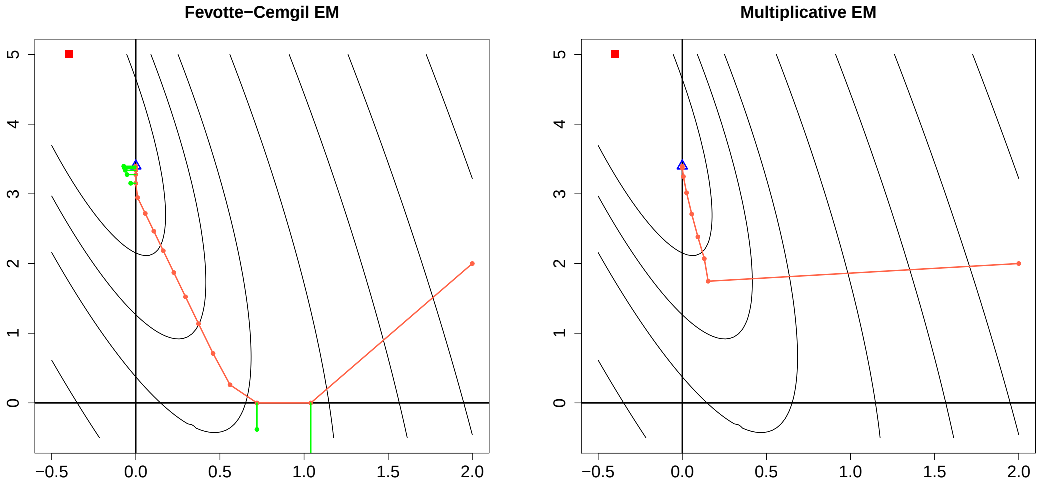

Figure 3: Left: Fevotte-Cemgil EM algorithm. Right: Multiplicative EM algorithm.

4.3 Projected gradient descent is a member of the EM family

Letting $\sigma_k^2 = A_{kk}$ for $k = 1,\dots,K$, we obtain the update rule

$$h_k^{t+1} = h_k^t - \frac{1}{\tau^2 + \operatorname{Tr} A}(\nabla f(h^t))_k,$$

where $\operatorname{Tr} A = \sum_{k=1}^{K} A_{kk}$ is the trace of A.

The projected gradient algorithm is an EM algorithm when the step-size s satisfies $s \in (0, 1/\operatorname{Tr} A]$. Since $1/(\tau^2 + \operatorname{Tr} A) \leq 1/L$, where L is the largest eigenvalue of A, the specific choices of step-sizes $s = r/L$ described in Section 3.3 does not give rise to EM algorithms when $r \in (1, 2)$.

4.4 Projected coordinate descent is a member of the EM family

We now consider an iteration scheme, where the variances τ^2 and $\sigma_k^2, k = 1, \ldots, K$, depend on the iteration. For a given iteration t where we want to update h_k, we let $\sigma_k^2 = 1$ and $\sigma_l^2 = 0$ for $l \neq k$. This yields the update rule

$$h_k^{t+1} = h_k^t - \frac{1}{\tau^2 + 1} \frac{(\nabla f(h^t))_k}{A_{kk}} \qquad \text{and} \qquad h_l^{t+1} = h_l^t \quad \text{for} \quad l \neq k.$$

Now we use τ^2 to ensure a non-negative update by defining

$$\tau^2 = \begin{cases} 0 & \text{if } h_k^t - \frac{(\nabla f(h^t))_k}{A_{kk}} \geq 0 \\ \frac{(\nabla f(h^t))_k}{A_{kk}} \frac{1}{h_k^t} - 1 & \text{if } h_k^t - \frac{(\nabla f(h^t))_k}{A_{kk}} < 0 \end{cases} \tag{14}$$

Note that $\tau^2 \geq 0$. The resulting update rule is exactly the coordinate descent updates (10).

4.5 Multiplicative EM updates and relation to the MM algorithm

The Majorize-Minimize algorithm described in Section 3.1 yields multiplicative update rules. From the EM update (12), we obtain a multiplicative rule by choosing σ_k proportional to h_k^t. In particular we get a mathematically appealing update rule by choosing

$$\sigma_k^2 = h_k^t A_{kk},$$

so that the update rule becomes

$$h_k^{t+1} = h_k^t \left(1 - \frac{(\nabla f(h^t))_k}{\tau^2 + \sigma^2}\right), \quad k = 1, \ldots, K, \tag{15}$$

where

$$\sigma^2 = \sum_{k=1}^{K} h_k^t A_{kk}.$$

We can choose τ^2 such that the update is always non-negative. One option is to define

$$\tau^2 = \begin{cases} 0 & \text{if } \ \max_k (\nabla f(h^t))_k \leq \sigma^2 \\ \max_k (\nabla f(h^t))_k - \sigma^2 & \text{if } \ \max_k (\nabla f(h^t))_k > \sigma^2. \end{cases} \tag{16}$$

In the right plot in Figure 3 we show this choice of multiplicative update rule.

Now consider the choice

$$\frac{\sigma_k^2}{\tau^2 + \sigma^2} = h_k^t \frac{A_{kk}}{(Ah^t)_k}. \tag{17}$$

Then (12) becomes

$$h_k^{t+1} = h_k^t - h_k^t \frac{(b + Ah^t)_k}{(Ah^t)_k} = h_k^t \frac{-b_k}{(Ah^t)_k},$$

and this is exactly the update rule (7) from (Lee and Seung, 2000, Equation (4)). However, and perhaps somewhat surprising, the choice (17) is not possible in the EM framework. This statement follows from an application of Cauchy-Schwarz inequality since

$$\sum_{k=1}^{K} h_k \frac{A_{kk}}{(Ah)_k} = \sum_{k=1}^{K} h_k \frac{\|W_k\|^2}{\sum_{l=1}^{K} h_l W_l' W_k} \geq \sum_{k=1}^{K} h_k \frac{\|W_k\|}{\sum_{l=1}^{K} h_l \|W_l\|} = 1,$$

where W_k denotes the kth column of W. This contradicts the fact that $\sum_{k=1}^{K} \sigma_k^2/(\tau^2 + \sigma^2) < 1$. In conclusion, the MM update rule is *not* a valid choice in the EM framework.

5 Cone Projection for non-negative least square regression

Recall that our basic problem is to

$$\text{minimize } ||v - Wh||^2 \text{ subject to } h_k \geq 0, \tag{18}$$

where $h = (h_1, \ldots, h_K)^T$ is a K-dimensional column vector ($K \times 1$ matrix), $v = (v_1, \ldots, v_M)^T$ is a M-dimensional column vector, and W is a $M \times K$ matrix with non-negative entries. Furthermore $\text{rank}(W) = K \leq M$.

The space

$$\mathcal{C} = \text{cone}(W) = \{Wh \in \mathbb{R}^M : h_j \geq 0\} = \Big\{ \sum_{j=1}^{K} h_j W_j \in \mathbb{R}^n : h_j \geq 0 \Big\} \tag{19}$$

consists of all linear combinations with non-negative coefficients of the columns W_j of W (sometimes called a conical combination). The origin belongs to the cone, the cone has edges $\mathcal{F}_j = \{h_j W_j \in \mathbb{R}^n : h_j > 0\}$ determined by W_j, and more general the faces of the cone are given by

$$\mathcal{F}_J = \Big\{ \sum_{j \in J} h_j W_j \in \mathbb{R}^n : h_j > 0 \Big\} \tag{20}$$

where $J \subseteq \{1, \ldots, K\}$. The interior of the cone has $J = \{1, \ldots, K\}$ and we define $\mathcal{F}_\emptyset$ to be the origin.

The basic problem (18) can be viewed as the problem of projecting v onto the cone $\mathcal{C}$. If the coefficient vector $(W'W)^{-1}W'v$ has positive entries, then v is in the interior of the cone, and the distance to the cone is zero. Otherwise the solution is on one of the faces of the cone. A brute force solution of the problem consists of an exhaustive search of all possible projections to the subspaces W_J generated by the columns of W that belong to the set J, investigate if the projection is on a face of the cone (i.e. if all the cofficients $(W_J'W_J)^{-1}W_J'y$ are non-negative), and then choosing the cone projection with the smallest distance.

Example 1. Consider again the situation from Section 2 where

$$W = \begin{bmatrix} 10 & 1 \\ 5 & 2 \end{bmatrix}. \tag{21}$$

Then the edges of the cone are generated by $W_1 = (10, 5)'$ and $W_2 = (1, 2)'$. In this example we only have four faces as illustrated in the left panel in Figure 4. The projections to the cone of the possible data points $v = (v_1, v_2)' \in \mathbb{R}^2$ are also indicated in the left panel in Figure 4.

With $v = (1, 8)'$ the projections on the edges of the cone are illustrated in the right panel in Figure 4. In Table 1 we show the coefficients for the projections on the subspaces generated by the columns of W. Non-negative coefficients are not allowed for conical projections, and therefore the first set of coefficients (corresponding to $J = \{1, 2\}$) must be discarded. The remaining coefficients are non-negative, and the smallest distance is obtained for the face with $J = \{2\}$. We conclude that the solution to the non-negative least square problem is $h = (0, 3.4)'$.

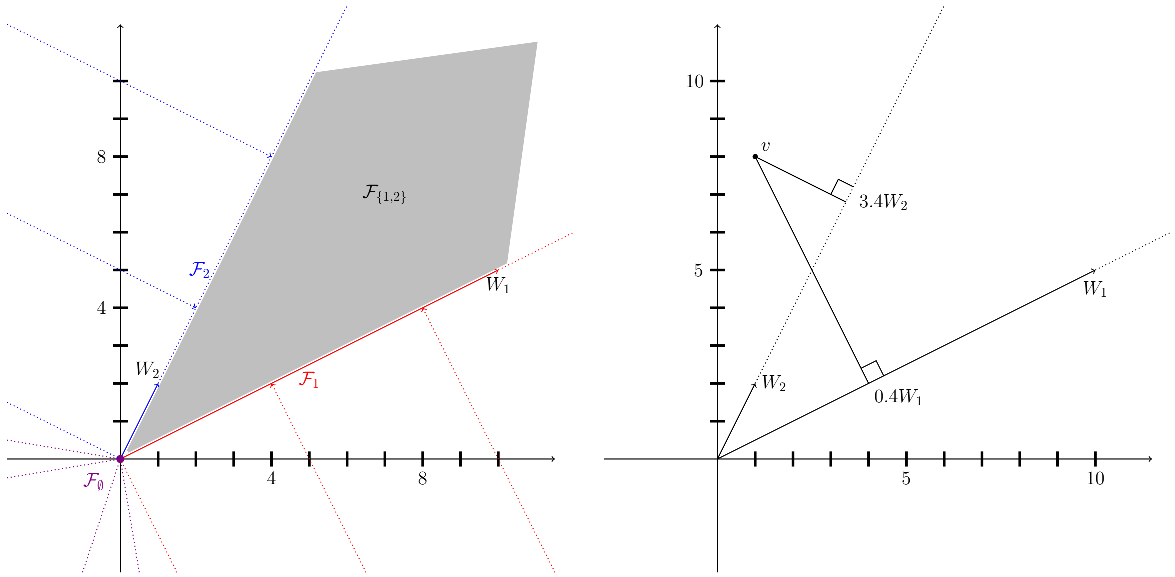

Figure 4: Left: The four faces and sketched projections to the cone for Example 1. Right: The projections to the two edges determined by W_1 and W_2 for the data point $v = (1, 8)'$.

Subset $J \subseteq \{1, 2\}$	Generator $X = W_J$	Coefficient $a = (X'X)^{-1}X'v$	Non-negative coefficients?	Squared distance $\|\|v - Xa\|\|^2$
$J = \{1, 2\}$	W	$(-0.4, 5)$	No	0.0
$J = \{1\}$	W_1	0.4	Yes	45.0
$J = \{2\}$	W_2	3.4	Yes	7.2
$J = \emptyset$	$(0, 0)^T$	0	Yes	65.0

Table 1: Coefficients for projections on all sub-spaces generated by W. Only non-negative coefficients (conical combinations) correspond to projections on the faces of the cone.

In general the number of faces is 2^K and K is most often of size 10 or more, so an exhaustive search would require at least $2^{10} = 1024$ projections. Meyer (2013) describe an efficient algorithm for a more general cone projection problem where the constraints on h are more complicated. We now present and prove a modification of Meyer's algorithm to our basic problem (18).

Algorithm 1. Cone projection for non-negative least square regression

Initial Step: Set $a = W'v$. If $\max_i a_i \leq 0$ the solution is $h = 0$. Otherwise let $J_0 = \{\arg\max_i a_i\}$.

Step 1: Set $a = (W'_{J_0} W_{J_0})^{-1} W'_{J_0} v$, $P_{J_0} = W_{J_0}(W'_{J_0} W_{J_0})^{-1} W'_{J_0}$ and $c = W'(I - P_{J_0})v$. If $\max_i c_i \leq 0$, the solution is $h = a$. Otherwise let $j = \arg\max_i c_i$, $J_1 = J_0 \cup \{j\}$ and $h_1 \in \mathbb{R}^{|J_1|}$ with $h_1 = a$ on J_0

and $h_{1j} = 0$. Go to *Step 2*.
Step 2: Let $d = (W'_{J_1} W_{J_1})^{-1} W'_{J_1} v$. If $\min_i d_i < 0$ define

$$\hat{x} = \min_{\{j : d_j < 0\}} \Big(\frac{h_{1j}}{h_{1j} - d_j} \Big), \tag{22}$$

and let $\hat{h}_1 = \hat{x} d + (1 - \hat{x}) h_1$. Define a new J_1 as $J_1 \setminus \{j : \hat{h}_{1j} = 0\}$ and a new h_1 as $\hat{h}_1$. Return to *Step 2*. Otherwise (that is, $\min_i d_i \geq 0$), define a new J_0 as $\{j \in J_1 : d_j > 0\}$ and return to *Step 1*.

A flow chart for the algorithm is provided in the Appendix, and an implementation in the programming language R is available in the Supplementary Material.

Proof of convergence. The proof of convergence for the algorithm is similar in spirit to the proof in Meyer (2013). The algorithm searches among the faces determined by the sets J_0 for which the coefficient vector $a = (W'_{J_0} W_{J_0})^{-1} W_{J_0} v$ of the projection on W_{J_0} has strictly positive entries. The final solution has been found when $W'_j (I - P_{J_0}) v \leq 0$ for $j \notin J_0$. The reason is that if $W'_j (I - P_{J_0}) v \leq 0$ then the angle between W_j and $(I - P_{J_0}) v$ is between $\pi/2$ and $3\pi/2$, and therefore adding the edge j to the face J_0 will never decrease the distance to the data point v.

Let $\text{SSD}(J_0) = ||v - W_{J_0} a||^2$ be the minimum quadratic distance when considering the subset J_0. It is clear, by construction, that on each entry of Step 1 we have a new set J_0 satisfying that a has strictly positive entries. The proof consists in showing that $\text{SSD}(J_0)$ is strictly decreasing from one entry of Step 1 to the next. Since there is a finite number of different sets J_0 the algorithm is bound to stop.

For illustration we return to Example 1. In the left panel in Figure 5 we illustrate the two cases where $J_0 = \{1\}$ or $J_0 = \{2\}$ when entering Step 1. In the case $J_0 = \{2\}$ we have found the solution, and indeed the vector product between $(I - P_{J_0}) v = (I - P_2) v$ and W_1 is negative. In the case $J_0 = \{1\}$ we find that the vector product between $(I - P_{J_0}) v = (I - P_1) v$ and W_2 is positive, and we need to update J_0.

In the initial step of Example 1 we get $a = (50, 17)$, and we therefore have $J_0 = \{1\}$ when entering Step 1. In Step 1 we get $a = 0.4$ and $c = (0, 9)$, and we thus get $j = 2$, $J_1 = \{1, 2\}$ and $h_1 = (0.4, 0)$. Furthermore consider the path of coefficients

$$\tilde{h}(x) = x d + (1 - x) h_1, \ \ 0 \leq x \leq 1, \tag{23}$$

where $d = (-0.4, 5)$ is the coefficient vector of the projection on W_{J_1}. Let $\text{ssd}(x) = ||v - W_{J_1} \tilde{h}(x)||^2$ be the quadratic distance from v to the to the point with coefficients $\tilde{h}(x)$ (see the right panel in Figure 5). For $x = 0$ we get $\tilde{h}(0) = h_1$ and for $x = 1$ we get $\tilde{h} = d$. The minimum of $\text{ssd}(x)$ is at $x = 1$, the maximum is at $x = 0$, and the function is strictly increasing. A valid solution must have non-negative coefficients, so we require $\tilde{h}_i(x) \geq 0$ for all i. For $i = 2$ this is clearly fulfilled, but for $i = 1$ this means that $x d_1 + (1 - x) h_{11} \geq 0$, or $x < \hat{x} = h_{11}/(h_{11} - d_1) = 0.4/(0.4 + 0.4) = 0.5$. The coefficient $\tilde{h}(\hat{x})$ lies on the edge determined by W_2, so now the face with the smallest distance has been identified.

Now consider the general situation. Let a, c and j be defined as in Step 1 and d as in Step 2 when going from Step 1 to Step 2. Let us write $W_j = W_{J_0} \epsilon + W^{\perp}$, where $W_{J_0} \epsilon$ is the projection of W_j on W_{J_0} and $W^{\perp}$ is the projection of W_j on the complementary subspace. (In the right panel in Figure 5 this construction is illustrated in the case $j = 2$ and $J_0 = 1$.) Define $z = (W^{\perp})' v / ||W^{\perp}||^2$. Then $d_{J_0} = a - \epsilon z$ and $d_j = z$. Furthermore, $0 < c_j = W'_j (I - P_{J_0}) v = (W^{\perp})' v = z ||W^{\perp}||^2$, so that $z = d_j > 0$. By construction a has strictly positive entries and so h_{1J_0} has strictly positive entries.

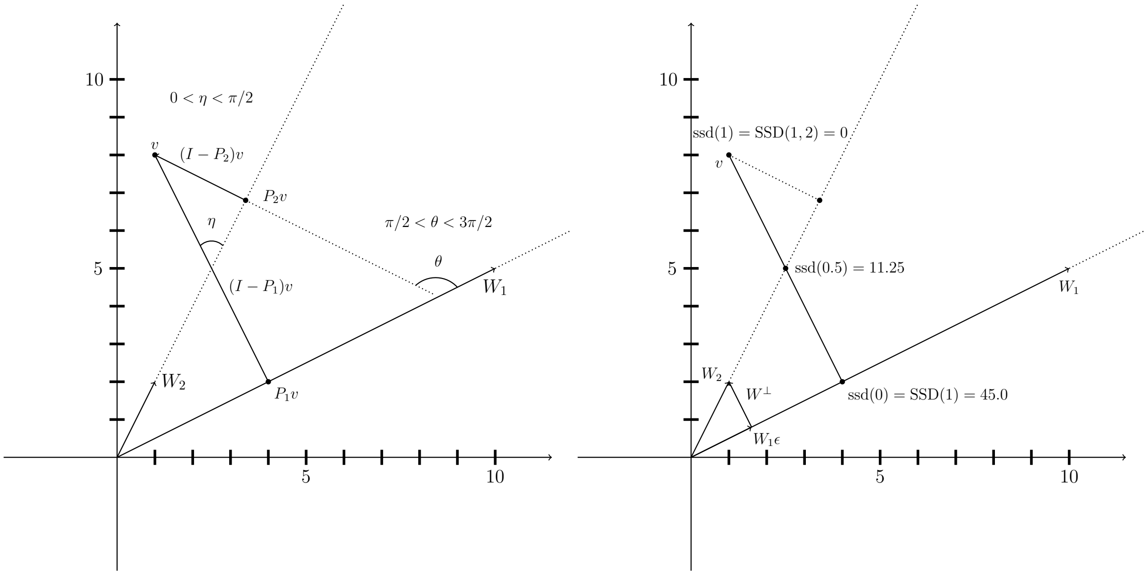

Figure 5: Illustration of the cone projection algorithm. Left panel: For $J = \{2\}$ we have $W_1'(I-P_2)v = |W_1|\,|(I-P_2)v|\cos\theta < 0$ and the algorithm terminates. For $J = \{1\}$ we have $W_2'(I-P_1)v = |W_2|\,|(I-P_1)v|\cos\eta > 0$ and the algorithm continues. Right panel: Minimizing ssd along a path from P_{J_0} to P_{J_1} corresponding to Step 2 of the algorithm.

Combining this with $d_j > 0$ we see that $\hat{x} > 0$ in the first run of Step 2. Consider the path of coefficients $\tilde{h}(x) = xd + (1-x)h_1$, $0 \le x \le 1$. The minimum of $\text{ssd}(x) = ||v - W_{J_1}\tilde{h}||^2$ is at $x = 1$ and the function is strictly decreasing. Thus, $\text{ssd}(\hat{x}) < \text{ssd}(0) = \text{SSD}(J_0)$.

In the subsequent passes of Step 2 we again consider a linear path from the present position towards a minimum so that the quadratic norm decreases even further. When leaving Step 2 we have therefore arrived at a new subset J_0 with a smaller value of $\text{SSD}(J_0)$. □

The present algorithm differs from that in Meyer (2013) in the possibility of returning to Step 2 within Step 2. With this modification the proof of convergence becomes easier and different from the proof in Meyer (2013) who refers to an unpublished note for this part of the proof.

6 Applications

In this section we compare selected algorithms from the three previous sections on a test problem and on mutation data from cancer genomics.

6.1 Test problem of Lange, Chi and Zhou (2014)

We applied our algorithms to a test problem similar to Lange et al. (2014). We simulated the $M \times K$ entries in W from independent and identically distributed (iid) exponential distributions with rate one. We let $M = 100$ and $K = 50$. We simulated the K entries in h_0 from iid standard uniform distributions, and we simulated a noise vector e of length M from iid standard normal distributions. The data vector v is given by $v = Wh_0 + e$, and the task is to estimate h as the minimizer of the residual sum of squares (RSS)

$$\text{RSS}(h) = ||v - Wh||^2,$$

where the entries in h must be non-negative.

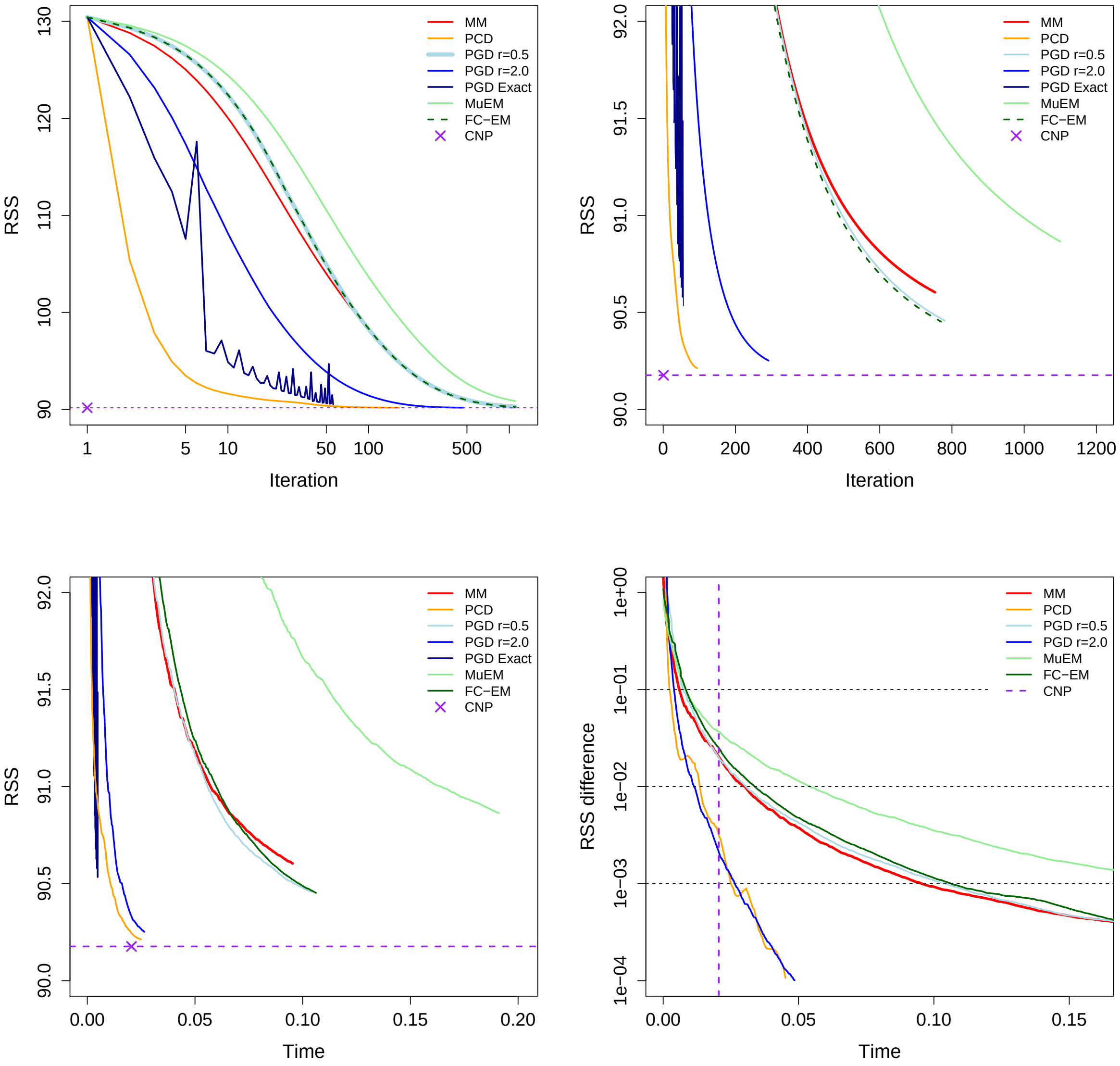

Figure 6: Application of our algorithms to a test problem similar to Lange et al. (2014). Upper left: RSS as a function of the number of iterations for seven iterative algorithms. The RSS decreases with the number of iterations for all algorithms except for PGD-Exact. We also include the minimum obtained from cone projection. Upper right: Same as the previous plot, but zooming in on the RSS value of convergence. The algorithms are stopped when the RSS difference between two iterations is smaller than a threshold of 0.001. Bottom left: It is more appropriate to use time (seconds) instead of iterations on the x-axis because the time for each iteration varies between the algorithms. Bottom right: Cone projection takes 0.020 seconds and is faster than all other algorithms if the threshold is 0.001. If the threshold 0.01 is satisfactory, then PCD or PGD with $r = 2$ are faster than CNP. All the iterative algorithms are faster than CNP is the threshold is 0.1.

The upper left plot in Figure 6 shows $\text{RSS}(h_t)$ as a function of the number of iterations t (in log-scale) for the seven iterative algorithms and cone projection (CNP). The initial point is h_0. The iterative procedure is stopped when the difference $\text{RSS}(h_{t+1}) - \text{RSS}(h_t)$ in the residual sum of squares between two iterations is below a threshold of 0.001. The seven iterative algorithms are the MM, PCD, PGD with $r = 0.5$, PGD with $r = 2.0$, PGD with exact line search, multiplicative EM (MuEM) and Fevotte-Cemgil EM (FC-EM). The number of iterations required for convergence varies with an order of magnitude between the algorithms. The Projected Coordinate Descent (PCD) and the Projected Gradient Descent algorithms with large step sizes (PGD-Exact and PGD with $r = 2.0$) converge in 100-300 iterations whereas the MM algorithm, the PGD with a small step size ($r = 0.5$), and the two EM algorithms need 700-1200 iterations. We also note that PGD Exact has large fluctuations in the objective function while all other algorithms have monotonically decreasing RSS values.

The upper right plot in Figure 6 shows that the algorithms that require many iterations before convergence also tend to stop the iterative process rather early. In the plot we have zoomed in on the final value of the residual sum of squares on the y-axis. The four algorithms that require many iterations (MM, PGD with $r = 0.5$, MuEM and FC-EM) and the PGD-Exact stops early compared to PCD and PGD with $r = 2.0$. The two latter algorithms have the best convergence properties with our choice of stopping criteria.

In the lower left plot in Figure 6 we have substituted the number of iterations with the actual time spent by the algorithm before convergence. The updating scheme for the MM-algorithm is faster to compute for each iteration, and this property can be seen in the plot; the MM algorithm is now below FC-EM and PGD with $r = 0.5$ in the beginning of the iterative process. We also see that PGD with $r = 2.0$ requires approximately as much time as PCD (0.029 seconds versus 0.022 seconds) even though many more iterations are required (292 iterations for PGD versus 94 iterations for PCD).

We have also included cone projection (CNP) in the plots. In this problem CNP takes 0.020 seconds. In the lower right plot in Figure 6 we show the time and corresponding difference in RSS. The CNP algorithm uses both the least amount of time and of course has the smallest RSS-value when the threshold is 0.001. The RSS-value for the six other algorithms would decrease with a lower threshold or with more iterations, but that would result in a longer running time. However, with a threshold of 0.01 PCD and PGD with $r = 2.0$ are faster. We conclude that for this test problem the best choice of algorithms are cone projection, PCD or PGD with $r = 2.0$.

6.2 Mutational signatures in cancer genomes

In a popular framework developed by Alexandrov et al. (2013), NMF is used for deciphering mutational signatures in human cancer from mutation counts. In this application, we investigate the performances of the algorithms by analysing signatures in the original data set of $M = 21$ breast cancer genomes (BRCA21 data) provided by Nik-Zainal et al. (2012).

Assuming strand-symmetry, the six possible base substitutions are C>A, C>G, C>T, T>A, T>C and T>G. We furthermore take the immediate flanking left and right nucleotides into account. The number of mutation types are therefore $N = 6 \cdot 4 \cdot 4 = 96$, and the data matrix V has dimension 96×21. Entry V_{mn} is the number of mutations of type m in genome n. Recall that in NMF we have $V \approx WH$. Alexandrov et al. (2013) find that the data should be analysed with 4 signatures, so we let $K = 4$. The 4 columns of the 96×4 $(N \times K)$ matrix W contains the signatures, and the 21 columns of the 4×21 $(K \times M)$ matrix H contains the loadings for each patient.

We use the alternating NLS algorithm to estimate the signatures and the loadings. In the alternating NLS algorithm we apply in sequential order an iterative algorithm for updating the 96 rows of H and

21 columns of W. The algorithm is stopped when the RSS difference

$$\triangle\mathrm{RSS}^{(t)} = \mathrm{RSS}^{(t-1)} - \mathrm{RSS}^{(t)} = \|V - W^{(t-1)}H^{(t-1}\|_{\mathrm{F}}^2 - \|V - W^{(t)}H^{(t)})\|_{\mathrm{F}}^2 \tag{24}$$

between two full updates of columns of H and rows of W is below 0.1. We apply the optimization algorithm on 100 different initial starting points. The values of the starting points for W and H are sampled uniformly at random.

We need to choose the number of iterations or a tolerance threshold for the iterative algorithms for each of the $96 + 21 = 117$ NLS problems. In the top panel in Figure 7 we show box plots of the final 100 RSS values when using 1, 10 or 100 iterations within each NLS problem. The left panel shows the box plots of the RSS values for the MM algorithm, and the right panel shows the box plots of the RSS values for the MuEM algorithm. For both algorithms, the traditional choice (one iteration per update) converges so slowly that the iterative updates stop before convergence. However, most of the 100 samples have converged for the MM algorithm with 10 iterations, while 100 iterations are needed for the MuEM algorithm. We conducted similar experiments for all the other iterative algorithms. We found that one iteration per update is sufficient for PCD and FC-EM, 10 iterations per update are needed for MM and PGD, and MuEM requires 100 iterations for each NLS problem.

Recall that if the unconstrained solution to the NLS problem has non-negative coefficients, then this solution applies. We therefore introduced a mix-and-match strategy where we before applying any of the iterative algorithms to the NLS problem checked if the unconstrained solution was valid. After a few iterations the unconstrained solution applies to roughly 100 of the 117 updates (see the left plot in the middle panel in Figure 7). For the trajectories that converge to a local minimum different from the global minimum, we find that the unconstrained solution applies much less often. Furthermore, adding this simple extension to the algorithm results in much faster convergence (see the right plot in the middle panel in Figure 7).

The advantage of applying the unconstrained solution when it is valid also applies to all the other algorithms. In the bottom panel in Figure 7 we show a comparison of all the eight algorithms MM, PCD, PGD with $r = 0.5$, PGD with $r = 2$, PGD Exact, MuEM, FC-EM and CNP. We find that CNP and PCD are the most efficient algorithms for the BRCA21 data.

The equation

$$WH = W\mathrm{diag}(s_1, \dots, s_K)\mathrm{diag}(1/s_1, \dots, 1/s_K)H = \tilde{W}\tilde{H}$$

shows that we can scale the signatures (rows of H) if we scale the loadings accordingly. It is natural to normalize the signatures such that the entries sum to one, i.e. let $s_k = \sum_{n=1}^{N} H_{kn}$. In the top plot of Figure 8 we show the normalized signatures. The first signature shows a global mutation pattern with moderate probabilities for all the mutation types. The other three signatures are much more focused on specific mutation types: Signature 2 has more C to G mutations at TCT, TCA and TCC sites, Signature 3 has more C to T mutations, and Signature 4 has more C to G mutations at TCA and TCT sites, and C to T mutation at TCA and TCT sites. The mutations at CpT sites are most likely related to the APOBEC signatures in breast cancer; see Nik-Zainal and Morganella (2017) for a recent discussion.

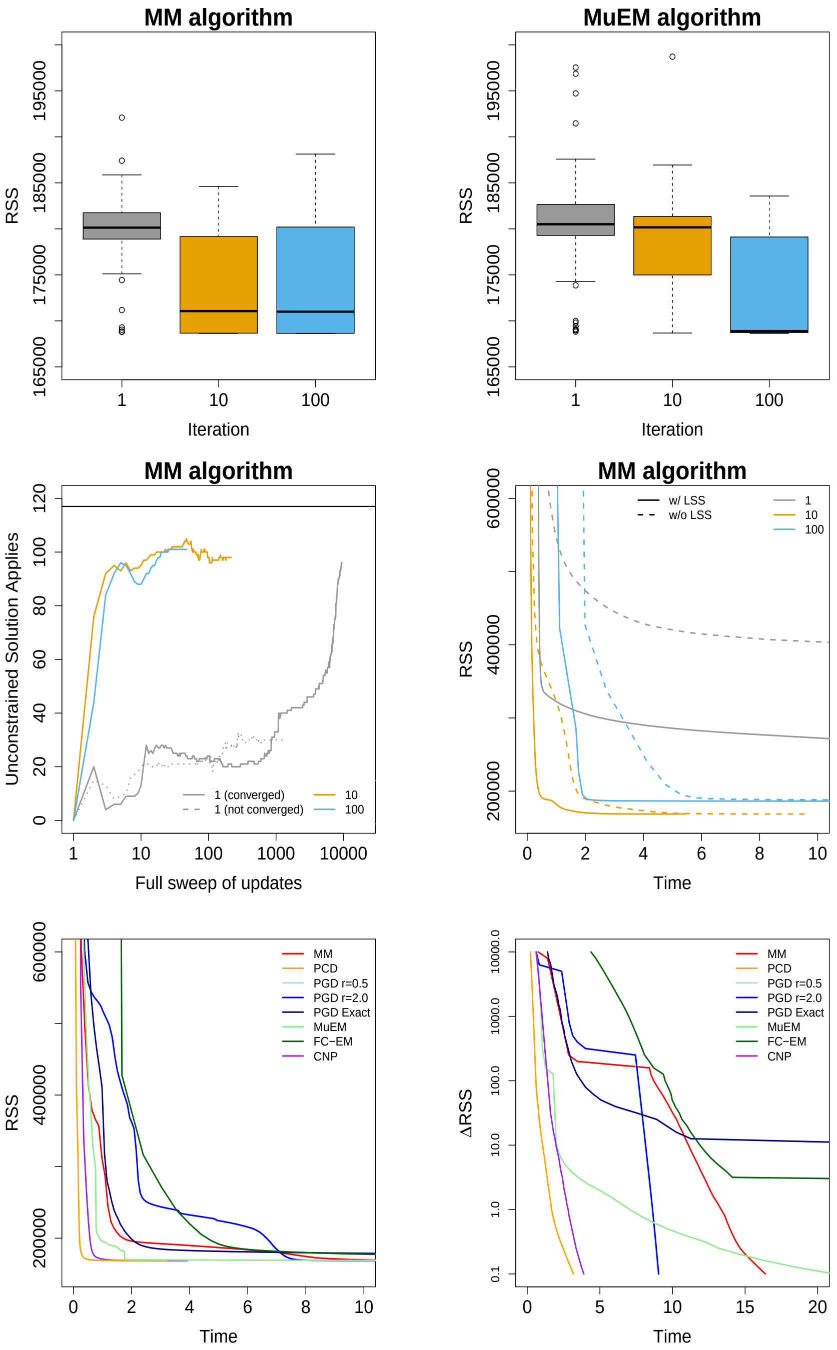

Figure 7: Convergence for the BRCA21 data. Top left: Box plots of the final RSS for 100 initial settings for the MM algorithm. Topr right: Box plots for the MuEM algorithm. Middle left: The number of times the unconstrained least square solutions (LSS) applies for the MM algorithm. After the first few iterations, in around 100 of the 117 updates (21 for the rows of W and 96 for the columns of H) the unconstrained LSS solution is valid for the NLS problem. Middle right: The convergence is faster when the mix-and-match (LSS) is applied. Right: Trajectories that converge to the global minimum tend to use more LSS updates (around 100 out of the 117 updates) than the trajectories that converge to a local minimum. Bottom left: RSS as a function of time for the different algorithms. Botttom right: Difference in RSS (24) as a function of time. CNP and PCD are faster than all other algorithms.

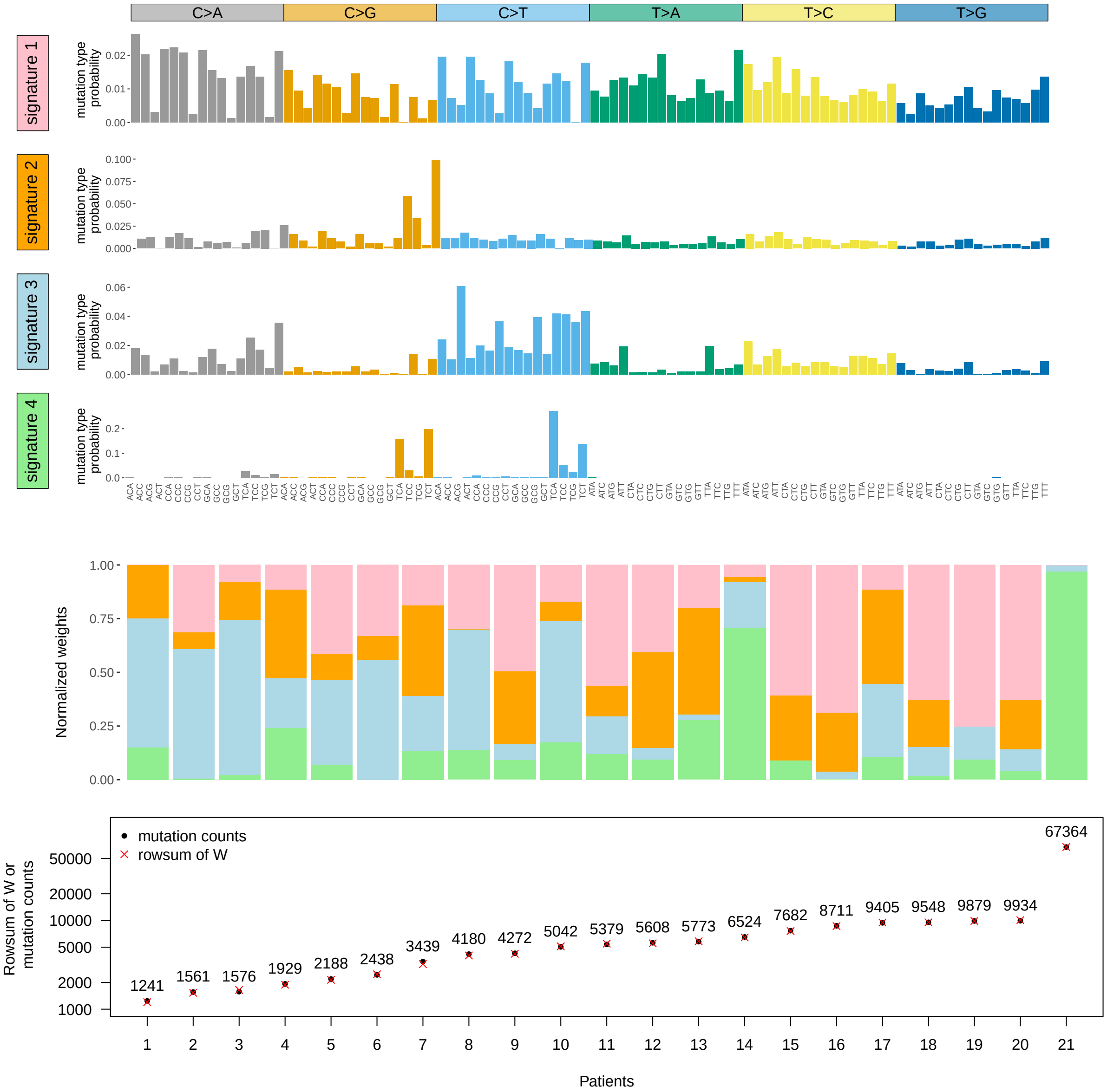

Figure 8: Top: The four estimated signatures (rows of H). Middle: Estimated normalized loadings of signatures (normalized rows of W) for each of the 21 patients. Bottom: Mutation counts and total loadings (row sums of W) for each patient.

7 Discussion and Recommendation

In this paper we have used the Frobenius norm

$$D_{\mathrm{F}}(V \mid WH) = \|V - WH\|_{\mathrm{F}} = \left(\sum_{m=1}^{M} \sum_{n=1}^{N} \left(V_{mn} - (WH)_{mn} \right)^2 \right)^{\frac{1}{2}} \tag{25}$$

as objective function. Another popular choice is the Kullback-Leibler divergence

$$D_{\mathrm{KL}}(V \mid WH) = \sum_{m=1}^{M}\sum_{n=1}^{N}\Big(V_{mn}\log\big(V_{mn}/(WH)_{mn}\big) - V_{mn} + (WH)_{mn}\Big).$$

Using this objective function corresponds to assuming that the entry V_{mn} in the data matrix follows a Poisson distribution with rate $(WH)_{mn}$. The problem of finding a solution to the minimization of the Kullback-Leibler divergence $D_{\mathrm{KL}}(V \mid WH)$ can be approximated by a NLS problem using a second-order Taylor expansion. The Taylor expansion of $x \mapsto x\log(x/\alpha)$ is $x - \alpha + (x-\alpha)^2/(2\alpha)$. Assuming that $V_{mn} > 0$ and $V_{mn} \approx (WH)_{mn}$ for all m and n, it follows that the Kullback-Leibler divergence is approximated by half the Goodness of Fit statistic,

$$D_{\mathrm{KL}}(V \mid WH) \approx \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{N}\frac{(V_{mn} - (WH)_{mn})^2}{V_{mn}}.$$

As in the case of the Frobenius norm, it is sufficient to consider updates for the columns of H. The equation

$$\sum_{m=1}^{M}\frac{(V_{mn} - (WH)_{mn})^2}{V_{mn}} = \sum_{m=1}^{M}\Big(\sqrt{V}_{mn} - \sum_{k=1}^{K}\frac{W_{mk}}{\sqrt{V_{mn}}}H_{kn}\Big)^2 = ||\tilde{v} - \tilde{W}h||^2,$$

where $\tilde{W}_{mk} = W_{mk}/\sqrt{V_{mn}}$, $\tilde{v}_m = \tilde{V}_{mn} = \sqrt{V_{mn}}$ and $h_k = H_{kn}$, shows that the Goodness of Fit can be written in the form of a NLS problem by appropriate scaling.

Varachan and Roland (2008) describe a squared extrapolation method (SQUAREM) to accelerate the convergence of EM, MM and other iterative algorithms. In future applications it could be interesting to include SQUAREM in our methodology. A challenge with applying SQUAREM for NMF problems is that in NMF we require non-negative parameters, and that we typically encounter many zero-valued parameters. We have found that SQUAREM is particularly useful and easy to apply when the parameters are away from the borders of the parameter space.

We have described non-negative matrix factorization and the non-negative least square problem in a general probabilistic setting and as a missing data problem. This formulation has enabled us to connect the coordinate descent, gradient descent and Févotte-Cemgil Expectation-Maximization algorithms as members of a general family of EM updates. Furthermore, we have shown that the majorize-minimize algorithm is *not* a member of the family. Cone projection is by far the most complex algorithm. Fortunately, the understanding and proof of the algorithm was worth the effort; the algorithm is very fast and we avoid difficult stopping criteria.

Finally, we generally recommend a mix-and-match strategy for the Frobenius Norm NMF problem. If the unconstrained solution is valid, then it should be applied.

8 Supplementary Material

The Supplementary Material consists of the following R code:

(i) NQPAlgorithms.R contains all the basic algorithms for the NQP problem: The MM algorithm, the PCD algorithm, the PGD algorithms with exact line search or choice of r, the MuEM algorithm, the FC-EM algorithm, and the CNP algorithm.

(ii) TestProblem.R contains the code for carrying out the test problem.

(iii) BRCA21.RData is a file containing the BRCA21 data. The data is loaded into R using the command load(BRCA21.RData). The data matrix is then given by V.

(iv) NMFanalysis.R is an analysis of the BRCA21 data using the alternating NLS algorithm.

References

L. B. Alexandrov, S. Nik-Zainal, D. C. Wedge, P. J. J. Campbell, and M. R. Stratton. Deciphering signatures of mutational processes operative in human cancer. *Cell Reports*, 3:246–259, 2013.

A. Baez-Ortega and K. Gori. Computational approaches for discovery of mutational signatures in cancer. *Briefings in Bioinformatics*, 1:1–12, 2017.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc., Ser. B*, 39:1–38, 1977.

M. Drton. *Maximum Likelihood Estimation in Gaussian AMP Chain Graph Models and Gaussian Ancestral Graph Models*. PhD. thesis, University of Washington, Department of Statistics, 2004.

C. Févotte and A. T. Cemgil. Nonnegative matrix factorization as probabilistic inference in composite models. In *Proc. 17th European Signal Processing Conference (EUSIPCO)*, pages 1913–1917, Glasgow, Scotland, 2009.

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

D.R. Hunter and K. Lange. A tutorial on mm algorithms. *The American Statistician*, 58:30–37, 2004.

K. Lange, E. C. Chi, and H. Zhou. A brief survey of modern optimization for statisticians. *Int. Stat. Rev.*, 82:46–70, 2014.

S. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, UK, 1996.

C.L. Lawson and R.J. Hanson. *Solving Least Squares Problems*. Englewood Cliffs NJ: Prentice Hall, 1974.

D. Lee and H. Seung. Learning the parts of objects by non-negative matix factorization. *Nature*, 401 (6755):788–791, 1999.

D. Lee and H. Seung. Algorithms for non-negative matrix factorization. In *Proc. Adv. Neural Info. Proc. Syst.*, volume 13, pages 556–562, 2000.

M. C. Meyer. A simple new algorithm for quadratic programming with applications in statistics. *Commun. Stat., Simulation Comput.*, 42(5):1126–1139, 2013.

K.M. Mollun and I.H.M. van Stokkum. Package nnls, available at https://cran.r-project.org/web/packages/nnls/nnls.pdf. 2015.

S. Nik-Zainal, L. B. Alexandrov, D. C. Wedge, P. Van Loo, D. Greenman, K. Raine, D. Jones, J. Hinton, J. Marshall, L. A. Stebbings, et al., and The Breast Cancer Working Group of International Cancer Genome Consortium. Mutational processes molding the genomes of 21 breast cancers. *Cell*, 149: 979–993, 2012.

Serena Nik-Zainal and Sandro Morganella. Mutational signatures in breast cancer: the problem at the dna level, 2017.

R. Varachan and C. Roland. Simple and globally convergent methods for accelerating the convergence of any em algorithm. *Scandinavian Journal of Statistics*, 35:335–353, 2008.

9 Appendix: Flow chart for cone projection

The flow chart for cone projection is shown in Figure 9.

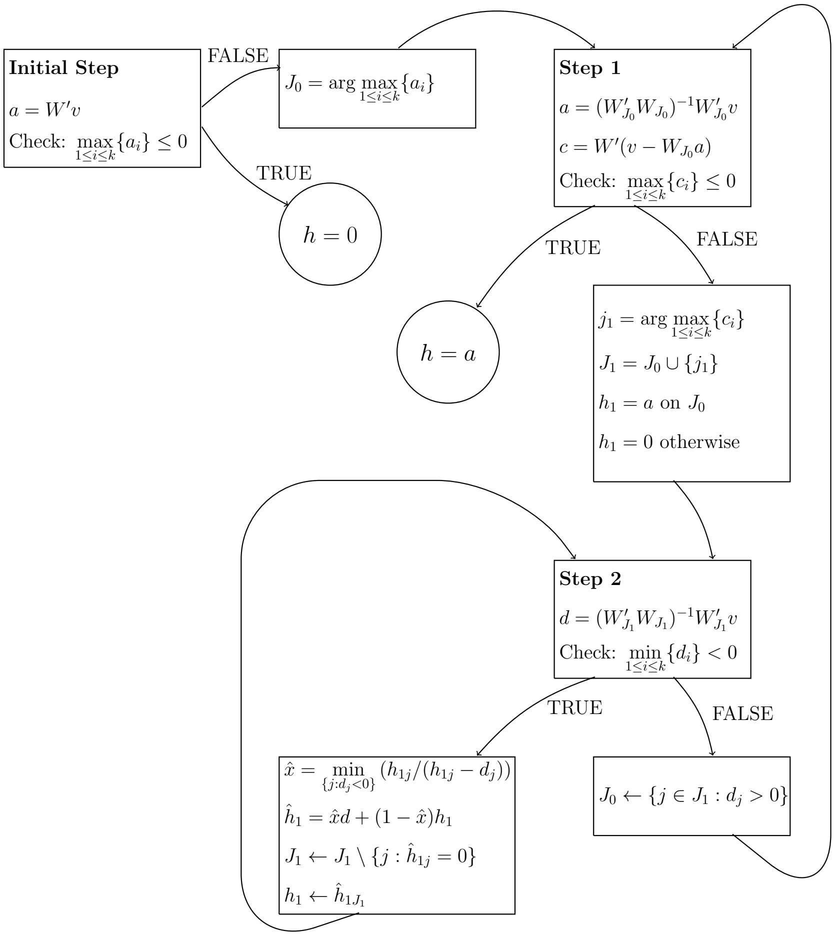
Initial Step
$a = W'v$
Check: $\max_{1\le i\le k}\{a_i\} \le 0$
FALSE
$J_0 = \arg\max_{1\le i\le k}\{a_i\}$
TRUE
$h = 0$
Step 1
$a = (W'_{J_0}W_{J_0})^{-1}W'_{J_0}v$
$c = W'(v - W_{J_0}a)$
Check: $\max_{1\le i\le k}\{c_i\} \le 0$
TRUE
FALSE
$h = a$
$j_1 = \arg\max_{1\le i\le k}\{c_i\}$
$J_1 = J_0 \cup \{j_1\}$
$h_1 = a$ on J_0
$h_1 = 0$ otherwise
Step 2
$d = (W'_{J_1}W_{J_1})^{-1}W'_{J_1}v$
Check: $\min_{1\le i\le k}\{d_i\} < 0$
TRUE
FALSE
$\hat{x} = \min_{\{j:d_j<0\}}(h_{1j}/(h_{1j}-d_j))$
$\hat{h}_1 = \hat{x}d + (1-\hat{x})h_1$
$J_1 \leftarrow J_1 \setminus \{j : \hat{h}_{1j} = 0\}$
$h_1 \leftarrow \hat{h}_{1J_1}$
$J_0 \leftarrow \{j \in J_1 : d_j > 0\}$

Figure 9: Flow chart for cone projection. The algorithm finds the vector of coefficients $\hat{h}$ with non-negative entries that minimizes $||v - Wh||^2$.