ClasSel
Délivrable 3.1
Algorithmes pour la classification croisée : choix des méthodes d'optimisation

A. Rakotomamonjy, J. Delporte et S. Canu

13 septembre 2010

Résumé

Le but de ce rapport est de faire le bilan des méthodes d'optimisation existantes pour extraire les techniques nous permettant une mise en oeuvre efficace des méthodes proposées dans les tâches de classification croisées et de sélection de modèle. Notre stratégie ici consiste pour obtenir une classification croisée consiste à calculer une factorisation parcimonieuse en utilisant une pénalité de type $\ell 1$. Le problème est de savoir quel type d'algorithme faut-il mettre en œuvre pour calculer efficacement la solution.

Pour répondre à cette question, le travail a été organisé en deux parties. Dans un premier temps nous avons cherché à comparer les méthodes permettant d'obtenir une factorisation parcimonieuse dans le but de pouvoir choisir quelle était la mieux adaptée à la classification croisée. Ce travail constitue le premier chapitre de ce rapport est soumis à une revue. Dans un second temps, nous nous sommes intéressé aux méthode de factorisation des matrices binaires, et notamment à travers une modélisation utilisant la régression logistique. Ce travail, encore en cours, fait l'objet du second chapitre de ce document.

Table des matières

Chapitre 1

Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms

1.1 Introduction

Since several years now, there has been a lot of interest about sparse signal approximation. This large interest comes from frequent wishes of practitioners to represent data in the most parsimonious way.

Recently, researchers have focused their efforts on a natural extension of sparse approximation problem which is the problem of finding jointly sparse representations of multiple signal vectors. This problem is also known as simultaneous sparse approximation and it can be stated as follows. Suppose we have several signals describing the same phenomenon, and each signal is contaminated by noise. We want to find the sparsest approximation of each signal by using the same set of elementary functions. Hence, the problem consists in finding the best approximation of each signal while controlling the number of functions involved in all the approximations.

Such a situation arises in many different application domains such as sensor networks signal processing [31, 13], neuroelectromagnetic imaging [23, 38, 58], source localization [32], image restoration [19], and distributed compressed sensing [28].

1.1.1 Problem formalization

Formally, the problem of simultaneous sparse approximation is the following. Suppose that we have measured L signals $\{\mathbf{s}_i\}_{i=1}^{L}$ where each signal is of the form $\mathbf{s}_i = \mathbf{\Phi}\mathbf{c}_i + \epsilon$ where $\mathbf{s}_i \in \mathbb{R}^N$, $\mathbf{\Phi} \in \mathbb{R}^{N\times M}$ is a matrix of unit-norm elementary functions, $\mathbf{c}_i \in \mathbb{R}^M$ a weighting vector and ϵ is a noise vector. $\mathbf{\Phi}$ will be denoted in the sequel as the dictionary matrix. Since we have several signals, the overall measurements can be written as :

$$\mathbf{S} = \mathbf{\Phi}\mathbf{C} + \mathcal{E} \tag{1.1}$$

with $\mathbf{S} = [\mathbf{s}_1\ \mathbf{s}_2\ \cdots\ \mathbf{s}_L]$ a signal matrix, $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2\ \cdots\ \mathbf{c}_L]$ a coefficient matrix and $\mathcal{E}$ a noise matrix. Note that in the sequel, we have adopted the following notations : $c_{i,\cdot}$ and $c_{\cdot,j}$ respectively denote the ith row and jth column of matrix $\mathbf{C}$ and $c_{i,j}$ is the ith element in the jth column of $\mathbf{C}$.

For the simultaneous sparse approximation (SSA) problem, the goal is then to recover the matrix $\mathbf{C}$ given the signal matrix $\mathbf{S}$ and the dictionary $\mathbf{\Phi}$ under the hypothesis that all signals $\mathbf{s}_i$ share the same sparsity profile. This latter hypothesis can also be translated into the coefficient matrix $\mathbf{C}$ having a minimal number of non-zero rows. In order to measure the number of non-zero rows of $\mathbf{C}$, a possible criterion is the so-called *row-support* or *row-diversity measure* of a coefficient matrix defined as

$$\text{rowsupp}(\mathbf{C}) = \{i \in [1 \cdots M] : c_{i,k} \neq 0 \text{ for some } k\}$$

The row-support of $\mathbf{C}$ tells us which atoms of the dictionary have been used for building the signal matrix. Hence, if the cardinality of the row-support is lower than the dictionary cardinality, it means that at least one atom of the dictionary has not been used for synthesizing the signal matrix. Then, the

row-ℓ_0 pseudo-norm of a coefficient matrix can be defined as : $\|\mathbf{C}\|_{row-0} = |\text{rowsupp}(\mathbf{C})|$. According to this definition, the simultaneous sparse approximation problem can be stated as

$$\begin{array}{ll} \min_{\mathbf{C}} & \frac{1}{2}\|\mathbf{S} - \mathbf{\Phi}\mathbf{C}\|_F^2 \\ \text{st.} & \|\mathbf{C}\|_{row-0} \leq T \end{array} \tag{1.2}$$

where $\|\cdot\|_F$ is the Frobenius norm and T a user-defined parameter that controls the sparsity of the solution. Note that the problem can also take the different form :

$$\begin{array}{ll} \min_{\mathbf{C}} & \|\mathbf{C}\|_{row-0} \\ \text{st.} & \frac{1}{2}\|\mathbf{S} - \mathbf{\Phi}\mathbf{C}\|_F \leq \epsilon \end{array} \tag{1.3}$$

For this latter formulation, the problem translates in minimizing the number of non-zero rows in the coefficient matrix $\mathbf{C}$ while keeping control on the approximation error. Both problems (1.2) and (1.3) are appealing for their formulation clarity. However, similarly to the single signal approximation case, solving these optimization problems are notably intractable because $\|\cdot\|_{row-0}$ is a discrete-valued function. Two ways of addressing these intractable problems (1.2) and (1.3) are possible : relaxing the problem by replacing the $\|\cdot\|_{row-0}$ function with a more tractable row-diversity measure or by using some suboptimal algorithms.

A large class of relaxed versions of $\|\cdot\|_{row-0}$ proposed in the literature are encompassed into the following form :

$$J_{p,q}(\mathbf{C}) = \sum_i \|c_{i,\cdot}\|_q^p \qquad \text{with } \|c_{i,\cdot}\|_q = \left(\sum_j |c_{i,j}|^q\right)^{1/q}$$

where typically $p \leq 1$ and $q \geq 1$. This penalty term can be interpreted as the ℓ_p quasi-norm of the sequence $\{\|c_{i,\cdot}\|_q\}_i$. Note that as p converges to 0, $J_{p,q}(\mathbf{C})$ provably converges towards $\sum_i \log(\|c_{i,\cdot}\|)$. According to this relaxed version of the row-diversity measure, most of the algorithms proposed in the literature try to solve the relaxed problem :

$$\min_{\mathbf{C}} \frac{1}{2}\|\mathbf{S} - \mathbf{\Phi}\mathbf{C}\|_F^2 + \lambda J_{p,q}(\mathbf{C}) \tag{1.4}$$

where λ is another user-defined parameter that balances the approximation error and the sparsity-inducing penalty $J_{p,q}(\mathbf{C})$. The choice of p and q results in a compromise between the row-support sparsity and the convexity of the optimization problem. Indeed, problem (1.4) is known to be convex when $p, q \geq 1$ while it is known to produce a row-sparse matrix $\mathbf{C}$ if $p \leq 1$ (due to the penalty function singularity at $\mathbf{C} = 0$ [17]).

The simultaneous sparse approximation problem as described here is equivalent to several other problems studied in other research communities. Problem (1.4) can be reformulated so as to make clear its relation with some other problems denoted as the $\ell_p - \ell_q$ *group lasso* [60] or *block-sparse regression* [29] in statistics or *block-sparse signal recovery* [15] in signal processing. Indeed, let us define :

$$\tilde{\mathbf{s}} = \begin{pmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_L \end{pmatrix} \qquad \tilde{\mathbf{\Phi}} = \begin{pmatrix} \mathbf{\Phi} & & (0) \\ & \ddots & \\ (0) & & \mathbf{\Phi} \end{pmatrix} \qquad \tilde{\mathbf{c}} = \begin{pmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_L \end{pmatrix}$$

then, problem (1.4) can be equivalently rewritten as :

$$\min_{\tilde{\mathbf{c}}} \frac{1}{2}\|\tilde{\mathbf{s}} - \tilde{\mathbf{\Phi}}\tilde{\mathbf{c}}\|_2^2 + \lambda J'_{p,q}(\tilde{\mathbf{c}}) \tag{1.5}$$

where $J'_{p,q}(\tilde{\mathbf{c}}) = \sum_{i=1}^M \|\tilde{\mathbf{c}}_{g_i}\|_q^p$ with g_i being all the indices in $\tilde{\mathbf{c}}$ related to the i-th element of the dictionary matrix $\mathbf{\Phi}$.

As we have already stated, simultaneous sparse approximation and equivalent problems have been investigated by diverse communities and have also been applied to various application problems. Since the literature on these topics has been overwhelming since the last few years, The aim of this work is to gather results from different communities (statistics, signal processing and machine learning) so as to survey, analyze and compare different proposed algorithms for solving optimization problem (1.4) for different

values of p and q. In particular, instead of merely summarizing existing results, we enrich our survey by providing results like proof of convergence, formal relations between different algorithms and experimental comparisons that were not available in the literature. These experimental comparisons essentially focus on the computational complexity of the different methods, on their ability of recovering the signal sparsity pattern and on the quality of approximation they provided evaluated through mean-square error.

Note that recently, there has been various works which addressed the simultaneous sparse approximation in the noiseless case [47, 45] and many works which aims at providing theoretical approximation guarantees in both noiseless and noisy cases [8, 15, 34, 16, 26, 30]. Surveying these works is beyond the scope of this paper and we suggest interested readers to follow for instance these pointers and references therein.

The most frequent case of problem (1.4) encountered in the literature is the one where $p = 1$ and $q = 2$. This case is the simpler one since it leads to a convex problem. We discuss different algorithms for solving this problem in Section 1.2. Then, we survey in Section 1.3 generic algorithms that are able to solve our approximation problems with different values of p and q. These algorithms are essentially iterative reweighted ℓ_1 or ℓ_2 algorithms. Many works in the literature have focused on one algorithm for solving a particular case of problem (1.4). These specific algorithms are discussed in section 1.4. Notably, we present some greedy algorithms and discuss a sparse bayesian learning algorithm. The experimental comparisons we carried out aim at clarifying how each algorithm behaves in terms of computational complexity, in sparsity recovery and in approximation mean-square error. These results are described in Section 1.5. For a sake of reproducibility, the code used in this paper has been made freely available [1]. Final discussions and conclusions are given in Section 1.6.

1.2 Solving the $\ell_1 - \ell_2$ optimization problem

The algorithm we propose in this section addresses the particular case of $p = 1$ and $q = 2$, denoted as the M-BP problem. We show that the specific structure of the problem leads to a very simple block coordinate descent algorithm named as M-BCD. We also show that if the dictionary is under-complete then it can be proved that the solution of the problem is equivalent to a simple shrinkage of the coefficient matrix. Proof of convergence of our M-BCD algorithm is also given in this section

Before delving into the algorithmic details, we should note that block-coordinate descent have already been considered by Sardy et al. [42] and Elad [14] for sparse signal approximations from orthogonal and redundant representations. Hence the numerical scheme we propose here can be seen as an extension of their works to vector-valued data.

1.2.1 Deriving optimality conditions

The M-BP optimization problem is the following

$$\min_{\mathbf{C}} W(\mathbf{C}) = \frac{1}{2}\|\mathbf{S} - \mathbf{\Phi C}\|_F^2 + \lambda \sum_i \|c_{i,\cdot}\|_2 \tag{1.6}$$

where the objective function $W(\mathbf{C})$ is a non-smooth but convex function. Since the problem is unconstrained a necessary and sufficient condition for a matrix $\mathbf{C}^\star$ to be a minimizer of (1.10) is that $\mathbf{0} \in \partial W(\mathbf{C}^\star)$ where $\partial W(\mathbf{C})$ denotes the subdifferential of our objective value $W(\mathbf{C})$ [4]. By computing the subdifferential of $W(\mathbf{C})$ with respect to each row $c_{i,\cdot}$ of $\mathbf{C}$, the KKT optimality condition of problem (1.10) is then

$$-\mathbf{r}_i + \lambda g_{i,\cdot} = 0 \qquad \forall i$$

where $\mathbf{r}_i = \phi_i^t(\mathbf{S} - \mathbf{\Phi C})$ and $g_{i,\cdot}$ is the i-th row of a subdifferential matrix $\mathbf{G}$ of $J_{1,2}(\mathbf{C}) = \sum_i \|c_{i,\cdot}\|_2$. According to the $J_{1,2}$'s subdifferential definition [39], the KKT optimality conditions can be rewritten as

$$\begin{aligned} -\mathbf{r}_i + \lambda \frac{c_{i,\cdot}}{\|c_{i,\cdot}\|_2} &= \mathbf{0} \quad \forall i, \quad c_{i,\cdot} \neq \mathbf{0} \\ \|\mathbf{r}_i\|_2 &\leq \lambda \quad \forall i, \quad c_{i,\cdot} = \mathbf{0} \end{aligned} \tag{1.7}$$

1. http://asi.insa-rouen.fr/enseignants/~arakotom/code/SSAindex.html

A matrix $\mathbf{C}$ satisfying these equations can be obtained after the following algebra. Let us expand each $\mathbf{r}_i$ so that

$$\begin{aligned} \mathbf{r}_i &= \phi_i^t(\mathbf{S} - \mathbf{\Phi}\mathbf{C}_{-i}) - \phi_i^t\phi_i c_{i,\cdot} \\ &= T_i - c_{i,\cdot} \end{aligned} \tag{1.8}$$

where $\mathbf{C}_{-i}$ is the matrix $\mathbf{C}$ with the i-th row being set to 0 and $T_i = \phi_i^t(\mathbf{S} - \mathbf{\Phi}\mathbf{C}_{-i})$. The second equality is obtained by remembering that $\phi_i^t\phi_i$=1. Then, equation (1.11) tells us that if $c_{i,\cdot}$ is non-zero, T_i and $c_{i,\cdot}$ have to be collinear. Plugging all these points into equation (1.11) yields to an optimal solution that can be obtained as :

$$c_{i,\cdot} = \left(1 - \frac{\lambda}{\|T_i\|}\right)_+ T_i \qquad \forall i \tag{1.9}$$

From this update equation, we can derive a simple algorithm which consists in iteratively applying the update (1.13) to each row of $\mathbf{C}$.

1.2.2 The algorithm and its convergence

Our block-coordinate descent algorithm is detailed in Algorithm (2). It is a simple and efficient algorithm for solving M-BP.

Basically, the idea consists in solving each row $c_{i,\cdot}$ at a time. By starting from a sparse solution like, $\mathbf{C} = 0$, at each iteration, we check for a given i whether row $c_{i,\cdot}$ is optimal or not based on conditions (1.11). If not, $c_{i,\cdot}$ is then updated according to equation (1.13).

Although, such a block-coordinate algorithm does not converge in general for non-smooth optimization problem, Tseng [54] has shown that for an optimization problem which objective value is the sum of a smooth and convex function and a non-smooth but block-separable convex function, block-coordinate optimization converges towards the global minimum of the problem. Our proof of convergence is based on such properties and follows the same line as the one proposed by Sardy et al. [42].

Theorem 1 *The M-BCD algorithm converges to a solution of the M-Basis Pursuit problem given in Equation (1.10), where convergence is understood as any accumulation point of the M-BCD algorithm is a minimum of problem (1.10) and the sequence of* $\{\mathbf{C}_k\}$ *generated by the algorithm is bounded.*

Proof 1 *Note that M-BP problem presents a particular structure with a smooth and differentiable convex function* $\|\mathbf{S} - \mathbf{\Phi}\mathbf{C}\|_F^2$ *and a row-separable penalty function* $\sum_i h_i(c_{i,\cdot})$ *where* $h(\cdot)$ *is a continuous and convex function with respects to* $c_{i,\cdot}$.

Also note that our algorithm considers a cyclic rule where within each loop, for each $i \in [1, \cdots, M]$, *each* $c_{i,\cdot}$ *is considered for optimization. The main particularity is that for some* i, *the* $c_{i,\cdot}$ *may be left unchanged by the block-coordinate descent if already optimal. This occurs especially for row* $c_{i,\cdot}$ *which are equal to* 0.

Then according to the special structure of the problem and the use of a cyclic rule, the results of Tseng [54] prove that our M-BCD algorithm converges.

Intuitively, we can understand this algorithm as an algorithm which tends to shrink to zero rows of the coefficient matrix that contribute poorly to the approximation. Indeed, T_i can be interpreted as the correlation between the residual when row i has been removed and ϕ_i. Hence the smaller the norm of T_i is, the less ϕ_i is relevant in the approximation. And according to equation (1.13), the smaller the resulting $c_{i,\cdot}$ is. Insight into this block-coordinate descent algorithm can be further obtained by supposing that $M \leq N$ and that $\mathbf{\Phi}$ is composed of orthonormal elements of $\mathbb{R}^N$, hence $\mathbf{\Phi}^t\mathbf{\Phi} = \mathbf{I}$. In such situation, we have

$$T_i = \phi_i^t\mathbf{S} \quad \text{and} \quad \|T_i\|_2^2 = \sum_{k=1}^{L}(\phi_i^t s_k)^2$$

and thus

$$c_{i,\cdot} = \left(1 - \frac{\lambda}{\sqrt{\sum_k^L(\phi_i^t s_k)^2}}\right)_+ \phi_i^t\mathbf{S}$$

This last equation highlights the relation between the single Basis Pursuit (when $L = 1$) and the Multiple-Basis Pursuit algorithm presented here. Both algorithms lead to a shrinkage of the coefficient projection.

Algorithm 1 Solving M-BP through block-coordinate descent

```
C = 0, Loop = 1
while Loop do
  for i = 1, 2, ..., M do
    Compute ||r_i||
    if optimality condition of c_{i,.} according to equations (1.11) is not satisfied then
      c_{i,.} = (1 - λ/||T_i||)_+ T_i
    end if
  end for
  if all optimality conditions are satisfied then
    Loop = 0
  end if
end while
```

With the inclusion of multiple signals, the shrinking factor becomes more robust to noise since it depends on the correlation of the atom ϕ_i to all signals.

1.2.3 Some relations with other works

As we have already stated, our M-BCD algorithm can be considered as an extension to simultaneous signal approximations of the works of Sardy et al. [42] and Elad [14]. However, here, we want to emphasize the importance of starting from a $\mathbf{C} = 0$. Indeed, since in the estimated $\hat{\mathbf{C}}$ is expected to be sparse, by doing so, only few updates are needed before convergence.

In addition to the works of Sardy et al. and Elad, many others authors have considered block-coordinate descent algorithm for related sparse approximation problems. For instance, it has also been used for solving the Lasso [21], and the elastic net [61]. Other works have also considered iterative thresholding algorithms for solving single signal sparse approximation problem [11, 19].

For recovering vector valued data with joint sparsity constraints, Fornasier et al. [**?**] have proposed an extension of the Landweber iterative approach of Daubechies et al. [11]. In their work, Fornasier et al. have also used an iterative shrinking algorithm (which has the flavor of a gradient projection approach) which is able to solve the general problem (1.4) with $p = 1$ and $q = \{1, 2, \infty\}$. For $q = 2$, the main difference between their algorithm and the one we propose here is that, by optimizing at each loop, only the $c_{i,\cdot}$'s that are not optimal yet, we have an algorithm that is more efficient than the one of Fornasier et al.

As we stated previously, the M-FOCUSS algorithm also solves the M-BP problem. In their M-FOCUSS approach, Cotter et al. [10] have proposed a factored gradient algorithm. That algorithm is related to iterative reweighted least-squares, which at each iteration updates the coefficient matrix $\mathbf{C}$. However, their factored gradient algorithm presents a important issue. Indeed, the updates they propose are not guaranteed to converge to a local minima of the problem (if the problem is not convex $p < 1$) or to the global minimum of the convex problem ($p = 1$). Indeed, their algorithm presents several fixed-points since when a row of $\mathbf{C}$ is equal to 0, it stays at 0 at the next iteration. Although such a point may be harmless if the algorithm is initialized with a "good" starting point, it is nonetheless an undesirable point when solving a convex problem. At the contrary, our M-BCD algorithm does not suffer from the presence of such fixed-points. However, such fixed-points in the M-FOCUSS algorithm can be handled by introducing a smoothing term ε in the weight so that the updated weight (according to Cotter's notation and for $p = 1$) becomes

$$\mathbf{W} = \operatorname{diag}\left(\sqrt{\|c_{i,\cdot}\| + \epsilon}\right)$$

where $\mathbf{W}$ is the diagonal weighting matrix and $\varepsilon > 0$. The use of ε avoids a given weight to be at zero and consequently it avoids the related $c_{i,\cdot}$ to stay permanently at zero. Then if we furthermore note that M-FOCUSS is not more than an iterative reweighted least-square. According to the very recent works of Daubechies et al. [12] and Chartrand et al. [7], it seems justified to iterate the M-FOCUSS algorithm using decreasing value of ε. In our numerical experimental, we will consider the M-FOCUSS algorithm with fixed and decreasing value of ε.

1.2.4 Evaluating complexity

From a computational complexity point of view, it is not possible to evaluate the exact number of iterations that will be needed before convergence of our algorithm. However, we can analyze the computational cost per each iteration. Although, this may not be relevant since the number of iterations needed for the considered algorithms to converge may be very different, such knowledge give an hint about the algorithm scaling with respects to parameters of the simultaneous sparse approximation problem.

For our M-BCD algorithm, we can note that each shrinking operation, in the worst case scenario, has to be done M times and the dominating cost for each update is the computation of T_i. This computation involves the matrix multiplication $\mathbf{\Phi C}_{-i}$ and a matrix-vector multiplication which respectively need $\mathcal{O}(NML)$ and $\mathcal{O}(NL)$ operations. On the overall, if we assume that at each iteration, all $c_{i,\cdot}$ are updated, we can consider that the computational cost of our algorithm is about $\mathcal{O}(M^2NL)$. This cost per iteration can be compared to the one of M-FOCUSS algorithm and second-order code programming of Malioutov et. al [32] which are respectively $\mathcal{O}(MN^2)$ and $\mathcal{O}(M^3L^3)$. Theoretically, it seems that our algorithm suffers more than M-FOCUSS from large dictionary size but it is far more efficient than the SOC programming.

Illustrations of how our algorithm behaves and empirical computational complexity evaluations are given in section 1.5.

When $p = 1, q = 2$, optimization problem (1.4) becomes a particular problem named as M-BP for Multiple Basis Pursuit in the sequel. It is a special case that deserves attention. Indeed, it seems to be the most natural extension of the so-called Lasso problem [49] or Basis Pursuit Denoising [9], since for $L = 1$, it can be easily shown that problem (1.4) reduced to these two problems. The key point of this case is that it yields to a convex optimization problem and thus it can benefit from all properties resulting from convexity *e.g* global minimum. One of the first algorithm addressing such the M-BP problem is the one proposed by Cotter et al. [10] known as M-FOCUSS. Such an algorithm based on factored gradient descent works for any $p \leq 1$ and has been proved to converge towards a local or global (when $p = 1$) minimum of problem (1.4) if it does not get stuck into a fixed point. Because, M-FOCUSS is better tailored for situations where $p \leq 1$, we post-pone its description to next section. The two algorithms on which we focus on herein, are based on a block-coordinate descent as proposed by Yuan et al. [60] for the group lasso and the one based on Landweber iterations of Fornasier et al. [19]. We have biased our survey towards these two approaches because of their efficiencies and their convergence properties.

1.2.5 Block Coordinate descent

The specific structure of the optimization problem (1.4) for $p = 1$ and $q = 2$ leads to a very simple block coordinate descent algorithm that we name as M-BCD. Here, we provide a more detailed derivation of the algorithm and we also give results that were not available in the Yuan et al. work [60]. We provide a proof of convergence of the M-BCD algorithm and we show that if the dictionary is under-complete then it can be proved that the solution of the problem is equivalent to a simple shrinkage of the coefficient matrix.

Deriving optimality conditions

The M-BP optimization problem is the following

$$\min_{\mathbf{C}} W(\mathbf{C}) = \frac{1}{2}\|\mathbf{S} - \mathbf{\Phi C}\|_F^2 + \lambda \sum_i \|c_{i,\cdot}\|_2 \tag{1.10}$$

where the objective function $W(\mathbf{C})$ is a non-smooth but convex function. Since the problem is unconstrained a necessary and sufficient condition for a matrix $\mathbf{C}^\star$ to be a minimizer of (1.10) is that $\mathbf{0} \in \partial W(\mathbf{C}^\star)$ where $\partial W(\mathbf{C})$ denotes the subdifferential of our objective value $W(\mathbf{C})$ [4]. By computing the subdifferential of $W(\mathbf{C})$ with respect to each row $c_{i,\cdot}$ of $\mathbf{C}$, the KKT optimality condition of problem (1.10) is then

$$-\mathbf{r}_i + \lambda g_{i,\cdot} = 0 \qquad \forall i$$

where $\mathbf{r}_i = \phi_i^t(\mathbf{S} - \mathbf{\Phi C})$ and $g_{i,\cdot}$ is the i-th row of a subdifferential matrix $\mathbf{G}$ of $J_{1,2}(\mathbf{C}) = \sum_i \|c_{i,\cdot}\|_2$. According to the $J_{1,2}$'s subdifferential definition given in the appendix, the KKT optimality conditions

can be rewritten as

$$\begin{aligned} -\mathbf{r}_i + \lambda \frac{c_{i,\cdot}}{\|c_{i,\cdot}\|_2} &= \mathbf{0} \quad \forall i, \quad c_{i,\cdot} \neq \mathbf{0} \\ \|\mathbf{r}_i\|_2 &\leq \lambda \quad \forall i, \quad c_{i,\cdot} = \mathbf{0} \end{aligned} \tag{1.11}$$

A matrix $\mathbf{C}$ satisfying these equations can be obtained after the following algebra. Let us expand each $\mathbf{r}_i$ so that

$$\begin{aligned} \mathbf{r}_i &= \phi_i^t(\mathbf{S} - \mathbf{\Phi}\mathbf{C}_{-i}) - \phi_i^t\phi_i c_{i,\cdot} \\ &= T_i - c_{i,\cdot} \end{aligned} \tag{1.12}$$

where $\mathbf{C}_{-i}$ is the matrix $\mathbf{C}$ with the i-th row being set to 0 and $T_i = \phi_i^t(\mathbf{S} - \mathbf{\Phi}\mathbf{C}_{-i})$. The second equality is obtained by remembering that $\phi_i^t\phi_i$=1. Then, equation (1.11) tells us that if $c_{i,\cdot}$ is non-zero, T_i and $c_{i,\cdot}$ have to be collinear. Plugging all these points into equation (1.11) yields to an optimal solution that can be obtained as :

$$c_{i,\cdot} = \left(1 - \frac{\lambda}{\|T_i\|}\right)_+ T_i \qquad \forall i \tag{1.13}$$

where $(x)_+ = x$ if $x > 0$ and 0 otherwise. From this update equation, we can derive a simple algorithm which consists in iteratively applying the update (1.13) to each row of $\mathbf{C}$.

The algorithm and its convergence

Our block-coordinate descent algorithm is detailed in Algorithm (2). It is a simple and efficient algorithm for solving M-BP.

Basically, the idea consists in solving each row $c_{i,\cdot}$ at a time. By starting from a sparse solution like, $\mathbf{C} = 0$, at each iteration, one checks for a given i whether row $c_{i,\cdot}$ is optimal or not based on conditions (1.11). If not, $c_{i,\cdot}$ is then updated according to equation (1.13).

Although, such a block-coordinate algorithm does not converge in general for non-smooth optimization problem, Tseng [54] has shown that for an optimization problem which objective value is the sum of a smooth and convex function and a non-smooth but block-separable convex function, block-coordinate optimization converges towards the global minimum of the problem. Our proof of convergence is based on such properties and follows the same lines as the one proposed by Sardy et al. [42] for a single signal approximation.

Theorem 2 *The M-BCD algorithm converges to a solution of the M-Basis Pursuit problem given in Equation (1.10), where convergence is understood as any accumulation point of the M-BCD algorithm is a minimum of problem (1.10) and the sequence of* $\{\mathbf{C}_k\}$ *generated by the algorithm is bounded.*

Proof 2 *Note that M-BP problem presents a particular structure with a smooth and differentiable convex function* $\|\mathbf{S} - \mathbf{\Phi}\mathbf{C}\|_F^2$ *and a row-separable penalty function* $\sum_i h_i(c_{i,\cdot})$ *where* $h(\cdot)$ *is a continuous and convex function with respects to* $c_{i,\cdot}$.

Also note that our algorithm considers a cyclic rule where within each loop, for any $i \in [1, \cdots, M]$, *each* $c_{i,\cdot}$ *is considered for optimization. The main particularity is that for some* i, *the* $c_{i,\cdot}$ *may be left unchanged by the block-coordinate descent if already optimal. This occurs especially for row* $c_{i,\cdot}$ *which are equal to* 0.

Then according to the special structure of the problem and the use of a cyclic rule, the results of Tseng [54] prove that our M-BCD algorithm converges.

Another point we want to emphasize and that has not been shed to light yet is that this algorithm should be initialized at $\mathbf{C} = 0$ so as to take advantage of the sparsity pattern of the solution. Indeed, by doing so, only few updates are needed before convergence.

Intuitively, we can understand this algorithm as an algorithm which tends to shrink to zero rows of the coefficient matrix that contribute poorly to the approximation. Indeed, T_i can be interpreted as the correlation between the residual when row i has been removed and ϕ_i. Hence the smaller the norm of T_i is, the less ϕ_i is relevant in the approximation. And according to equation (1.13), the smaller the resulting $c_{i,\cdot}$ is. Insight into this block-coordinate descent algorithm can be further obtained by supposing that

Algorithm 2 Solving M-BP through block-coordinate descent

t=1,$\mathbf{C}^{(0)} = 0$, Loop = 1,
while Loop **do**
  for $i = 1, 2, \cdots, M$ **do**
    Compute $\|\mathbf{r}_i\|$
    if optimality condition of $c_{i,\cdot}$ according to equations (1.11) is not satisfied **then**
      $T_i \leftarrow \phi_i^t(\mathbf{S} - \mathbf{\Phi}\mathbf{C}_i^{(t-1)})$
      $c_{i,\cdot}^{(t)} \leftarrow \left(1 - \frac{\lambda}{\|T_i\|}\right)_+ T_i$
    end if
    $t \leftarrow t + 1$
  end for
  if all optimality conditions are satisfied **then**
    Loop = 0
  end if
end while

$M \leq N$ and that $\mathbf{\Phi}$ is composed of orthonormal elements of $\mathbb{R}^N$, hence $\mathbf{\Phi}^t\mathbf{\Phi} = \mathbf{I}$. In such a situation, we have

$$T_i = \phi_i^t \mathbf{S} \quad \text{and} \quad \|T_i\|_2^2 = \sum_{k=1}^{L} (\phi_i^t s_k)^2$$

and thus

$$c_{i,\cdot} = \left(1 - \frac{\lambda}{\sqrt{\sum_k^L (\phi_i^t s_k)^2}}\right)_+ \phi_i^t \mathbf{S}$$

This last equation highlights the relation between the single Basis Pursuit (when $L = 1$) and the Multiple-Basis Pursuit algorithm presented here. Both algorithms lead to a shrinkage of the coefficient projection when considering orthonormal dictionary elements. With the inclusion of multiple signals, the shrinking factor becomes more robust to noise since it depends on the correlation of the atom ϕ_i to all signals.

This M-BCD algorithm can be considered as an extension to simultaneous signal approximations of the works of Sardy et al. [42] and Elad [14] which also considered block coordinate descent for single signal approximation. In addition to the works of Sardy et al. and Elad, many others authors have considered block-coordinate descent algorithm for related sparse approximation problems. For instance, block coordinate descent has also been used for solving the Lasso [21], and the elastic net [61].

1.2.6 Landweber iterations

For recovering vector valued data with joint sparsity constraints, Fornasier et al. [19] have proposed an extension of the Landweber iterative approach introduced by Daubechies et al. [11]. In their work, Fornasier et al. used an iterative shrinking algorithm (which has the flavor of a gradient projection approach) which is able to solve the general problem (1.4) with $p = 1$ and $q = \{1, 2, \infty\}$. We summarize here the details of their derivations for general q.

The problem to solve is the one given in Equation (1.4) with $p = 1$:

$$\min_{\mathbf{C}} \frac{1}{2}\|\mathbf{S} - \mathbf{\Phi}\mathbf{C}\|_F^2 + \lambda \sum_i \|c_{i,\cdot}\|_q \tag{1.14}$$

The iterative algorithm can be derived first by supposing that the objective function given above is strictly convex then by defining the following surrogate objective function :

$$J_{sur}(\mathbf{C}, \mathbf{A}) = \frac{1}{2}\|\mathbf{S} - \mathbf{\Phi}\mathbf{C}\|_F^2 + \lambda \sum_i \|c_{i,\cdot}\|_q + \frac{1}{2}\|\mathbf{\Phi}\mathbf{C} - \mathbf{\Phi}\mathbf{A}\|_F^2 - \frac{1}{2}\|\mathbf{C} - \mathbf{A}\|_F^2$$

From this surrogate function, Fornasier et al. considers the solution of problem (1.4) as the limit of the sequence $\mathbf{C}^{(t)}$:

$$\mathbf{C}^{(t+1)} = \arg\min_{\mathbf{C}} J_{sur}(\mathbf{C}, \mathbf{C}^{(t)})$$

Algorithm 3 Solving M-BP through Landweber iterations

$\mathbf{C}^{(0)} = 0$, Loop = 1, t=0
while Loop **do**
  $\mathbf{C}^{(t+\frac{1}{2})} \leftarrow \mathbf{C}^{(t)} + \mathbf{\Phi}^t(\mathbf{S} - \mathbf{\Phi}\mathbf{C}^{(t)})$
  for $i = 1, 2, \cdots, M$ **do**
    Compute $\|c_{i,\cdot}^{(t+1/2)}\|$
    $c_{i,\cdot}^{(t+1)} = \left(1 - \frac{\lambda}{\|c_{i,\cdot}^{(t+\frac{1}{2})}\|}\right)_+ c_{i,\cdot}^{(t+\frac{1}{2})}$
  end for
  $t \leftarrow t + 1$
  if stopping criterions are satisfied **then**
    Loop = 0
  end if
end while

Note that according to the additional terms (the last two ones) in the surrogate function, $C^{(t+1)}$ is a coefficient matrix that minimizes the desired objective function plus terms that constraints $\mathbf{C}^{(t+1)}$ and the novel signal approximation $\mathbf{\Phi}\mathbf{C}^{(t+1)}$ to be "closed" respectively to $\mathbf{C}^{(t)}$ and the previous signal approximation $\mathbf{\Phi}\mathbf{C}^{(t)}$.

Now, the minimizer of J_{sur} for fixed $\mathbf{A}$ can be obtained by expanding J_{sur} as

$$\begin{aligned} J_{sur}(\mathbf{C}, \mathbf{A}) &= \frac{1}{2}\|(\mathbf{A} + \mathbf{\Phi}^t(\mathbf{S} - \mathbf{\Phi}\mathbf{A})) - \mathbf{C}\|_F^2 + \lambda \sum_i \|c_{i,\cdot}\|_q \\ &\quad -\frac{1}{2}\|\mathbf{A} + \mathbf{\Phi}^t(\mathbf{S} - \mathbf{\Phi}\mathbf{A})\|^2 + \frac{1}{2}\|\mathbf{S}\|_F^2 - \frac{1}{2}\|\mathbf{\Phi}\mathbf{A}\|_F^2 + \frac{1}{2}\|\mathbf{A}\|_F^2 \end{aligned}$$

and noticing that the second line of this equation does not depend on $\mathbf{C}$. Thus we have

$$\arg\min_{\mathbf{C}} J_{sur}(\mathbf{C}, \mathbf{A}) = U_q(\mathbf{A} + \mathbf{\Phi}^t(\mathbf{S} - \mathbf{\Phi}\mathbf{A}))$$

with $U_q(\mathbf{X})$ being defined as

$$U_q(\mathbf{X}) = \arg\min_{\mathbf{C}} \frac{1}{2}\|\mathbf{X} - \mathbf{C}\|_F^2 + \lambda \sum_i \|c_{i,\cdot}\|_q \tag{1.15}$$

Thus the iterative algorithm we get simply reads, at iteration t as :

$$\mathbf{C}^{(t+1)} = U_q(\mathbf{C}^{(t)} + \mathbf{\Phi}^t(\mathbf{S} - \mathbf{\Phi})\mathbf{C}^{(t)}) \tag{1.16}$$

The Landweber algorithm proposed by Fornasier et al. involves two steps : a first step which is similar to a gradient descent with fixed step and which does not take into account the sparsity constraint and a thresholding step through the operator $U_q(\cdot)$, which updates the solution according to the penalty $\sum_i \|c_{i,\cdot}\|_q$. Fornasier et al. give detailed information of the $U_q(\cdot)$ closed form for $q = \{1, 2, \infty\}$. We can note that the problem (1.15) decouples with respect to the row of $\mathbf{C}$ and the thresholding operator $U_q(\cdot)$ acts independently on each row.

According to Fornasier et al. [19], for $q = 2$, $U_2(\mathbf{C})$ writes for each row $c_{i,\cdot}$ as :

$$[U_2(\mathbf{C})]_{i,\cdot} = \left(1 - \frac{\lambda}{\|c_i\|}\right)_+ c_{i,\cdot}$$

and the full algorithm is summarized in Algorithm 3. We note that block-coordinate descent and Landweber iterations yield to algorithms that are very similar. Two differences can be noted. The first one is that the shrinking operation for one case is based on T_i while in the other case it directly considers $c_{i,\cdot}$. At the moment, it is not clear what are the implications of these two different schemes in term of convergence rate and more investigations are needed to clarify this point. The other main difference between the two approaches is that, by optimizing at each loop, only the $c_{i,\cdot}$'s that are not optimal yet, the BCD algorithm is more efficient than the one of Fornasier et al.

Note that Fornasier et al. have also shown that this iterative scheme converges towards the minimizer of problem (1.4) for $1 \geq q \geq \infty$.

1.2.7 Other works

Besides the M-FOCUSS algorithm of Cotter et al., another seminal work which considered the SSA problem is the one of Malioutov et al. [32]. While dealing with a source localization problem, they had to consider a sparse approximation problem with joint sparsity constraints. From the *block-sparse* signal approximation problem

$$\min_{\tilde{\mathbf{c}}} \frac{1}{2}\|\tilde{\mathbf{s}} - \tilde{\mathbf{\Phi}}\tilde{\mathbf{c}}\|_2 + \lambda \sum_{i}^{M} \|\tilde{\mathbf{c}}_{g_i}\|_2$$

they derived an equivalent problem which writes as :

$$\begin{array}{ll} \min_{p,q,\mathbf{r},\tilde{\mathbf{c}}} & p + \lambda q \\ \text{st.} & \frac{1}{2}\|\tilde{\mathbf{s}} - \tilde{\mathbf{\Phi}}\tilde{\mathbf{c}}\|_2^2 \leq p \\ & \sum_{i=1}^{M} r_i \leq q \\ & \|\mathbf{c}_{g_i}\| \leq r_i \qquad \forall i = 1, \cdots, M \end{array}$$

Note that this above problem has a linear objective function and linear and second-order cone constraints. Such a problem is known as a second-order cone programming problem and there exists interior point method for solving it [46]. While very attractive because of its simple formulation and the global convergence of the algorithm, this approach suffers from a high complexity which is of the order $\mathcal{O}(M^3L^3)$.

Very recently, two other algorithms have been proposed for addressing our jointly sparse approximation problem. The first one, derived by van den Berg et al. [55] is very similar to the one of Fornasier et al. although from a different perspective. Indeed, they develop a method based on spectral gradient projection for solving the *Group lasso* problem which is the one given in Equation (1.5) with $p = 1$ and $q = 2$. However, instead of the regularized version, they considered the following constrained version

$$\begin{array}{ll} \min_{\tilde{\mathbf{c}}} & \frac{1}{2}\|\tilde{\mathbf{s}} - \tilde{\mathbf{\Phi}}\tilde{\mathbf{c}}\|_2 \\ \text{st.} & \sum_{i}^{M} \|\tilde{\mathbf{c}}_{g_i}\|_2 \leq \tau \end{array}$$

that they iteratively solve owing to spectral gradient projection. Basically, their solution iterates write as

$$\tilde{\mathbf{c}}^{(t+1)} = P\left(\tilde{\mathbf{c}}^{(t)} + \alpha\tilde{\mathbf{\Phi}}^t(\tilde{\mathbf{s}} - \tilde{\mathbf{\Phi}}\tilde{\mathbf{c}}^{(t)})\right) \tag{1.17}$$

where α is some step size to be optimized for instance by backtracking and $P(\cdot)$ is the projection operator defined as :

$$P(\mathbf{z}) = \left\{ \arg\min_{\mathbf{x}} \|\mathbf{z} - \mathbf{x}\| \text{subject to} \sum_{i} \|\mathbf{x}_{g_i}\|_2 \leq \tau \right\}$$

van den Berg et al. then proposed an algorithm that computes this projection in linear time. Note how similar the iterates given in Equation (1.16) and (1.17) are. Actually, approaches of Fornasier et al. and van den Berg et al. are equivalent and the only point in which they differ is the way the projection are computed. This difference essentially comes from the problem formulation : Fornasier et al. use a penalized problem with known λ while van den Berg consider the constrained optimization problem. Hence, the latter can not directly use the analytic solution of the projection $P(\cdot)$ but have to compute the appropriate value of λ given their value of τ. However, the algorithm they derive for this projection is efficient. Note that if one has some prior knowledge on the value of τ, then this algorithm would be preferable than the one of Fornasier et al.

The other recent work that is noteworthy is the one of Obozinski et al. [36]. They address the problem of SSA within the context of multi-task classification problem. Indeed, they want to recover a common set of covariates that are simultaneously relevant for all classification tasks. For achieving this aim, they propose to solve problem (1.4) with $p = 1$, $q = 2$ and a differentiable loss function such as the logistic loss or the square loss function. One of their most prominent contribution is to have proposed a path-following algorithm that is able to compute the regularization path of the solution with respects to λ. Their algorithm uses a continuation method based on prediction-correction steps. The prediction step is built upon the optimality conditions given in equation (1.11) while the correction step needs an algorithm which solves problem (1.4) but only for a small subset of covariates. The main advantage of such an algorithm is its efficiency for selecting an appropriate value of λ in a model selection context.

1.3 Generic algorithms for large classes of p and q

In this section, we present several algorithms that are able to solve the simultaneous sparse approximation problem for a large classes of p and q. In the first part of the section, we review the M-FOCUSS algorithm of Cotter et al.[**?**] that addresses the case where $0 < p \leq 1$ and $q = 2$. In a second part, we make clear how the penalization term $J_{p,q}(\mathbf{C})$ with $p = 1$ and $1 \leq q \leq 2$, is related to automatic relevance determination. From this novel insight, we then propose an iterative reweighted least-square algorithm that solves this general case. The last part of the section is devoted to the extension of reweighted ℓ_1 algorithms [62, 6] to the SSA problem. We describe how algorithms tailored for solving the SSA problem with $p = 1$ and any q can be re-used for solving a problem with the same q but $p < 1$.

1.3.1 M-FOCUSS algorithm

We first detail how the M-FOCUSS algorithm proposed by Cotter et al. [10] can be derived, then we discuss some of its properties and relation with other works.

Deriving the algorithm

M-FOCUSS is, as far as we know, the first algorithm that has been introduced for solving simultaneous sparse approximation problem. M-FOCUSS addresses the general case where $q = 2$ and $p \leq 1$. This algorithm can be understood as a fixed-point algorithm which can be derived through an appropriate factorization of problem (1.4) objective function gradient.

Indeed, since the partial derivative of $J_{p,2}(\mathbf{C})$ with respects to an entry $c_{m,n}$ is :

$$\begin{aligned} \frac{\partial J_{p,2}(\mathbf{C})}{\partial c_{m,n}} &= \frac{\partial}{\partial c_{m,n}} \sum_i \left(\sum_j c_{i,j}^2 \right)^{p/2} \\ &= p \|c_{m,\cdot}\|_2^{p-2} c_{m,n} \end{aligned} \tag{1.18}$$

the gradient of the objective function writes :

$$-\boldsymbol{\Phi}^t(\mathbf{S} - \boldsymbol{\Phi}\mathbf{C}) + \lambda \mathbf{P}\mathbf{C}$$

where $\mathbf{P} = \text{diag}(p\|c_{i,\cdot}\|_2^{p-2})$. Then, we define the weighting matrix $\mathbf{W}$ as $\mathbf{W} = \text{diag}(p^{-1/2}\|c_{i,\cdot}\|_2^{1-p/2})$. After having replaced $\mathbf{W}^{-2} = \mathbf{P}$ in the above gradient, and after simple algebras, we have the following necessary optimality condition

$$\left((\boldsymbol{\Phi}\mathbf{W})^t(\boldsymbol{\Phi}\mathbf{W}) + \lambda \mathbf{I}\right) \mathbf{W}^{-1}\mathbf{C} = (\boldsymbol{\Phi}\mathbf{W})^t \mathbf{S}.$$

Hence, the optimum solution writes as

$$\mathbf{C}^\star = \mathbf{W} \left((\boldsymbol{\Phi}\mathbf{W})^t(\boldsymbol{\Phi}\mathbf{W}) + \lambda \mathbf{I}\right)^{-1} (\boldsymbol{\Phi}\mathbf{W})^t \mathbf{S}. \tag{1.19}$$

Note that since $\mathbf{W}$ also depends on $\mathbf{C}$, the above closed-form expression of the optimal matrix $\mathbf{C}$ can be interpreted as a fixed-point algorithm. From this insight on $\mathbf{C}$, Cotter et al. suggest the following iterates for solving the simultaneous sparse approximation problem :

$$\mathbf{C}^{(t+1)} = \mathbf{W}^{(t)} \left((\boldsymbol{\Phi}\mathbf{W}^{(t)})^t(\boldsymbol{\Phi}\mathbf{W}^{(t)}) + \lambda \mathbf{I}\right)^{-1} (\boldsymbol{\Phi}\mathbf{W}^{(t)})^t \mathbf{S} \tag{1.20}$$

where $\mathbf{W}^{(t)} = \text{diag}(p^{-1/2}\|c_{i,\cdot}^{(t)}\|_2^{1-p/2})$. Now, since we have :

$$\left((\boldsymbol{\Phi}\mathbf{W}^{(t)})^t(\boldsymbol{\Phi}\mathbf{W}^{(t)}) + \lambda \mathbf{I}\right)^{-1} (\boldsymbol{\Phi}\mathbf{W}^{(t)})^t = (\boldsymbol{\Phi}\mathbf{W}^{(t)})^t \left((\boldsymbol{\Phi}\mathbf{W}^{(t)})(\boldsymbol{\Phi}\mathbf{W}^{(t)})^t + \lambda \mathbf{I}\right)^{-1}$$

Cotter et al. actually consider the following iterative scheme

$$\mathbf{C}^{(t+1)} = \mathbf{W}^{(t)} (\boldsymbol{\Phi}\mathbf{W}^{(t)})^t \left((\boldsymbol{\Phi}\mathbf{W}^{(t)})(\boldsymbol{\Phi}\mathbf{W}^{(t)})^t + \lambda \mathbf{I}\right)^{-1} \mathbf{S} \tag{1.21}$$

Resulting algorithm is detailed in Algorithm 4.

Algorithm 4 M-FOCUSS with $J_{p\leq 1,q=2}$ penalty

Initialize $\mathbf{C}^{(0)}$ to a matrix of $\mathbf{1}$, $t = 1$
while Loop **do**
  $\mathbf{W}^{(t)} \leftarrow \text{diag}(p^{-1/2}\|c_{i,\cdot}^{(t-1)}\|_2^{1-p/2})$
  $\mathbf{A}^{(t)} \leftarrow \mathbf{\Phi}\mathbf{W}^{(t)}$
  $\mathbf{C}^{(t)} \leftarrow \mathbf{W}^{(t)}(\mathbf{A}^{(t)})^t \left(\mathbf{A}^{(t)}(\mathbf{A}^{(t)})^t + \lambda\mathbf{I}\right)^{-1}\mathbf{S}$
  $t \leftarrow t + 1$
  if stopping condition is satisfied **then**
    Loop = 0
  end if
end while

Discussing M-FOCUSS

Rao et al [40] has provided an interesting interpretation on the FOCUSS algorithm for a single signal approximation which can be readily extend to M-FOCUSS. Indeed, M-FOCUSS can be viewed as an iterative reweighted least-square algorithm. Indeed, From the updating equation of $\mathbf{C}^{(t)}$ (see Equation (1.20)) , we can note that $\mathbf{C}^{(t)} = \mathbf{W}^{(t)}\mathbf{Z}^{(t)}$ where $\mathbf{Z}$ is the minimizer of

$$\frac{1}{2}\|\mathbf{S} - \mathbf{\Phi}\mathbf{W}^{(t)}\mathbf{Z}\|_F^2 + \frac{\lambda}{2}\|\mathbf{Z}\|_F^2$$

and thus $\mathbf{C}^{(t)}$ can be understood as the minimizer of

$$\frac{1}{2}\|\mathbf{S} - \mathbf{\Phi}\mathbf{C}\|_F^2 + \frac{\lambda}{2}\|(\mathbf{W}^{(t)})^{-1}\mathbf{C}\|_F^2.$$

This above equation makes clear the relation between sparse approximation problem and iterative reweighted least-square. Such an connection has already been highlighted in other context. Indeed, while sparsity is usually induced by using ℓ_1 norm penalty, it has been proved that solving the problem in which the ℓ_1 norm has been replaced by an adaptive ℓ_2 norm leads to equivalent solutions [12, 41, 24].

Despite this nice insight, M-FOCUSS presents an important issue. Indeed, the updates proposed by Cotter et al. are not guaranteed to converge to a local minimum of the problem (if the problem is not convex $p < 1$) or to the global minimum of the convex problem ($p = 1$). Their algorithm presents several fixed-points since when a row of $\mathbf{C}$ is equal to 0, it stays at 0 at the next iteration. Although such a point may be harmless if the algorithm is initialized with a "good" starting point, it is nonetheless an undesirable point when solving a convex problem. At the contrary, the M-BCD and the Landweber iteration based algorithms do not suffer from the presence of such fixed points. However, in the M-FOCUSS algorithm, this pathological behavior can be handled by introducing a smoothing term ε in the weight so that the updated weight becomes

$$\mathbf{W}^{(t)} = \text{diag}\left(p^{-1/2}\left(\|c_{i,\cdot}^{(t)}\| + \epsilon\right)\right)^{1-p/2}$$

where $\varepsilon > 0$. The use of ε avoids a given weight to be at zero and consequently it avoids the related $c_{i,\cdot}$ to stay permanently at zero. If we furthermore note that M-FOCUSS is not more than an iterative reweighted least-square, then according to the very recent works of Daubechies et al. [12] and Chartrand et al. [7], it seems justified to iterate the M-FOCUSS algorithm using decreasing value of ε. In our numerical experimental, we will consider the M-FOCUSS algorithm with fixed and decreasing value of ε.

1.3.2 Automatic relevance determination approach

In this section, we focus on the relaxed optimization problem given in (1.4) with the general penalization $J_{p,q}(\mathbf{C})$. Our objective here is to clarify the connection between such a form of penalization and the automatic relevance determination of $\mathbf{C}$'s rows, which has been the keystone of the Bayesian approach of Wipf et al [59]. We will show that for some values of p and q, the mixed-norm $J_{p,q}(\mathbf{C})$ has an equivalent variational formulation. Then, by using this novel formulation in problem (1.4), instead of $J_{p,q}(\mathbf{C})$, we exhibit the relation between our sparse approximation problem and ARD. We then propose an iterative reweighted least-square approach for solving this resulting ARD problem.

Exhibiting the relation with ARD

For this purpose, we first consider the following formulation of the simultaneous sparse approximation problem :

$$\min_{\mathbf{C}} \frac{1}{2}\|\mathbf{S}-\mathbf{\Phi C}\|_F^2 + \lambda' \left(J_{p,q}(\mathbf{C})\right)^{\frac{2}{p}}. \tag{1.22}$$

In the convex case (for $p \geq 1$ and $q \geq 1$), since the power function is strictly monotonically increasing, problems (1.4) and (1.22) are equivalent, in the sense that for a given value λ', there exists a λ so that solutions of the two problems are equivalent. When $J_{p,q}$ is not convex, this equivalence does not strictly apply. However, due to the nature of the problem, the problem formulation (1.22) is more convenient for exhibiting the relation with ARD.

Let us introduce the key lemma that allows us to derive the ARD-based formulation of the problem. This lemma gives a variational form of the $\ell_{p,q}$ norm of a sequence $\{a_{t,k}\}$.

Lemma 1 *if $s > 0$ and $\{a_{t,k} : k \in \mathbb{N}_n, t \in \mathbb{N}_T\} \in \mathbb{R}$ such that at least one $a_{t,k} > 0$, then*

$$\min_{\mathbf{d}} \left\{ \sum_{t,k} \frac{|a_{t,k}|^2}{d_{t,k}} : d_{t,k} \geq 0, \sum_k \left(\sum_t d_{t,k}^{1/s} \right)^{\frac{s}{r+s}} \leq 1 \right\} = \left(\sum_k \left(\sum_t |a_{t,k}|^q \right)^{\frac{p}{q}} \right)^{\frac{2}{p}} \tag{1.23}$$

where $q = \frac{2}{s+1}$ and $p = \frac{2}{s+r+1}$. Furthermore, at optimality, we have :

$$d_{t,k}^{\star} = \frac{|a_{t,k}|^{\frac{2s}{s+1}} \left(\sum_u |a_{u,k}|^{\frac{2}{s+1}} \right)^{\frac{r}{s+r+1}}}{\left(\sum_v \left(\sum_u |a_{u,v}|^{\frac{2}{s+1}} \right)^{\frac{s+1}{s+r+1}} \right)^{r+s}} \tag{1.24}$$

Proof : the proof has been post-poned to the appendix.

According to this lemma, the $\ell_{p,q}$ norm of a sequence can be computed through a minimization problem. Hence, applying this lemma to $\left(J_{p,q}(\mathbf{C})\right)^{\frac{2}{p}}$ by defining $a_{t,k} = c_{t,k}$, we get a variational form of the penalization term. We can also note that the mixed-norm on the matrix coefficients has been transformed to a mixed-norm on weight matrix $\mathbf{d}$. Plugging the above variational formulation of the penalization term in problem (1.22) yields to the following equivalent problem :

$$\begin{array}{ll} \min\limits_{\mathbf{C},\mathbf{d}} & \frac{1}{2}\|\mathbf{S}-\mathbf{\Phi C}\|_F^2 + \lambda \sum_{t,k} \frac{c_{t,k}^2}{d_{t,k}} \\ \text{s.t.} & \sum_k \left(\sum_t d_{t,k}^{1/s} \right)^{\frac{s}{r+s}} \leq 1 \\ & d_{t,k} \geq 0 \qquad \forall t,k \end{array} \tag{1.25}$$

This problem is the one which makes clear the automatic relevance determination interpretation of the original formulation (1.4). Indeed, we have transformed problem (1.4) into a problem with a smooth objective function at the expense of some additional variables $d_{t,k}$. These parameters $d_{t,k}$ actually aim at determining the relevance of each element of $\mathbf{C}$. Indeed, in the objective function, each squared-value $c_{t,k}$ is now inversely weighted by a coefficient $d_{t,k}$. By taking the convention that $\frac{x}{0} = \infty$ if $x \neq 0$ and 0 otherwise, the objective value of the optimization problem becomes finite only if $d_{t,k} = 0$ for $c_{t,k}^2 = 0$. Then the smaller $d_{t,k}$ is, the smaller the $c_{t,k}$ norm should be. Furthermore, optimization problem (1.25) also involves some constraints on $\{d_{t,k}\}$. These constraints impose the matrix $\mathbf{d}$ to have positive elements and to be so that its $\ell_{\frac{1}{r+s},\frac{1}{s}}$ mixed-norm is smaller than 1. Note that this mixed-norm on $\mathbf{d}$ plays an important role since it induces the row-norm sparsity on $\mathbf{C}$. According to the relation between p, r and s, for $p < 1$, we also have $r + s > 1$, making the $\ell_{\frac{1}{r+s},\frac{1}{s}}$ non-differentiable with respect to the first norm. Such singularities favor row-norm sparsity of the matrix $\mathbf{d}$ at optimality, inducing row-norm sparsity of $\mathbf{C}$. As we have noted above, when a row-norm of $\mathbf{d}$ is equal to 0, the corresponding row-norm of $\mathbf{C}$ should also be equal to 0 which means that the corresponding element of the dictionary is "irrelevant" for the approximation of all signals. Problem (1.25) proposes an equivalent formulation of problem (1.4) for which the row-diversity measure has been transformed in another penalty function owing to an ARD formulation. The trade-off between convexity of the problem and the sparsity of the solution has been transferred from p and q to r and s.

Algorithm 5 Iterative Reweighted Least-Square for addressing $J_{1,1\leq q\leq 2}$ penalty

1: Initialize $\mathbf{d}^{(0)}$ to a strictly positive matrix, $t = 1$
2: **while** Loop **do**
3: $\quad\mathbf{C}^{(t)} \leftarrow$ solution of problem (1.26) with fixed $\mathbf{d} = \mathbf{d}^{(t-1)}$ as given by Equation (1.27)
4: $\quad\mathbf{d}^{(t)} \leftarrow$ solution of problem (1.26) with fixed $\mathbf{C} = \mathbf{C}^{(t)}$ as given by Equation (1.28)
5: $\quad t \leftarrow t + 1$
6: $\quad$**if** stopping condition is satisfied **then**
7: $\quad\quad$Loop $= 0$
8: $\quad$**end if**
9: **end while**

From a Bayesian perspective, we can interpret the mixed-norm on $\mathbf{d}$ as the diagonal term of the covariance matrix of a Gaussian prior over the row-norm on $\mathbf{C}$ distribution. This is typically the classical Bayesian Automatic Relevance Determination approach as proposed for instance by Tipping [50]. This novel insight on the ARD interpretation of $J_{p,q}(\mathbf{C})$ clarifies the connection between the M-FOCUSS algorithm of Cotter et al. [10] and the Multiple Sparse Bayesian Learning (M-SBL) algorithm of Wipf et al. [59] for any value of $p < 1$. In their previous works, Wipf et al. have proved that these two algorithms were related when $p \approx 0$. Here, we refine their result by enlarging the connection to other values of p by showing that both algorithms actually solve a problem with automatic relevance determination on the row-norm of $\mathbf{C}$.

1.3.3 Solving the ARD formulation for $p = 1$ and $1 \leq q \leq 2$

Herein, we propose a simple iterative algorithm for solving problem (1.25) for $p = 1$ and $1 \leq q \leq 2$. This algorithm, named as M-EM$_q$, is based on an iterative-reweighted least squares where the weights are updated according to equation (1.24). Thus, it can be seen as an extension of the M-FOCUSS algorithm of Cotter et al. for $q \leq 2$. Note that we have restricted ourselves to $p = 1$ since we will show in the next section that the case $p < 1$ can be handled using another reweighted scheme.

Since $p = 1$, thus $s + r = 1$, the problem we are considering is :

$$\begin{array}{ll}\min\limits_{\mathbf{C},\mathbf{d}} & \sum_k \frac{1}{2}\left(\|\mathbf{s}_k - \mathbf{\Phi} c_{\cdot,k}\|_2^2 + \lambda \sum_t \frac{c_{t,k}^2}{d_{t,k}}\right) = J_{obj}(\mathbf{C},\mathbf{d}) \\ \text{s.t.} & \sum_k \left(\sum_t d_{t,k}^{1/s}\right)^s \leq 1 \\ & d_{t,k} \geq 0\end{array} \tag{1.26}$$

Since, we consider that $1 \leq q \leq 2$ hence $0 \leq s \leq 1$ [2], this optimization problem is convex with a smooth objective function. We propose to address this problem through a block-coordinate algorithm which alternatively solves the problem with respects to $\mathbf{C}$ with the weight $\mathbf{d}$ being fixed, and keeping $\mathbf{C}$ fixed and computing the optimal weight $\mathbf{d}$. The resulting algorithm is detailed in Algorithm 5.

Owing to the problem structure, step 4 and 5 of this algorithm has a simple closed form. Indeed, for fixed $\mathbf{d}$, each vector $c_{\cdot,k}^{(t)}$ at iteration t is given by :

$$c_{\cdot,k}^{(t)} = \left(\mathbf{\Phi}^t\mathbf{\Phi} + 2\lambda\mathbf{D}_k^{(t-1)}\right)^{-1}\mathbf{\Phi}^t\mathbf{s}_k \tag{1.27}$$

where $\mathbf{D}_k^{(t-1)}$ is a diagonal matrix of entries $1/d_{\cdot,k}^{(t-1)}$. In a similar way, for fixed $\mathbf{C}$, step 5 boils down in solving problem (1.23). Hence, by defining $a_{t,k} = c_{t,k}^{(t)}$, we also have a closed-form for $\mathbf{d}^{(t)}$ as

$$d_{t,k}^{(t)} = \frac{|a_{t,k}|^{\frac{2s}{s+1}}\left(\sum_u |a_{u,k}|^{\frac{2}{s+1}}\right)^{\frac{1-s}{2}}}{\sum_v \left(\sum_u |a_{u,v}|^{\frac{2}{s+1}}\right)^{\frac{s+1}{2}}} \tag{1.28}$$

Note that similarly to the M-FOCUSS algorithm, this algorithm can also be seen as an iterative reweighted least-square approach or as an Expectation-Minimization algorithm, where the weights are

2. for $s = 0$, we have explicitly used the sup norm of vector $d_{\cdot,k}$ in the constraints.

defined in equation (1.28). Furthermore, it can be shown that if the weights $\mathbf{d}$ are initialized to non-zero values then at each loop involving step 4 and 5, the objective value of problem (1.26) decreases. Hence, since the problem is convex, our algorithm should converge towards the global minimum of the problem.

Theorem 3 *If the objective value of problem (1.26) is strictly convex (for instance when $\mathbf{\Phi}$ is full-rank), and if for the t-th loop, after the step 5, we have $\mathbf{d}^{(t)} \neq \mathbf{d}^{(t-1)}$, then the objective value has decreased, i.e :*

$$J_{obj}(\mathbf{C}^{(t+1)}, \mathbf{d}^{(t)}) < J_{obj}(\mathbf{C}^{(t)}, \mathbf{d}^{(t)}) < J_{obj}(\mathbf{C}^{(t)}, \mathbf{d}^{(t-1)}).$$

Proof : The right inequality $J_{obj}(\mathbf{C}^{(t)}, \mathbf{d}^{(t)}) < J_{obj}(\mathbf{C}^{(t)}, \mathbf{d}^{(t-1)})$ comes from $\mathbf{d}^{(t)}$ being the optimal value of the optimization problem resulting from step 5 of algorithm (5). The strict inequality yields from the hypothesis that $\mathbf{d}^{(t)} \neq \mathbf{d}^{(t-1)}$ and from the strict convexity of the objective function. A similar reasoning allows us to derive the left inequality. Indeed, since $\mathbf{C}^{(t)}$ is not optimal with respects to $\mathbf{d}^{(t)}$ for the problem given by step (4), invoking the strict convexity of the associated objective function and optimality of $\mathbf{C}^{(t+1)}$ concludes the proof.

As stated by the above theorem, the decrease in objective value is actually guaranteed unless, the algorithm get stuck in some fixed points (*e.g* all the elements of $\mathbf{d}$ being zero expect for one entry $\{t_1, k_1\}$). In practice, we have experienced, by comparing for $q = 2$ with the M-BCD algorithm, that if $\mathbf{d}$ is initialized to non-zero entries, algorithm (5) converges to the global minimum of problem (1.26). Numerical experiments will illustrate this point.

1.3.4 Iterative reweighted $\ell_1 - \ell_q$ algorithms for $p < 1$

This section introduces an iterative reweighted M-Basis Pursuit (IrM-BP) algorithm and proposes a way for setting the weight. Through this approach, we are able to provide an iterative algorithm which solves problem (1.4) with $p < 1$ and $1 \leq q \leq 2$.

Iterative reweighted algorithm

Recently, several works have advocated that sparse approximations can be recovered through iterative algorithms based on a reweighted ℓ_1 minimization [62, 6, 7, 58, 20]. Typically, for a single signal case, the idea consists in iteratively solving the following problem

$$\min_{\mathbf{c}} \frac{1}{2}\|\mathbf{s} - \mathbf{\Phi c}\|_2^2 + \lambda \sum_i z_i |c_i|$$

where z_i are some positive weights, and then to update the positive weights z_i according to the solution $\mathbf{c}^\star$ of the above problem. Besides providing empirical evidences that reweighted ℓ_1 minimization yields to sparser solutions than a simple ℓ_1 minimization, the above cited works theoretically support such a claim. These results for the single signal approximation case suggest that in the simultaneous sparse approximation problem, reweighted M-Basis Pursuit would also lead to sparser solutions than the classical M-Basis Pursuit.

The iterative reweighted M-Basis Pursuit algorithm is defined as follows. We iteratively construct a sequence $\mathbf{C}^{(t)}$ defined as

$$\mathbf{C}^{(t)} = \arg\min_{\mathbf{C}} \frac{1}{2}\|\mathbf{S} - \mathbf{\Phi C}\|_F^2 + \lambda \sum_i z_i^{(t)} \|c_{i,\cdot}\|_q \tag{1.29}$$

where the positive weight vector $\mathbf{z}^{(t)}$ depends on the previous iterate $\mathbf{C}^{(t-1)}$. For $t = 1$, we typically define $\mathbf{z}^{(1)} = 1$ and for $t > 1$, we will consider the following weighting scheme

$$z_i^{(t)} = \frac{1}{(\|c_{i,\cdot}^{(t-1)}\|_q + \varepsilon)^r} \quad \forall i \tag{1.30}$$

where $\{c_{i,\cdot}^{(t-1)}\}$ is the i-th row of $\mathbf{C}^{(t-1)}$, r a user-defined positive constant and ε a small regularization term that prevents from having an infinite regularization term for $c_{i,\cdot}$as soon as $c_{i,\cdot}^{(t-1)}$ vanishes. This is a classical trick that has been used for instance by Candès et al. [6] or Chartrand et al. [41]. Note

Algorithm 6 Majorization-Minimization algorithm leading to iterative reweighted ℓ_1 for addressing $J_{p\leq 1,q}$ penalty.

Initialize $z_i^{(1)} = 1$, $r = 1 - p$, $t = 1$
while Loop **do**
  $\mathbf{C}^{(t)} \leftarrow$ solution of problem (1.29)
  $t \leftarrow t + 1$
  $z_i^{(t)} \leftarrow \frac{1}{(\|c_{i,\cdot}^{(t-1)}\|_q + \varepsilon)^r} \quad \forall i$
  if stopping condition is satisfied **then**
    Loop = 0
  end if
end while

that for any positive weight vector $\mathbf{z}$, problem (1.29) is a convex problem that does not present local minima. Furthermore, for $1 \leq q \leq 2$, it can be solved by our block-coordinate descent algorithm or by our M-EM$_q$ given in Algorithm 5, by simply replacing λ with $\lambda_i = \lambda \cdot z_i$. This reweighting scheme is similar to the *adaptive lasso* algorithm of Zou et al. [62] but uses a larger number of iterations and addresses the simultaneous approximation problem.

Connections with Majorization-Minimization algorithm

This IrM-BP algorithm can be interpreted as an algorithm for solving an approximation of problem (1.4) when $0 < p < 1$ and $1 \leq q \leq 2$. Indeed, similarly to the reweighted ℓ_1 scheme of Candès et al. [6] or the one-step reweighted lasso of Zou et al. [63], this algorithm falls into the class of majorize-minimize (MM) algorithms [27]. MM algorithms consists in replacing a difficult optimization problem with a easier one, for instance by linearizing the objective function, by solving the resulting optimization problem and by iterating such a procedure.

The connection between MM algorithms and our reweighted scheme can be made through linearization. Let us first define $J_{p,q,\varepsilon}(\mathbf{C})$ as an approximation of the penalty term $J_{p,q}(\mathbf{C})$:

$$J_{p,q,\varepsilon}(\mathbf{C}) = \sum_i g\left(\|c_{i,\cdot}\|_q + \varepsilon\right)$$

where $g(\cdot) = |\cdot|^p$. Since $g(\cdot)$ is concave for $0 < p < 1$, a linear approximation of $J_{p,q,\varepsilon}(\mathbf{C})$ around $\mathbf{C}^{(t-1)}$ yields to the following majorizing inequality

$$J_{p,q,\varepsilon}(\mathbf{C}) \leq J_{p,q,\varepsilon}(\mathbf{C}^{(t-1)}) + \sum_i \frac{p}{\left(\|c_{i,\cdot}^{(t-1)}\|_q + \varepsilon\right)^{1-p}} (\|c_{i,\cdot}\|_q - \|c_{i,\cdot}^{(t-1)}\|_q)$$

then for the minimization step, replacing in problem (1.4) $J_{p,q}$ with the right part of the inequality and dropping constant terms lead to our optimization problem (1.29) with appropriately chosen z_i and r. Note that for the weights given in equation (1.30), $r = 1$ corresponds to the linearization of a log penalty $\sum_i \log(\|c_{i,\cdot}\| + \varepsilon)$ whereas setting $r = 1 - p$ corresponds to a ℓ_p penalty ($0 < p < 1$).

MM algorithms have already been considered in optimization problems with sparsity-inducing penalties. For instance, an MM approach has been used by Figueiredo et al. [18] for solving a least-square problem with a ℓ_p sparsity-inducing penalty, whereas Candès et al. [6] have addressed the problem for exact sparse signal recovery. In a context of simultaneous approximation, Simila [43, 44] has also considered MM algorithms while approximating the non-convex penalty with a quadratic term. In the light of all these previous works, what we have exposed here is merely an extension of iterative reweighted ℓ_1 algorithm to simultaneous sparse approximation problem. Through this iterative scheme, we can solve problem (1.4) with $p < 1$ and any q provided that we have an algorithm that is able to solve problem (1.4) with $p = 1$ and the desired q.

Analyzing the convergence of the sequence $\mathbf{C}^{(t)}$ towards the global minimizer of problem (1.4) is a challenging issue. Indeed, several points make a formal proof of convergence difficult. At first, in order to avoid a row $c_{i,\cdot}^{(t)}$ to be permanently at zero, we have introduced a smoothing term ε, thus we are only solving a ε-approximation of problem (1.4). Furthermore, the penalty we use is non-convex, thus using

Algorithm 7 Group bridge Lasso based on Iterative reweighted ℓ_1 for addressing $J_{p\leq 1,q}$ penalty.

Initialize $\tilde{\mathbf{c}}^{(0)}$, $\tau = \frac{1-p}{p}\lambda^{-\frac{1}{p-1}}$, $t = 1$
while Loop **do**
  $z_i^{(t)} \leftarrow \tau^{-p}\|\tilde{\mathbf{c}}_{g_i}^{(t-1)}\|_1^p \left(\frac{1-p}{p}\right)^p$
  $\tilde{\mathbf{c}}^{(t)} \leftarrow \arg\min_{\tilde{\mathbf{c}}} \frac{1}{2}\|\tilde{\mathbf{s}} - \tilde{\mathbf{\Phi}}\tilde{\mathbf{c}}\|^2 + \sum_i (z_i^{(t)})^{1-1/p}\|\tilde{\mathbf{c}}_{g_i}\|_1$
  $t \leftarrow t + 1$
  if stopping condition is satisfied **then**
    Loop = 0
  end if
end while

a monotonic algorithm like a MM approach which decreases the objective value at each iteration, can not guarantee convergence to the global minimum of our ε-approximate problem. Hence, due to these two major obstacles, we have left this convergence proof for future works. Note however that few works have addressed the convergence issue of reweighted ℓ_1 or ℓ_2 algorithms for single sparse signal recovery. Notably, we can mention the recent work of Daubechies et al. [12] which provide a convergence proof of iterative reweighted least square for exact sparse recovery. In the same flavor, Foucart et al. [20] have proposed a tentative of rigorous convergence proof for reweighted ℓ_1 sparse signal recovery. Although, we do not have any rigorous proof of convergence, in practice, we will show that the reweighted algorithm provides good sparse approximations.

As already noted by several authors [6, 41, 12], ε plays a major role in the quality of the solution. In the experimental results presented below, we have investigated two methods for setting ε : the first one is to set it to a fixed value $\varepsilon = 0.001$, the other one, denoted as an annealing approach, consists in gradually decreasing ε after having solved problem (1.29).

Connection with group bridge Lasso

Recently, in a context of variable selection through *Group Lasso*, Huang et al. [26] have proposed an algorithm for solving problem (1.5) with $p < 1$ and $q = 1$:

$$\min_{\tilde{\mathbf{c}}} \frac{1}{2}\|\tilde{\mathbf{s}} - \tilde{\mathbf{\Phi}}\tilde{\mathbf{c}}\|^2 + \lambda \sum_i \|\tilde{\mathbf{c}}_{g_i}\|_1^p. \tag{1.31}$$

Instead of directly addressing this optimization problem, they have shown that the above problem is equivalent to the minimization problem :

$$\begin{array}{ll} \min_{\tilde{\mathbf{c}},\mathbf{z}} & \frac{1}{2}\|\tilde{\mathbf{s}} - \tilde{\mathbf{\Phi}}\tilde{\mathbf{c}}\|^2 + \sum_i \frac{\|\tilde{\mathbf{c}}_{g_i}\|_1}{z_i^{1/p-1}} + \tau \sum_i z_i \\ \text{st.} & z_i \geq 0 \qquad \forall i. \end{array} \tag{1.32}$$

where τ is a regularization parameter. Here equivalence is understood as follows : $\tilde{\mathbf{c}}^\star$ minimizes equation (1.31) if and only if the pair $(\tilde{\mathbf{c}}^\star, \mathbf{z}^\star)$ minimizes Equation (1.31) with an appropriate value of τ. This value of τ will be made clear in the sequel.

The connection between iterative reweighted ℓ_1 approach and the algorithm proposed by Huang et al. comes from the way problem (1.32) has been solved. Indeed, Huang et al. have considered a block-coordinate approach which consists in minimizing the problem with respects to $\mathbf{z}$ and then, after having plugged the optimal $\mathbf{z}$ in the equation (1.32) in minimizing the problem with respects to $\tilde{\mathbf{c}}$. For the first step, the optimal $\mathbf{z}$ is derived by minimizing the convex problem

$$\begin{array}{ll} \min_{\mathbf{z}} & \sum_i \frac{\|\tilde{\mathbf{c}}_{g_i}\|_1}{z_i^{1/p-1}} + \tau \sum_i z_i \\ \text{st.} & z_i \geq 0 \qquad \forall i. \end{array} \tag{1.33}$$

Using Lagrangian theory, simple algebras yield to a closed-form solution of the optimal $\mathbf{z}$:

$$z_i^\star = \tau^{-p}\|\tilde{\mathbf{c}}_{g_i}\|_1^p \left(\frac{1-p}{p}\right)^p$$

Plugging these z_i's in Equation (1.32) and using τ such that $\lambda = \left(\frac{1-p}{p\tau}\right)^{p-1}$ proves the equivalence of the problems (1.31) and (1.32). The relation with iterative reweighted ℓ_1 is made clear in the second step of the block-coordinate descent. Indeed, in problem (1.32), since we only optimize with respects to $\tilde{\mathbf{c}}$ with fixed z_i, this step is equivalent to

$$\min_{\tilde{\mathbf{c}}} \quad \frac{1}{2}\|\tilde{\mathbf{s}} - \tilde{\mathbf{\Phi}}\tilde{\mathbf{c}}\|^2 + \sum_i \frac{\|\tilde{\mathbf{c}}_{g_i}\|_1}{z_i^{1/p-1}} \tag{1.34}$$

which is clearly a weighted ℓ_1 problem. The full iterative reweighted algorithm is detailed in Algorithm (7). Note that although the genuine work on Group bridge Lasso only addresses the case $q = 1$ (because of theoretical developments considered in the paper), the algorithm proposed by Huang et al. can be applied to any choice of q provided that one is able to solve the related $\ell_1 - \ell_q$ problem (for instance using the algorithm described in Section 1.3).

1.4 Specific case algorithms

In this section, we survey some algorithms that provide solutions of the SSA problem (1.4) when $p = 0$. These algorithms are different in nature and provide different qualities of approximation. The first two algorithms we detail are greedy algorithms which have the flavor of Matching Pursuit [33]. The third one is based on a sparse bayesian learning and is related to iterative reweighted ℓ_1 algorithm.

1.4.1 S-OMP

Simultaneous Orthogonal Matching Pursuit (S-OMP) [53] is an extension of the Orthogonal Matching Pursuit [37, 52] algorithm to the multiple approximation problem.

S-OMP is a greedy algorithm which is based on the idea of selecting, at each iteration, an element of the dictionary and on building all signal approximations as the projection of the signal matrix $\mathbf{S}$ on the span of these selected dictionary elements. Since, according to problem (1.2), one looks for a for dictionary elements that can simultaneously provide good approximation of all signals, Tropp et al. have suggested to select the dictionary element that maximizes the sum of absolute correlation between the basis element and the signal residuals. This greedy selection criterion is formalized as follows :

$$\max_i \sum_{k=1}^{L} |\langle \mathbf{s}_k - P^{(t)}(\mathbf{s}_k), \mathbf{\Phi}_i \rangle|$$

where $P^{(t)}(\cdot)$ is the projection operator on the span of the t selected dictionary elements. While conceptually very simple, this algorithm which details are given in Algorithm (8), provide theoretical guarantees about the quality of its approximation. For instance, Tropp et al. [53] have shown that if the algorithm is stopped when T dictionary elements have been selected then, under some mild conditions on the matrix $\mathbf{\Phi}$, there exists a upper bound on $\|\mathbf{S} - \mathbf{\Phi}\mathbf{C}^{(T)}\|_F$. This upper bound depends on $\|\mathbf{S} - \mathbf{\Phi}\mathbf{C}^{\star}\|_F$ with $\mathbf{C}^{\star}$ being the solution of problem (1.2) and multiplicative constant which is a function of T, L and the coherence of $\mathbf{\Phi}$ [53]. In this sense, S-OMP can be understood as an algorithm that approximately solves problem (1.4) with $p = 0$.

1.4.2 M-CosAmp

Very recently, Needell and al. have developed a novel signal reconstruction algorithm known as CoSAmp [35]. This algorithm uses ideas from orthogonal matching pursuit but also takes advantage of ideas from the compressive sensing literature. This novel approach provides stronger theoretical guarantees on the quality of signal approximation. The main particularities of CosAmp are the following. Firstly, unlike Orthogonal matching pursuit, CosAmp selects many elements of the dictionary at each iteration, the ones that have the largest correlation with respects to the signal residual. These novel dictionary elements are thus incorporate into the set of current dictionary elements. The other specificity of CosAmp is a pruning dictionary element step after signal estimation. This step is justified by the fact that the signal to approximate is supposed to be sparse.

This CosAmp algorithm have been extended so as to handle signals with specific structures such as *block sparse* signal Extension to jointly sparse approximation has also been provided by Duarte et al.

Algorithm 8 S-OMP algorithm. T number of dictionary elements to select

1: Initialize : $t \leftarrow 0, \mathbf{R}^{(t)} \leftarrow \mathbf{S}, \Omega \leftarrow \emptyset, \text{Loop} \leftarrow 1$
2: **while** Loop **do**
3: $\mathbf{E} \triangleq \mathbf{\Phi}^t \mathbf{R}^{(t)}$
4: $i \leftarrow \arg\max_k \sum_j |e_{j,k}|$
5: $\Omega \leftarrow \Omega \cup i$
6: $t \leftarrow t + 1$
7: $\mathbf{C}^{(t)} \leftarrow (\mathbf{\Phi}_\Omega^t \mathbf{\Phi}_\Omega)^{-1} \mathbf{\Phi}_\Omega^t \mathbf{S}$
8: $\mathbf{R}^{(t)} = \mathbf{S} - \mathbf{\Phi}\mathbf{C}^{(t)}$
9: **if** t= T **then**
10: Loop = 0
11: **end if**
12: **end while**

Algorithm 9 M-CosAmp algorithm. T number of dictionary elements to select.

1: Initialize $\mathbf{C}^{(0)} = \mathbf{0}, \textit{Loop} \leftarrow, \mathbf{R}^{(t)} = \mathbf{S}, \Omega_{res} \leftarrow \emptyset$
2: **while** Loop **do**
3: $\mathbf{E} \triangleq \mathbf{\Phi}^t \mathbf{R}^{(t)}$
4: $e_i = \|\mathbf{E}_{i,.}\|_2^2 \qquad i = 1, \cdots, M$
5: $\Omega_{res} \leftarrow \text{supp}(\mathbf{e}, 2T)$
6: $\Omega \leftarrow \Omega \cup \Omega_{res}$
7: $\hat{\mathbf{C}} \leftarrow (\mathbf{\Phi}_\Omega^t \mathbf{\Phi}_\Omega)^{-1} \mathbf{\Phi}_\Omega^t \mathbf{S}$
8: $v_i = \|c_{i,.}\|_2^2 \qquad i = 1, \cdots, M$
9: $\Omega \leftarrow \text{supp}(\mathbf{v}, T)$
10: $t \leftarrow t + 1$
11: $\mathbf{C}^{(t)} = \mathbf{0}$
12: $\mathbf{C}_\Omega^{(t)} = \hat{\mathbf{C}}_\Omega$
13: $\mathbf{R}^{(t)} = \mathbf{S} - \mathbf{\Phi}\mathbf{C}^{(t)}$
14: **if** stopping condition is satisfied **then**
15: Loop = 0
16: **end if**
17: **end while**

[13]. Such an extension of the CosAmp algorithm to simultaneous sparse approximation is detailed in Algorithm 9. Note that lines 4-5 of the algorithm estimate the dictionary elements which have the largest correlations over all the signals while lines 8-9 prune the set of current dictionary elements so as to keep the matrix $\mathbf{C}$ row-sparse. This algorithm is denoted as M-CosAmp in the experimental section.

1.4.3 Sparse Bayesian Learning and Reweighted algorithm

The approach proposed by Wipf et al. [59], denoted in the sequel as Multiple-Sparse Bayesian Learning (M-SBL), for solving the sparse simultaneous approximation is somewhat related to the optimization problem in equation (1.4) but from a very different perspective. Indeed, if we consider that the approaches described in Section 1.3 are equivalent to a Maximum a Posteriori estimation procedures, then Wipf et al. have explored a Bayesian model which prior encourages sparsity. In this sense, their approach is related to the relevance vector machine of Tipping et al. [50]. Algorithmically, they proposed an empirical bayesian learning approach based on Automatic Relevance Determination (ARD). The ARD prior over each row they have introduced is

$$p(c_{i,.}; d_i) = \mathcal{N}(0, d_i \mathbf{I}) \qquad \forall i$$

where $\mathbf{d}$ is a vector of non-negative hyperparameters that govern the prior variance of each coefficient matrix row. Hence, these hyperparamaters aim at catching the sparsity profile of the approximation. Mathematically, the resulting optimization problem is to minimize according to $\mathbf{d}$ the following cost function :

$$L \log |\Sigma_t| + \sum_{j=1}^{L} \mathbf{s}_j^t \Sigma_t^{-1} \mathbf{s}_j \tag{1.35}$$

where $\Sigma_t = \sigma^2\mathbf{I} + \mathbf{\Phi}\mathbf{D}\mathbf{\Phi}^t$, $\mathbf{D} = \text{diag}(\mathbf{d})$ and σ^2 a parameter of the algorithm related to the noise level presented in the signals to be approximated. The algorithm is then based on a likelihood maximization which is performed through an Expectation-Minimization approach. Very recently, a very efficient algorithm for solving this problem has been proposed [28]. However, the main drawback of this latter approach is that due to its greedy nature, and as any EM algorithm where the objective function is not convex, the algorithm can be easily stuck in local minima.

Recently, Wipf et al. [56] have proposed some new insights on Automatic Relevance Determination and Sparse Bayesian Learning. They have shown that, for the vector regression case, ARD can be achieved by means of iterative reweighted ℓ_1 minimization. Furthermore, in that paper, they have sketched an extension of such results for matrix regression in which ARD is used for automatically selecting the most relevant covariance components in a dictionary of covariance matrices. Such an extension is more related to learning with multiple kernels in regression as introduced by Girolami et al. [22] or Rakotomamonjy et al. [39] although some connections with simultaneous sparse approximation can be made. Here, we build on the works on Wipf et al. [56] and give all the details about how M-SBL and iterative reweighted M-BP are related.

Recall that the cost function minimized by the M-SBL of Wipf et al. [59] is

$$\mathcal{L}(\mathbf{d}) = L \log |\Sigma_t| + \sum_{j=1}^{L} \mathbf{s}_j^t \Sigma_t^{-1} \mathbf{s}_j \tag{1.36}$$

where $\Sigma_t = \sigma^2\mathbf{I}+\mathbf{\Phi}\mathbf{D}\mathbf{\Phi}^t$ and $\mathbf{D} = \text{diag}(\mathbf{d})$, with $\mathbf{d}$ being a vector of hyperparameters that govern the prior variance of each coefficient matrix row. Now, let us define $g^\star(z)$ as the conjugate function of the concave $\log|\Sigma_t|$. Since, that log function is concave and continuous on $\mathbb{R}_+^M$, according to the scaling property of conjugate functions we have [5]

$$L \cdot \log |\Sigma_t| = \min_{\mathbf{z} \in \mathbb{R}^M} \mathbf{z}^t \mathbf{d} - L g^\star \left(\frac{\mathbf{z}}{L} \right)$$

Thus, the cost function $\mathcal{L}(\mathbf{d})$ in equation (1.36) can then be upper-bounded by

$$\mathcal{L}(\mathbf{d}, \mathbf{z}) \triangleq \mathbf{z}^t \mathbf{d} - L g^\star \left(\frac{\mathbf{z}}{L} \right) + \sum_{j=1}^{L} \mathbf{s}_j^t \Sigma_t^{-1} \mathbf{s}_j \tag{1.37}$$

Hence when optimized over all its parameters, $\mathcal{L}(\mathbf{d}, \mathbf{z})$ converges to a local minima or a saddle point of (1.36). However, for any fixed $\mathbf{d}$, one can optimize over $\mathbf{z}$ and get the tight optimal upper bound. If we denote as $\mathbf{z}^\star$ such an optimal $\mathbf{z}$ for any fixed $\mathbf{d}^\dagger$, since $L \cdot \log|\Sigma_t|$ is differentiable, we have, according to conjugate function properties, the following closed form of $\mathbf{z}^\star$

$$\mathbf{z}^\star = L \cdot \nabla \log |\Sigma_t|(\mathbf{d}^\dagger) = \text{diag}(\mathbf{\Phi}^t \Sigma_t^{-1} \mathbf{\Phi}) \tag{1.38}$$

Similarly to what proposed by Wipf et al., Equations (1.37) and (1.38) suggest an alternate optimization scheme for minimizing $\mathcal{L}(\mathbf{d}, \mathbf{z})$. Such a scheme would consist, after initialization of $\mathbf{z}$ to some arbitrary vector, in keeping $\mathbf{z}$ fixed and in computing

$$\mathbf{d}^\dagger = \arg\min_{\mathbf{d}} \mathcal{L}_z(\mathbf{d}) \triangleq \mathbf{z}^t \mathbf{d} + \sum_{j=1}^{L} \mathbf{s}_j^t \Sigma_t^{-1} \mathbf{s}_j \tag{1.39}$$

then to minimize $\mathcal{L}(\mathbf{d}^\dagger, \mathbf{z})$ for fixed $\mathbf{d}^\dagger$, which can be analytically done according to equation (1.38). This alternate scheme is then performed until convergence to some $\mathbf{d}^\star$.

Owing to this iterative scheme proposed for solving M-SBL, we can now make clear the connection between M-SBL and an iterative reweighted M-BP according to the following lemma. Again this is an extension to the multiple signals case of a Wipf's lemma.

Lemma 2 *The objective function in equation (1.39) is convex and can be equivalently solved by computing*

$$\mathbf{C}^\star = arg\min_{\mathbf{C}} \mathcal{L}_z(\mathbf{C}) = \frac{1}{2}\|\mathbf{S} - \mathbf{\Phi}\mathbf{C}\|_F^2 + \sigma^2 \sum_i z_i^{1/2} \|c_{i,.}\| \tag{1.40}$$

and then by setting

$$d_i = z_i^{-1/2} \|c^\star_{i,\cdot}\| \qquad \forall i$$

Proof 3 *Convexity of the objective function in equation (1.39) is straightforward since it is just a sum of convex functions [4]. The key point of the proof is based on the equality*

$$\mathbf{s}_j^t \Sigma_t^{-1} \mathbf{s}_j = \frac{1}{\sigma^2} \min_{c_{\cdot,j}} \|\mathbf{s}_j - \mathbf{\Phi} c_{\cdot,j}\|_2^2 + \sum_i \frac{c_{i,j}^2}{d_i} \tag{1.41}$$

which proof is given in the appendix. According to this equality, we can upper-bound $\mathcal{L}_z(\mathbf{d})$ with

$$\mathcal{L}_z(\mathbf{d}, \mathbf{C}) = \mathbf{z}^t \mathbf{d} + \sum_j \frac{1}{\sigma^2} \|\mathbf{s}_j - \mathbf{\Phi} c_{\cdot,j}\|_2^2 + \sum_{i,j} \frac{c_{i,j}^2}{d_i} \tag{1.42}$$

The problem of minimizing $\mathcal{L}_z(\mathbf{d}, \mathbf{C})$ is smooth and jointly convex in its parameters $\mathbf{C}$ and $\mathbf{d}$ and thus an iterative coordinate-wise optimization scheme (iteratively optimizing over $\mathbf{d}$ with fixed $\mathbf{C}$ and then optimizing over $\mathbf{C}$ with fixed $\mathbf{d}$) yields to the global minimum. It is easy to show that for any fixed $\mathbf{C}$, the minimal value of $\mathcal{L}_z(\mathbf{d}, \mathbf{C})$ with respects to $\mathbf{d}$ is achieved when

$$d_i = z_i^{-1/2} \|c_{i,\cdot}\| \qquad \forall i$$

Plugging these solutions back into (1.42) and multiplying the the resulting objective function with $\sigma^2/2$ yields to

$$\mathcal{L}_z(\mathbf{C}) = \frac{1}{2} \sum_j \|\mathbf{s}_j - \mathbf{\Phi} c_{\cdot,j}\|_2^2 + \sigma^2 \sum_i z_i^{1/2} \|c_{i,\cdot}\| \tag{1.43}$$

Making the relation between ℓ_2 and Frobenius norms concludes the proof.

Minimizing $\mathcal{L}_z(\mathbf{C})$ boils down to minimize the M-BP problem with an adaptive penalty $\lambda_i = \sigma^2 \cdot z_i^{1/2}$ on each row-norm. This latter point makes the alternate optimization scheme based on equation (1.38) and (1.39) equivalent to our iterative reweighted M-BP for which weights z_i would be given by equation (1.38).

The impact of this relation between M-SBL and iterative reweighted M-BP is essentially methodological. Indeed, its main advantage is that it turns the original M-SBL optimization problem into a serie of convex optimization problems. In this sense, the iterative reweighted algorithm described here, can again be viewed as an application of MM approach for solving problem (1.36). Indeed, we are actually iteratively minimizing a proxy function which has been obtained by majorizing each term of equation (1.36). Owing to this MM point of view, convergence of our iterative algorithm towards a local minimum of equation (1.36) is guaranteed [27]. Convergence for the single signal case using other arguments has also been shown by Wipf et al. [56]. Note that similarly to M-FOCUSS, the original M-SBL algorithm based on EM approach suffers from presence of fixed points (when $d_i = 0$). Hence, such an algorithm is not guaranteed to converge towards a local minimum of (1.36). This is then another argument for preferring IrM-BP.

1.5 Numerical experiments

Some computer simulations have been carried out in order to evaluate the algorithms proposed in the above sections. Results that have been obtained from these numerical studies are detailed in this section.

1.5.1 Experimental set-up

In order to quantify the performance of our algorithms and compare them to other approaches, we have used simulated datasets with different redundancy factors $\frac{M}{N}$, number k of active elements and number L of signals to approximate. The dictionary $\mathbf{\Phi}$ is based on M vectors sampled from the unit hypersphere of $\mathbb{R}^N$. The true coefficient matrix $\mathbf{C}^\star$ has been obtained as follows. The positions of the k non-zero rows in the matrix are randomly drawn. The non-zero coefficients of $\mathbf{C}^\star$ are then drawn from a zero-mean unit variance Gaussian distribution. The signal matrix $\mathbf{S}$ is obtained as in equation (1.1)

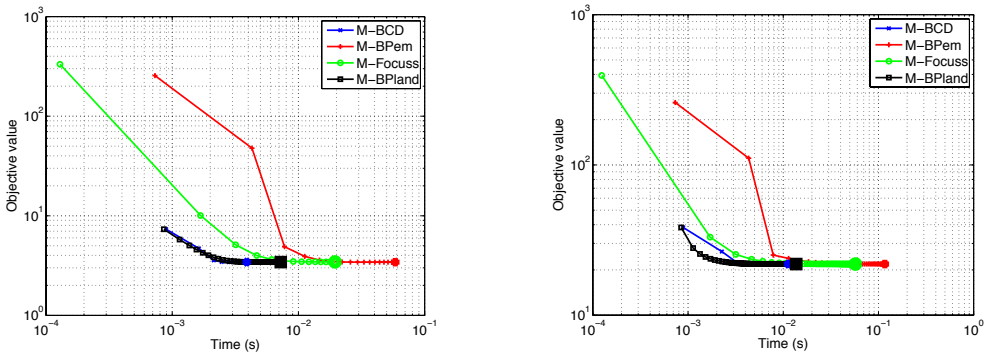

FIGURE 1.1 – Examples of objective value evolution with respects to computational time. Here we have, $M = 128$, $N = 64$, $L = 3$. The number of active elements is : left) $k = 5$. right) $k = 32$. For each curve, the large point corresponds to the objective value at convergence.

with the noise matrix being drawn i.i.d from a zero-mean Gaussian distribution and variance so that the signal-to-noise ratio of each single signal is 10 dB. For a given experiment, when several trials are needed, we only resample the dictionary $\mathbf{\Phi}$ and the additive noise $\mathcal{E}$.

Each algorithm is provided with the signal matrix $\mathbf{S}$ and the dictionary $\mathbf{\Phi}$ and will output an estimate of $\mathbf{C}$. The performance criterion we have considered are the mean-square error between the true and the approximate signals and the sparsity profile of the coefficient matrix that has been recovered. For the latter, we use as a performance criterion the F-measure between the row-support of the true matrix $\mathbf{C}^\star$ and the estimate one $\hat{\mathbf{C}}$. In order to take into account numerical precisions, we have overloaded the row support definition as :

$$\text{rowsupp}(\mathbf{C}) = \{i \in [1 \cdots M] : \|c_{i,\cdot}\| < \mu\}$$

where μ is a threshold coefficient that has been set by default to 0.01 in our experiments. From $\text{rowsupp}(\hat{\mathbf{C}})$ and $\text{rowsupp}(\mathbf{C}^\star)$ respectively the estimated and true sparsity profile, we define :

$$\text{F-measure} = 2 \cdot \frac{|\text{rowsupp}(\hat{\mathbf{C}}) \cap \text{rowsupp}(\mathbf{C}^\star)|}{|\text{rowsupp}(\hat{\mathbf{C}})| + |\text{rowsupp}(\mathbf{C}^\star)|}.$$

Note that the F-measure is equal to 1 when the estimated sparsity profile coincides exactly with the true one.

Regarding the stopping criterion, in the experiments presented below, we have considered convergence of our M-BCD algorithm when the optimality conditions given in equation (1.11) are satisfied up to a tolerance of 0.001 and when all matrix coefficient $c_{i,j}$ variations are smaller than 0.001. This latter condition has also been used as a stopping criterion for the Landweber iteration, M-EM, IrM-BP, M-FOCUSS and M-SBL algorithms. When the annealing approach is in play for IrM-BP and M-FOCUSS, the annealing loop is also stopped under the same condition *e.g* $\max_{i,j} |C_{i,j}^{(t+1)} - C_{i,j}^{(t)}| < 0.001$ where (t) and $(t+1)$ denotes two consecutive annealing solution.

1.5.2 Comparing $\ell_1 - \ell_2$ M-BP problem solvers

In this first experiment, we have compared different algorithms which solves the M-BP problem with $p = 1$ and $q = 2$. Besides our M-BCD and M-EM algorithms, we have also used the M-FOCUSS of Cotter et al. [10] and the approach of Fornasier et al. [19] based on Landweber iterations and denoted in the sequel as M-BPLand. Note that for M-FOCUSS, we have modified the genuine algorithm by introducing a ε parameter, set to 0.001, which helps in avoiding a row-norm of $\mathbf{C}$ to be permanently at 0. For all compared problems, regularization parameter λ has been so as to make the problem equivalent. In this experiment, we have set $\lambda = \frac{1}{5} \max_i \|\phi_i^t \mathbf{S}\|_2$ for problem (1.10).

Figure 1.1 shows two examples of how the objective value of the different algorithms evolves with respects to computational time. We can note that the two iterative reweighted least-square algorithms (M-EM and M-FOCUSS) are the most computationally demanding. Furthermore, we also see that the Landweber iteration approach of Fornasier et al. quickly reduces its objective value but compared to our

TABLE 1.1 – Summary of M-BP solvers comparison. Comparisons have been carried out for two values of k, the number of active elements in the dictionary and have been averaged over 100 trials. Comparison measures are the time needed before convergence, the difference in objective value and the largest coefficient matrix difference. For the two latter measure, the baseline algorithm is considered to be the M-BCD one.

	k=5		
	Time (ms)	Δ ObjVal (10^{-3})	$\|\Delta\mathbf{C}\|_\infty(10^{-3})$
M-BCD	4.2 ± 2.7	-	-
M-EM	57.2 ± 11.7	2.6 ± 2.2	0.8 ± 1.0
M-Focuss	24.1 ± 4.2	3.6 ± 1.8	16.6 ± 1.0
M-BPLand	7.5 ± 1.3	5.3 ± 1.1	0.04 ± 1.1

	k=32		
	Time (ms)	Δ ObjVal (10^{-3})	$\|\Delta\mathbf{C}\|_\infty(10^{-3})$
M-BCD	11.5 ± 2.87	-	-
M-EM	130.9 ± 17.0	23.1 ± 5.9	15.4 ± 8.0
M-Focuss	60.0 ± 6.2	28.8 ± 5.8	38.8 ± 5.4
M-BPLand	14.1 ± 2.7	16.9 ± 3.8	0.41 ± 3.8

M-BCD method, it needs more time before properly converging. Table 1.1 summarizes more accurately the difference between the four algorithms. These results have been averaged over 100 trials and consider two values of k. As comparison criteria, we have considered the computational time before convergence, the difference (compared to the M-BCD algorithm) in objective values and the maximal absolute difference in the coefficient matrix $c_{i,j}$. The table clearly shows that the M-BCD algorithm is clearly faster than M-BPLand (especially when signals to approximate are highly sparse) and the two iterative reweighted least-square approaches. We can also note from the table that, although M-FOCUSS and our M-EM are not provided with a formal convergence proof, these two algorithms seem to empirically converge to the problem global minimum.

1.5.3 Computational performances

We have also empirically assessed the computational complexity of our algorithms (we used $s = 0.2$, thus $q = \frac{5}{3}$ for M-EM and $r = 1$ for IrM-BP). We varied one of the different parameters (dictionary size M, signal dimensionality N) while keeping the others fixed. All matrices $\mathbf{\Phi}$, $\mathbf{C}$ and $\mathbf{S}$ are created as described above. Experiments have been run on a Pentium D-3 GHz with 4 GB of RAM using Matlab code. The results in Figure 1.2, averaged over 20 trials, show the computational complexity of the different algorithms for different experimental settings. Note that we have also experimented on the M-SBL computational performances owing to the code of Wipf et al. [59] and have implemented the M-FOCUSS of Cotter et al. [10], the CosAmp block-sparse approach of Baraniuk et al. [] and the Landweber iteration method of Fornasier et al. [19] [3]. All algorithms need one hyperparameter to be set, for M-SBL and CosAmp, we were able to choose the optimal one since the hyperparameter respectively depends on a known noise level and a known number of active elements in the dictionary. For other algorithms, we have reported the computational complexity for the λ that yields to the best sparsity recovery. Note that our aim here is not give an exact comparison of computational complexity of the algorithms but just to give an order of magnitude of these complexities. Indeed, accurate comparisons are difficult since the different algorithms do not solve the same problem and do not use the same stopping criterion.

We can remark in Figure 1.2 that with respects to the dictionary size, all algorithms present an empirical exponent between 1.3 and 2.7. Interestingly, we have theoretically evaluate the complexity of the M-BCD algorithm as quadratic whereas we measure a sub-quadratic complexity. We suppose that this happens because at each iteration, only the non-optimal $c_{i,.}$'s are updated and thus the number of updates drastically reduces along iterations. We can note that among all approaches that solve the $\ell_1 - \ell_q$ problem (left plots), M-BCD, Landweber iteration approach and M-CosAmp have similar complexity

3. All the implementations are included in the toolbox.

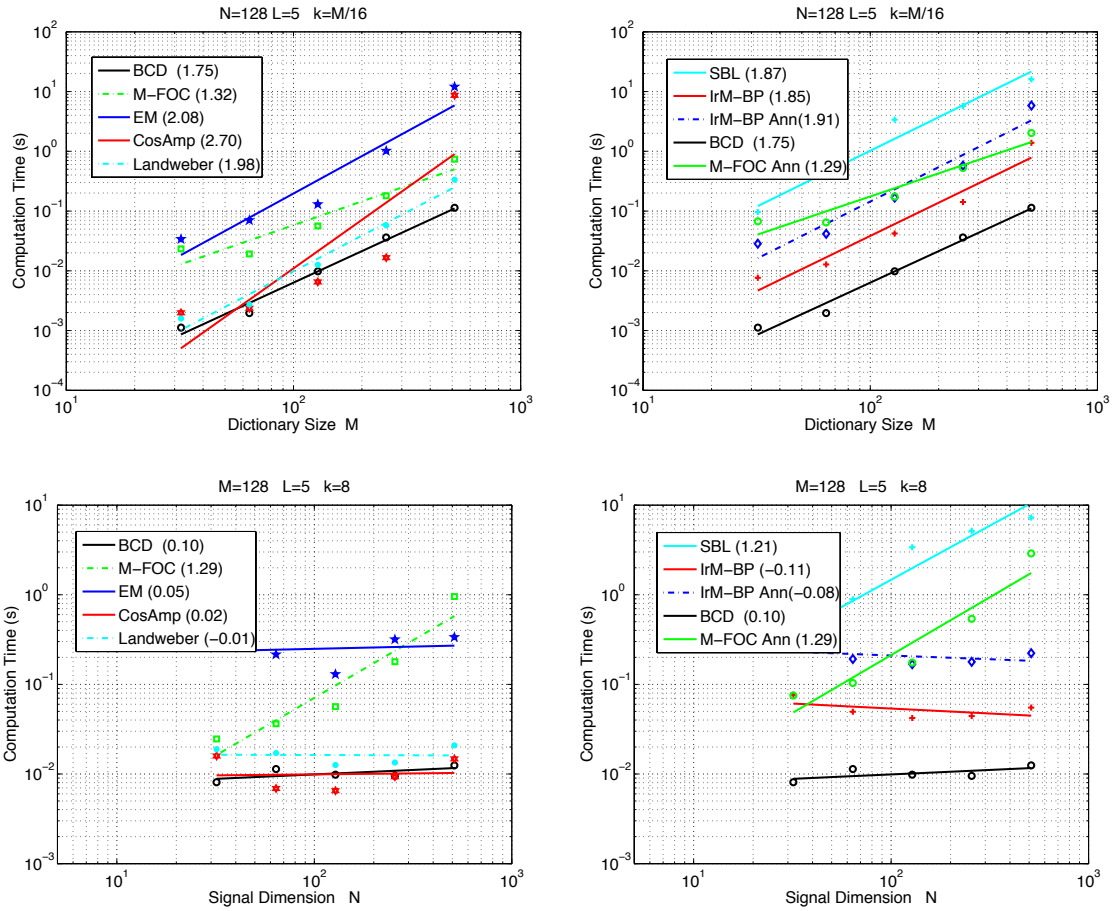

FIGURE 1.2 – Estimating the empirical exponent, given in parenthesis, of the computational complexity of different algorithms (M-BCD, IrM-BP, M-SBL, M-FOCUSS, CosAmp, Landweber iterations). The top plots give the computation time of the algorithms with respects to the dictionary size. The bottom plots respectively depict the computational complexity with respects to the signal dimensionality. For a sake of readability, we have separated the algorithms in two groups :(left) the ones that solve $\ell_1 - \ell_q$ problem. (right) the ones that solve $\ell_p - \ell_2$ problem (M-BCD result provided for baseline comparison). The "IrM-BP Ann" and "M-FOC Ann" refers to the Ir-MBP and M-FOCUSS algorithm using an annealing approach for iteratively decreasing ε as described in Algorithm (**??**).

with a slight advantage to M-BCD for large dictionary size. However, we have to note that the M-CosAmp algorithm sometimes suffers from lack of convergence and thus stop only when the maximal number of allowed iterations is reached. This is the reason why for large dictionary size M-CosAmp is computationally expensive. However, for small and medium dictionary size, M-CosAmp is slighly more efficient than M-BCD. When considering the algorithms that solve the $\ell_p - \ell_2$ problem (right plots), they all have similar complexity, with a slightly better constant for IrM-BP while M-SBL seems to be the most demanding algorithm.

Bottom plots of Figure 1.2 depicts the complexity dependency of all algorithms with respects to signal dimension N. Interestingly, the results show that except for M-SBL and M-FOCUSS algorithms, all algorithms do not suffer from the signal dimension increase. We assume that this is due to the fact that as dimension increases, the approximation problem becomes easier and thus faster convergence of those algorithms occurs.

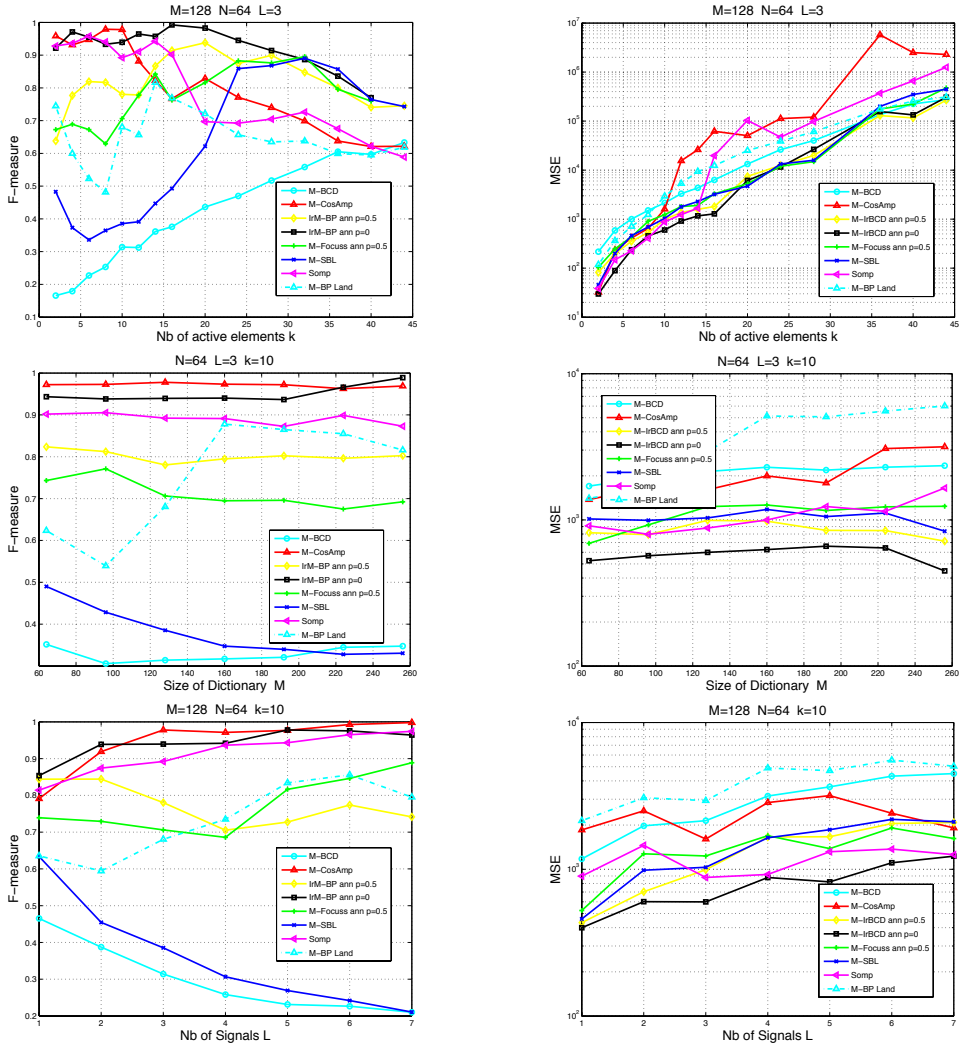

FIGURE 1.3 – Results comparing performances of different simultaneous sparse algorithms. We have varied (top) the number k of active elements in the dictionary. (middle) the dictionary size M and (bottom) the number of signal to approximate L. On the left columns are given the F-measure of all methods while the average mean-square errors are on the right column.

1.5.4 Comparing performances

The objective of the next empirical study is to compare the performances of the algorithms we surveyed : M-SBL, M-CosAmp, Landweber iterations, S-OMP, M-FOCUSS with an annealing decreasing of ε, the M-BCD algorithm and the IrM-BP approach with two values of p and an annealing decrease of ε.

The baseline experimental context is $M = 128$, $N = 64$, $k = 10$ and $L = 3$. For this experiment, we have considered an agnostic context with no prior knowledge about the noise level being available. Hence, for all models, we have performed model selection (either for selecting λ, the noise level σ for M-SBL or the number of elements for M-CosAmp and S-OMP). Model selection procedure is the following. Training signals $\mathbf{S}$ are randomly splitted in two parts of $N/2$ samples. Each algorithm is then trained on one part of the signal and the mean-square error of the resulting model is evaluated on the second part. This splitting and training is run 5 times and the hyperparameter yielding to the minimal averaged mean-square error is considered as optimal. Each method is then run on the full signals with that parameter. Performances, averaged over 50 trials of all methods have been evaluated according to the F-measure and a mean-square error computed on 10000 samples.

Figure 1.3 shows, from top to bottom, these performances when k increases from 2 to 40, when M goes

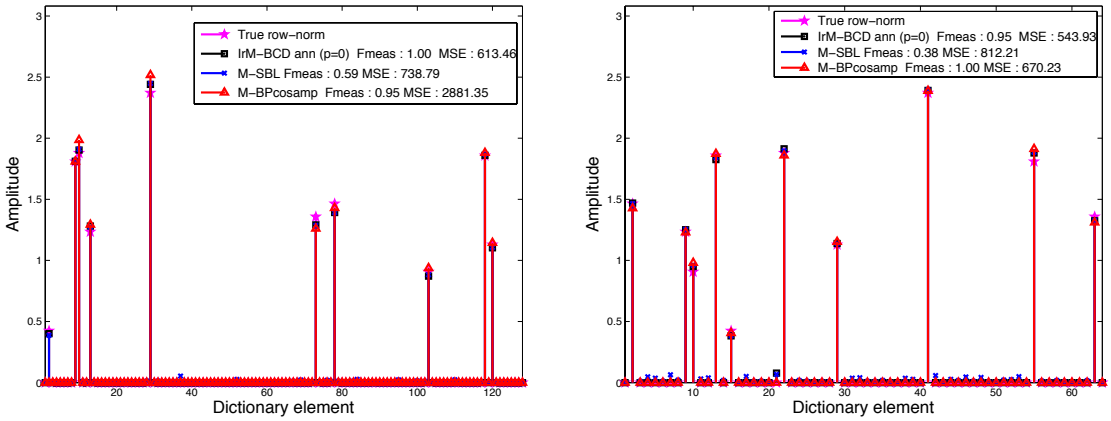

FIGURE 1.4 – Examples of estimated row-norm using 3 different algorithms. left) $M = 128$, $N = 64$, $k = 10$ and $L = 3$. right) $M = 64$, $N = 64$, $k = 10$ and $L = 3$. Here, we want to illustrate cases where a "good" sparsity recovery does not necessary lead to low mean-square error.

from 64 to 256 and when $L = 2, \cdots, 7$. When varying k, we can note that across the range of variation, our IrM-BP method with $p = 0$ is competitive compared to all other approaches both with respects to the F-measure and the mean-square error criterion. When k increases, IrM-BP and M-FOCUSS with $p = 0.5$ perform also very good. This may be explained by the fact that as k increases, the optimal solution becomes less and less sparse thus the need for a less aggressive penalty. CosAmp and S-OMP are very competitive for small k but as soon as the latter increases these two methods are not able anymore to recover a "reasonable" sparsity pattern. Interestingly, we remark that M-SBL yields to a poor sparsity recovery measure while the resulting model achieves good mean-square error. A reason for this is that the model selection procedure tends to under-estimate the noise level and thus it leads to a model which keeps many spurious dictionary elements as illustrated in Figure 1.4 and detailed in the sequel. From Figure 1.3, we can also notice that the two M-BP solvers, our M-BCD and the Landweber iteration approach perform poorly compared to other methods. However, the Fornasier's method seems to be less sensitive to noise and model selection since it provides a better sparsity pattern recovery. It is worth noting that M-SBL and these two latter methods always correctly select all the true dictionary elements but they also have the tendency to include other spurious ones.

In the middle and bottom plots, similar behavior as above can be highlighted. M-CosAmp yields to very good sparsity recovery while the resulting mean-square error is rather poor. Again our IrM-BP with $p = 0$ yields the best mean-square error while providing a good sparsity pattern recovery. M-SBL and M-BCD keeps too many spurious dictionary elements. All other methods provide in-between performances both in term of F-measure and mean-square error.

Figure 1.4 illustrates the behaviour of M-CosAmp, M-SBL and the IrM-BP with $p = 0$ for two different experimental situations. On the left plot, we have a case where on one hand, M-CosAmp misses to recover the first active dictionary element yielding thus to high mean-square error. On the other hand, M-SBL achieves lower mean-square error while keeping few spurious dictionary elements in the model. In the meantime, IrM-BP recovers perfectly the sparsity pattern and yields to low mean-square error. In the right plot, we have another case where M-CosAmp achieves perfect sparsity recovery but provides a model with higher mean-square error than IrM-BP.

In most of the experimental situations presented here, M-CosAmp and the IrM-BP seems to be the two algorithms that perform the best, with however a slight advantage for the IrM-BP. These two methods are actually related since both approaches solve a simultaneous sparse approximation with a $J_{0,2}(\mathbf{C})$ penalty. The main difference lies in the algorithms since our IrM-BP owing to the ε term provides a smooth approximation of the ℓ_0 quasi-norm whereas M-CosAmp directly solves the approximation problem with the $J_{0,2}(\mathbf{C})$ penalty.

To summarize, we suggest the following rule : in applications where computational complexities are critical and where signals to be approximated are highly sparse, M-CosAmp is to be preferred. In other situations, IrM-BP should be used.

1.6 Conclusions

This paper aimed at surveying and comparing different simultaneous sparse signal approximation algorithms available in the statistical learning and signal processing communities.

These algorithms usually solve a relaxed version of an intractable problem where the relaxation is based on sparsity-inducing mixed norm on the coefficient approximation. In the first part of the paper, we have described several algorithms which addresses the most common SSA problem which is a convex problem with a $\ell_1 - \ell_2$ penalty function. For other choices of penalty, one usually considers iterative reweighted algorithms. These algorithms have been detailed in a second part of the paper. We have also brought our attention on greedy algorithms such as S-OMP or M-CosAmp since they have both have the abilities to be very efficient and to provide theoretical guarantees on the quality of their solutions.

The last part of the paper is devoted to experimental comparisons of the different algorithms we surveyed. The lesson learned from these comparisons is that the two algorithms that seem to be the best performing either in terms of efficiency, sparsity recovery or low mean-square error are the M-CosAmp or an iterative reweighted ℓ_1 approach.

1.7 Appendix

1.7.1 $J_{1,2}(\mathbf{C})$ subdifferential

By definition, a matrix $\mathbf{G}$ lies in $\partial J_{1,2}(\mathbf{B})$ if and only if for every matrix $\mathbf{Z}$, we have

$$J_{1,2}(\mathbf{Z}) \geq J_{1,2}(\mathbf{B}) + \langle \mathbf{Z} - \mathbf{B}, \mathbf{G} \rangle_F \tag{1.44}$$

If we expand this equation we have the following equivalent expression

$$\sum_i \|z_{i,\cdot}\|_2 \geq \sum_i \|b_{i,\cdot}\|_2 + \sum_i \langle z_{i,\cdot} - b_{i,\cdot}, g_{i,\cdot} \rangle \tag{1.45}$$

From this latter equation, we understand that, since both $J_{1,2}$ and the Frobenius inner product are row-separable, a matrix $\mathbf{G} \in \partial J_{1,2}(\mathbf{B})$ if and only if each row of $\mathbf{G}$ belongs to the subdifferential of the ℓ_2 norm of the corresponding row of $\mathbf{B}$.

Indeed, suppose that $\mathbf{G}$ is so that any row of $\mathbf{G}$ belongs to the subdifferential of the ℓ_2 norm of the corresponding row of $\mathbf{B}$. We thus have for any row i

$$\forall \mathbf{z}, \quad \|\mathbf{z}\|_2 \geq \|b_{i,\cdot}\|_2 + \langle z - b_{i,\cdot}, g_{i,\cdot} \rangle \tag{1.46}$$

A summation over all the rows then proves that $\mathbf{G}$ satisfies equation (1.45) and thus belongs to the subdifferential of $J_{1,2}(\mathbf{B})$.

Now, let us show that a matrix $\mathbf{G}$ for which there exists a row that does not belong to the subdifferential of the ℓ_2 norm of the corresponding row of $\mathbf{B}$ can not belong to the subdifferential of $J_{1,2}(\mathbf{B})$. Let us consider $g_{i,\cdot}$ the i-th row of $\mathbf{G}$, since we have supposed that $g_{i,\cdot} \notin \partial \|b_{i,\cdot}\|_2$, the following equation holds

$$\exists \mathbf{z}_0 \text{ st.} \quad \|\mathbf{z}_0\|_2 < \|b_{i,\cdot}\|_2 + \langle \mathbf{z} - b_{i,\cdot}, g_i \rangle$$

Now let us construct $\mathbf{Z}$ so that $\mathbf{Z} = \mathbf{B}$ except for the i-th row where $z_{i,\cdot} = \mathbf{z}_0$. Then it is easy to show that this matrix $\mathbf{Z}$ does not satisfy equation (1.45), which means that $\mathbf{G}$ does not belong to $\partial J_{1,2}(\mathbf{B})$. In conclusion, we get $\partial J_{1,2}(\mathbf{B})$ by applying the ℓ_2 norm subdifferential to each row of $\mathbf{B}$. And it is well known [4] that

$$\partial \|\mathbf{b}\|_2 = \begin{cases} \{\mathbf{g} \in \mathbb{R}^L : \|\mathbf{g}\|_2 \leq 1\} & \text{if } \mathbf{b} = \mathbf{0} \\ \frac{\mathbf{b}}{\|\mathbf{b}\|_2} & \text{otherwise} \end{cases} \tag{1.47}$$

1.7.2 Proof of Lemma 2

We aim at proving that

$$\min_{\mathbf{d}} \left\{ \sum_{t,k} \frac{|a_{t,k}|^2}{d_{t,k}} : d_{t,k} \geq 0, \sum_k \left(\sum_t d_{t,k}^{1/s} \right)^{\frac{s}{r+s}} \leq 1 \right\} = \left(\sum_k \left(\sum_t |a_{t,k}|^q \right)^{\frac{p}{q}} \right)^{\frac{2}{p}}$$

where $q = \frac{2}{s+1}$ and $p = \frac{2}{s+r+1}$. The proof proceeds by writing the Lagrangian of the optimization problem :

$$\mathcal{L} = \sum_{t,k} \frac{|a_{t,k}|^2}{d_{t,k}} + \lambda \left(\sum_k \left(\sum_t d_{t,k}^{1/s} \right)^{\frac{s}{r+s}} - 1 \right) - \sum_{t,k} \nu_{t,k} d_{t,k}$$

where λ and $\{\nu_{t,k}\}$ are the Lagrangian multipliers associated to the inequality constraint and the positivity constraints on $d_{t,k}$. By deriving the first-order optimality conditions, we get :

$$\frac{\partial \mathcal{L}}{\partial d_{m,n}} = -\frac{|a_{m,n}|^2}{d_{m,n}^2} - \nu_{m,n} + \frac{\lambda s}{r+s} \left(\sum_t d_{t,n}^{1/s} \right)^{\frac{-r}{r+s}} \cdot \frac{1}{s} \cdot d_{m,n}^{\frac{1-s}{s}}$$

According to these optimality conditions, at a stationary point, we have either $d_{m,n} = 0$ or

$$d_{m,n} = \left(\frac{\lambda}{r+s} \right)^{-s/(s+1)} |a_{m,n}|^{2s/(s+1)} \left(\sum_t d_{t,n}^{1/s} \right)^{rs/[(r+s)(s+1)]} \tag{1.48}$$

Then, we can derive

$$\left(\sum_m d_{m,n}^{1/s}) \right)^{(s+1)} = \left(\frac{r+s}{\lambda} \right) \left(\sum_m |a_{m,n}|^{2/(s+1)} \right)^{s+1} \left(\sum_m d_{m,n}^{1/s} \right)^{r/(r+s)} \tag{1.49}$$

and thus

$$\left(\sum_m d_{m,n}^{1/s}) \right)^{s} = \left(\frac{r+s}{\lambda} \left(\sum_m |a_{m,n}|^{2/(s+1)} \right)^{s+1} \right)^{(r+s)/(r+s+1)} \tag{1.50}$$

As $\lambda \neq 0$, the inequality on the mixed-norm on $d_{t,k}$ becomes an equality. Hence, after powering each side of Equation (1.50) to $1/(r+s)$ and summing each side over n, we have :

$$\frac{\lambda}{r+s} = \left(\sum_n g_n^{(s+1)/(r+s+1)} \right)^{r+s+1} \tag{1.51}$$

where $g_n = \sum_m |a_{m,n}|^{2/(s+1)}$. Then, plugging equations (1.51) and (1.50) into (1.48) gives the desired result :

$$d_{m,n} = \frac{|a_{m,n}|^{\frac{2s}{s+1}} g_n^{\frac{r}{s+r+1}}}{\left(\sum_n g_n^{\frac{s+1}{s+r+1}} \right)^{r+s}} \tag{1.52}$$

1.7.3 Proof of equation (1.41)

We want to show that at optimality which occurs at $\mathbf{C}^\star$, we have

$$\mathbf{s}_j^t \Sigma_t^{-1} \mathbf{s}_j = \frac{1}{\sigma^2} \mathbf{s}_j^t (\mathbf{s}_j - \mathbf{\Phi C}^\star)$$

which is equivalent, after factorizing with $\mathbf{s}^t$, to show that

$$\sigma^2 \mathbf{s}_j = \Sigma_t \mathbf{s}_j - \Sigma_t \mathbf{\Phi C}^\star$$

This last equation can be proved using simple algebra

$$\begin{aligned} \Sigma_t \mathbf{s}_j - \Sigma_t \mathbf{\Phi C} &= \sigma^2 \mathbf{s}_j + \mathbf{\Phi D \Phi}^t \mathbf{s} - (\sigma^2 I + \mathbf{\Phi} D \mathbf{\Phi}^t) \mathbf{\Phi C}^\star \\ &= \sigma^2 \mathbf{s}_j + \mathbf{\Phi D \Phi}^t \mathbf{s} - \mathbf{\Phi} (\sigma^2 I + D \mathbf{\Phi}^t \mathbf{\Phi}) \mathbf{C}^\star \\ &= \sigma^2 \mathbf{s}_j + \mathbf{\Phi D \Phi}^t \mathbf{s} - \mathbf{\Phi D \Phi}^t \mathbf{s} \\ &= \sigma^2 \mathbf{s}_j \end{aligned}$$

Chapitre 2

Factorisation de très grandes matrices creuses pour la classification croisée

2.1 Modélisation

On dispose d'une matrice A de n lignes et p colonnes. Notre but est de trouver deux matrices orthogonales U et V qui minimisent

$$\|A - UV\| - F^2 = \sum_i \sum_j (a_{ij} - \mathbf{u}_i^\top \mathbf{v}_j)^2$$

Nous allons supposer cette matrice A est à valeurs dans $\{0, 1\}$ avec $A_{ij} = 1$ si le client i a acheté l'article j et $A_{ij} = 0$ sinon. Une manière de modéliser ce phénomène consiste à poser un modèle de Bernoulli de paramètre naturel θ_{ij} sur chaque terme a_{ij} de la matrice, vu comme la réalisation d'une variable aléatoire A_{ij} de probabilité P_{ij}

$$P_{ij} = \mathbb{P}(A_{ij} = 1) = \frac{\exp^{\theta_{ij}}}{1 + \exp^{\theta_{ij}}} \qquad \text{et} \qquad \mathbb{P}(A_{ij} = a_{ij}) = \frac{\exp^{a_{ij}\theta_{ij}}}{1 + \exp^{\theta_{ij}}}$$

C'est le modèle de l'analyse en composante principale logistique proposé par [2]. Dans ce modèle, θ_{ij} est le paramètre fondamental qui représente l'appétence du client i pour le produit j. la variable aléatoire X_{ij} est donc liée au « passage à l'acte » du client i sur le produit j. Quand θ_{ij} est grand, la probabilité d'achat (le passage à l'acte) est proche de un alors que quand θ_{ij} tend vers moins l'infini la probabilité d'achat tend vers zéro. A la matrice d'achat A sont associées deux autres matrices de même taille : la matrice P des probabilités d'achat et la matrice Θ des scores d'appétence.

L'hypothèse fondamentale concerne la structure de la matrice d'appétence. Nous allons supposer que cette matrice est structurée autour de variables non observables (cachées ou latentes) qui représentent un certain nombre k de profils type. Ces profils sont des vecteurs $U_{\bullet\kappa}$ et $V_{\bullet\kappa}$ qui caractérisent respectivement l'appétence des utilisateurs pour un produit type et l'appétence d'un client type pour tous les produits. A partir de ces profils, l'appétence d'un client i pour un produit j s'écrit comme une combinaison des différents profils, soit :

$$\theta_{ij} = \sum_{\kappa=1}^{k} U_{i\kappa} V_{j\kappa} = U_{i\bullet} V_{j\bullet}^\top \qquad \text{et} \quad \theta_{i\bullet} = \sum_{\kappa=1}^{k} U_{i\kappa} V_{\bullet\kappa}^\top \,, \quad \theta_{\bullet j} = \sum_{\kappa=1}^{k} U_{\bullet\kappa} V_{j\kappa}^\top$$

L'estimation des paramètres U et V peut s'obtenir grâce au principe du maximum de vraisemblance qui, si l'on fait l'hypothèse discutable d'indépendance des données, s'écrit :

$$\max_{U,V} \prod_i^n \prod_j^p \frac{\exp^{a_{ij} U_{i\bullet} V_{j\bullet}^\top}}{1 + \exp^{U_{i\bullet} V_{j\bullet}^\top}}$$

en prenant le logarithme et en changeant le signe on obtient la fonction score à minimiser :

$$\min_{U,V} S(U,V) \qquad \text{avec} \quad S(U,V) = \sum_{i}^{n}\sum_{j}^{p} -a_{ij}U_{i\bullet}V_{j\bullet}^{\top} + \log\big(1+\exp^{U_{i\bullet}V_{j\bullet}^{\top}}\big)$$

Une façon de résoudre ce problème de minimisation consiste à utiliser une approche de relaxation de type EM (*expectation maximization*) et de résoudre itérativement le problème de minimisation en U à V fixé puis le problème de minimisation en V à U fixé. Nous allons étudier le problème de minimisation en V à U fixé en suivant [1]. Dans ce cas le gradient de la fonction score s'écrit :

$$\nabla_{V_{j\bullet}} S(U,V) = \sum_{i=1}^{n} -U_{i\bullet}a_{ij} + U_{i\bullet}\frac{\exp^{U_{i\bullet}V_{j\bullet}^{\top}}}{1+\exp^{U_{i\bullet}V_{j\bullet}^{\top}}} = \sum_{i=1}^{n}(P_{ij}-a_{ij})U_{i\bullet}$$

de même :

$$\nabla_{U_{i\bullet}} S(U,V) = \sum_{j=1}^{p}(P_{ij}-a_{ij})V_{j\bullet}$$

Soit H la hessienne de $S(U,V)$ et $X \in \mathbb{R}^{(n+p)\times k}$. X est de même taille qu'une matrice de la forme $\begin{pmatrix} U \\ V \end{pmatrix}$.

$$\begin{array}{rccc} H: & \mathbb{R}^{(n+p)\times k} & \longmapsto & \mathbb{R}^{(n+p)\times k} \\ & X & \longmapsto & H(X) \end{array}$$

Ainsi $H \in \mathbb{R}^{(n+p)^2\times k^2}$. La Hessienne est en 4 partie :

$$H = \begin{pmatrix} \nabla_U^2 S & \nabla_U\nabla_V S \\ \nabla_V\nabla_U S & \nabla_V^2 S \end{pmatrix}$$

Calculons chaque partie indépendamment.

$$\begin{aligned} \nabla_{V_{j_2\bullet}}\nabla_{V_{j_1\bullet}} S(U,V) &= \nabla_{V_{j_2\bullet}}[\sum_{i=1}^{n}(P_{ij_1}-a_{ij_1})U_{i\bullet}] \\ &= \begin{cases} \sum_{i=1}^{n} U_{i\bullet}^{\top}P_{ij_1}(1-P_{ij_1})U_{i\bullet} & \text{si } i_1 = i_2 \\ 0 & \text{si } i_1 \neq i_2 \end{cases} \\ \nabla_{V_{j\bullet}}^2 S(U,V) &= \sum_{i=1}^{n} U_{i\bullet}^{\top}P_{ij}(1-P_{ij})U_{i\bullet} \\ &= U^{\top}W_{V_{j\bullet}}U \end{aligned}$$

avec $W_{V_{j\bullet}}(i,i) = P_{ij}(1-P_{ij})$ pour tout i.

$$\begin{aligned} \nabla_{U_{i\bullet}}^2 S(U,V) &= \sum_{j=1}^{p} V_{j\bullet}^{\top}P_{ij}(1-P_{ij})V_{j\bullet} \\ &= V_{j\bullet}^{\top}W_{U_{i\bullet}}V_{j\bullet} \end{aligned}$$

avec $W_{U_{i\bullet}}(j,j) = P_{ij}(1-P_{ij})$ pour tout j. Notons que $W_{V_{j\bullet}} \in \mathbb{R}^{n\times n}$ et $W_{U_{i\bullet}} \in \mathbb{R}^{p\times p}$.

$$\begin{aligned} \nabla_{V_{j\bullet}}\nabla_{U_{i\bullet}} S(U,V) &= \nabla_{V_{j\bullet}}[\sum_{i=1}^{n}(P_{ij}-a_{ij})V_{j\bullet}] \\ &= \frac{U_{i\bullet}^{\top}e^{U_{i\bullet}V_{j\bullet}^{\top}}(1+e^{U_{i\bullet}V_{j\bullet}^{\top}}) - U_{i\bullet}^{\top}e^{U_{i\bullet}V_{j\bullet}^{\top}}e^{U_{i\bullet}V_{j\bullet}^{\top}}}{(1+e^{U_{i\bullet}V_{j\bullet}^{\top}})^2}V_{j\bullet} + (P_{ij}-a_{ij})I_k \\ &= U_{i\bullet}^{\top}P_{ij}(1-P_{ij})V_{j\bullet} + (P_{ij}-a_{ij})I_k \end{aligned}$$

où I_k est la matrice identité de taille $k \times k$.

$$\begin{aligned}\nabla_{U_{i\bullet}} \nabla_{V_{j\bullet}} S(U,V) &= \nabla_{U_{i\bullet}} [\sum_{j=1}^{p} (P_{ij} - a_{ij}) U_{i\bullet}] \\ &= V_{j\bullet}^\top P_{ij}(1 - P_{ij}) U_{i\bullet} + (P_{ij} - a_{ij}) I_k\end{aligned}$$

Ainsi pour résumé :

$$\nabla^2_{U_{i\bullet}} S(U,V) = V_{j\bullet}^\top W_{U_{i\bullet}} V_{j\bullet} \quad (2.1)$$

$$\nabla^2_{V_{j\bullet}} S(U,V) = U_{i\bullet}^\top W_{V_{j\bullet}} U_{i\bullet} \quad (2.2)$$

$$\nabla_{V_{j\bullet}} \nabla_{U_{i\bullet}} S(U,V) = U_{i\bullet}^\top P_{ij}(1 - P_{ij}) V_{j\bullet} + (P_{ij} - a_{ij}) I_k \quad (2.3)$$

$$\nabla_{U_{i\bullet}} \nabla_{V_{j\bullet}} S(U,V) = V_{j\bullet}^\top P_{ij}(1 - P_{ij}) U_{i\bullet} + (P_{ij} - a_{ij}) I_k \quad (2.4)$$

Nous choisissons de réduire l'expression de la matrice hessienne aux équations numéro 1 et 2. Une itération de l'algorithme de Newton appliqué à ce problème s'écrit alors en partant du vecteur $V_{\bullet j}^{\text{old}}$:

$$\begin{aligned}V_{j\bullet}^{\text{new}} &= V_{j\bullet}^{\text{old}} - (U^\top W_{V_{j\bullet}} U)^{-1} U^\top (P_{\bullet j} - A_{\bullet j}) \\ &= (U^\top W_{V_{j\bullet}} U)^{-1} \big((U^\top W_{V_{j\bullet}} U) V_{j\bullet}^{\text{old}} - U^\top (P_{\bullet j} - A_{\bullet j}) \big) \\ &= (U^\top W_{V_{j\bullet}} U)^{-1} U^\top W_{V_{j\bullet}} (U V_{\bullet j}^{\text{old}} - W_{V_{j\bullet}}^{-1} (P_{\bullet j} - A_{\bullet j})) \\ &= (U^\top W_{V_{j\bullet}}^{\frac{1}{2}\top} W_{V_{j\bullet}}^{\frac{1}{2}} U)^{-1} U^\top W_{V_{j\bullet}}^{\frac{1}{2}\top} \mathbf{z}_{V_{j\bullet}} \\ &= (W_{V_{j\bullet}}^{\frac{1}{2}} U)^\dagger \mathbf{z}_{V_{j\bullet}}\end{aligned}$$

où $M_{ij}^{\frac{1}{2}\top} = \sqrt{M_{ij}}$ pour tout i et j.
avec $\mathbf{z}_{V_{j\bullet}} = W_{V_{j\bullet}}^{\frac{1}{2}} U V_{\bullet j}^{\text{old}} - W_{V_{j\bullet}}^{-\frac{1}{2}} (P_{\bullet j} - A_{\bullet j})$.

Algorithm 10 Factorisation Binomiale

Require: $U \in \mathbb{R}^{n \times k}$, $V \in \mathbb{R}^{p \times k}$
while not convergent **do**
  for all i and j **do**
    $P_{ij} \leftarrow \frac{e^{U_{i\bullet} V_{j\bullet}^\top}}{1 + e^{U_{i\bullet} V_{j\bullet}^\top}}$
  end for
  for $i = 1$ to n **do**
    $U_{i\bullet}^{new} \leftarrow (W_{U_{i\bullet}}^{\frac{1}{2}} V)^\dagger \mathbf{z}_{U_{i\bullet}}$
  end for
  for $j = 1$ to p **do**
    $V_{j\bullet}^{new} \leftarrow (W_{V_{j\bullet}}^{\frac{1}{2}} U)^\dagger \mathbf{z}_{V_{j\bullet}}$
  end for
  $U \leftarrow U^{new}$
  $V \leftarrow V^{new}$
end while
return U,V

Il existe des approximations de $(X)^\dagger$. Notamment celle qui consiste à effectuer la SVD de X, en posant $X = u\sigma v^\top$, alors $(X)^\dagger = v\sigma^{-1}u^\top$

2.2 Expériences

2.2.1 Les données

La base de données que nous avons utilisé pour tester les performances de l'algorithme de factorisation est celle de Netflix. Nous avons réduit la taille de la base en ne conservant que 3000 utilisateurs choisi aléatoirement, et en ne gardant que les 2000 articles pour lesquels ces 3000 utilisateurs ont le plus voté. Comme pour le challenge KDD « *who rated what* », les données de vote sont transformées en matrice de 1 et de 0, où la valeur 1 à la ligne i et à la colonne j signifie que l'utilisateur i a voté pour l'article j, et la valeur 0 signifie l'inverse.

2.2.2 Les critères

Pour chaque utilisateurs, nous cachons 5 valeurs tirés au hasard, ce seront les votes à retrouver. La reconstruction de la matrice factorisée nous donne une matrice avec des scores d'appétence, plus le score est haut plus l'article aura de chance d'avoir un vote. Nous recommandons 5 articles par utilisateur, et nous calculons le pourcentage de ces articles présents dans les 5 articles cachés. Ceci correspond à la fois à un calcul du rappel à 5 et de la précision à 5, qui dans ce cadre sont égaux. Étant donné le cadre que nous nous fixons, nous ne recommandons pas les articles déjà achetés par les clients, ainsi nous recommandons les 5 articles (parmi ceux qui n'ont pas de vote) ayant le plus haut score d'appétence pour chaque utilisateur. Cette expérience peut être répétée, avec 3000 autres utilisateurs tirés aléatoirement.

2.2.3 Les résultats

Pour évaluer les performances de notre méthodes, nous la comparons avec une méthode standard qui est la factorisation SVD.

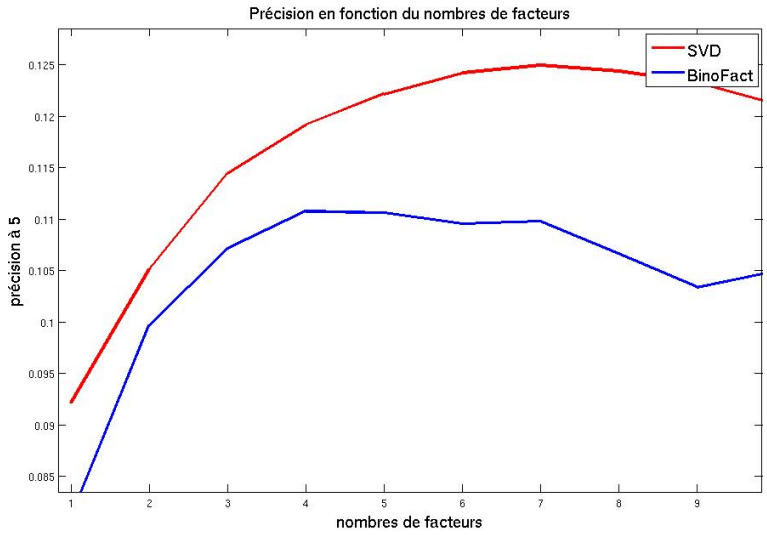

FIGURE 2.1 – Précision en fonction du nombre de facteurs

La figure 2.1 représente la précision à 5 en fonction du nombre de facteurs pour les méthodes de factorisation SVD et binomiale. On note tout d'abord que pour chaque courbe, la précision augmente jusqu'à atteindre un maximum et pour enfin redescendre. On voit clairement que la précision de la méthode de factorisation binomiale est inférieure à celle de la SVD. Non seulement la complexité de l'algorithme est supérieure, mais en plus les performances sont moindres. Cette méthode ne semble malheureusement pas à retenir, un algorithme de Newton ne semble pas approprié pour ce genre de problème. Le seul point positif serait que le maximum de précision est atteint pour un nombre de paramètres inférieur à celui de la SVD.

Bibliographie

[1] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning : data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2) :83–85, 2005.

[2] A.I. Schein, L.K. Saul, and L.H. Ungar. A generalized linear model for principal component analysis of binary data. In *In Proceedings of the 9 th International Workshop on Artificial Intelligence and Statistics*, 2003.

[3] Baraniuk, R. G., Cevher, V., Duarte, M., Hegde, C., 2009. Model-based compressive sensing. IEEE Transactions on Information Theory to appear.

[4] Bertsekas, D., Nedic, A., Ozdaglar, A., 2003. Convex Analysis and Optimization. Athena Scientific.

[5] Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.

[6] Candès, E., Wakin, M., Boyd, S., 2008. Enhancing sparsity by reweighted ℓ_1 minimization. J. Fourier Analysis and Applications 14, 877–905.

[7] Chartrand, R., Yin, W., 2008. Iteratively reweighted algorithms for compressive sensing. In : 33rd International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

[8] Chen, J., Huo, X., 2005. Sparse representations for multiple measurements vectors (mmv) in an overcomplete dictionary. In : Proc IEEE Int. Conf Acoustics, Speech Signal Processing. Vol. 4. pp. 257–260.

[9] Chen, S., Donoho, D., Saunders, M., 1999. Atomic decomposition by basis pursuit. SIAM Journal Scientific Comput. 20 (1), 33–61.

[10] Cotter, S., Rao, B., Engan, K., Kreutz-Delgado, K., 2005. Sparse solutions to linear inverse problems with multiple measurement vectors. IEEE Transactions on Signal Processing 53 (7), 2477–2488.

[11] Daubechies, I., Defrise, M., Mol, C. D., 2004. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Communication Pure Applied Mathematics 57, 1413–1541.

[12] Daubechies, I., DeVore, R., Fornasier, M., Gunturk, S., 2009. Iteratively reweighted least squares minimization for sparse recovery. Commun. Pure Appl. Math to appear.

[13] Duarte, M., Cevhler, V., Baraniuk, R., 2009. Model-based compressive sensing for signal ensembles. In : Proceeding of the 47th Allerton Conference on Communication, Control and Computing.

[14] Elad, M., 2006. Why simple shrinkage is still relevant for redundant representations ? IEEE Trans. on Information Theory 52 (12), 5559–5569.

[15] Eldar, Y., Kuppinger, P., Blocskei, H., 2009. Compressed sensing of block-sparse signals : Uncertainty relations and efficient recovery. IEEE Trans. Signal Processing to appear.

[16] Eldar, Y., Mishali, M., To appear. Robust recovery of signals from a structured union of subspaces. IEEE Trans. Information Theory.

[17] Fan, J., Li, R., 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American Statistical Association 96 (456), 1348–1360.

[18] Figueiredo, M., Bioucas-Dias, J., R., N., 2007. Majorization-minimization algorithms for wavelet-based image for restoration. IEEE Trans. on Image Processing 16 (12), 2980–2991.

[19] Fornasier, M., Rauhut, H., 2008. Recovery algorithms for vector valued data with joint sparsity constraints. SIAM Journal of Numerical Analysis 46 (2), 577–613.

[20] Foucart, S., Lai, M.-J., 2009. Sparsest solutions of underdetermined linear systems via ℓ_q-minimization for $0 < q \leq 1$. Applied and Computational Harmonic Analysis 26 (3), 395–407.

[21] Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., 2007. Pathwise coordinate optimization. The Annals of Applied Statistics 1 (2), 302–332.

[22] Girolami, M., Rogers, S., 2005. Hierarchic bayesian models for kernel learning. In : Proc. of 22nd International Conference on Machine Learning. pp. 241–248.

[23] Gorodnitsky, I., George, J., Rao, B., 1995. Neuromagnetic source imaging with FOCUSS : a recursive weighted minimum norm algorithm. J. Electroencephalogr. Clin. Neurophysiol. 95 (4), 231–251.

[24] Grandvalet, Y., 1998. Least absolute shrinkage is equivalent to quadratic penalization. In : Niklasson, L., Bodén, M., Ziemske, T. (Eds.), ICANN'98. Vol. 1 of Perspectives in Neural Computing. Springer, pp. 201–206.

[25] Gribonval, R., Nielsen, M., 2006. Sparse approximations in signal and image processing. Signal Processing 86 (3), 415–416.

[26] Huang, J., Ma, S., Xie, H., Zhang, C., 2009. A group bridge approach for variable selection. Biometrika 96 (2), 339–355.

[27] Hunter, D., Lange, K., 2004. A tutorial on MM algorithms. The American Statistician 58, 30–37.

[28] Ji, S., Dunson, D., Carin, L., 2008. Multi-task compressive sensing. IEEE Trans. Signal Processing 57 (1), 92–106.

[29] Kim, Y., Kim, J., Kim, Y., 2006. blockwise sparse regression. Statistica Sinica 16, 375–290.

[30] Liu, H., Zhang, J., 2009. On the estimation and variable selection consistency of the sum of q-norm regularized regression. Tech. rep., Department of Statistics, Carnegie Mellon University.

[31] Luo, Z., Gaspar, M., Liu, J., Swami, A., 2006. Distributed signal processing in sensor networks. IEEE Signal Processing magazine 23 (4), 14–15.

[32] Malioutov, D., Cetin, M., Willsky, A., 2005. Sparse signal reconstruction perspective for source localization with sensor arrays. IEEE Trans. Signal Processing 53 (8), 3010–3022.

[33] Mallat, S., Zhang, Z., 1993. Matching pursuit with time-frequency dictionaries. IEEE Trans Signal Processing 41 (12), 3397–3415.

[34] Mishali, M., Eldar, Y., 2008. Reduce and boost : Recovering arbitrary sets of jointly sparse vectors. IEEE Trans. On Signal Processing 56 (10), 4692–4702.

[35] Needell, D., Tropp, J., 2009. Cosamp : Iterative signal recovery from incomplete and inaccurate samples. Applied and Computational Harmonic Analysis 26 (3), 301–321.

[36] Obozinski, G., Taskar, B., Jordan, M., 2009. Joint covariate selection and joint subspace selection for multiple classification problems. Statistics and Computing to appear.

[37] Pati, Y., Rezaiifar, R., Krishnaprasad, P., 1993. Orthogonal matching pursuit : Recursive function approximation with applications to wavelet decomposition. In : Proc. of the 27th Annual Asilomar Conference on Signals, Systems and Computers.

[38] Phillips, C., Mattout, J., Rugg, M., Maquet, P., Friston, K., 2005. An empirical Bayesian solution to the source reconstruction problem in EEG. NeuroImage 24, 997–1011.

[39] Rakotomamonjy, A., Bach, F., Grandvalet, Y., Canu, S., 2008. SimpleMKL. Journal of Machine Learning Research 9, 2491–2521.

[40] Rao, B., Engan, K., Cotter, S., Palmer, J., Kreutz-Delgado, K., 2003. Subset selection in noise based on diversity measure minimization. IEEE Trans. Signal Processing 51 (3), 760–770.

[41] Saab, R., Chartrand, R., Özgür Yilmaz, 2008. Stable sparse approximations via nonconvex optimization. In : 33rd International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

[42] Sardy, S., Bruce, A., Tseng, P., 2000. Block coordinate relaxation methods for non-parametric wavelet denoising. Journal of Computational and Graphical Statistics 9 (2), 361–379.

[43] Simila, T., 2007. Majorize-minimize algorithm for multiresponse sparse regression. In : IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). pp. 553–556.

[44] Simila, T., Tikka, J., 2007. Input selection and shrinkage in multiresponse linear regression. Computational Statistics and Data analysis 52 (1), 406–422.

[45] Stojnic, M., Parvaresh, F., Hassibi, B., 2008. On the reconstruction of block-sparse signals with an optimal number of measurements. Tech. Rep. ArXiv 0804.0041v1, California Instite of Technology.

[46] Sturm, J., 1999. Using SeDuMi 1.02 a Matlab toolbox for optimization over symmetric codes. Optimization Methods and Software 11 (12), 625–653.

[47] Sun, L., Liu, J., Chen, J., Ye, J., 2009. Efficient recovery of jointly sparse vectors. In : Advances in Neural Information Processing Systems 23.

[48] Theis, F., Garcia, G., 2006. On the use of sparse signal decomposition in the analysis of multi-channel surface electromyograms. Signal Processing 86 (3), 603–623.

[49] Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society 46, 267–288.

[50] Tipping, M., 2001. Sparse Bayesian Learning and the Relevance Vector Machine. Journal of Machine Learning Research 1, 211–244.

[51] Tropp, J., 2006. Algorithms for simultaneous sparse approximation. part II : Convex relaxation. Signal Processing 86, 589–602.

[52] Tropp, J., Gilbert, A., 2007. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Information Theory 53 (12), 4655–4666.

[53] Tropp, J., Gilbert, A., Strauss, M., 2006. Algorithms for simultaneous sparse approximation. part I : Greedy pursuit. Signal Processing 86, 572–588.

[54] Tseng, P., 2001. Convergence of block coordinate descent method for nondifferentiable minimization. Journal of Optimization Theory and Application 109, 475–494.

[55] van den Berg, E., Schmidt, M., Friedlander, M., Murphy, K., 2008. Group sparsity via linear-time projection. Tech. Rep. TR-2008-09, University of British Columbia, Department of Computer Science.

[56] Wipf, D., Nagarajan, S., 2008. A new view of automatic relevance determination,. In : Advances in Neural Information Processing Systems. Vol. 20. MIT Press, Cambridge, MA.

[58] Wipf, D., Nagarajan, S., 2010. Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions. Journal of Selected Topics in Signal Processing, Special Issue on Compressive Sensing to appear.

[58] Wipf, D., Owen, J., Attias, H., Sekihara, K., Nagarajan, S., 2010. Robust bayesian estimation of the location orientation and time course of multiple correlated neural sources using meg. NeuroImage 49 (1), 641–655.

[59] Wipf, D., Rao, B., Jul. 2007. An empirical bayesian strategy for solving the simultaneous sparse approximation problem. IEEE Trans on Signal Processing 55 (7), 3704–3716.

[60] Yuan, M., Lin, Y., 2006. Model selection and estimation in regression with grouped variables. Journal of Royal Statisctics Society B 68, 49–67.

[61] Zhou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistics Society Ser. B 67, 301–320.

[62] Zou, H., 2006. The adaptive lasso and its oracle properties. Journal of the American Statistical Association 101 (476), 1418–1429.

[63] Zou, H., Li, R., 2008. One-step sparse estimates in nonconcave penalized likelihood models. The Annals of Statistics 36 (4), 1509–1533.